



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
COLEGIO DE CIENCIAS Y HUMANIDADES
SECRETARÍA ACADÉMICA**



**SEMINARIO CENTRAL PROPUESTA EDUCATIVA PARA APOYAR A
LA ACTUALIZACIÓN DE LOS PROGRAMAS DE ESTUDIO DE**

Cibernética y Computación I y Cibernética y Computación II

CICLO ESCOLAR 2021-2022

COORDINADOR (ES):

Ignacio Rafael Vázquez Torre

Jeanett Figueroa Martínez

INTEGRANTES:

Nombre del docente	Plantel de adscripción
Atzin Antonio Acosta Zavala	Azcapotzalco
Dulce Patricia Domínguez Arias	Azcapotzalco
Edgar Omar García Vázquez	Oriente
Gabriela López Vargas	Oriente
Javier Millán Martínez	Sur
Héctor Gabriel Rivera Vargas	Naucalpan
Claudia del Socorro Rodríguez Saldívar	Naucalpan
Yazmin Toeyotl Calderon	Oriente
Angie Terán González	Azcapotzalco
Mario Alberto Terrés Palacios	Vallejo

Julio 2022



Índice	
Propuesta Educativa para apoyar a la actualización de los programas de estudios de las asignaturas de Cibernética y Computación I y Cibernética y Computación II	
Introducción donde se indique el Campo de Actividad, los objetivos y los alcances a los que se llegó al finalizar el proyecto.	54
Marco teórico metodológico.	54
La información en que se basa.	62
La contribución disciplinaria, didáctica y pedagógica para el mejoramiento de la docencia.	64
Referencias de donde se fundamentó la información del marco teórico metodológico.	65
Propuesta de aplicación.	68

Introducción

El objetivo principal del seminario ha sido crear una propuesta teórico-metodológica fundamentada e innovadora, basada en un estudio o investigación formal, con el propósito de mejorar el proceso de enseñanza-aprendizaje que propone aspectos disciplinarios, didácticos y pedagógicos centrales de la docencia de las asignaturas de Cibernética y Computación I y II, de forma coherente con los propósitos y el Modelo Educativo, el Plan y Programas de Estudio del Colegio.

Por lo tanto en esta propuesta educativa no intenta encontrar el hilo negro de la educación, solo pretende innovar en otras palabras, mejorar lo que existe, aportando nuevas opciones desde la perspectiva de los resultados del trabajo elaborado desde el diagnóstico hasta la evaluación al programa de las asignaturas de Cibernética y Computación I y II de años anteriores, en los cuales ya existen propuestas de mejoras, pero consideramos que quedó inconcluso el rubro de estrategias sugeridas, e incluso fue punto de debate del coloquio realizado... donde se mostraron los avances de estas evaluaciones a programas de estudios de distintas asignaturas, se debatió el que si eran estrategias de enseñanza o de aprendizaje, que si el Modelo es centrado en el estudiante debieran las estrategias ser enfocadas hacia él, sin embargo, después de estudiar el papel crucial que tiene el profesor no como orador y solo transmisor de conocimiento, sino como guía, orientador, experto que ayuda en el andamiaje de la construcción del conocimiento del estudiante y muchas más acciones que se platicaran a través del marco teórico conceptual que a continuación se presenta, junto con la referencia de la que se recuperó dicha información, agregando la contribución disciplinar, pedagógica y teórica de esta propuesta para terminar con la aplicación de esta propuesta didáctica.

a) Marco Teórico Conceptual

Esta propuesta educativa pretende enfocarse en la construcción de estrategias de enseñanza-aprendizaje, al ser éstas, el motor del quehacer docente para guiar el aprendizaje del estudiante durante su proceso educativo.

Para M. Esteban, O, Valera y A. Ruiz (1999, en Zilberteín, 2005. p. 9) la educación es “un proceso conscientemente organizado, dirigido, sistematizado sobre la base de una concepción pedagógica determinada que se plantea como



objetivo más general la formación multilateral y armónica del educando para que se integre a la sociedad en que vive, contribuye a su desarrollo y perfeccionamiento”; mientras que la enseñanza es el proceso de organización de la actividad cognoscitiva de los escolares que implica la apropiación de la experiencia científico, matemático, histórico social y lenguaje para la asimilación del conocimiento que mediatiza toda su actividad y contribuye a la formación integral.



Para el logro del objetivo del seminario, fue determinante el conocimiento acerca de las principales teorías del aprendizaje, que enmarcan el Modelo Educativo de la Escuela Nacional Colegio de Ciencias y Humanidades.

Las teorías del aprendizaje son la base principal de éste proceso, cabe señalar que el principio fundamental del Modelo Educativo del Colegio de ciencias y Humanidades es el “Aprender a Aprender” que es una teoría del aprendizaje que tiene como base orientar la formación de las capacidades del sujeto para que, de forma consciente, se apropie de los cuatro grupos de Habilidades Conformadoras del Desarrollo de la Personalidad (HCDP).

Podemos decir que las teorías del aprendizaje son la pauta del surgimiento del modelo educativo de nuestro colegio, entendiendo por aprendizaje la definición de Gonzales, (2000) “al proceso de interacción que produce cambios internos, modificación de los procesos en la configuración psicológica del sujeto de forma activa y continua” y de ahí que la teoría cognitiva se basa en este aprendizaje, porque el estudiante posee estructuras organizativas, cognitivas en las que va integrando tanto la información ya adquirida como la que va adquiriendo a lo largo de su vida, para luego formar conceptos significativos, en otras palabras que tienen sentido para él.

En esta teoría, el alumno se vuelve un participante activo de su propio proceso de aprendizaje, tal y como lo mencionó en su momento el Rector de la Universidad Nacional Autónoma de México (UNAM), Pablo González Casanova “...ubica al alumno en el centro del acto educativo y lo concibe como una persona capaz de transformar su medio y a sí mismo, convirtiendo a la educación en un acto vivo y dinámico” (Gaceta UNAM, 1971. p. 7).

Dentro del cognitivismo, el trabajo del docente consiste en indagar las diferentes experiencias y conocimientos previos de los estudiantes, con el fin de planificar

y estructurar los contenidos disciplinares y procedimentales para adaptarlos al estilo de aprendizaje de sus alumnos.

En el CCH el principio de aprender a aprender refiere a un concepto “multidimensional que incluye aspectos metacognitivos, habilidades complejas del pensamiento, autorregulación y autoestima, que son algunos elementos base para aprender a lo largo de la vida y propician que el alumno sea capaz de adquirir nuevos conocimientos por cuenta propia; es decir, que asuma la responsabilidad sobre su propio aprendizaje, situándolo como el actor central del acto educativo” (Gaceta UNAM, 1971, p. 4).



Lo anterior va de la mano con la teoría psicológica del cognitivismo de Piaget en Flavell, (1992) sentó las bases diciendo que la **construcción del conocimiento** es indispensable la acción del organismo. Pero no cualquier tipo de acción conduce a la construcción del conocimiento. Las acciones que se relacionan con el conocimiento son acciones que tienen una regularidad y una organización interna producen ciertos esquemas que van conformando estructuras cognitivas **entendidas como el desarrollo de un proceso de construcción gradual** que permiten mantener niveles superiores de equilibrio con el medio. Y aplicables a cualquier ámbito del conocimiento; el proceso de construcción de las estructuras operatorias) haciendo énfasis en el desarrollo como un proceso interno del organismo, que sigue un curso universal y que se fundamenta en los **mecanismos de equilibración** (capacidad de autorregulación del organismo que tiende a mantener un equilibrio con el medio), **asimilación (lo que se percibe en el mundo exterior se incorpora en el mundo interno)** y **acomodación (el mundo interno tiene que acomodarse a la evidencia externa** con la que se enfrenta y, por lo tanto, adaptarse a ella) y dividió en cuatro etapas dicho desarrollo cognoscitivo: etapa sensoriomotora, etapa preoperacional, etapa de las operaciones concretas y etapa de las operaciones formales, de aquí Norte América adopta el sistema K12 designación utilizada en algunos sistemas educativos para la escolarización preprimaria, primaria y secundaria, sistema que integra México también.

Otros autores basados en Piaget pero en el marco pedagógico de la educación durante el constructivismo, como Bruner (1960) refiere que el aprendizaje es la esencia activa del proceso educativo, lo cual significa que una persona que

aprende algo nuevo, lo incorpora a sus experiencias previas y a sus propios esquemas mentales mientras que el profesor guía la predisposición del estudiante hacia el aprendizaje relacionándolo con los contenidos, con las experiencias y contextos que hacen que los alumnos estén desahogados de aprender, para otro de los exponentes de esta teoría Ausubel (1968) el aprendizaje busca promover los procesos de crecimiento del alumno en su propio entorno construyendo el conocimiento a través del descubrimiento de contenidos y al ser retomados desde su contexto, desde lo que comprende entonces el aprendizaje se vuelve significativo, por lo que el autor refiere que el concepto que el estudiante va aprender de forma activa, constructiva, colaborativa, intencional, conversacional, contextualizado y reflexiva, sin perder de vista lo que apuntala el sentido del principio de *aprender a aprender* en el modelo educativo del colegio, como bien lo menciona la filosofía del modelo educativo del colegio, “Aprender a aprender va más allá del aprender a conocer y se manifiesta de distintas formas, una de ellas se presenta cuando el alumno es consciente tanto del proceso que lo llevó a formular un nuevo conocimiento, como de la manera en que se vincula éste con otros conocimientos...” (p. 4), esto quiere decir que el estudiante se da cuenta de sus logros y dificultades, reflexiona sobre lo que aprendió y su nuevo pensamiento, adquiere la capacidad de investigar, se vuelve participativo, creativo y crítico.

El papel del profesor en la teoría ausubeliana, se sitúa como orientador o guía durante el proceso de aprendizaje, así como de promotor de una enseñanza activa.

Otro de los principios del modelo educativo del colegio es el *aprender a hacer*, donde se pretende que el estudiante sea capaz de leer y analizar cualquier tipo de texto, escribir y expresar de forma adecuada sus ideas, resolver problemas, llevar a cabo proyectos o experimentos, trabajar en equipo, entre otras cosas, el principio de aprender a hacer busca evaluar el proceso,

Desde estas teorías se sugieren ideas para emplear distintas técnicas o modelos e implementarlas en el quehacer del docente para sus estrategias didácticas, por ejemplo el modelo pedagógico Aula Invertida creado por Sams y Bergaman (2014), la cual refiere a un método de enseñanza cuyo principal objetivo consiste en que el estudiante asuma un rol activo en su proceso de aprendizaje, que estudie los conceptos teóricos por sí mismo a través de las diversas



herramientas como la tecnología o materiales didácticos, que le son proporcionados por el profesor que a su vez ya organizó y estructuro con la finalidad de que el estudiante obtenga el aprendizaje y durante el tiempo de clase dedicarle a esclarecer dudas relacionadas con la temática investigada, realiza práctica, abrir foros de discusión, crear líneas de tiempo, infografías, presentaciones, videos, de tal forma que el aprendizaje sea más activo y cooperativo. El modelo intenta fomentar el interés y el compromiso en los estudiantes, lo que, lo hace protagonista de su propio aprendizaje y además fomenta su participación en clase, el pensamiento crítico, analítico y su creatividad.



Otra Metodología didácticas es el trabajo colaborativo, el aprendizaje basado en problemas (ABP)¹, al igual busca desarrollar competencias profesionales y capacidades para el aprendizaje autónomo a través de una adecuación centrada en el estudiante y basada en el aprendizaje activo, a la vez que adquiere conocimientos propios de la disciplina de forma integral, se crea un ambiente propicio para el desarrollo de habilidades sociales y colaborativas.

Aunado a la Lúdica, como actividad que propicia el desarrollo de las aptitudes, las relaciones y el sentido del humor en las personas y predispone la atención del estudiante en motivación para su aprendizaje, generando experiencias significativas, dando como resultado el desarrollo de habilidades y fortalezas y además la disposición a trabajar en el aula, despertando ese interés a la curiosidad y creatividad, así como la convivencia en el aula (Pomare y Steele, 2018).Y que hoy en día se traspola a la gamificación Gabe Zichermann y Christopher Cunningham (2011) lo definen en su obra Gamification by Design como “un proceso relacionado con el pensamiento del jugador y las técnicas de juego para atraer a los usuarios y resolver problemas” (p.11). y para Karl. M. Kapp (2012) es, junto a Zichermann y Cunnigham, otro de los autores que estudian la gamificación señalan en su obra The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education que la gamificación es “la utilización de mecanismos, la estética y el uso del pensamiento, para atraer a las personas, incitar a la acción, promover el aprendizaje y resolver problemas” (p.9), conceptos que desde los conocimientos

¹ ABP: Aprendizaje Basado en Problemas. Método de aprendizaje iniciado en la Escuela de Medicina de la Universidad de McMaster (Canadá), basado en el principio de usar problemas como punto de partida para la adquisición e integración de los nuevos conocimientos.

que se adquieren en la asignatura de Cibernética y Computación están muy relacionados.

Sin olvidar las herramientas tecnológicas, en los programas de estudios solicitan incluir las Tecnologías de la Información y Comunicación TIC, las que engloban a cualquier dispositivo de comunicación (televisión, móviles, ordenadores, redes...) tecnológico que se utilice para la transmisión de información, han sido una herramienta indispensable para la educación, lo que dio pie a las Tecnologías de aprendizaje y conocimiento TAC, entonces es aquí donde se liga íntimamente las TICS y el aprendizaje, por lo tanto las TAC son todos aquellos recursos digitales orientados a la enseñanza que fomentan el aprendizaje, donde el rol del docente, es planear y seleccionar aquellas herramientas digitales más adecuadas para su uso en el aula con el fin de lograr el aprendizaje en el estudiante, más sin embargo el desarrollo de tantas aplicaciones en la web crea un nuevo término Tecnología del Empoderamiento y la Participación TEP, lo que permite al estudiante a interactuar y crear contenido propio agregando un modelo pedagógico que motiva la educación y, especialmente, fomenta el espíritu crítico y participativo entre los alumnos. (Latorre et. al., 2018).

“En el colegio, se reconoce que la educación del estudiante tiene una doble finalidad: personal y social. Ambas orientadas al desarrollo armónico del alumno y al mejoramiento de la sociedad en la que se desenvuelve” (El modelo educativo del CCH, p.2), con base en lo anterior, alcanzamos el tercer principio del colegio, el cual refiere al aprender a ser, lo que representa para el alumno, la afirmación de valores personales aplicados en su vida diaria, en cuyo caso, el profesor coadyuva a la adquisición de dichos valores, mediante su ejemplo, valores como la responsabilidad, la justicia, la honestidad, el compromiso, la solidaridad, el respeto, entre otros. La expresión de aprender a ser es en el campo social, que se encuentra relacionado con el aprender a convivir, es decir, la relación con sus pares, éste último principio es fundamentalmente importante para el desarrollo integral del estudiante, así como los anteriores el aprender a aprender, aprender a hacer, todo constituye parte esencial de la construcción del conocimiento, por lo que se relaciona con la teoría del constructivismo sociocultural de Vygostky.

La teoría Lev Vygostky menciona que la interacción social desempeña un papel formador y constructor, lo que para él representaba que algunas categorías de funciones mentales superiores (atención, aprendizaje, memoria, lenguaje,



emociones, conciencia, pensamiento y razonamiento) no podrían surgir y construirse en el proceso de desarrollo sin la contribución constructora de las interacciones sociales, en un sistema de relaciones sociales entre alumnos y docentes, entre los alumnos entre sí, entre la escuela y el entorno social.

Mientras que la construcción del conocimiento a partir de dicha interacción permite que lo que ya aprendió el estudiante se desarrolle y pase a una zona de desarrollo próximo, que a la mejor en un inicio debe ser directiva por parte del experto², mediante la creación de un sistema de apoyo denominado andamiaje y, posteriormente con los avances del estudiante en la adquisición o internalización del contenido con apoyo de las herramientas proporcionadas, para que el estudiante desarrolle la siguiente zona potencial de desarrollo, en otras palabras comprenda lo que no sabía y pueda modificar su entorno físico y social con lo aprendido.

Cabe señalar que el profesor buscará de forma similar como lo expresa Silvestre (1999), ser un proceso activo a la par del estudiante, vinculado con la vida, desarrollador de la inteligencia, que contribuya a la formación de cualidades y valores positivos de la personalidad, y al autoaprendizaje, expectativas cercana a la de los padres, además crear un proceso de interacción dinámico con principios didácticos dirigidos a un proceso de enseñanza-aprendizaje que instruya, eduque y desarrolle a través de varias pautas:

La realización de un diagnóstico Integral momento inicial donde se reconoce al estudiante como punto de inicio para transformar sus conocimientos sabiendo qué sabe, cómo lo hace, su comportamiento, la forma de aprendizaje, cómo piensa, cuáles son sus cualidades, su valoración y autoconocimiento, con todo este conocimiento más profundo de él, da la posibilidad al docente de un mejor panorama para la planeación y ejecución de la actividad docente.

Estructurar el proceso de enseñanza-aprendizaje hacia la búsqueda activa del conocimiento por el alumno, teniendo en cuenta las acciones a realizar por este en los momentos de orientación, ejecución y evaluación de la actividad y el uso de medios de enseñanza que favorezcan la actividad independiente y la búsqueda de información, desde la concepción de un sistema de actividades para la búsqueda y exploración del conocimiento por el alumno, tomando

² El experto puede ser un par, un profesor o alguien más con mayores conocimientos y tenga la posibilidad de transmitirlos.



posiciones analítico-reflexivas, que estimule y propicie el desarrollo del pensamiento, además que sean tareas portadoras de pequeñas metas que vayan impulsando el avance del estudiante de menor éxito, como de tareas de mayor nivel de exigencia que impulsen el desarrollo también de aquellos más saben, orientar la motivación hacia el objeto de la actividad de estudio y mantener su constancia, la motivación, ya que el estudiante motivado, interesado por la actividad, tendrá una disposición positiva por su realización, por alcanzar el resultado para lograr su actuación independiente e incluso de forma autodidáctica (Luna, 2015).



La instrucción escolar debe preocuparse menos por las conductas, conocimientos fosilizados o automatizados y más por aquellos en procesos de cambio, por lo tanto, el profesor aunque ya no es el centro del aprendizaje es el que crea estrategias innovadoras que coadyuven al aprendizaje del estudiante de una manera mucha más dinámica, motivador y atractiva para el estudiante, lo que se verá reflejado en su capacidad de actuar, hacer por la vida.

Después de un recorrido sobre algunas teorías, modelos, paradigmas pedagógicos que confluyen en la forma de enseñar y aprender, que integran intrínsecamente el modelo educativo del Colegio, se puede observar que los docentes toman decisiones de cómo van a enseñar con respecto a sus bases epistemológicas³ sobre el aprendizaje, de cómo les enseñaron, de una visión y misión institucional, un sistema educativo, etc., por ejemplo el docente que tiene convicción de que el conocimiento es construido de forma social y el estudiante es el centro del aprendizaje, en el que hay que generar aprendizajes significativos para que los asimile y acomode para que queden perenes en su conocimiento, probablemente el profesor seleccionará estrategias como foros, el trabajo colaborativo, donde el alumno investigará y realizará actividades que le permitan razonar, reflexionar de forma crítica; en cambio otro que piense que conocimiento debe ser memorístico porque está dentro de una escuela eclesiástica, donde todo se debe aprender como dogma, esté profesor se sentirá mejor presentando una cátedra magistral y en el examen deberán escribir las respuestas tal cual él las dijo en dicha cátedra.

³ La epistemología es la rama de la filosofía encargada del estudio del conocimiento; y el conocimiento tiene intrínsecamente al aprendizaje.

Entonces el sistema educativo de la UNAM, así como los docentes de cada una de las áreas que constituyen el Colegio:

“Se han dado a la tarea de diseñar estrategias didácticas de enseñanza aprendizaje pertinentes que aporten los conocimientos idóneos, las herramientas de estudio necesarias para formar un alumno competente, sujeto de la cultura y no un mero receptor que no solo debe comprender los conocimientos sino juzgarlos, relacionarlos con su propia experiencia y realidad, asimilarlos crítica y personalmente y, si fuera el caso, trascenderlos, relacionarlos o sustituirlos por otros mejor fundados, una formación no solo en la esfera del conocimiento, sino en valores humanos, éticos, cívicos y de sensibilidad estética” (Bazán, 1996).



b) Información en que basa el Marco Teórico

Después de un recorrido sobre algunas teorías, modelos, paradigmas pedagógicos que confluyen en la forma de enseñar y aprender, que integran intrínsecamente el modelo educativo del Colegio, los integrantes del Seminario Institucional consideramos que lo dicho por Ericksen “El aprendizaje real en la clase depende de la habilidad del profesor para mantener y mejorar la motivación que tengan los estudiantes al inicio del curso. Sea cual sea el nivel motivación que traen los estudiantes, será cambiado, a mejor, por lo que ocurre en el aula”, es una gran verdad, que debemos tener en cuenta en nuestro quehacer docente.

Muchos de nuestros alumnos tienen años de experiencia en clases en las que se les ha obligado a estar sentados, callados, escuchando. Para ello el profesor era la fuente del conocimiento, de manera que el aprendizaje era algo que se inyectaba en algún momento sin la participación de su conciencia.

La realidad en la que se encuentra inmerso el docente del colegio es que lo importante no es la enseñanza, sino lo que los alumnos aprenden. La calidad del aprendizaje está relacionada directamente, aunque no de manera exclusiva, con la calidad de la enseñanza. Una de las mejores maneras de mejorar el aprendizaje es mejorar la enseñanza.

El aprendizaje real en la clase depende de la habilidad del profesor para incentivar del estudiante. Pero no existe una fórmula ni remedio para

incentivarlos. Muchos factores afectan, como por ejemplo el interés en la materia, la percepción de su utilidad, la paciencia del alumno, y no todos los estudiantes se comportan de igual manera, lo que sí está claro es que los estudiantes incentivados son más receptivos y aprenden más, que la incentivación tiene una influencia importantísima en el aprendizaje.



Lo que sí parece ser cierto es que la mayoría de los estudiantes responden de una manera positiva a una asignatura bien organizada, enseñada por un docente entusiasta que tiene un interés destacado en los estudiantes y en lo que aprenden. Si queremos que aprendan, debemos crear condiciones que promuevan la incentivación.

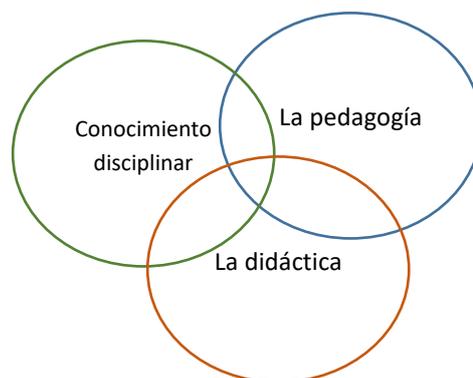
Con el propósito de incentivar a nuestros estudiantes, debíamos continuamente comentarles que tienen la capacidad necesaria para desarrollar una actividad y que pueden hacerlo bien, ayudarles a sentirse miembros valorados de una comunidad que aprende, crear en la clase una atmosfera abierta y positiva, si somos profesores, nuestro trabajo podría decirse que consiste en enseñar, pero tal vez debiéramos decir que se trata de que nuestros alumnos aprendan. Es decir, nosotros les vamos a proporcionar ladrillos para que ellos construyan la casa del conocimiento, el docente no puede construir la casa, el alumno es el que debe de aprender, ahí ésta la palabra aprendizaje, pero claramente el docente es el responsable que se construya mejor o peor esa casa del conocimiento.

En consecuencia, como ya se comentó anteriormente, nuestro modelo educativo se fundamenta en diversas teorías, modelos, paradigmas pedagógicos que confluyen en la forma de enseñar y aprender, así pues, el conjunto de estrategias de enseñanza aprendizaje desarrolladas por los integrantes del Seminario Institucional se fundamentan en las diferentes teorías del aprendizaje, pero teniendo siempre presente lo que establece el modelo educativo del Colegio de Ciencias y Humanidades y considerando dos cuestiones fundamentales, que nuestros alumnos aprendan, y que el docente proporcione herramientas claras y suficientes que propicien dicho aprendizaje y por ende la base de la información se sustenta con la información aquí citada y puesta en las referencias de ésta propuesta educativa.

d) La contribución disciplinaria, didáctica y pedagógica para el mejoramiento de la docencia.

En esta sección lo que pretendemos es identificar cual es la contribución disciplinaria, didáctica y pedagógica de nuestra propuesta educativa para el mejoramiento de la docencia, primero hay que definir que es conocimiento didáctico del contenido desde la definición de Shulman en 1987 (Acevedo, 2009; Medina y Jarauta, 2013) quien menciona que es, el conocimiento específico que poseen los docentes expertos, y se trata de una combinación adecuada entre el conocimiento de la materia a enseñar y el conocimiento pedagógico y didáctico para promover en el estudiante la comprensión y adquisición del aprendizaje, por ejemplo Medina y Jarauta (2013) mencionan que se puede observar en la experiencia de los docente que llevan dando la asignatura y que además tienen el dominio del contenido por la forma en que adecuan las estrategias de enseñanza en las que integra la disciplina, el saber pedagógico para darle sentido didáctico a la enseñanza.

Para comprenderlo mejor el esquema se ve la integración entre los conocimientos del profesor sobre tema a enseñar y la didáctica necesaria para enseñarlo de manera eficaz en un contexto determinado desde un enfoque pedagógico.



Sabiendo que la Pedagogía es la Ciencia que estudia la educación y la enseñanza (Teoría educativa), su objeto de estudio la educación. (Diccionario pedagógico AMEI WAECE).



Didáctica: es una rama de la pedagogía que se encarga de la forma de enseñar a través de métodos, técnicas o estrategias; en otras palabras es la práctica docente. (Diccionario pedagógico AMEI WAECE).

Conocimiento disciplinar Zambrano (2006) lo describe como tres saberes que tienen lugar en la práctica y están vinculados con tres preguntas ¿Qué sé?, ¿cómo comunico lo que sé? y ¿cómo aprende con la retroalimentación del proceso enseñanza-aprendizaje? Dichos saberes que permiten reflexionar sobre su propio ser, su identidad, la especificidad de la profesión, la práctica, la vocación conjugando las prácticas institucionales que se transforman en saber, para trascender desde la enseñanza para que el estudiante aprenda.

Entonces después de la revisión de distintas teorías, metodologías, paradigmas sobre el proceso enseñanza-aprendizaje donde el centro del aprendizaje es el estudiante y **el profesor** que enseña tiene un gran **reto y responsabilidad**: guiar la predisposición del estudiante hacia el aprendizaje relacionándolo con los contenidos, promover los procesos de construcción de conocimiento a través del descubrimiento y que a su vez sea significativo, de forma activa, colaborativa, intencional, conversacional, contextualizado y reflexiva, siendo el estudiante consciente del proceso que lo llevó a formular el nuevo conocimiento, pero además vincular al estudiante con la vida, desarrollador de la inteligencia, que contribuya a la formación de cualidades y valores positivos de la personalidad, y al autoaprendizaje; desde los momentos de diagnóstico, orientación, ejecución y evaluación, donde identifica como el estudiante está comprendiendo, obtiene la habilidades así como si saber cuándo, y por qué utilizar dicho conocimiento lo anterior son resultantes del aprendizaje, para que con estos resultados pueda adecuar, remarcar, corregir y/o profundizar el área de oportunidad a desarrollar con el uso de medios de enseñanza que favorezcan el aprendizaje significativo como: preguntas problematizadoras, la solución de problemas e incluso el conflicto cognitivo con ayuda de mediadores como la discusión en clase, foros, redes sociales, debate, el juego de roles y el descubrimiento guiado.

Entonces aunque el profesor no es el centro del aprendizaje es el que crea, modifica y readecua estrategias didácticas que derivan del programa de estudios de las asignaturas y que coadyuvan al aprendizaje del estudiante, en todo lo planteado anteriormente y referido en el marco teórico.



Entonces se vuelve a insistir la importancia del diseño de las estrategias didácticas sabiendo que son el guion a seguir dentro de la escena de enseñanza del profesor, el profesor necesita contestar la siguiente pregunta ¿qué necesito conocer y saber hacer para transmitir los conocimientos de forma adecuada y coherente y los estudiantes puedan aprender? (Conocimiento disciplinar pedagógico y didáctico).



Y para contestar la pregunta y poderla plasmar en dichas estrategias didácticas el profesor debe seguir renovando sus conocimientos y habilidades disciplinares conocimiento adquirido por su formación formal y algunas veces la experiencia laboral, sin embargo, algunos no cuentan con la formación pedagógica pero la institución se preocupa por impartir cursos que den cuenta de ella junto con la técnicas didácticas que se necesitan para poder lograr esta noble y ardua tarea de la enseñanza-aprendizaje.

Concluyendo los conocimientos disciplinares los otorgará el profesor con la profesión estudiada y la experiencia laboral sobre los contenidos de la asignatura, y es importante remarcar sólo es **una propuesta de estrategias didácticas** bajo las teorías pedagógicas y didácticas ya explicadas con antelación que guiaran el que hacer del profesor a través del programa de estudios de las asignatura de Cibernética y Computación I y Cibernética y Computación II que a su vez tienen como base el Modelo educativo del Colegio que a continuación se presentaran como propuesta de aplicación, las cuales cubren la mayoría de los aprendizajes de cada unidad del programa de estudios de cada asignatura.

Referencias

- Acevedo, J. (2009). Conocimiento didáctico del contenido para la enseñanza de la ciencia (I): El marco Teórico. Revista Eureka sobre enseñanza y divulgación de las Ciencias, 6(2), 164-189. Recuperado de <https://revistas.uca.es/index.php/eureka/article/view/3698/3286>
- Asociación Mundial de Educadores Infantiles AMEI WAECE (2022) Diccionario pedagógico. Recuperado de: <http://www.waece.org/diccionario/index.php>
- Ausubel, D.P. (1968). Educational psychology: a cognitive view. New York, Holt, Rinehart and Winston

- CCH. (2021). "Modelo educativo del Colegio de Ciencias y Humanidades". México: CCH/UNAM. Recuperado de: <https://www.cch.unam.mx/modelo>
- Gaceta UNAM (1971) "Se crea el Colegio de Ciencias y Humanidades". *Gaceta UNAM, tercera época vol. III (número extraordinario)* México: UNAM
- Huerta de Soto (2022) El método didáctico e investigador en el método para transmitir los conocimientos 279. España. Recuperado de: <https://www.jesushuertadesoto.com/articulos/articulos-en-espanol/proyecto-docente/3-el-metodo-para-transmitir-los-conocimientos-279/#ocho>
- Latorre, E. L., Castro, K. P. y Potes, I. D. (2018). Las TIC, las TAC y las TEP: innovación educativa en la era conceptual. Colombia: Universidad Sergio Arboleda.
- Flavell J. H. (1992). La psicología evolutiva de Jean Peaget. España: Paidós.
- González S., D. (2000). «Una concepción integradora del aprendizaje humano», en *Revista Cubana de Psicología*, v.17, n.2
- Kapp, K. (2012). *The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education*. San Francisco: John Wiley & Sons.
- Luna, C. (2015). El futuro del aprendizaje 2 ¿Qué tipo de aprendizaje se necesita en el siglo XXI?. Investigación y prospectiva en educación documentos de trabajos. UNESCO
- Medina, J. L. y Jarauta, B. (2013). Análisis del Conocimiento Didáctico del Contenido de tres profesores universitarios. *Revista de Educación*, 360, 600-623. Recuperado de <http://www.mecd.gob.es/dctm/revista-de-educacion/articulosre360/re36027.pdf?documentId=0901e72b814a77fb>
- Pomare, K. A. y Steele, J. O. (2018). La didáctica lúdica, mediadora en el aprendizaje significativo [tesis de maestría]. Colombia: UCC
- Sams, A. y Bergaman, J. (2014) Dale la vuelta a tu clase, lleva tu clase a cada estudiante, en cualquier momento y cualquier lugar. España: SM
- Shulman, L. (1987). Knowledge and teaching: foundations of the New Reform. *Harvard Educational Review*, 57(1), 1-22.
- Zichermann, G. y Cunningham, C. (2011). *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. Cambridge, MA: O'Reilly Media.
- Zambrano, A. (2006). Tres tipos de saber del profesor y competencias: una relación compleja. *Educere [online].10(33), pp.225-232* Colombia: Universidad de Santiago de Cali.





**SEMINARIO CENTRAL PROPUESTA EDUCATIVA PARA APOYAR A LA
ACTUALIZACIÓN DE LOS PROGRAMAS DE ESTUDIO DE CIBERNÉTICA
Y COMPUTACIÓN I-II
Estrategias de Aprendizaje**

Cibernética y Computación I

Unidad 1: La Cibernética

Propósito: Al finalizar la unidad el alumno:

Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.



ESTRATEGIA DIDÁCTICA 1

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación I
SEMESTRE ESCOLAR	Semestre escolar: Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 10 de enero de 2022
NO DE ALUMNOS	Número de alumnos: 25

II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad I La Cibernética
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.

APRENDIZAJE(S)	Comprende la influencia de la cibernética en el desarrollo de la ciencia.
TEMA(S)	<ul style="list-style-type: none"> Definición del concepto de cibernética. Antecedentes de la cibernética. Relación de la cibernética con otras ciencias. Aplicaciones de la cibernética en la actualidad.



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> Aula en el laboratorio de cómputo Aula en condiciones adecuadas de ventilación y limpieza Espacio entre computadoras de al menos 1 metro Pizarrón Plumones Computadora con acceso a internet Forma de trabajar individual
DESARROLLO Y ACTIVIDADES	<p>Inicio (20 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> ✓ Comenta a los alumnos del grupo los aprendizajes a estudiar relativos a: ✓ Definición del concepto de cibernética. ✓ Antecedentes de la cibernética. ✓ Relación de la cibernética con otras ciencias. ✓ Aplicaciones de la cibernética en la actualidad. ✓ Comenta brevemente los antecedentes de la cibernética, su relación con otras ciencias, y su vertiginosa evolución. <p>Desarrollo (80 minutos)</p> <p>Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:</p> <ul style="list-style-type: none"> • Empleando el internet realiza la búsqueda de los siguientes conceptos (20 minutos). ✓ Definición del concepto de cibernética. ✓ Su relación con otras ciencias ✓ Aplicaciones de la cibernética.

- ✓ Con base en la búsqueda realizada y bajo la supervisión del profesor, el alumno llena las siguientes tablas (40 minutos).

Definición y antecedentes de la cibernética.



La cibernética y su relación con otras ciencias.

Aplicaciones de la cibernética

- ✓ **Aplicación 1.**
 - Descripción.

 - Funcionamiento

- ✓ **Aplicación 2.**
 - Descripción.

 - Funcionamiento

✓ **Aplicación 3.**

- Descripción.

- Funcionamiento



- El profesor proporciona las siguientes ligas, solicita al estudiante que vea los siguientes videos y responda las siguientes preguntas (20 minutos).

✓ <https://www.youtube.com/watch?v=t2GgF5BWPkl>
el robot Da Vinci

✓ https://www.youtube.com/watch?v=YO0V_nAnYow
un corredor sin piernas

Con relación a los videos observados se le pide al alumno que conteste lo siguiente.

- ¿Ciencias que intervienen?

- ¿En estos dispositivos interviene la electrónica?

Cierre (20 minutos)

1. Bajo la guía y supervisión del profesor, en reunión plenaria solicita a los alumnos del grupo que expongan su trabajo.

El Alumno:

- ✓ Expone el trabajo realizado en torno a
 - Definición y antecedentes de la cibernética.
 - La cibernética y su relación con otras ciencias.
 - Aplicaciones de la cibernética.

Los alumnos del grupo, en forma crítica, ordenada y responsable, comparten sus comentarios y retroalimentan el trabajo de sus compañeros.

- Como parte de su evaluación, el profesor solicita a los alumnos del grupo que, de acuerdo a su área de interés busquen un video sobre la cibernética y sus aplicaciones y empleando un documento en Word escriben una reseña que entregarán en la clase siguiente.



Lista de cotejo

Concepto	Si	No
El documento tiene carátula.		
En la carátula del documento se muestra el título del video.		
El alumno indica la razón del por qué selecciono el video.		
La reseña escrita del video es adecuada y se aprecia su relación con la cibernética.		
La reseña no tiene errores ortográficos		
Total		

ESTRATEGIA DIDÁCTICA 2

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
--------------------	-------------------------

ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación I
SEMESTRE ESCOLAR	Semestre escolar: Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 12 de enero de 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad I La Cibernética
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.
APRENDIZAJE(S)	Describe el trabajo científico sobre la cibernética de Norbert Wiener, Arturo Rosenblueth, Claude Shannon, entre otros.
TEMA(S)	Trabajo científico sobre la cibernética de <ul style="list-style-type: none"> • Norbert Wiener. • Arturo Rosenblueth. • Claude Shannon • Otros

III.ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón

	<ul style="list-style-type: none"> • Plumeros • Computadora con acceso a internet • Forma de trabajar individual
<p>DESARROLLO Y ACTIVIDADES</p>	<p>Inicio (10 minutos) El profesor:</p> <ul style="list-style-type: none"> • Comenta a los alumnos del grupo los aprendizajes a estudiar relativos a el trabajo científico sobre la cibernética de: <ul style="list-style-type: none"> ✓ Norbert Wiener. ✓ Arturo Rosenblueth. ✓ Claude Shannon ✓ Otros • Comenta brevemente el trabajo en torno a la cibernética realizado por algunos de los notables científicos que intervinieron en su evolución. <p>Desarrollo (70 minutos) Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:</p> <ul style="list-style-type: none"> • Empleando el internet realiza la búsqueda de los siguientes conceptos (20 minutos). <p>Trabajo científico en torno a la cibernética realizado por:</p> <ul style="list-style-type: none"> ✓ Norbert Wiener ✓ Arturo Rosenbluth ✓ Claude Shannon ✓ John Von Neumann ✓ William Ross Ashby <ul style="list-style-type: none"> ▪ Con base en la búsqueda realizada y bajo la supervisión del profesor, el alumno llena las siguientes tablas (50 minutos). <p>Aportaciones Norbert Wiener.</p>





Aportaciones Arturo Rosenbluth

Aportaciones Claude Shannon

Entre otros (se puede citar cualquier personaje adicional que desde el punto de vista del estudiante se considere importante).

John Von Neumann

William Ross Arhby

Manuel Sandoval Vallarta

George Boole

Cierre (40 minutos)

3. Como parte de su evaluación, el profesor plantea a los alumnos del grupo que usando algún software elaboren una línea de tiempo, en donde deben considerar a los autores que se enlistan a continuación y sus aportaciones a la cibernética.

André Marie Ampere, George Boole, Gottfried Wilhelm Leibniz, Claude E. Shannon, Alan M. Turing, John Von Neumann, Norbert Wiener, Manuel Sandoval Vallarta, Arturo Rosenbluth Stearns

4. Bajo la guía y supervisión del profesor, en reunión plenaria solicita a los alumnos del grupo que expongan su trabajo.

El Alumno:

- ✓ Presenta su línea del tiempo en donde se observa trabajo realizado en torno.

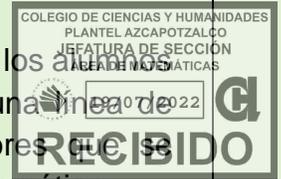
Trabajo científico en torno a la cibernética realizado por:

- Norbert Wiener
- Arturo Rosenbluth
- Claude Shannon
- Otros

Los alumnos del grupo, en forma crítica, ordenada y responsable, comparten sus comentarios y retroalimentan el trabajo de sus compañeros.

Lista de cotejo

Concepto	Si	No
En la línea del tiempo los autores son presentados en forma correcta.		



	Se describe en forma correcta y adecuada las aportaciones realizadas de los diferentes científicos a la cibernética.		
	Los elementos empleados en la línea del tiempo son adecuados y permiten visualizar las aportaciones realizadas de los diferentes científicos.		
	Las fechas presentadas en la línea del tiempo son acordes al trabajo realizado por cada uno de los científicos.		
	Total		



I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación I
SEMESTRE ESCOLAR	Semestre escolar: Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 17 de enero de 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad I La Cibernética
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.
APRENDIZAJE(S)	Comprende los componentes de un sistema.
TEMA(S)	Sistemas: <ul style="list-style-type: none">• Concepto.• Elementos.• Ambiente.• Clasificación.

III.ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none">• Aula en el laboratorio de cómputo• Aula en condiciones adecuadas de ventilación y limpieza• Espacio entre computadoras de al menos 1 metro• Pizarrón

	<ul style="list-style-type: none"> • Plumeros • Computadora con acceso a internet • Forma de trabajar individual
<p>DESARROLLO Y ACTIVIDADES</p>	<p>Inicio (10 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> ✓ Comenta a los alumnos del grupo los aprendizajes a estudiar relativos a sistemas: ✓ Concepto. ✓ Elementos. ✓ Ambiente. ✓ Clasificación. ✓ Comenta brevemente conceptos relativos a: concepto de sistema, elementos de un sistema, ambiente de un sistema y su clasificación. <p>Desarrollo (80 minutos)</p> <p>Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:</p> <ul style="list-style-type: none"> ✓ Empleando el internet realiza la búsqueda de los siguientes conceptos (20 minutos). ✓ Concepto de sistema ✓ Elementos de un sistema ✓ Ambiente y clasificación de los sistemas. ✓ Con base en la búsqueda realizada y bajo la supervisión del profesor, el alumno llena las siguientes tablas (40 minutos). <div data-bbox="462 1549 1437 1921" style="border: 1px solid black; padding: 10px; margin-top: 20px;"> <p>Conceptos de un sistema.</p> </div>



Elementos de un sistema



Propuestas de 2 sistemas

El alumno de acuerdo a su área de interés elige 2 sistemas, explica las partes que lo conforman (entradas, salidas), y el funcionamiento del mismo (proceso).

Sistema 1

Sistema 2

Clasificación de los sistemas

- ✓ El alumno clasifica los sistemas en Abierto o cerrado y fundamenta su respuesta (20 minutos).

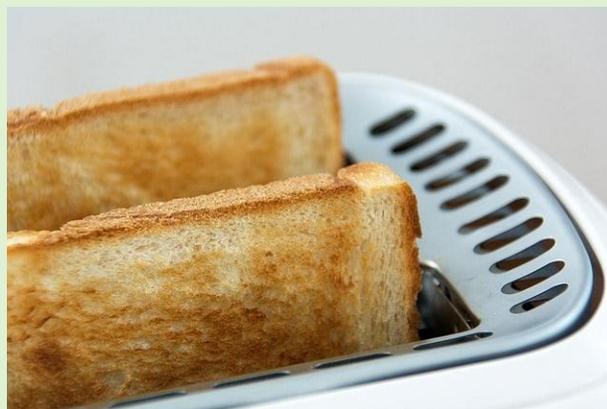
Sistema	Clasificación	Fundamentación
Careta de buzo		
El cuerpo humano		
Termo		
Olla de agua hirviendo		
Hoguera		
Motor a combustión		
Planta		
Reloj		
Sauna		
Caja fuerte		
Sol		
Planeta tierra		
Batería		
Refrigerador		
Bombilla		



Cierre (30 minutos)

- ✓ El profesor propone a los alumnos del grupo el siguiente sistema.

- ✓ Tostador de pan



Le solicita al alumno que identifique y describa.

Entrada

Proceso

Salida

2. Bajo la guía y supervisión del profesor, en reunión plenaria solicita a los alumnos del grupo que expongan su trabajo.

El Alumno:

- ✓ Expone el trabajo realizado en torno a
 - Concepto de sistema
 - Elementos de un sistema
 - Ambiente y clasificación de los sistemas.

Los alumnos del grupo, en forma crítica, ordenada y responsable, comparten sus comentarios y retroalimentan el trabajo de sus compañeros.

Lista de cotejo

Concepto	Si	No
El alumno estableció las entradas del sistema.		
El alumno estableció las salidas del sistema.		
El alumno estableció en forma clara y concreto el funcionamiento del sistema.		
El alumno indica la clasificación del sistema.		
Total		



ESTRATEGIA DIDÁCTICA 4

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
--------------------	-------------------------

ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación I
SEMESTRE ESCOLAR	Semestre escolar: Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 19 de enero de 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad I La Cibernética
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.
APRENDIZAJE(S)	Comprende los componentes esenciales de un sistema de control
TEMA(S)	Sistemas de control: <ul style="list-style-type: none"> • Lazo abierto. • Lazo cerrado. • Retroalimentación.

III.ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Forma de trabajar individual

**DESARROLLO Y
ACTIVIDADES**

Inicio (10 minutos)

El profesor:

- Comenta a los alumnos del grupo los aprendizajes relativos a:

Sistemas de control:

- ✓ Lazo abierto.
- ✓ Lazo cerrado.
- ✓ Retroalimentación.

- Comenta brevemente conceptos relativos a los sistemas de control y la retroalimentación.

Desarrollo (70 minutos)

Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

- Empleando el internet realiza la búsqueda de los siguientes conceptos (20 minutos).

Sistemas de control:

- ✓ Lazo abierto.
- ✓ Lazo cerrado.
- ✓ Retroalimentación.

- Con base en la búsqueda realizada y bajo la supervisión del profesor, el alumno llena las siguientes tablas (40 minutos).

¿Qué es un Sistema de control?

Elementos de un sistema de control





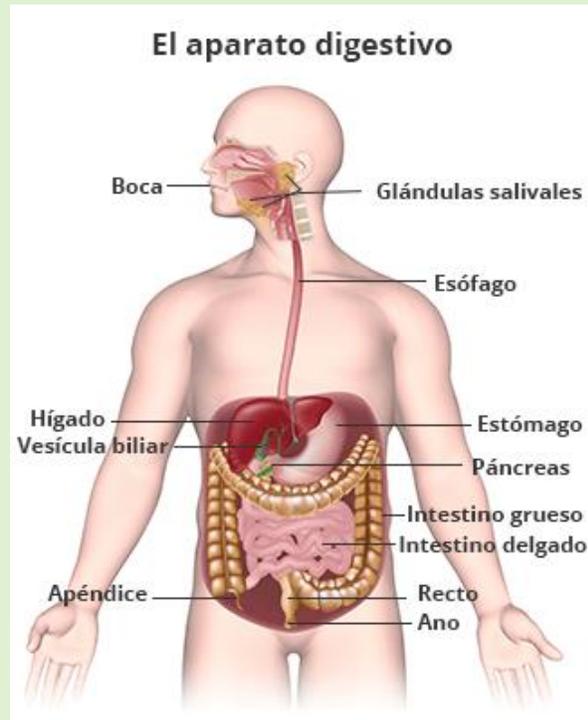
Sistema de control de lazo abierto

Ejemplo

Sistema de control de lazo cerrado

Ejemplo

- El profesor proporciona el siguiente sistema.



Con referencia al sistema proporcionado, solicita a los alumnos que expresen lo siguiente (10 minutos).

Entradas

Proceso

Salidas

Cierre (40 minutos)

Como parte de la evaluación, el profesor solicita a los alumnos del grupo que resuelvan lo siguiente.

	<p>1. Relaciona las siguientes columnas, escribiendo en el paréntesis la letra de la respuesta correcta.</p>	
<p>a) Intercambian energía (calor, trabajo) con el exterior, pero nunca materia su masa permanece intacta.</p> <p>b) Matemático y filósofo norteamericano que sentó en 1948 las bases de la Cibernética, teoría sobre el control y la comunicación en máquinas y animales.</p> <p>c) En 1936 publicó el artículo “Sobre números computables, con una aplicación al Entscheidungs problema” (traducible como “problema de decisión”), que resultó ser el origen de la informática teórica. En el que definía que era computable y que no lo era.</p> <p>d) Nació en la agonía del siglo XIX, en octubre de 1900, en Chihuahua. A los 17 años ingresó a la Escuela Nacional de Medicina.</p> <p>e) No se mide la salida ni se realimenta para compararla con la entrada.</p> <p>f) Señal generada por algún tipo de fenómeno electromagnético que codifica el contenido de la misma en término de algunas magnitudes que representan valores discretos.</p> <p>g) Fue un matemático e ingeniero eléctrico recordado como el padre</p>	<p>() Señal analógica</p> <p>() Arturo Rosenbluth</p> <p>() Sensor</p> <p>() Claude Shannon</p> <p>() Sistemas artificiales</p> <p>() Sistema compuestos</p> <p>() Norbert Wiener</p> <p>() Retroalimentación</p> <p>() Señal digital</p> <p>() Sistema control lazo abierto</p> <p>() Sistema de control de lazo cerrado</p> <p>() Sistema cerrado</p>	



de la teoría de la información (hoy internet).

- h)** Son aquellos que fueron logrados por la intervención directa de la raza humana.
- i)** Los resultados obtenidos de una tarea o actividad son reintroducidos nuevamente en el sistema con el fin de controlar y optimizar su comportamiento.
- j)** Ocurren cuándo en un sistema natural existe la participación de una fuerza externa manipulada por el ser humano de forma directa ó indirecta.
- k)** Permiten conocer los valores de las variables medidas del sistema (salidas).



Valor 1 punto

- 2.** Una familia habita en una zona rural en donde escasea agua, la forma de obtener este preciado líquido es a través del agua de lluvia, diseña un sistema que le permita a esta familia captar el agua y solucionar en la medida de lo posible el problema.





CAPTACIÓN DE AGUA DE LLUVIA

Entradas

Proceso

¿hay control?

Salidas

Valor 2 puntos

ESTRATEGIA DIDÁCTICA 5

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación I
SEMESTRE ESCOLAR	Semestre escolar: Quinto semestre

PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades	
FECHA DE ELABORACIÓN	22 enero 2022	
NO DE ALUMNOS	Número de alumnos: 25	

II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad I La Cibernética
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.
APRENDIZAJE(S)	Comprende el concepto y la importancia del modelo.
TEMA(S)	Modelos: <ul style="list-style-type: none"> • Concepto. • Tipos. <ul style="list-style-type: none"> ○ Naturales y artificiales. ○ Analógicos y digitales.- ○ Matemáticos. ○ Conceptuales. • Relación

III.ORGANIZACIÓN DE ACTIVIDADES

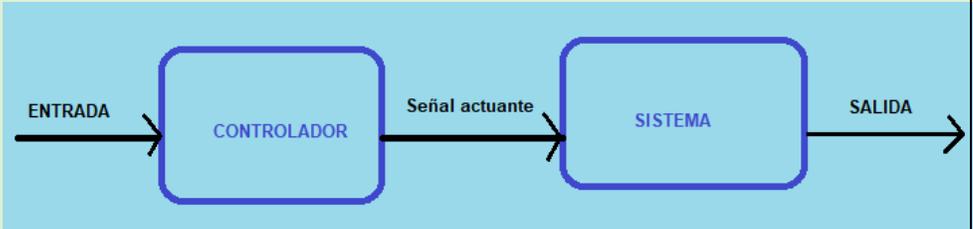
TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet

DESARROLLO Y ACTIVIDADES

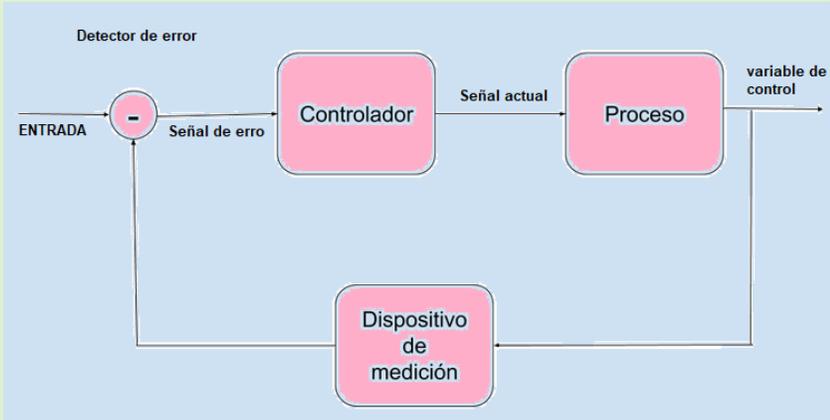
- Forma de trabajar individual

Inicio (30 minutos)

- El profesor realiza un repaso menciona cuales son las características de los sistema abierto, entonces se sabe que son sistemas de control en los que la salida o resultado de proceso no tiene ningún efecto sobre la acción de control, es decir, en un sistema de control abierto la salida no se mide. Estos sistemas generalmente son controlados manualmente. Por ejemplo, cuando enciendes con la llave o interruptor el motor de un carro para ir. El carro por sí solo no se “prende” o se apaga; tiene un controlador (el interruptor), pero necesita ser encendido por un usuario.



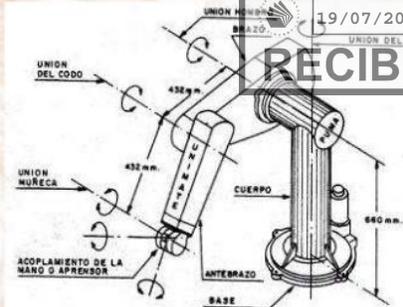
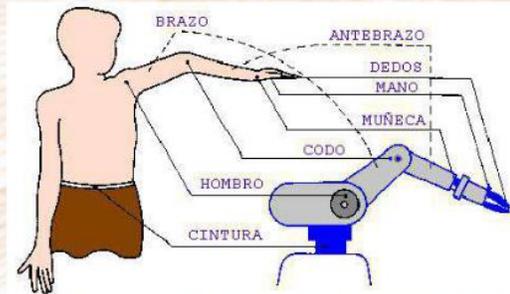
- Sistemas de lazo cerrado. Son aquellos en los que la señal de salida tiene efecto sobre la acción de control, es decir, en principio es un sistema de lazo abierto en el que la salida se “retroalimenta”. Por lo tanto, el sistema se “cierra”. Estos sistemas corresponden a las máquinas automáticas. La retroalimentación o medición de la salida se realiza por medio de un sensor.



El profesor explica la importancia de la representación de un sistema desde un MODELO:



El sistema robótico, está formado por los siguientes elementos: estructura mecánica, transmisiones, actuadores, sensores, elementos terminales y controlador.



Fuente: González, V. (2003). Estructura de un robot industrial. Componentes de un robot. [Imagen]. Disponible en: http://platea.pntic.mec.es/vgonzale/cyr_0204/ctrl_rob/robotica/sistema/morfologia.htm

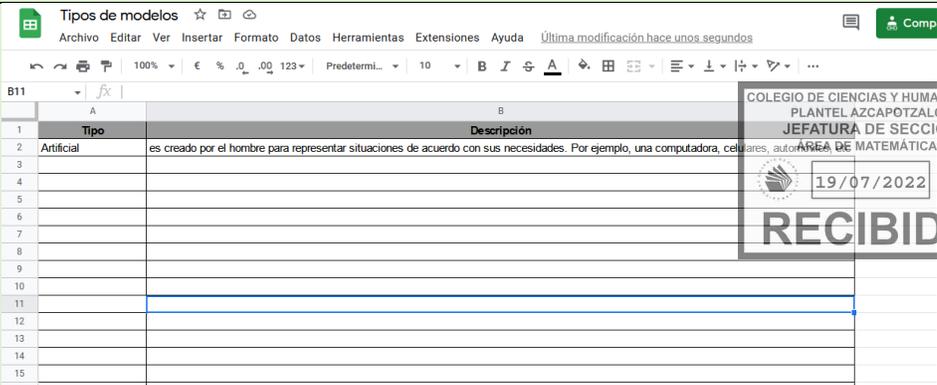
COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALCO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS
19/07/2022
RECIBIDO

Desarrollo (50 minutos)

- El profesor les pedirá a los estudiantes que contesten ¿Qué es para ellos un MODELO? para que lo llene en el siguiente espacio o lo exprese en el salón de clase. puede utilizar un Jamboard para que coloque de manera escrita con notas lo que piensa que es un modelo proporcionando la siguiente liga:
<https://jamboard.google.com/d/1z2y2JuBXQCDRjYzDtEBsYVpAdL4y3KFnTFFUBHnGTBk/edit?usp=sharing>

MODELO

Instrucciones: se les pide que realicen la búsqueda de un tipo de modelo e irán llenando la siguiente hoja electrónica en forma colaborativa <https://docs.google.com/spreadsheets/d/1eiPc-c6qg9hGTI45kzB9q31IO99aVAjwlWnkdUEvPw/edit?usp=sharing>, con el tipo de modelo encontrado, respetando el trabajo del otro.

	
	<p>Cierre (30 minutos)</p> <p>Los alumnos concluyen la sesión dando ejemplo de los sistemas encontrados y descritos en la hoja electrónica de cálculo.</p>

ESTRATEGIA DIDÁCTICA 6

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación I
SEMESTRE ESCOLAR	Semestre escolar: Quinto semestre

PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades	
FECHA DE ELABORACIÓN	Enero 2022	
NO DE ALUMNOS	Número de alumnos: 25	

II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad I La Cibernética
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.
APRENDIZAJE(S)	Desarrolla el modelo de un sistema
TEMA(S)	Elementos para modelar un sistema: <ul style="list-style-type: none"> • Entrada y salida. • Proceso.

III.ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Forma de trabajar individual

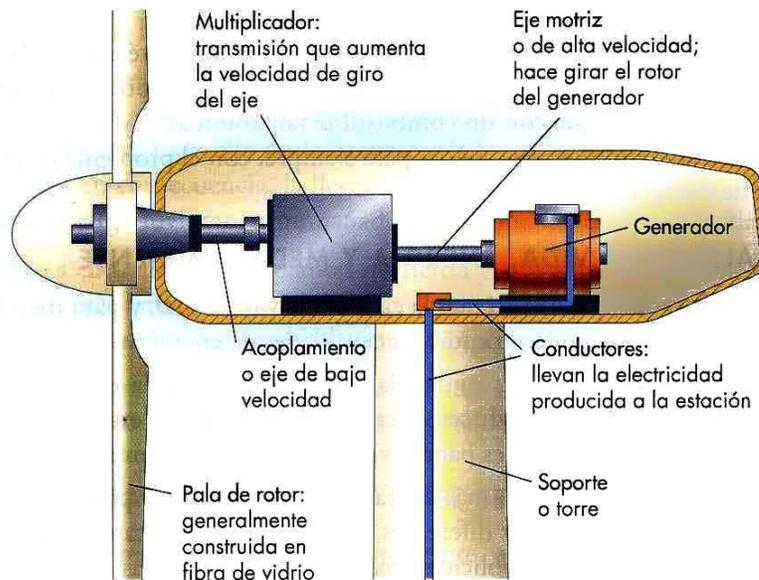
**DESARROLLO Y
ACTIVIDADES**

Inicio (20 minutos)

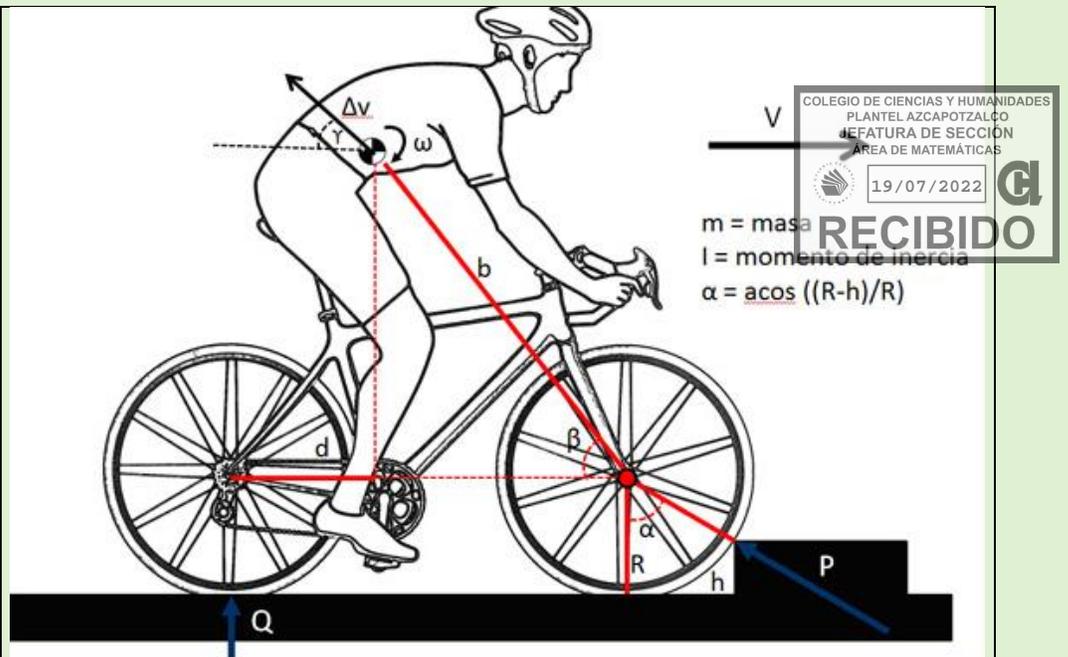
- El profesor menciona la importancia del aprendizaje a lograr en la sesión
- El profesor continua mostrando distintos tipos de modelos y les pide a los estudiantes que identifiquen cuáles son sus entradas, procesos y salidas.



PARTES DE UN GENERADOR EÓLICO



RF ELECTRONICS EN FACEBOOK



LINEA de Encajado JW-001-b

Linea completa de encajado



Desarrollo (40 minutos)

- Los estudiantes organizados en equipos de 3 o 4 integrantes realizarán una presentación con diapositivas usando (Genealy, Canva, PowerPoitn, etc.) donde elijarán un sistema al cual representarán a través de un modelo para identificar sus entradas procesos y salidas

Cierre (60 minutos)

Los equipos expondrán sus presentaciones y los compañeros podrán intervenir respetuosa y ordenadamente para retroalimentar el trabajo, donde el profesor enfatizará que en ese proceso también hay un sistema de comunicación el cual es de lazo cerrado y se necesita de la retroalimentación para poder mejorar nuestros trabajos.

Unidad 2: Circuitos lógicos

Propósito: Al finalizar la unidad el alumno:

Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.



ESTRATEGIA DIDÁCTICA 1

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTELE	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022

II. PROGRAMA

UNIDAD TEMÁTICA	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Convierte números entre sistemas de numeración binario, octal, decimal y hexadecimal.
TEMA(S)	Sistemas de numeración. <ul style="list-style-type: none">• Binario, octal, decimal y hexadecimal.

	<ul style="list-style-type: none"> • Conversiones numéricas entre los sistemas binario, octal, decimal y hexadecimal.
--	--



III. ESTRATEGIA

El alumno conoce los sistemas: binario, octal, decimal y hexadecimal y realiza conversiones numéricas.

IV. SECUENCIA

TIEMPO DIDÁCTICO	2 horas
DESARROLLO Y ACTIVIDADES	<p>Inicio</p> <p>Tiempo: 10 minutos</p> <ol style="list-style-type: none"> 1. El docente indica al grupo cual será el aprendizaje que se trabajará en esa sesión. 2. El docente da una introducción acerca de los sistemas posicionales y su importancia. <p>Desarrollo</p> <p>Tiempo: 80 minutos</p> <ol style="list-style-type: none"> 3. El docente explica el procedimiento para hacer las conversiones de sistema binario a sistema decimal y de decimal a binario. 4. El docente comparte con los alumnos los ejercicios que deben resolver de manera individual del sistema binario. 5. El alumno resuelve las conversiones en su libreta. 6. El docente explica el procedimiento para hacer las conversiones de sistema octal a sistema decimal y de decimal a octal. 7. El docente comparte con los alumnos los ejercicios que deben resolver de manera individual del sistema octal. 8. El alumno resuelve las conversiones en su libreta. 9. El docente explica el procedimiento para hacer las conversiones de sistema hexadecimal a sistema decimal y de decimal a hexadecimal. 10. El docente comparte con los alumnos los ejercicios que deben resolver de manera individual del sistema hexadecimal.

	<p>11. El alumno resuelve las conversiones en su libreta.</p> <p>Cierre 30 minutos</p> <p>12. El docente selecciona aleatoriamente a alumnos, que pasen al pizarrón o muestren por cámara, el procedimiento y resultado que obtuvieron de uno de los ejercicios.</p> <p>13. El alumno seleccionado comparte con el docente y compañeros su procedimiento y resultado.</p> <p>14. El docente da una retroalimentación al ejercicio que está presentando el alumno</p> <p>15. El alumno realiza una autoevaluación de sus ejercicios y hace la entrega de la actividad de forma presencial o en el aula asignada.</p> <p>16. El docente recibe las actividades, evalúa cada ejercicio y retroalimenta al alumno</p>
ORGANIZACIÓN	<p>La secuencia se llevará a cabo con actividades de forma individual.</p> <p>Actividad diseñada para 30 alumnos.</p>
MATERIALES Y RECURSOS DE APOYO	<p>Equipo de cómputo o smartphone.</p> <p>Internet</p> <p>Aula virtual para la materia de Cibernética y Computación I</p>
EVALUACIÓN	<p>Puntaje por ejercicio resuelto correctamente</p>



V. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS.	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación.</p> <p>Diseño digital, 3ra Edición (ollintec.com)</p> <p>Vasconcelos, J. (2018). Introducción a la Computación. México: Grupo Editorial.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación.</p> <p>Diseño digital, 3ra Edición (ollintec.com)</p>
COMENTARIOS ADICIONALES	<p>Esta estrategia puede ser llevada a cabo de manera presencial o en línea.</p>

ESTRATEGIA DIDÁCTICA 2

III. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022



IV. PROGRAMA

UNIDAD TEMÁTICA	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Realiza operaciones aritméticas con el sistema de numeración binario.
TEMA(S)	Aritmética del sistema de numeración binario.

III. ESTRATEGIA

El profesor explica el procedimiento de suma, resta multiplicación y división en binario

IV. SECUENCIA

TIEMPO DIDÁCTICO	4 horas
DESARROLLO Y ACTIVIDADES	<div data-bbox="1247 289 1511 464" style="border: 1px solid black; padding: 5px; text-align: center;"> COLEGIO DE CIENCIAS Y HUMANIDADES PLANTEL AZCAPOTZALCO JEFATURA DE SECCIÓN ÁREA DE MATEMÁTICAS  19/07/2022  RECIBIDO </div> <p>(Clase 1) Inicio</p> <p>Tiempo: 10 minutos</p> <ol style="list-style-type: none"> 1. El docente indica al grupo cual será el aprendizaje que se trabajará en esa sesión. 2. Menciona el trasfondo de las operaciones aritméticas con diferentes bases a manera de introducción. <p>Desarrollo</p> <p>Tiempo: 100 minutos</p> <ol style="list-style-type: none"> 3. El docente explica la suma binaria comparándola con una suma en decimal para denotar las similitudes y diferencias 4. El docente resuelve 3 sumas diferentes y las explica paso a paso 5. El docente deja a los alumnos 10 sumas a que las resuelvan individualmente mientras pasa a revisar y resolver dudas individualmente alumno por alumno. 6. El docente pide a los alumnos que vayan terminando que anoten el resultado de las sumas en el pizarrón para que el grupo compruebe el resultado. Si es necesario se anotan más operaciones hasta que el docente compruebe que el aprendizaje sea asimilado 7. El docente realiza el mismo procedimiento desde el paso 3, pero ahora con restas binarias <p style="text-align: center;">Cierre 10 minutos</p> <ol style="list-style-type: none"> 8. El docente deja de tarea 10 sumas y 10 restas, adicionales a las realizadas en clase cuyo grado de complejidad se incrementa. <p>Clase 2 (Inicio) 110 minutos</p> <ol style="list-style-type: none"> 1. Se ocupan los primeros 10 minutos de la clase para revisar las operaciones dejadas de tarea. 2. Posteriormente se siguen los mismos pasos desde el paso 3 pero ahora con multiplicaciones y divisiones binarias. 3. El profesor deja 5 ejercicios de multiplicación y división binaria. <p>Cierre 10 minutos.</p> <ol style="list-style-type: none"> 4. Aleatoriamente el docente elige a 2 integrantes del grupo, el primero para que muestre el desarrollo de un ejemplo de la

	<p>multiplicación binaria, el segundo para que muestre el desarrollo de un ejemplo de la división binaria.</p>
ORGANIZACIÓN	Individual
MATERIALES Y RECURSOS DE APOYO	Pizarrón y plumón
EVALUACIÓN	Puntaje por ejercicio resuelto correctamente



V. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS.	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación. Diseño digital, 3ra Edición (ollintec.com)</p> <p>Vasconcelos, J. (2018). Introducción a la Computación. México: Grupo Editorial.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación. Diseño digital, 3ra Edición (ollintec.com)</p>
COMENTARIOS ADICIONALES	<p>Esta estrategia puede ser llevada a cabo de manera presencial o en línea.</p>

ESTRATEGIA DIDÁCTICA 3

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022
NO DE ALUMNOS	30



II. PROGRAMA I

TÍTULO DE LA UNIDAD	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Construye tablas de verdad de funciones booleanas. Simplifica funciones booleanas utilizando postulados y teoremas básicos.
TEMA(S)	Elementos del álgebra de Boole <ul style="list-style-type: none"> • Variable booleana • Operaciones básicas: conjunción, disyunción y negación. Función booleana <ul style="list-style-type: none"> • Concepto • Expresiones booleanas Tablas de verdad Simplificación de funciones básicas <ul style="list-style-type: none"> • Postulados Teoremas básicos

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	4 horas	
MATERIALES Y/O RECURSOS	Equipo de cómputo o smartphone Internet Aula virtual para la materia de Cibernética y Computación I Aplicación de Hoja Electrónica de Cálculo	
DESARROLLO Y ACTIVIDADES	<p>Inicio</p> <p>Tiempo: 30 minutos</p> <ol style="list-style-type: none"> 1. El docente indica al grupo cuáles serán los aprendizajes que se trabajarán en esta sesión y da una breve introducción del álgebra Booleana. 2. El docente solicita al alumno que ingrese al enlace http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_index.htm en donde debe consultar: <ul style="list-style-type: none"> * El punto 3 Tabla de verdad, identificar el concepto. * El punto 4 Funciones lógicas en donde deberá ingresar a cada una de las operaciones básicas * El procedimiento para obtener la tabla de verdad a través de la función lógica. 3. El alumno ingresa al enlace, consulta el material y anota las dudas que le surgieron. <p>Desarrollo</p> <p>Tiempo: 150 minutos</p> <ol style="list-style-type: none"> 4. El docente resuelve las dudas planteadas, consulta en conjunto con los estudiantes el ejemplo que se presenta en la sección, haciendo énfasis en los pasos del procedimiento. 	

5. El alumno realiza el ejercicio interactivo, comprueba sus resultados.

6. El docente presenta al alumno tres funciones booleanas, solicita que conformen equipo de tres integrantes, obtengan la tabla de verdad de cada una de ellas, utilizando una HEC.



$$Z(X, Y) = XY + XY'$$

$$M(A, B) = AB + A'B$$

$$F(A, B, C, D) = A' + B' + CD' + C'D$$

7. Los alumnos conforman sus equipos y realizan las tablas de verdad de las tres funciones booleanas en una Hoja Electrónica de Cálculo (HEC).

8. El docente solicita al alumno que en equipo, investiguen en Internet cuales son los postulados y teoremas básicos del Álgebra Booleana y los incluya en su libreta física o cuaderno digital.

9. Cada equipo entra a su navegador deseado, busca en una fuente confiable los postulados y teoremas solicitados, y los incluye en su libreta.

10. El docente da una breve explicación de las ventajas de utilizar los postulados y teoremas básicos del Álgebra Booleana, realiza algunos ejemplos y solicita que de las funciones hechas anteriormente apliquen postulados y teoremas.

11. Los alumnos en equipo simplifican las funciones, indicando que teorema o postulado aplicaron, comentan cuales son las ventajas del uso de postulados y teoremas básicos.

Cierre

Tiempo: 60 minutos

	<p>12.El docente selecciona aleatoriamente algunos equipos que presentarán la función antes de la simplificación y la simplificada.</p> <p>13.El docente proporcionará retroalimentación a cada equipo con las soluciones presentadas.</p> <p>14.El docente solicita que entreguen la actividad en el aula virtual de manera individual.</p> <p>El docente revisa las actividades con base a la lista de cotejo.</p>
EVALUACIÓN	Lista de cotejo (Anexo 1)



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ingeniería Electrónica. UNAM (s.f). Álgebra Booleana. Diseño Digital. https://uapa.cuaieed.unam.mx/sites/default/files/minisite/static/7aeaa4d5-346d-4a31-91d9-ad11549328c5/Algebra-Booleana/index.html</p> <p>Landin (2017). Electrónica Digital. https://drive.google.com/file/d/0BxOrdGiYZyv5OFQ5WIR0V2ZKaGM/view?usp=sharing</p> <p>Ministerio de Educación, Cultura y Deporte (2010). Función lógica a partir de la tabla de la verdad. http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_index.htm</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Ministerio de Educación, Cultura y Deporte (2010). Función lógica a partir de la tabla de la verdad. http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_index.htm</p>

Ingeniería Electrónica. UNAM (s.f). Álgebra Booleana. Diseño Digital.
<https://uapa.cuaieed.unam.mx/sites/default/files/minisite/static/7aeaa4d5-346d-4a31-91d9-ad11549328c5/Algebra-Boleana/index.html>

Micelti (s.f). Electrónica Digital.
https://angelmicelti.github.io/4ESO/EDI/33_lgebra_de_boole.html

Landin (2017). Electrónica Digital.
<https://drive.google.com/file/d/0BxOrdGiYZyv5OFQ5WIR0V2ZKaGM/view?usp=sharing>



V. ANEXOS

Anexo 1: Lista de cotejo

Concepto	Si	No
El alumno presenta de forma correcta la tabla de verdad 1		
El alumno presenta de forma correcta la tabla de verdad 2		
El alumno presenta de forma correcta la tabla de verdad 3		
El alumno presenta los postulados y teoremas del Álgebra Booleana		
El alumno presenta de forma correcta la función 1 simplificada		
Indica que postulados o teoremas utilizó para la simplificación en función 1		
El alumno presenta de forma correcta la función 2 simplificada		
Indica que postulados o teoremas utilizó para la simplificación en función 2		
El alumno presenta de forma correcta la función simplificada		
Indica que postulados o teoremas utilizó para la simplificación en función 3		
Total:		

ESTRATEGIA DIDÁCTICA 4



I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTELE	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022
NO DE ALUMNOS	30

II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Describe los conceptos de interruptor, circuito eléctrico, compuerta y circuito lógicos.
TEMA(S)	Interruptor y circuito eléctrico. Compuertas y circuitos lógicos: Compuerta: And, Or y Not

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 horas
MATERIALES Y/O RECURSOS	Equipo de cómputo o smartphone Internet Aula virtual para la materia de Cibernética y Computación El juego de memoria. El juego de arrastrar palabras. Aplicación de Presentación.
DESARROLLO Y ACTIVIDADES	<p>Inicio</p> <p>Tiempo: 30 minutos</p> <ol style="list-style-type: none"> 1. El docente indica al grupo cual será el aprendizaje que se trabajará en esa sesión. 2. El profesor realiza una serie de preguntas al grupo en general para ejemplificar el aprendizaje <ul style="list-style-type: none"> ¿Cómo funciona el interruptor para prender y apagar un foco? ¿Cómo consideras que funciona una máquina expendedora de golosinas? ¿Cómo consideras que funciona una máquina para realizar el pago del estacionamiento? 3. Los alumnos participan para responder las preguntas. <p>Desarrollo</p> <p>Tiempo: 60 minutos</p> <ol style="list-style-type: none"> 4. El docente explica brevemente los conceptos con los que van a realizar la actividad y solicita se formen grupos de trabajo. Y le asigna a cada equipo un concepto: interruptor, circuito eléctrico, compuerta And, Or, Not, Nand, Nor, Xor, Xnor, deberán investigar las características. 5. Los alumnos en equipo buscan información acerca de la compuerta y sus características, realizan una diapositiva. 6. El docente realiza en una aplicación de presentaciones un archivo nuevo, lo comparte con los alumnos para que por



equipo suban la diapositiva de la compuerta que les fue asignada. De esta manera todos tendrán la presentación completa.



7. Los alumnos exponen el concepto que les fue asignado, el docente hace comentarios complementarios.

Cierre

Tiempo: 30 minutos

8. El docente solicita a los alumnos que entren al aula virtual, en donde encontrará dos actividades de gamificación, la primera relacionada a las compuertas básicas, es un juego de memoria



9. Y la segunda relacionada con arrastrar palabras de acuerdo con las compuerta que le corresponda.

	<div style="text-align: center;"> <p>NOT</p> <p>XOR</p> <p>NAND</p> <p>AND</p> <p>XNOR</p> <p>NOR</p> </div> <div style="text-align: right; margin-top: 20px;">  </div> <div style="text-align: center; margin-top: 20px;"> <table border="0" style="width: 100%;"> <tr> <td style="width: 50%; text-align: center;">$A \otimes B$</td> <td style="width: 50%; text-align: center;">$A \oplus B$</td> </tr> <tr> <td style="text-align: center;"><input type="text"/></td> <td style="text-align: center;"><input type="text"/></td> </tr> <tr> <td style="text-align: center;">\overline{AB}</td> <td style="text-align: center;">AB</td> </tr> <tr> <td style="text-align: center;"><input type="text"/></td> <td style="text-align: center;"><input type="text"/></td> </tr> <tr> <td style="text-align: center;">$\overline{A+B}$</td> <td style="text-align: center;">\overline{A}</td> </tr> <tr> <td style="text-align: center;"><input type="text"/></td> <td style="text-align: center;"><input type="text"/></td> </tr> </table> </div> <p style="text-align: center; margin-top: 20px;">10.El docente pregunta a los alumnos si hay alguna duda o pregunta, hace la retroalimentación.</p>	$A \otimes B$	$A \oplus B$	<input type="text"/>	<input type="text"/>	\overline{AB}	AB	<input type="text"/>	<input type="text"/>	$\overline{A+B}$	\overline{A}	<input type="text"/>	<input type="text"/>
$A \otimes B$	$A \oplus B$												
<input type="text"/>	<input type="text"/>												
\overline{AB}	AB												
<input type="text"/>	<input type="text"/>												
$\overline{A+B}$	\overline{A}												
<input type="text"/>	<input type="text"/>												
EVALUACIÓN	Lista de cotejo (Anexo 1)												

IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA	Ministerio de Educación, Cultura y Deporte (2010). Puertas lógicas básicas.
---------------------------------	---

PARA LOS ALUMNOS	http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena6/index_4quincena6.htm
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Ministerio de Educación, Cultura y Deporte (2010). Puertas lógicas básicas. http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena6/index_4quincena6.htm</p> <p>Landín (s.f). Unidad temática 4. Electrónica Digital. https://iesvillalbahervastecnologia.files.wordpress.com/2009/06/electrc3b3nica-digital-teoriayejercicios.pdf</p>



V. ANEXOS

Anexo 1: Lista de cotejo

Concepto	Si	No
Realiza la investigación del tema correspondiente		
Realiza la diapositiva del concepto que se le solicitó		
Expone el tema que le fue asignado		
Realiza el juego de memoria		
Realiza el juego de arrastrar palabras		
Total:		

ESTRATEGIA DIDÁCTICA 5



I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTELE	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022
NO DE ALUMNOS	30

II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Aprende a utilizar el protoboard o simulador.
TEMA(S)	Uso de protoboard o un simulador

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 hrs
MATERIALES Y/O RECURSOS	Equipo de cómputo o smartphone Internet

Simulador disponible en <https://logic.ly/>
 Aplicación para hacer vídeo
 Tablas de verdad de las compuertas (And, Or, Not)
 Aula virtual para la materia de Cibernética y Computación



DESARROLLO Y ACTIVIDADES

Inicio

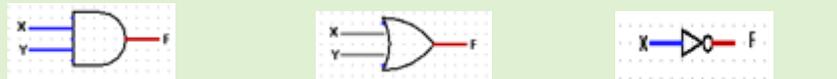
Tiempo: 30 minutos

1. El docente indica al grupo cual será el aprendizaje que se trabajará en esa sesión.
2. Solicita a los alumnos que consulten las tablas de verdad de las compuertas lógicas antes vistas y realiza una serie de preguntas para recordar el tema.

Pregunta 1. Realiza la tabla de verdad para cada compuerta lógica digital.

AND			OR			X	
X	Y	F	X	Y	F	X	F
0	0		0	0		0	
0	1		0	1		1	
1	0		1	0		1	
1	1		1	1			

Pregunta 2. Identifica el símbolo gráfico de cada compuerta lógica digital.



Pregunta 3. ¿Cuál es la expresión algebraica para cada compuerta lógica digital?

AND _____ OR _____ NOT _____

3. El alumno consulta sus tablas de verdad y participa con base a las diferentes combinaciones y resultados que se obtienen.
4. El docente solicita al alumno abrir el simulador **logic.ly** disponible en: <https://logic.ly/> en el navegador de su preferencia.
5. El alumno ingresa al simulador.

Desarrollo

Tiempo: 60 minutos

	<p>6. El docente explica brevemente el funcionamiento del simulador, indicando los principales componentes del ambiente de trabajo a utilizar controles de entrada, controles de salida, compuertas lógicas, etiquetas, (Anexo 1)</p> <p>7. El alumno va identificando cada parte del ambiente de trabajo, que el docente le está indicando.</p> <p>8. El docente solicita al alumno que a través del simulador realice la comprobación de las tres compuertas lógicas básicas And, Or y Not.</p> <p>9. El alumno deberá integrar su circuito lógico con: un switch para cada variable, la cual deberá identificar con su nombre a través de una etiqueta, la compuerta lógica que le corresponda y un foco en la salida.</p> <p>10. Una vez terminado el circuito el alumno deberá comprobar la tabla de verdad a través de la manipulación de sus switches (encendido, apagado) y observando el foco de salida, hasta completar las combinaciones posibles.</p>
EVALUACIÓN	Lista de cotejo (Anexo 2)



IV. REFERENCIAS DE APOYO

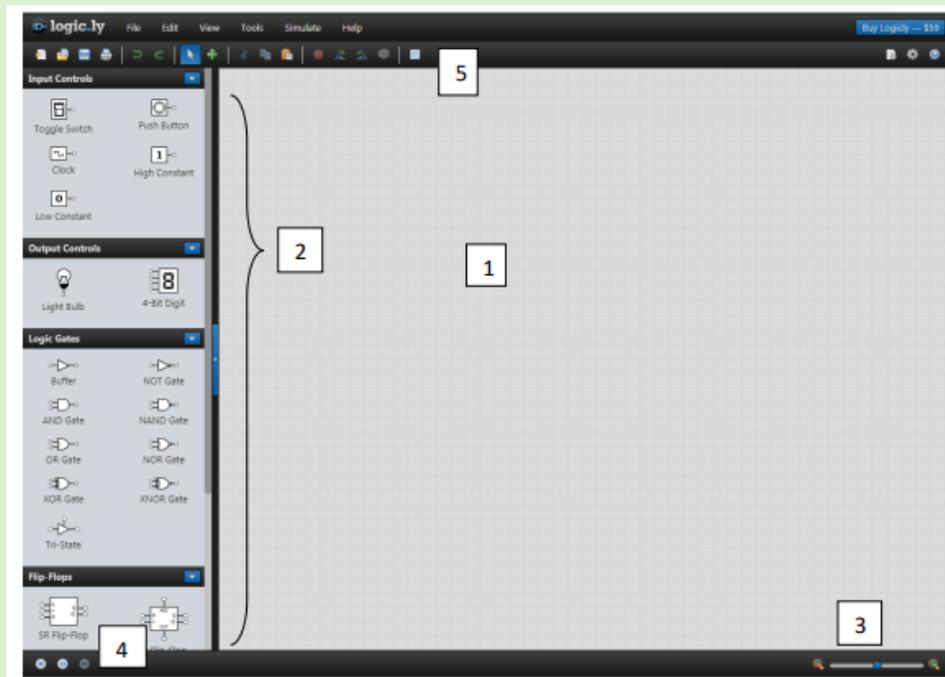
BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación. http://ollintec.com/SistemasDigitales/libros/Dise%C3%B1o%20digital,%203ra%20Edici%C3%B3n%20-%20M.%20Morris%20Mano-FREELIBROS.ORG.pdf</p> <p>Tynjala (2020). Simulador Logic.ly. https://logic.ly/</p>
BIBLIOGRAFÍA DE CONSULTA	Andel. Tecnología, Programación y Robótica (2019). Tutorial Logicly puertas lógicas.

<p>PARA EL PROFESOR</p>	<p>https://andeltecnologia.files.wordpress.com/2019/02/guic3b3n-tutorial-logicly.pdf</p> <p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación.</p> <p>http://ollintec.com/SistemasDigitales/libros/Dise%C3%B1oDigital,%203ra%20Edici%C3%B3n-%20M.%20Morris%20Mano-FREELIBROS.ORG.pdf</p> <p>Tynjala (2020). Simulador Logic.ly. https://logic.ly/</p>
--------------------------------	---



V. ANEXOS

Anexo 1: Ambiente de trabajo



1. Este es el lugar donde vas a arrastrar los elementos que tenemos a la izquierda para construir tu propio circuito con puertas lógicas.
2. Este es lugar de donde vas a sacar las puertas lógicas, las entradas y las salidas, para después arrastrarlas al punto 1
3. Zoom para agrandar o empequeñecer la imagen
4. Para pausar o resetear la simulación
5. Barra de herramientas.



Anexo 2: Lista de cotejo

Concepto	Si	No
El video tiene resolución aceptable para distinguir los componentes (10pts)		
Las variables están representadas con un switch y una etiqueta (10pts)		
Las salidas están representadas con un foco (10pts)		
Las compuertas están correctamente conectadas (10pts)		
El diagrama que contiene la compuerta <u>And</u> responde según su tabla de verdad (20pts)		
El diagrama contiene la compuerta <u>Or</u> responde según su tabla de verdad (20pts)		
El diagrama contiene la compuerta <u>Not</u> responde según su tabla de verdad (20pts)		
Total:		

ESTRATEGIA DIDÁCTICA 6

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022
NO DE ALUMNOS	30



II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Construye la función booleana a partir de la tabla de verdad, empleando suma de productos.
TEMA(S)	Obtención de la función booleana. Suma de Productos.

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 hora
MATERIALES Y/O RECURSOS	Equipo de cómputo o smartphone Internet

	Aula virtual para la materia de Cibernética y Computación I
DESARROLLO Y ACTIVIDADES	<p>Inicio Tiempo: 10 minutos</p> <ol style="list-style-type: none"> 1. El docente indica al grupo cual será el aprendizaje que se trabajará en esa sesión. 2. El profesor realiza una serie de preguntas al grupo en general para recordar el aprendizaje de tabla de verdad. <p>Pregunta 1. ¿Qué es una tabla de verdad?</p> <p>Pregunta 2. ¿Cómo determinar la cantidad de combinaciones posibles?</p> <p>Pregunta 3. ¿Cómo construir la tabla con base a la función?</p> 3. El alumno participa para responder las preguntas. <p>Desarrollo Tiempo: 40 minutos</p> <ol style="list-style-type: none"> 4. El docente explica brevemente como obtener la función booleana a partir de la tabla de verdad. 5. El docente solicita al alumno que ingrese a la página http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_index.htm, en donde deberá consultar los ejemplos que se presentan ahí. 6. El alumno consulta el recurso web http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/4q2_index.htm, en donde revisa los tres ejemplos. Posteriormente en conjunto alumno- el docente (en plenaria o videollamada), van analizando el procedimiento paso a paso con los ejemplos que se presentan.



7. El docente solicita que individualmente realicen los cinco ejercicios que se presenta en la plataforma.

8. El alumno localiza los dos primeros ejercicios que están en el siguiente enlace

<http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/ejercicios/ejercicios4b.htm>



EJERCICIOS

1. Indica en el recuadro cuál es la función lógica representada en la siguiente tabla:

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

S_1

S_4

$$S = \overset{S_1}{-} \overset{S_1}{-} + \overset{S_4}{-} \overset{S_4}{-}$$

comprobar

Borrar

2. Indica en el recuadro cuál es la función lógica representada en la siguiente tabla:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

S_3

S_3

$$S = \overset{S_2}{-} \overset{S_2}{-} + \overset{S_3}{-} \overset{S_3}{-}$$

comprobar

Borrar

COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALÉO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS



19/07/2022



RECIBIDO

y los siguientes tres ejercicios que se encuentran en:

http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena5/q5_ejercicios_d.php

 **Para practicar**

Repasa los **contenidos** de esta sección:
Obtención de la función lógica a partir de la tabla

Obtención de la función lógica a partir de la tabla de verdad

Obtén la función de las siguientes tablas de verdad fijándote en los valores que toman la entradas A, B y C que S=1.
Las funciones obtenidas no están simplificadas

COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALCO
CAPACITACIÓN DE SECCIÓN
ÁREA DE MATEMÁTICAS
19/07/2022

RECIBIDO

a)

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$$S = \overset{S_1}{-} \overset{S_1}{-} \overset{S_1}{-} + \overset{S_3}{-} \overset{S_3}{-} \overset{S_3}{-} + \overset{S_6}{-} \overset{S_6}{-} \overset{S_6}{-}$$

b)

A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$S = \overset{S_5}{-} \overset{S_5}{-} \overset{S_5}{-} + \overset{S_6}{-} \overset{S_6}{-} \overset{S_6}{-} + \overset{S_8}{-} \overset{S_8}{-} \overset{S_8}{-}$$

c)

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$$S = \overset{S_2}{-} \overset{S_2}{-} \overset{S_2}{-} + \overset{S_4}{-} \overset{S_4}{-} \overset{S_4}{-} + \overset{S_7}{-} \overset{S_7}{-} \overset{S_7}{-}$$

Cierre

Tiempo: 10 minutos

9. El docente pregunta a los alumnos si hay alguna duda o pregunta, solicita al alumno que entregue su actividad en el aula virtual.
10. El alumno comprobará los resultados, generará un archivo pdf con los resultados obtenidos y los subirá al aula virtual.
11. El docente revisa las actividades y hace una retroalimentación a cada una de las actividades que recibió.

EVALUACIÓN

Lista de cotejo (Anexo 1)

IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	Ministerio de Educación, Cultura y Deporte (2010). Función Lógica a partir de la tabla de la verdad. http://recursostic.educacion.es/secundaria/edad/4esotecnologia/q2/index.htm
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	Instituto Tecnológico de Querétaro (s.f). Lógica secuencial y combinatoria. http://www.itq.edu.mx/carreras/IngElectronica/archivos_contenido/Apuntes%20de%20materias/Apuntes_Log_Sec_Comb/Sesion_06_LSC.pdf



V. ANEXOS

Anexo 1: Lista de cotejo

Concepto	Si	No
La función obtenida de la tabla de verdad inciso 1 es correcta		
La función obtenida de la tabla de verdad inciso 2 es correcta		
La función obtenida de la tabla de verdad inciso A es correcta		
La función obtenida de la tabla de verdad inciso B es correcta		
La función obtenida de la tabla de verdad inciso C es correcta		
Total:		

ESTRATEGIA DIDÁCTICA 7

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022

COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALCO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS

19/07/2022

RECIBIDO

II.PROGRAMA

UNIDAD TEMÁTICA	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Construye un semisumador.
TEMA(S)	Semisumador.

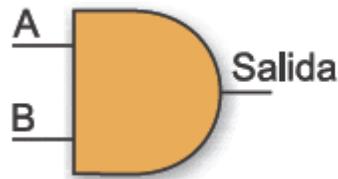
III. ESTRATEGIA

El profesor presenta al grupo el problema de diseñar un circuito que se comporte como una suma binaria, después guía el grupo a la respuesta del problema para finalizar con la elaboración del circuito en un simulador.

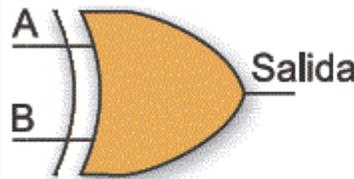
IV.SECUENCIA

TIEMPO DIDÁCTICO	2 hrs.																								
DESARROLLO Y ACTIVIDADES	<p>Inicio Tiempo: 30 minutos.</p> <ol style="list-style-type: none"> 1. El docente indica al grupo cual será el aprendizaje que se trabajará en esa sesión. 2. El profesor realiza una recuperación de conocimientos para retomar el tema de operaciones con el sistema de numeración binario, en particular la suma de 2 números binarios 3. Los alumnos identifican nuevamente los resultados de la suma y los guía para que lleguen a la elaboración de la siguiente tabla: <table border="1" data-bbox="799 934 1193 1138"> <thead> <tr> <th colspan="2">Entradas</th> <th colspan="2">Salidas</th> </tr> <tr> <th>A</th> <th>B</th> <th>Acarreo</th> <th>Suma</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> <ol style="list-style-type: none"> 4. El profesor aclara la diferencia entre suma binaria y suma booleana para que los alumnos no se confundan entre las 2 formas de sumar que han aprendido hasta el momento <p>Desarrollo Tiempo: 60 minutos</p> <ol style="list-style-type: none"> 5. El docente pide a los alumnos que observen los resultados de la tabla obtenida por ellos mismos y plantea la siguiente incógnita: ¿A qué compuertas lógicas se asemejan los resultados observados? 6. Los alumnos analizan las tablas de verdad de las compuertas lógicas que han aprendido hasta el momento y con la guía del profesor llegan a la conclusión de que se trata del mismo comportamiento de las compuertas AND y XOR 	Entradas		Salidas		A	B	Acarreo	Suma	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	0
Entradas		Salidas																							
A	B	Acarreo	Suma																						
0	0	0	0																						
0	1	0	1																						
1	0	0	1																						
1	1	1	0																						



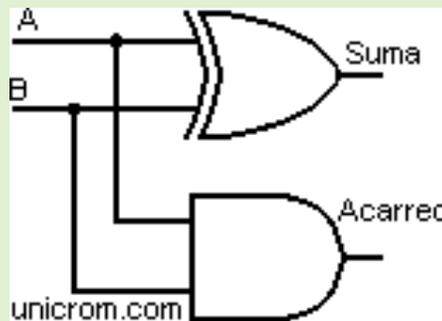


A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

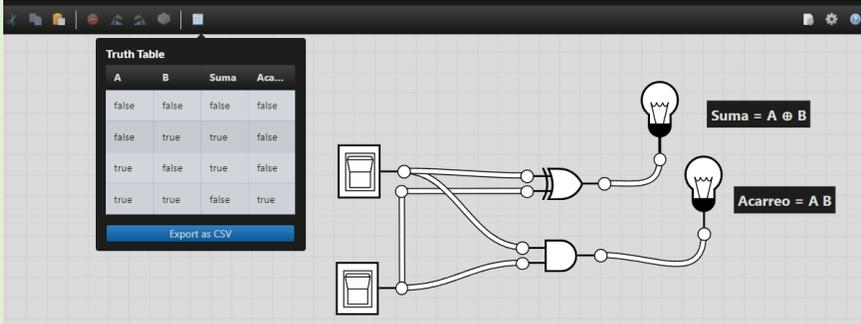


A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	0

7. El profesor Dibuja estás 2 compuertas juntas en el pizarrón o en la pizarra electrónica y explica que se trata de un circuito con 2 entradas A y B que representan los números a ser sumados y 2 salidas “Acarreo” y “suma” Que representan el resultado de la suma, une las compuertas para formar el siguiente diagrama:



8. El profesor solicita a los alumnos que realicen las fórmulas del circuito anterior con el objetivo de repasar funciones booleanas y obtener las 3 partes fundamentales del circuito.
9. El profesor recibe las respuestas de los alumnos y verifica que concuerden con las siguientes funciones:
 Suma = $A \oplus B$
 Acarreo = AB
10. El profesor solicita a los alumnos que realicen el diagrama en el simulador logic.ly de tal forma que se muestre el diagrama, las funciones y la tabla de verdad.

	<p>Cierre</p> <p>Tiempo: 30 minutos</p> <ol style="list-style-type: none"> 11.El docente resuelve dudas mientras los alumnos trabajan individualmente 12.El profesor solicita a los alumnos que prueben el circuito y verifiquen su comportamiento. 13.El profesor recibe los trabajos de los alumnos y comprueba que sean entregados de la siguiente forma:  <ol style="list-style-type: none"> 14.El docente evalúa con base a la lista de cotejo y proporciona retroalimentación.
ORGANIZACIÓN	Individual
MATERIALES Y RECURSOS DE APOYO	Equipo de cómputo o smartphone Internet
EVALUACIÓN	Lista de cotejo (Anexo 1)



V. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS.	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación.</p> <p>http://ollintec.com/SistemasDigitales/libros/Dise%C3%B1o%20digital,%203ra%20Edici%C3%B3n%20-%20M.%20Morris%20Mano-FREELIBROS.ORG.pdf</p>
---	---

	Vasconcelos, J. (2018). Introducción a la Computación. México: Grupo Editorial
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación.</p> <p>http://ollintec.com/SistemasDigitales/libros/Dise%C3%B1o%20digital,%203ra%20Edici%C3%B3n%20-%20M.%20Morris%20Mano-FREELIBROS.ORG.pdf</p>
COMENTARIOS ADICIONALES	



VI. ANEXOS

Anexo 1: Lista de cotejo

	sí	n o
1.- El diagrama está elaborado en un simulador (Logic.ly o similar)		
2.- El diagrama presenta 2 entradas A y B		
3.- El diagrama presenta 2 compuertas XOR y AND		
4.- El diagrama Presenta 2 salidas (focos/leds)		
5.- El diagrama está correctamente alambrado		
6.- El diagrama presenta la función de la suma		
7.- El diagrama presenta la función del acarreo		
8.- Las entradas están debidamente identificadas con su nombre (A y B)		
9.- Las salidas están debidamente identificadas con su nombre (Suma y Acarreo)		
10.- El diagrama presenta la tabla de verdad		
Total:		

ESTRATEGIA DIDÁCTICA 8

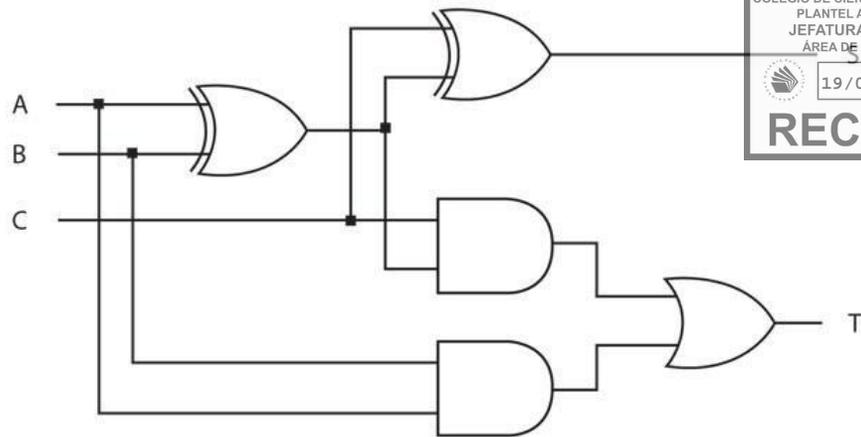
I.DATOS GENERALES	
PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	8 de abril de 2022



II.PROGRAMA	
UNIDAD TEMÁTICA	Unidad 2. Circuitos Lógicos
PROPÓSITO(S) DE LA UNIDAD	Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.
APRENDIZAJE(S)	Construye un sumador completo.
TEMA(S)	Construcción del circuito lógico.

III. ESTRATEGIA	
<p>El profesor proporciona el diagrama del sumador completo para que los alumnos analicen y comprendan el funcionamiento de un sumador completo, Después el profesor guía a los alumnos para representar el diagrama en un simulador de circuitos lógicos.</p>	

IV.SECUENCIA	
TIEMPO DIDÁCTICO	2 hrs.
DESARROLLO Y ACTIVIDADES	<p>Inicio</p> <p>Tiempo: 15 minutos.</p> <p>1. El profesor comparte el aprendizaje con el que se va a trabajar y proporciona el siguiente diagrama a los alumnos:</p>



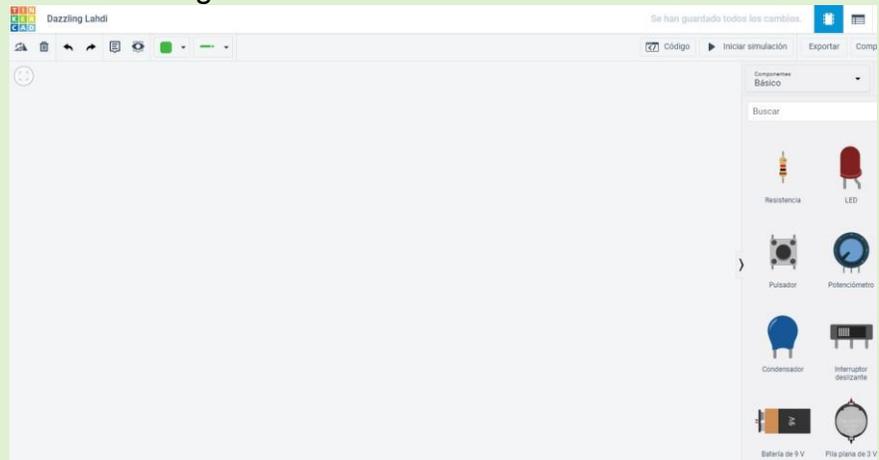
2. Solicita a los alumnos que realicen las funciones del diagrama para que se familiaricen con el
3. El profesor da la respuesta de las funciones para que los alumnos verifiquen sus resultados
4. El profesor explica que el diagrama corresponde a un sumador completo que contempla 3 entradas que corresponden a los números a ser sumados y un supuesto acarreo de entrada y 2 salidas que corresponden al resultado de la suma y un acarreo de salida.
5. Proporciona la tabla de verdad que corresponde al circuito:

A	B	C	T	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

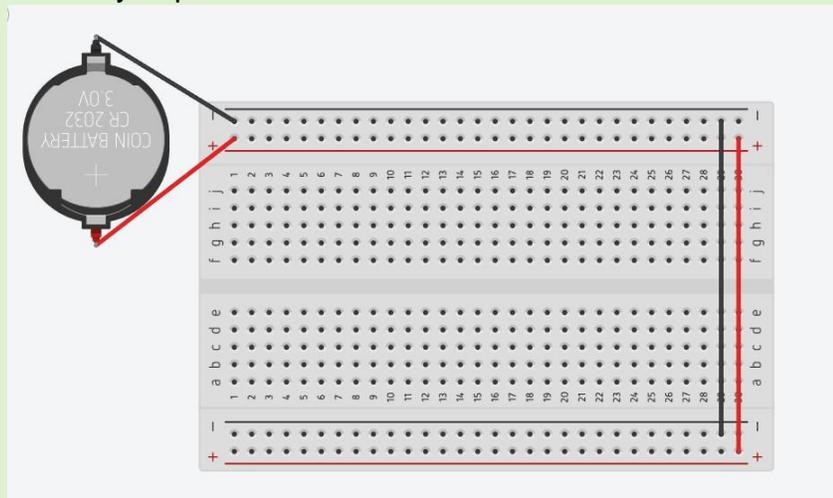
Desarrollo

Tiempo: 90 minutos

6. El docente Solicita a los alumnos que abran el simulador <https://www.tinkercad.com/>, creen una cuenta e ingresen al apartado de circuitos para desplegar el área de trabajo que se ve de la siguiente forma:



7. El profesor añade una placa de pruebas pequeña, una batería y explica como conectarlos:



8. Solicita a los alumnos que hagan el mismo procedimiento
9. El docente explica brevemente el funcionamiento de la tableta protoboard.

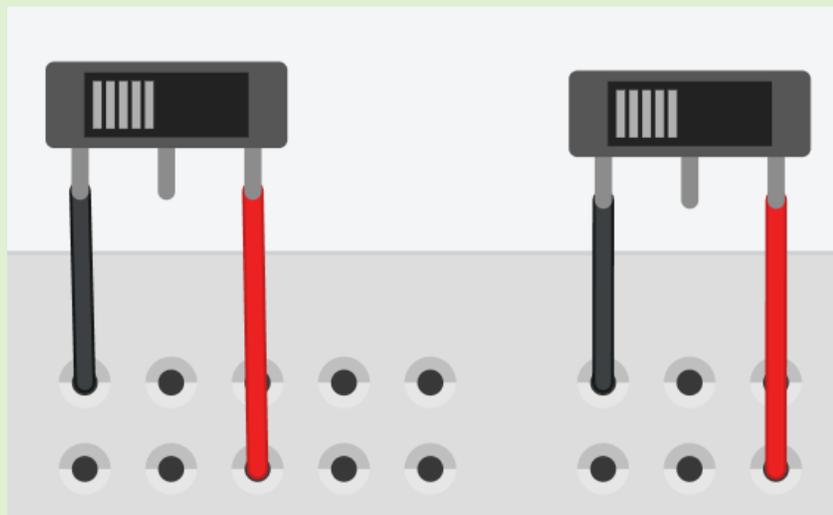


10. El profesor agrega las compuertas necesarias en forma de circuitos integrados y explica su funcionamiento:



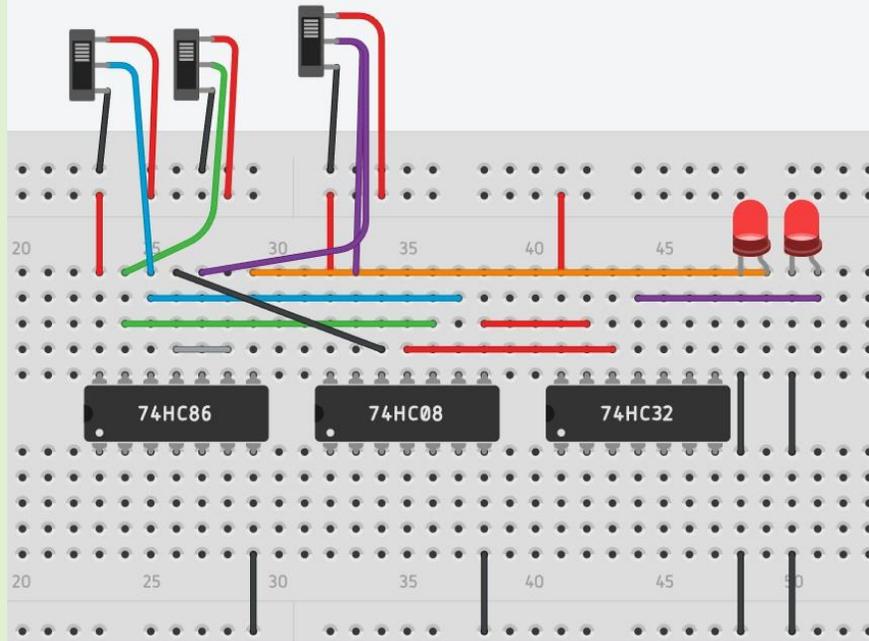
11. Solicita a los alumnos que hagan el mismo procedimiento

12. El profesor agrega 2 switches y 2 leds y explica como conectarlos:



13. Solicita a los alumnos que hagan el mismo procedimiento.

14. Ya con los elementos necesarios el profesor procede a alambrear el circuito según el diagrama, esto para que los alumnos puedan seguir el paso de la actividad y se resuelvan las dudas necesarias. El Circuito de la siguiente forma:



Cierre

Tiempo: 15 minutos

- 15. El docente termina el circuito y deja en pantalla el resultado para que los alumnos terminen
- 16. Califica a los alumnos que vayan terminando de acuerdo con la lista de cotejo

ORGANIZACIÓN

Individual

MATERIALES Y RECURSOS DE APOYO	Equipo de cómputo o smartphone Internet	
EVALUACIÓN	Lista de cotejo (Anexo 1)	

V. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS.	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación. http://ollintec.com/SistemasDigitales/libros/Dise%C3%B1o%20digital,%203ra%20Edici%C3%B3n%20-%20M.%20Morris%20Mano-FREELIBROS.ORG.pdf</p> <p>Vasconcelos, J. (2018). Introducción a la Computación. México: Grupo Editorial</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Morris, Mano. (2003). Diseño Digital, tercera edición. México: PEARSON Educación. http://ollintec.com/SistemasDigitales/libros/Dise%C3%B1o%20digital,%203ra%20Edici%C3%B3n%20-%20M.%20Morris%20Mano-FREELIBROS.ORG.pdf</p>
COMENTARIOS ADICIONALES	Breves comentarios o aclaraciones.

VI. ANEXOS

Anexo 1: Lista de cotejo

	sí	no
1.- Termina el Circuito		
2.- El circuito presenta una pila de reloj		
3.- El circuito presenta una protoboard bien alambrada		
4.- El circuito presenta 3 switches		
5.- El circuito presenta 2 leds		
6.- El circuito presenta un circuito integrado 74HC86		
7.- El circuito presenta un circuito integrado 74HC08		
8.- El circuito presenta un circuito integrado 74HC32		
9.- Todos los componentes están debidamente aterrizados		

10.- El circuito funciona de acuerdo con su tabla de verdad		
	Total:	



Unidad 3: Metodología de solución de problemas e introducción al lenguaje de programación Java

Propósito: Al finalizar la unidad el alumno:

Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.



ESTRATEGIA DIDÁCTICA 1

I. DATOS GENERALES

PROFESOR(A)	Seminario Institucional de Propuesta educativa
ASIGNATURA	Cibernética y Computación
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades.
FECHA DE ELABORACIÓN	Junio de 2022
NO DE ALUMNOS	25

II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • El alumno define el concepto de problema. • El alumno identifica los elementos de un problema.
TEMA(S)	<ul style="list-style-type: none"> • Definiciones y conceptos generales de un problema. • Elementos y relaciones del problema: <ul style="list-style-type: none"> ○ Entrada.

	<ul style="list-style-type: none"> ○ Proceso. ○ Salida.
--	---



III.ESTRATEGIA

Las actividades de aprendizaje se apoyarán en el uso de la plataforma educativa, se realizará videoconferencia si las circunstancias sanitarias lo exigen y/o en su caso en forma presencial.
 El estudiante abordará los conceptos generales de un problema, mediante ejemplos de la vida real modelará al problema y sus elementos.

IV.ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 clase de 120 minutos. <ul style="list-style-type: none"> • Inicio: 25 minutos • Desarrollo: 70 minutos • Cierre: 25 minutos.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Computadora con acceso a internet. • Plataforma educativa. • Cuenta del profesor y alumno de Google. • Presentación electrónica “Conceptos generales de un problema”. • Encuesta: ¿Qué preocupa a los jóvenes? en https://www.questionpro.com/ • Diagrama EPS (plantilla, ejemplo, ejercicio). • Diagramas árbol (plantilla, ejemplo, ejercicio). • Documento colaborativo en Google drive.

**DESARROLLO Y
ACTIVIDADES**

Actividad previa a la sesión:

El Profesor: configura las fechas de apertura y cierre del tema

“Conceptos generales de un problema” en plataforma educativa para que se asegure que los materiales y recursos estén disponibles para su uso en plataformas seleccionadas.



Inicio

Tiempo estimado: 30 minutos.

El Profesor: (Grupal)

Tiempo estimado: 10 minutos

- Da la bienvenida a la clase.
- Realiza el encuadre de la sesión.
- Brinda introducción al tema y da paso a la actividad 1.

Actividad 1. Encuesta: ¿Qué preocupa a los jóvenes?

Tiempo estimado: 15 minutos

Inicio - Actividad 1. ¿Qué preocupa a los jóvenes?

- El Profesor solicita a los alumnos que ingresen a la plataforma educativa e ingresen al enlace <https://questionpro.com/t/AVXnsZsYBI> o escaneen el código QR para contestar la encuesta. **(ANEXO 1)**



Desarrollo - Actividad 1.

Los estudiantes: (Grupal)

- Contestan la encuesta.

El profesor (Grupal):

- Comparte los resultados obtenidos de la encuesta.
- A partir de los resultados obtenidos define en conjunto con los alumnos sí las temáticas expuestas representan algún problema.

Cierre - Actividad 1.

Los alumnos (grupal):

- Responden y argumentan ante las preguntas: ¿Estas temáticas abordadas son un problema? ¿Si o no? ¿Por qué son un problema? O ¿Por qué no? ¿Qué es un problema?

Fin Actividad 1.



Desarrollo

El profesor (Grupal):

Tiempo estimado: 20 minutos.

- Con ayuda de una presentación digital se exponen algunas definiciones y conceptos generales del problema y los elementos que lo conforman.

Actividad 2. Diagramas de resolución de problemas.

Tiempo estimado: 50 minutos.

Inicio - Actividad 2. Diagramas de resolución de problemas.

El profesor (Grupal):

- Proporciona 4 ejercicios a través de la plataforma educativa (ANEXO 2), así como los formatos Diagrama EPS (ANEXO 3) y/o Diagrama de árbol (ANEXO 4) del problema para ser llenados por el alumno.
- Brinda las indicaciones para el llenado del diagrama.
- Presenta un ejemplo resuelto de los diagramas (ANEXOS 5 Y 6)
- De ser necesario crea salas de equipos reducidos, permite la integración de equipos o los asigna directamente.

Desarrollo - Actividad 2. Diagramas de resolución de problemas.

El alumno (En equipos reducidos):

- Forma equipos de 5 alumnos
- Un integrante comparte el esquema en drive con sus compañeros
- Anotan los nombres de los integrantes del equipo y asignan un nombre de equipo.
- Seleccionan el diagrama más útil para el tipo de problema.
- Identifican las causas, insumos o entradas; operaciones o tratamiento efectuados; así como los efectos, productos o salidas para cada problema.
- En caso de requerirse, agregan cuadros de textos necesarios en cada sección.

- Seleccionan a un integrante quien expondrá el producto obtenido.

El profesor (En equipos reducidos):

- Visita de forma presencial o virtual (salas de equipos reducidos) a los equipos.
- Resuelve dudas, comenta y realiza aportaciones integrante de los equipos de trabajo.



Cierre - Actividad 2. Diagramas de resolución de problemas.

El profesor (Grupal):

- Mediante participación voluntaria, al azar, o designación; exhorta a los equipos a exponer su diagrama en sesión plenaria.

El alumno (Grupal por equipo):

- Expone su diagrama de árbol.
- Comenta y realiza aportaciones al material presentado por otros equipos.

Cierre

Tiempo estimado: 25 minutos.

El alumno (individual - grupal):

- Realiza un escrito donde define al problema y argumenta la importancia de la identificación de los elementos del problema. **(ANEXO 7)**.
- De manera voluntaria, al azar o por designación, lee en voz alta el texto escrito al grupo.
- Complementa la participación de otros compañeros.

El profesor (Grupal):

- Brinda una conclusión genérica sobre el tema.

Extra - clase

Tiempo estimado: 10 minutos.

El alumno (individual):

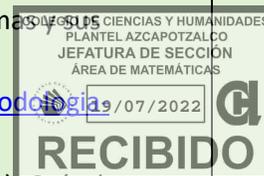
- Sube a la plataforma educativa sus ejercicios para evaluación:
 - Diagramas de resolución de problemas.
 - Síntesis y argumentación del tema en la plataforma educativa.

EVALUACIÓN	Rúbrica: Diagramas de resolución de problemas			
	Criterios	Totalmente adecuado 2pts.	Parcialmente adecuado 1 pt.	Inadecuado
	Nombre asignado al problema			
	Ortografía, puntuación y gramática			
	Identifica los elementos de entrada			
	Identifica los procesos			
	Identifica los elementos de salida			
	Rúbrica: Síntesis y argumentación del tema en la plataforma educativa			
	Criterio	Excelente (2 pts.)	Bien (1 pt.)	Insuficiente (1 pt.)
	Conceptos	Los conceptos fueron expresados de una manera clara y organizada. Fue fácil entender la información.	Las ideas fueron expresadas de una manera muy clara, pero presentó algunas fallas en la organización.	La síntesis es una colección de oraciones no organizadas que dificultan la comprensión.
	Descripción	Hace una descripción completa y detallada del contenido y los elementos vistos.	Hace una descripción detallada de la mayoría del contenido y los elementos vistos.	No hace una descripción detallada del contenido ni de los elementos vistos.
	Redacción y ortografía	Utiliza correctamente los signos de puntuación y no presenta errores ortográficos.	Presenta de uno a dos errores en los signos de puntuación y/o en la ortografía.	Presenta 6 ó más errores en la redacción y/o en la ortografía.
	Conclusión	La conclusión incluye interpretaciones personales y los que se aprendió del tema.	La conclusión incluye los descubrimientos que apoyan su interpretación personal, así como lo que se aprendió del tema.	No hay conclusión incluida en la síntesis.



IV. REFERENCIAS DE APOYO

<p>BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS</p>	<ul style="list-style-type: none"> Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill. Portal Académico del CCH (18 de mayo de 2022). Problemas y sus elementos. https://portalacademico.cch.unam.mx/cibernetica1/metodologia/resolucion-problemas/problemas-elementos Colegio de Ciencias y Humanidades. (18 de mayo de 2022). Guía de estudio de Cibernética y Computación I y II. https://www.unamenlinea.unam.mx/recurso/83615-guia-para-el-estudiante-de-cibernetica-y-computacion Universidad de los andes (18 de mayo de 2022). Fundamentos de programación. https://universidad-de-los-andes.gitbooks.io/fundamentos-de-programacion/content/Nivel1/3_ProblemasYSoluciones.html Legalidad por México (18 de mayo de 2022). Árbol de problemas. https://legalidadpormexico.org/mcl/modulo1/arboldeproblemas.html
<p>BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR</p>	<ul style="list-style-type: none"> Colegio de Ciencias y Humanidades. (18 de mayo de 2022). Guía de estudio de Cibernética y Computación I y II. https://www.unamenlinea.unam.mx/recurso/83615-guia-para-el-estudiante-de-cibernetica-y-computacion Castillo, A. S., Berenguer, I. A., Sánchez, A. G., & Fernández, Y. T. (2013). Didáctica de la resolución de problemas de Programación Computacional. <i>Pedagogía Universitaria</i>, 18(4), 62-75. Ortega Ruipérez, B. (2018). Pensamiento computacional y resolución de problemas.



V. ANEXOS

Anexo 1. Encuesta en línea.

¿Qué preocupa a los jóvenes?



Cambio climático



Muy insatisfecho



Insatisfecho



Neutral



Satisfecho



Muy satisfecho

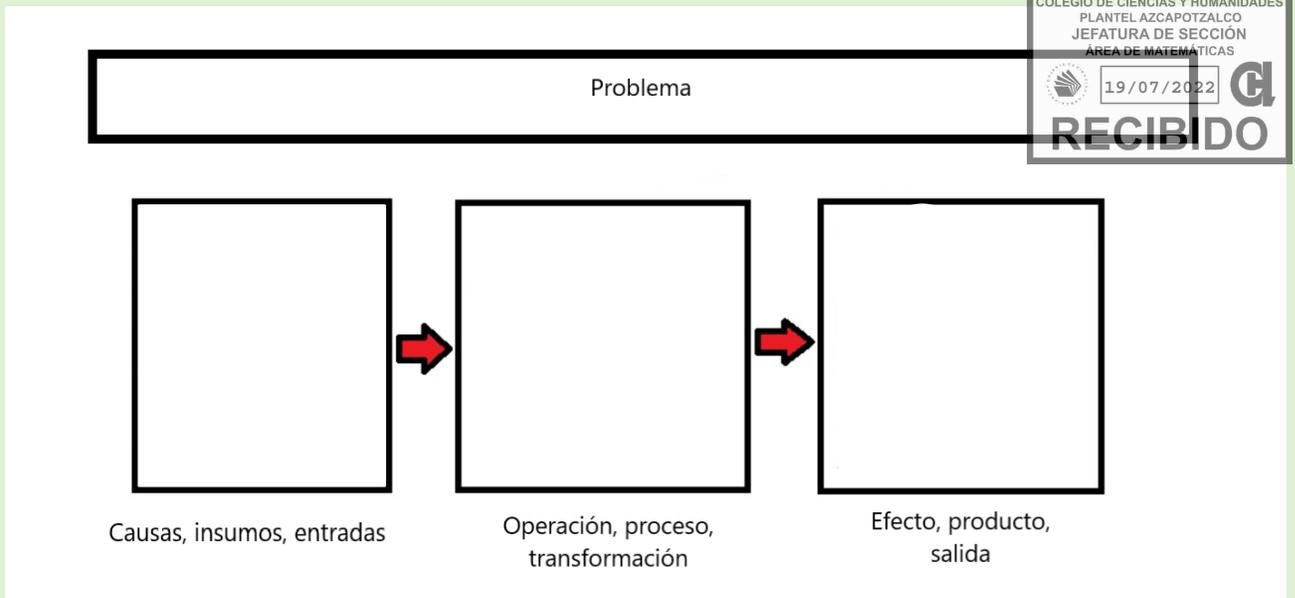
<https://www.questionpro.com/a/TakeSurvey?tt=kjxqWtYBP5I%3D>

Anexo 2. Ejercicios Propuestos

1. Un estudiante quiere viajar a los Estados Unidos si sabe que el tipo de cambio está en 19 pesos por dólar ¿Cuántos dólares obtendrá por un monto de pesos?
2. Se necesita calcular el sueldo de un asesor, este trabajó tres meses en la empresa.
3. Un cliente ordena 4 libros, que a menudeo cuestan %630 cada uno, menos un descuento de 25%, 14 separadores de libros con un costo de \$20 cada uno, con un descuento del 10%, ¿Cuánto pagará el cliente por su compra?
4. Una joven decide preparar unos molletes, ¿cuáles serían los ingredientes y como elaboraría los molletes para tener un exquisito desayuno?

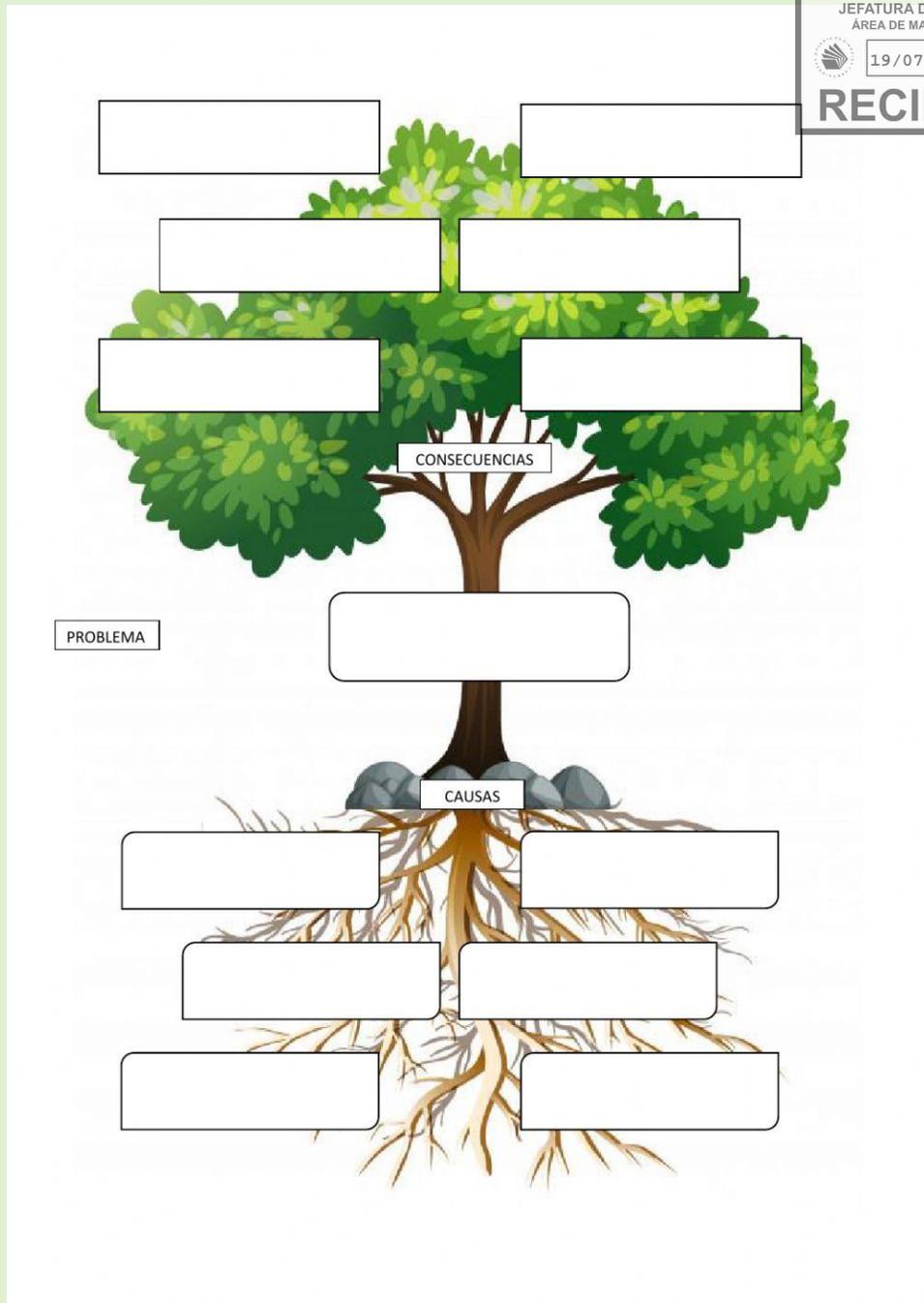


Anexo 3. Diagrama EPS

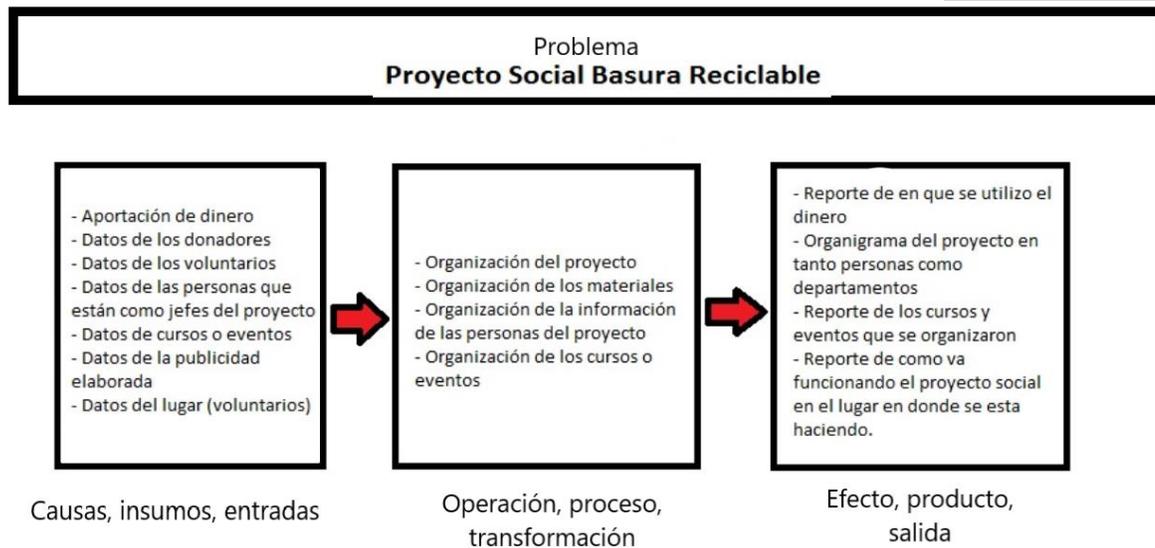


Anexo 4. Diagrama de Árbol

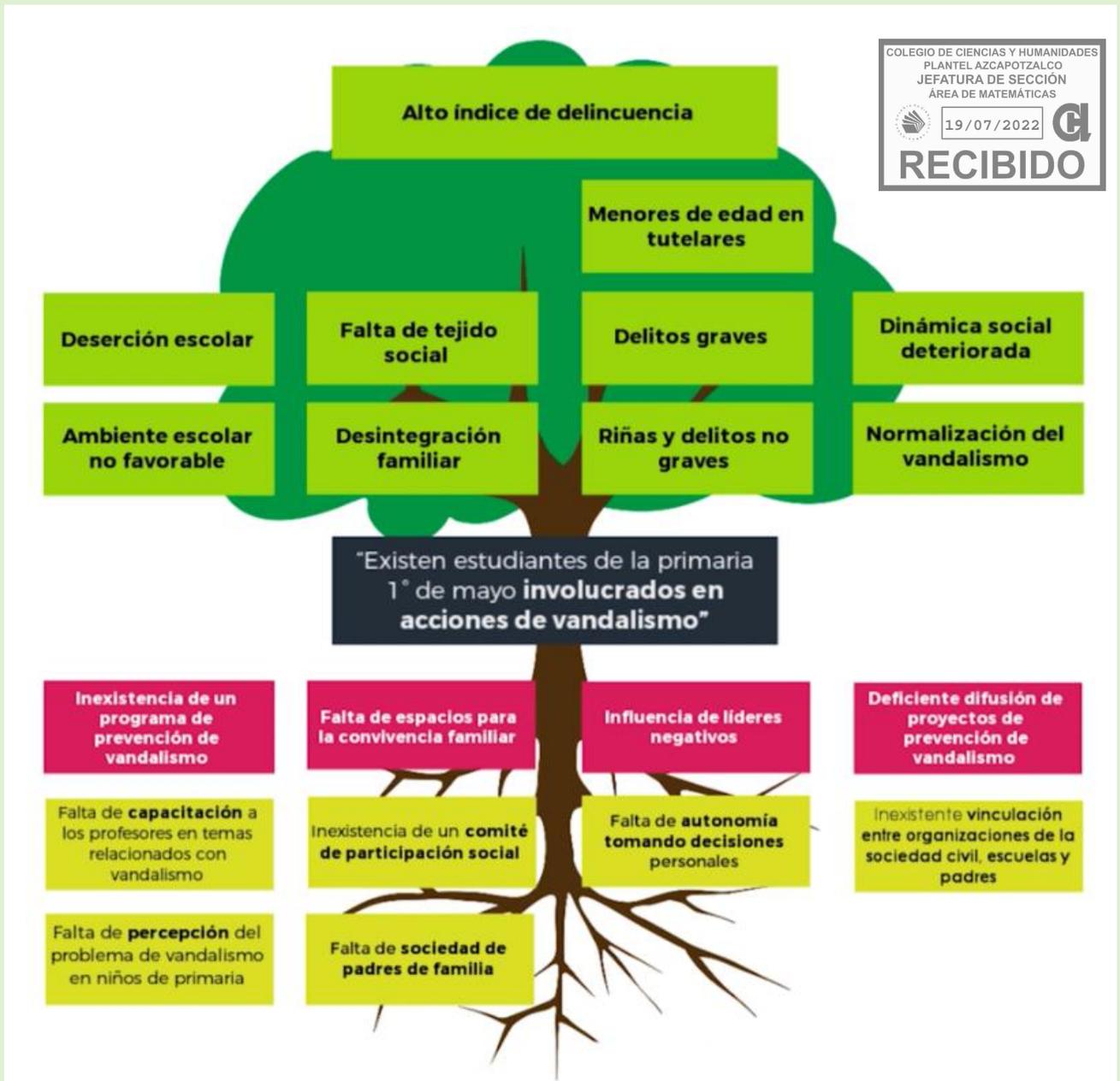
COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALCO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS
19/07/2022
RECIBIDO



Anexo 5. Ejemplo Diagrama EPS



Anexo 6. Ejemplo Diagrama de árbol



COLEGIO DE CIENCIAS Y HUMANIDADES
 PLANTEL AZCAPOTZALCO
 JEFATURA DE SECCIÓN
 ÁREA DE MATEMÁTICAS

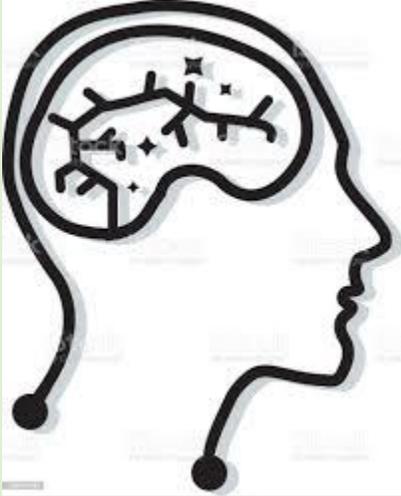
19/07/2022

RECIBIDO

Anexo 7. Formato Conclusión

Es momento de hacer un alto para que tu cabeza ponga orden sobre las actividades que acabamos de realizar,

Realiza un breve escrito donde desarrolles con tus propias palabras los Conceptos generales de un problema.



¡A escribir!

¡OJO! Se revisará que utilices los conceptos claves para desarrollar la temática, considerando que:

- Los conceptos sean expresados de una manera clara y organizada.
- La redacción facilite entender la información.
- Hagas una descripción completa y detallada del contenido y los elementos vistos.
- Utilices correctamente los signos de puntuación y no tu texto no presente errores ortográficos.

ESTRATEGIA DIDÁCTICA 2

I. DATOS GENERALES

PROFESOR(A)	Seminario Institucional de Propuesta educativa
ASIGNATURA	Cibernética y Computación
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Mayo de 2022
NO DE ALUMNOS	25



II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Describe la diferencia entre problemas determinísticos, probabilísticos, secuenciales y condicionales
TEMA(S)	<ul style="list-style-type: none"> • Tipos de problema: <ul style="list-style-type: none"> ○ Determinísticos. ○ Probabilísticos. ○ Secuenciales. ○ Condicionales.

III. ESTRATEGIA

Las actividades de aprendizaje se apoyarán en el uso de la plataforma educativa, se realizará videoconferencia si las circunstancias sanitarias lo exigen y/o en su caso en forma presencial.

A través de ejercicios y juegos, el estudiante analiza e identifica en ejemplos de la vida real, los tipos de problemas y describe los motivos por los que asigna dicha categoría.



IV. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	120 minutos. <ul style="list-style-type: none">• Inicio: 30 minutos• Desarrollo: 60 minutos• Cierre: 30 minutos.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none">• Computadora con acceso a internet.• Plataforma educativa.• Cuenta del profesor y alumno de Google.• Presentación electrónica "Tipos de problemas".• Documento. Actividad 1. Análisis de problemas inicial• Documento Actividad 2. Identificación de tipos de problemas y descripción• Objeto: Bomba de Problemas ó app Boom Quiz La Bomba

**DESARROLLO Y
ACTIVIDADES**

Actividad previa a la sesión:

El Profesor: configura las fechas de apertura y cierre del tema

“Tipos de problemas” en plataforma educativa y se aseguran que los materiales y recursos estén disponibles para su uso en plataformas seleccionadas.



Inicio

Tiempo estimado: 30 minutos.

El Profesor: (Grupal)

Tiempo estimado: 10 minutos

- Da la bienvenida a la clase.
- Realiza el encuadre de la sesión.

Actividad 1. Análisis de problemas inicial

Tiempo estimado: 20 minutos

Inicio - Actividad 1. Análisis de problemas inicial

El profesor: (Grupal)

- Presenta los Diagramas EPS (ANEXO 1) y solicita al alumno que para caso indique el problema, las entradas y procesos, a partir de la salida esperada.

Desarrollo - Actividad 1. Análisis de problemas inicial

Los estudiantes: (Individual)

- Resuelven el ejercicio.

Cierre - Actividad 1. Análisis de problemas inicial

El profesor (Grupal):

- Mediante participación voluntaria, al azar, o designación; exhorta a los estudiantes a exponer su diagrama en sesión plenaria.

El alumno (Grupal por equipo):

- Explica su actividad
- Comenta y realiza aportaciones a la actividad presentada por otros alumnos.

Fin Actividad 1.

Desarrollo

Tiempo estimado: 60 minutos.

El profesor (Grupal):

Tiempo estimado: 20 minutos.

- Con ayuda de una presentación digital expone los tipos de problemas computacionales y presenta ejemplos

Actividad 2. Identificación de tipos de problemas y descripción

Tiempo estimado: 40 minutos.

Inicio - Actividad 2. Identificación de tipos de problemas y descripción

El profesor (Grupal):

- Proporciona una serie de ejercicios a través de la plataforma educativa (**ANEXO 2**), para ser llenados por el alumno.
- Brinda las indicaciones para el llenado del documento.
- Presenta un ejemplo resuelto de los diagramas (**ANEXO 3**).
- De ser necesario crea salas de equipos reducidos, permite la integración de equipos o los asigna directamente.

Desarrollo - Actividad 2. Identificación de tipos de problemas y descripción

El alumno (En equipos reducidos):

- Forma equipos de 5 alumnos.
- Un integrante comparte el esquema en drive con sus compañeros
- Anotan los nombres de los integrantes del equipo y asignan un nombre de equipo.
- Identifican el tipo de problema para cada caso.
- Identifican las causas, insumos o entradas; operaciones o tratamiento efectuados; así como los efectos, productos o salidas para cada problema.
- Describen las razones o motivos de porque asignaron ese tipo de problema.
- Realizan un mapa mental del tema.
- Seleccionan a un integrante, quien expondrá el producto obtenido.

El profesor (En equipos reducidos):

- Visita de forma presencial o virtual (salas de equipos reducidos) a los equipos.
- Resuelve dudas, comenta y realiza aportaciones como integrante de los equipos de trabajo.

Cierre - Actividad 2. Identificación de tipos de problemas y descripción

El profesor (Grupal):



- Mediante participación voluntaria, al azar, o designación; exhorta a los equipos a compartir una sección de su actividad en plenaria.

El alumno (Grupal, en equipos reducidos):

- Expone su actividad.
- Comenta y realiza aportaciones al material presentado por otros equipos.



Cierre

Tiempo estimado: 30 minutos.

El profesor (Grupal):

- Bomba de Problemas (Versión presencial) (**Anexo 3**).
 - Crea la “Bomba de problemas”.
 - Escribir en diferentes papeles preguntas y ejemplos del tema.
 - Con papel periódico crear una esfera y pegar una pregunta sobre ella, se agrega una capa de periódico, sobre la nueva capa se pega otra pregunta.
 - Podrá agregar tantas capas como desee. Al final se pinta de negro para que luzca como una bomba o puede pegar una imagen para que luzca más realista.
 - Brinda indicaciones del juego “Bomba de problemas”.
 - Lanza la bomba al alumno, quien deberá descubrir la pregunta y contestarla.
 - En caso de que la respuesta no sea correcta, recibirá un castigo.
 - El castigo lo determina la persona que le lanzó la bomba y que fue el último que contestó correctamente.
- Bomba de Problemas (Versión en línea) (**Anexo 4**)
 - Descarga la app Boom Quiz La Bomba
 - Captura las preguntas y posibles respuestas.
 - Solicita a los estudiantes la activación de Cámara y/o audio (para los castigos o retos).
 - Cada bomba tiene 4 hilos, solo uno de ellos hace explotar la bomba.
 - Selecciona las 3 respuestas incorrectas para quitar los cables que no hacen explotar la bomba.

El alumno (individual):

- Lanza o recibe la bomba de problemas.
- Participa contestando y cubriendo los retos.

Extra - clase

Tiempo estimado: 10 minutos.

El alumno (individual):

- Sube a la plataforma educativa sus ejercicios para evaluación:
 - Actividad 1. Análisis de problema Inicial.
 - Actividad 2. Identificación de tipos de problemas y descripción.



EVALUACIÓN

Rúbrica: Actividad 1. Análisis de problemas inicial

Criterios	Totalmente adecuado 2pts.	Parcialmente adecuado 1 pt.	Inadecuado 0 pts.
Nombre asignado al problema			
Ortografía, puntuación y gramática			
Identifica los elementos de entrada			
Identifica los procesos			
Identifica los elementos de salida			

Rúbrica Actividad 2. Identificación de tipos de problemas y descripción

Criterios	Totalmente adecuado 2pts.	Parcialmente adecuado 1 pt.	Inadecuado 0 pts.
Tipo de problema			
Identifica los elementos de entrada			

	Identifica los procesos Identifica los elementos de salida Argumentos para la elección del problema	
--	---	--



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<ul style="list-style-type: none"> Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill. Portal Académico del CCH (18 de mayo de 2022). Tipos de problemas https://portalacademico.cch.unam.mx/cibernetica1/metodologia-resolucion-problemas/tipos-problemas Colegio de Ciencias y Humanidades. (18 de mayo de 2022). Guía de estudio de Cibernética y Computación I y II. https://www.unamenlinea.unam.mx/recurso/83615-guia-para-el-estudiante-de-cibernetica-y-computacion Curso Java: UdeMy [en línea] Programación para principiantes: Lógica para resolver problemas. (2022) https://www.udemy.com/course/programacion-para-principiantes/
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<ul style="list-style-type: none"> Wu, C. (2001). Introducción a la programación orientada a objetos con Java. Madrid: McGraw Hill/ Interamericana de España. Wu, C. (2008). Programación en Java. México: McGraw Hill

V. ANEXOS

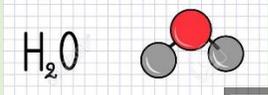
ANEXO 1. Actividad Inicial.

A partir de la imagen de salida, determina las entradas y el proceso

Ejercicio 1.

¿Cuál es el problema?: _____

Entrada	Proceso	Salida
---------	---------	--------

		
--	--	---



Ejercicio 2.

¿Cuál es el problema?: _____

Entrada	Proceso	Salida
		

Ejercicio 3.

¿Cuál es el problema?: _____

Entrada	Proceso	Salida
		

¿Fue igual la forma de resolver los problemas? ¿Si, No? ¿Porqué?:

ANEXO 2. Actividad 2. Identificación de tipos de problemas y descripción

Nombre del
Equipo: _____

Fecha: / /

Integrantes:

1.-

2.-

3.-

4.-

5.-



Ejercicio 1.

Problema: Si un maratonista participara en una maratón 35K (35 kilómetros) ¿cuál sería el tiempo esperado para culminar la carrera en horas, minutos y segundos

Tipo de Problema asignado:

Entrada	Proceso	Salida

Argumentos para seleccionar el tipo de problema:

Ejercicio 2.

Problema: En una tienda departamental se hace un descuento del 5% a los clientes cuya compra supera el monto de \$100,000. ¿Cuál será la cantidad que pagará una persona por su compra?

Tipo de Problema asignado:

Entrada	Proceso	Salida



Argumentos para seleccionar el tipo de problema:

ANEXO 3. Bomba de periódico.



Es una característica de los problemas determinísticos

ANEXO 4. App Boom Quiz La Bomba en Google Play

Boom Quiz La Bomba

Cadev Games Preguntas y respuestas

E Apto para todo público

Contiene anuncios · Ofrece compras directas desde la app

Esta app está disponible para algunos de tus dispositivos

Puedes compartir este contenido con tu grupo familiar.
[Más información sobre la Biblioteca familiar](#)

Agregar a la lista de deseos

Instalar

COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALCO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS
19/07/2022
RECIBIDO

281

284

284

140

¿Cómo se llamaba el villano de la clásica serie animada "Los pitufos"?

00:16:24

1

Gargamel Skeletor

Mumm-Ra Asrael

Gargamel

Mumm-Ra

ESTRATEGIA DIDÁCTICA 3

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	2 de abril de 2022
NO DE ALUMNOS	25



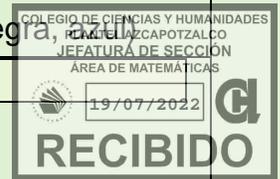
II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java.
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Conoce las etapas de la metodología de solución de problemas. • Conoce el concepto de algoritmo, diagrama de flujo y pseudocódigo. • Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas secuenciales.
TEMA(S)	<ul style="list-style-type: none"> • Etapas de la metodología de solución de problemas. • Concepto de algoritmo, diagrama de flujo y pseudocódigo. • Elaboración de algoritmos secuenciales. • Representación de algoritmos secuenciales a través de diagramas de flujo y pseudocódigo.

III.ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 sesiones de clase, de 120 minutos cada una.
---------------	---

MATERIALES Y/O RECURSOS	Computadora y servicio de internet Software Power Point, PSeInt Cuaderno, lápiz, goma, regla, colores, plumas (roja, negra, azul)
DESARROLLO Y ACTIVIDADES	<p>Sesión 1</p> <p>Inicio (30 minutos)</p> <p>El profesor:</p> <ol style="list-style-type: none"> 1. Expone la metodología de solución de problemas. <p>El alumno:</p> <ol style="list-style-type: none"> 1. Toman nota en su cuaderno sobre la metodología de solución de problemas. <p>Desarrollo (60 minutos)</p> <p>El profesor:</p> <ol style="list-style-type: none"> 1. Enuncia los siguientes problemas secuenciales para que los alumnos en equipo los resuelvan utilizando la metodología descrita y solicita entreguen un documento con la solución desarrollando los siguientes aspectos: <ol style="list-style-type: none"> 1. Planteamiento. 2. Análisis. 3. Diseño de la solución. <ul style="list-style-type: none"> • Algoritmo • Diagrama de flujo <p>Problema 1: Se deja caer un objeto desde la azotea de un edificio que tiene una altura de 12 m. Calcular en que tiempo toca el piso utilizando la ecuación $h = (g \cdot t^2) / 2$</p> <p>Problema 2: Calcular la altura de un árbol si un niño se encuentra a 3 metros de distancia de él volando un papalote que tiene un hilo de 5 metros.</p> <p>Problema 3. Convertir 100 °C a grados Kelvin.</p> <p>Problema 4: Calcular el perímetro y el área de un heptágono sabiendo que el lado mide 8 cm. y la apotema 8.30 cm.</p> <p>El alumno:</p> <ol style="list-style-type: none"> 1. Resuelve en equipo los problemas utilizando la metodología descrita. <p>Cierre (30 minutos)</p> <p>El profesor:</p> <ol style="list-style-type: none"> 1. Solicita a cada equipo explicar la solución de los problemas. <p>El alumno:</p>



1. Explica en equipo la solución de los problemas y compara resultados.

Actividad extra clase

El profesor:

1. Indica al alumno realizar un mapa mental de la metodología de solución de problemas, en alguna aplicación o en el cuaderno incluyendo imágenes, consultar las referencias y entregar un archivo pdf legible.

El alumno:

1. Elabora un mapa mental de la metodología en alguna aplicación o en el cuaderno, debe incluir el nombre del alumno y grupo, algunas imágenes, ser legible y entregarse en formato pdf.

Sesión 2

Inicio (40 minutos)

El profesor:

1. Solicita al alumno ver los siguientes videos que explican el programa PSeInt y presenta la documentación que tiene el programa PSeInt:

- <https://www.youtube.com/watch?v=8dHk2LJDYug>
- <https://www.youtube.com/watch?v=b-DhlwRQ9nk&list=PLAzISdU-KYwXIIxcUCW-BylQZemcDV798&index=8>

El alumno:

1. Ve los videos y revisa la documentación de PSeInt guiado por el profesor.





Desarrollo (50 minutos)

El profesor:

1. Solicita al alumno elaborar el algoritmo de los problemas resueltos en la sesión pasada en PSeInt y entregar un documento con las capturas de pantalla del algoritmo, ejecución y diagrama de flujo.

El alumno:

1. Elabora el algoritmo de los problemas en PSeInt, los ejecuta y revisa el diagrama de flujo.

Cierre (30 minutos)

El profesor:

1. Solicita a algunos alumnos que presenten su algoritmo elaborado en PSeInt.

El alumno:

1. Presenta el algoritmo ante el grupo y envía el documento al profesor para ser evaluado.

EVALUACIÓN

Evaluación formativa	Evidencia	Porcentaje
Procedimental	Solución del problema	40%
	Algoritmo en PSeInt	30%

Conceptual	Mapa mental	20%
Actitudinal	Trabajo en equipo y participación	10%



Lista de cotejo para la solución del problema

El alumno	Completa (1 punto)	Parcial (0.5 puntos)	Incompleta (0 puntos)
Elabora el planteamiento del problema con un dibujo.			
Elabora el análisis del problema con el modelo de caja negra			
Elabora el algoritmo del problema			
Elabora el diagrama de flujo del problema			

Lista de cotejo para la elaboración del algoritmo en PSeInt

El alumno	Completa (1 punto)	Parcial (0.5 puntos)	Incompleta (0 puntos)
Elabora el algoritmo del problema en PSeInt			
Elabora el diagrama de flujo del problema en PSeInt			
Realiza la ejecución del algoritmo en PSeInt			

IV. REFERENCIAS DE APOYO

<p>BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS</p>	<p>Academia Play, (2016). 10 consejos para hacer un mapa mental. [en línea]. Desarrollo personal y Liderazgo. Recuperado el 20 de octubre de 2021 en https://www.facebook.com/watch/?v=1123770017653918&hashtag=mapa-mental</p> <p>Ávila, S. et. Al. (2019). Paquete didáctico para la asignatura de Cibernética y Computación I. México: ENCCH Plantel Oriente, UNAM. p. 166 – 169. Recuperado el 20 de octubre de 2021 en http://www.cch-oriente.unam.mx/material_didactico/Matematicas/paqueteCyC.pdf</p> <p>Code Labs Chile, (2018). Pseudocódigo: Creando nuestro primer programa en PseINT. [video en línea]. Recuperado el 21 de octubre de 2021 en https://www.youtube.com/watch?v=8dHk2LJDYug</p> <p>Danisable, (2019). Aprende a programar desde cero con PseInt! Asignación Parte 8. [video en línea] Recuperado el 21 de octubre de 2021 en https://www.youtube.com/watch?v=b-DhlwRQ9nk&list=PLAzISdU-KYwXIIxcUCW-BylQZemcDV798&index=8</p>
<p>BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR</p>	<p>Ávila, S. et. Al. (2019). Paquete didáctico para la asignatura de Cibernética y Computación I. México: ENCCH Plantel Oriente, UNAM. p. 166 – 169. Recuperado el 20 de octubre de 2021 en http://www.cch-oriente.unam.mx/material_didactico/Matematicas/paqueteCyC.pdf</p> <p>Danisable, (2019) Curso PseInt. [video en línea]. Recuperado el 21 de octubre de 2021 en https://www.youtube.com/playlist?list=PLAzISdU-KYwXIIxcUCW-BylQZemcDV798</p>



ESTRATEGIA DIDÁCTICA 4

I. DATOS GENERALES

PROFESOR(A)	Seminario Institucional de Propuesta educativa
ASIGNATURA	Cibernética y Computación
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades.
FECHA DE ELABORACIÓN	mayo de 2022
NO DE ALUMNOS	25



II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Analiza el resultado de expresiones aritméticas utilizando la jerarquía de las operaciones.
TEMA(S)	<ul style="list-style-type: none"> • Expresiones y operadores aritméticos: <ul style="list-style-type: none"> ○ Asignación. ○ Operadores aritméticos. ○ Jerarquía de operadores aritméticos. ○ Evaluación de expresiones aritméticas.

III. ESTRATEGIA

Las actividades de aprendizaje se apoyarán en el uso de la plataforma educativa, se realizará videoconferencia si las circunstancias sanitarias lo exigen y/o en su caso en forma presencial.

A través de ejercicios y juegos, el estudiante identifica los símbolos utilizados como operadores aritméticos realizar operaciones matemáticas y cálculos, asimismo, describe el orden en el que estos se resuelven para llegar a un resultado.



IV. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	<p>90 minutos.</p> <ul style="list-style-type: none"> • Inicio: 15 minutos • Desarrollo: 60 minutos • Cierre: 15 minutos.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Computadora con acceso a internet. • Plataforma educativa. • Cuenta del profesor y alumno de Google. • Presentación electrónica “Expresiones y Operadores en algoritmos”. • Lectura: “Operadores para algoritmos” de Verónica Márquez en la plataforma educativa (https://www.uaeh.edu.mx/scige/boletin/prepa2/n7/m2.html). • Vídeo: “Operadores”. (https://www.youtube.com/watch?v=6cprJILuARw)
DESARROLLO Y ACTIVIDADES	<p>Actividad previa a la sesión:</p> <p>El Profesor: configura las fechas de apertura y cierre del tema “Expresiones y operadores aritméticos” en plataforma educativa y se asegura que los materiales y recursos estén disponibles para su uso en plataformas seleccionadas.</p> <p>Inicio</p> <p>Tiempo estimado: 15 minutos.</p> <p>El Profesor: (Grupal)</p> <ul style="list-style-type: none"> • Da la bienvenida a la clase. • Realiza el encuadre de la sesión. <p>Desarrollo</p>

Actividad 1. Interacción entre datos

Tiempo estimado: 20 minutos

Inicio - Actividad 1. Interacción entre datos

El profesor: (Grupal)

- Solicita a los estudiantes iniciar la lectura de los “Operadores para algoritmos” de Verónica Márquez en la plataforma educativa (<https://www.uaeh.edu.mx/scige/boletin/prepa2/n7/m2.html>)
- A continuación, solicita que se observe el vídeo “Operadores” dispuesto en la plataforma educativa. (<https://www.youtube.com/watch?v=6cprJILuARw>).
- Al finalizar el vídeo, resuelve dudas y amplía la información.

Los estudiantes: (Grupal)

- Realizan la lectura “Operadores para algoritmos”.
- Observan el vídeo ¿Qué es un operador?

Desarrollo - Actividad 1. Interacción entre datos

El profesor: (Grupal)

- Da lectura a las siguientes preguntas y selecciona al estudiante que deberá responder.
 - ¿Cuáles son las formas para representar datos en un algoritmo o programa?
 - ¿Cómo asignas un valor a una variable o una constante?
 - ¿Qué es un operador?
 - ¿Qué uso se le confiere a un operador en la programación?
 - ¿A qué se refiere la jerarquía de los operadores?
 - ¿Por qué crees que sea importante la jerarquía de los operadores aritméticos?

Los estudiantes: (Grupal)

- En sesión plenaria dan respuesta y complementan las participaciones de los compañeros.

El profesor (Grupal):



- Realiza una síntesis de los materiales vistos hasta el momento
- Comparte en pantalla el documento colaborativo con los nombres de los operadores aritméticos y la completa con ayuda de los estudiantes.



Operador Aritmético	Operación	Ejemplo con Literales	Ejemplo con variables	Asignación	Tipo dato	Jerarquía
	Asignación					
	Suma					
	Resta					
	Multiplicación					
	División					
	Módulo					
	Potencia					
	Paréntesis					

Con base a la tabla completada pide a los estudiantes a evaluar las siguientes expresiones aritméticas, utilizando la tabla de jerarquía de operadores aritméticos.

Expresión	Orden en que se resuelve	Resultado
$10 + 9 - 4$	a. En este caso la suma y la resta se encuentran en el mismo nivel de jerarquía, por ello se comienza de izquierda a derecha. b. Efectúas la operación $10 + 9$ igual a 19. c. Al resultado le restas 4 y obtienes 15. d. 15 es el resultado de la operación.	15
$(5^3 * 4) - (6 + 9)$	a. Tienes dos grupos de operaciones, comenzarás de izquierda a derecha. b. En el primer grupo $(5^3 * 4)$ la potencia tiene mayor jerarquía que el producto, 5^3 es igual a 125. c. 125 lo multiplicas por 4 y da 500. d. Ahora el siguiente grupo de operaciones $6 + 9$ obtienes 15.	485

	e. Juntas ambos resultados, 500 menos 15 es igual a 485.	
	f. 485 es el resultado de la operación.	
$A = A + 1$		
$x = (12 + 6) / 2 * 3$		
$y = x = (2 + 3) / 4$		
$3x9 + (6 + 5 - 3) - 12/4$		
$40 - [30 + 6(19-12)]$		



Cierre - Actividad 1. Interacción entre los datos

El profesor (Grupal):

- Mediante participación voluntaria, al azar, o designación; exhorta a los estudiantes a compartir sus resultados.
- Pública la tabla de operadores aritméticos en la plataforma para consulta de los estudiantes.

El alumno (Grupal por equipo):

- Explica su actividad
- Comenta y realiza aportaciones a la actividad presentada por otros alumnos.

Fin Actividad 1.

Cierre

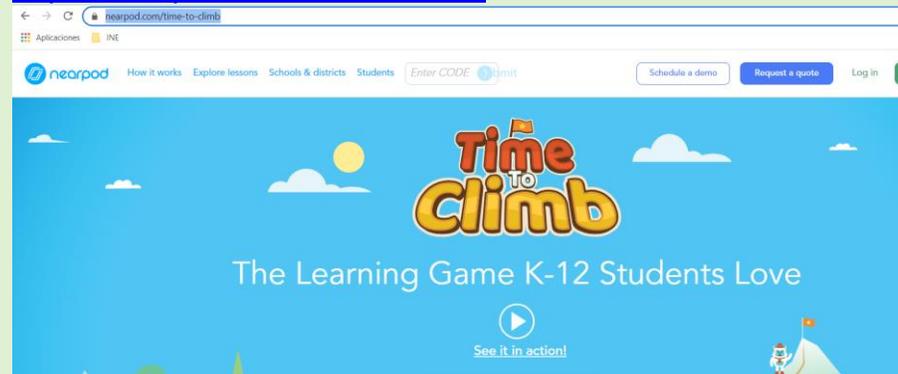
Tiempo estimado: 15 minutos.

Juego Time to Climb (Carrera virtual de avatars)

El profesor (Grupal):

Crea cuenta en aplicación Time to Climb en

<https://nearpod.com/time-to-climb>



Crea el escenario y carga las preguntas en la aplicación.

	<p>Configura la apertura de la sala. Explica la dinámica a los estudiantes y da el banderazo de salida.</p> <p>El alumno (individual):</p> <ul style="list-style-type: none"> • Se registra en nearpod con cuenta de trabajo de Google. • Selecciona y configura su personaje. • Participa contestando y cubriendo los retos. <p>Extra - clase</p> <p>Tiempo estimado: 10 minutos.</p> <p>El alumno (individual):</p> <ul style="list-style-type: none"> • Sube a la plataforma educativa sus ejercicios para evaluación: <ul style="list-style-type: none"> ○ Ejercicios de operaciones aritméticas. 												
EVALUACIÓN	<p>Rúbrica: Actividad 1. Análisis de problemas inicial</p> <table border="1" data-bbox="457 869 1399 1060"> <thead> <tr> <th data-bbox="457 869 695 974">Criterios</th> <th data-bbox="695 869 954 974">Correcto 2pts.</th> <th data-bbox="954 869 1214 974">Parcialmente Correcto 1 pt.</th> <th data-bbox="1214 869 1399 974">Incorrecto 0 pts.</th> </tr> </thead> <tbody> <tr> <td data-bbox="457 974 695 1060">El resultado anotado</td> <td data-bbox="695 974 954 1060"></td> <td data-bbox="954 974 1214 1060">NA</td> <td data-bbox="1214 974 1399 1060"></td> </tr> <tr> <td data-bbox="457 1060 695 1060">Orden con que se resuelve</td> <td data-bbox="695 1060 954 1060"></td> <td data-bbox="954 1060 1214 1060"></td> <td data-bbox="1214 1060 1399 1060"></td> </tr> </tbody> </table>	Criterios	Correcto 2pts.	Parcialmente Correcto 1 pt.	Incorrecto 0 pts.	El resultado anotado		NA		Orden con que se resuelve			
Criterios	Correcto 2pts.	Parcialmente Correcto 1 pt.	Incorrecto 0 pts.										
El resultado anotado		NA											
Orden con que se resuelve													



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • Márquez, Verónica (2017). Operadores para algoritmos. Universidad Autónoma del Estado de Hidalgo. • Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill. • Colegio de Ciencias y Humanidades. (18 de mayo de 2022). Guía de estudio de Cibernética y Computación I y II. https://www.unamenlinea.unam.mx/recurso/83615-guia-para-el-estudiante-de-cibernetica-y-computacion
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • Duran, F.J. (2007). Programación Orientada a Objetos con Java. España: Paraninfo.

- Wu, C. (2001). Introducción a la programación orientada a objetos con Java. Madrid: McGraw Hill/ Interamericana de España.
- Wu, C. (2008). Programación en Java. México: McGraw Hill



V. ANEXOS

ESTRATEGIA DIDÁCTICA 5

DATOS GENERALES

PROFESOR(A)	Seminario Institucional de Propuesta educativa
ASIGNATURA	Cibernética y Computación
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades.
FECHA DE ELABORACIÓN	mayo de 2022
NO DE ALUMNOS	25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Construye expresiones lógicas utilizando operadores relacionales y lógicos.
TEMA(S)	<ul style="list-style-type: none"> • Expresiones y operadores relacionales y lógicos: <ul style="list-style-type: none"> ○ Operadores relacionales. ○ Operadores lógicos. ○ Jerarquía de operadores lógicos. ○ Evaluación de expresiones lógicas.

III. ESTRATEGIA

Las actividades de aprendizaje se apoyarán en el uso de la plataforma educativa, se realizará videoconferencia si las circunstancias sanitarias lo exigen y/o en su caso en forma presencial.

A través de ejercicios y juegos, el estudiante identifica los símbolos utilizados como operadores relacionales y lógicos, asimismo, describe el orden en el que estos se resuelven para llegar a un resultado.



IV. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	<p>90 minutos.</p> <ul style="list-style-type: none"> • Inicio: 15 minutos • Desarrollo: 50 minutos • Cierre: 15 minutos.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Computadora con acceso a internet. • Plataforma educativa. • Cuenta del profesor y alumno de Google. • Presentación electrónica “Expresiones y operadores en algoritmos”. • Lectura: “Operadores para algoritmos” de Verónica Márquez en la plataforma educativa (https://www.uaeh.edu.mx/scige/boletin/prepa2/n7/m2.html). • Vídeo: “Operadores lógicos” intervenido por h5p. (https://www.youtube.com/watch?v=Ly8sJepm9Ko) • Vídeo “Operadores relacionales” intervenido por h5p (https://www.youtube.com/watch?v=puWMS6qbdjM)
DESARROLLO Y ACTIVIDADES	<p>Actividad previa a la sesión: El Profesor: configura las fechas de apertura y cierre del tema “Expresiones y operadores aritméticos” en plataforma educativa y se asegura que los materiales y recursos estén disponibles para su uso en plataformas seleccionadas.</p> <p>Inicio Tiempo estimado: 15 minutos. El Profesor: (Grupal)</p> <ul style="list-style-type: none"> • Da la bienvenida a la clase. • Realiza el encuadre de la sesión. • Retoma el diagrama de la lectura “Operadores para algoritmos” de Verónica Márquez en la plataforma educativa

(<https://www.uaeh.edu.mx/scige/boletin/prepa2/n7/m2.html>)

Desarrollo

Tiempo estimado: 50 minutos



El profesor: (Grupal)

- Interviene los vídeos mediante h5p y los publica en la plataforma educativa para integrar preguntas y reflexiones durante la emisión de los vídeos, tales como:
 - ¿Cuántos valores pueden tomar los operadores lógicos?
 - ¿Cuántos y cuáles son los operadores lógicos?
 - ¿Qué símbolo está asociado a la operación disyunción?
 - ¿Cuáles son los tipos de condiciones que se pueden presentar?
 - ¿Cuál es la condición para que la operación conjunción devuelva 1 como resultado?
 - ¿Cuáles son los posibles valores que pueden tomar las expresiones lógicas?
 - ¿Cuál es la diferencia entre el símbolo “=” y “==” ?Etc.
- Solicita que se observe el vídeo “Operadores lógicos” dispuesto en la plataforma educativa. (<https://www.youtube.com/watch?v=Ly8sJepm9Ko>) y llegado el momento, selecciona a uno de los estudiantes para que conteste la pregunta.
- Al finalizar el vídeo, resuelve dudas y amplía la información.
- A continuación, solicita que se observe el vídeo “Operadores Relacionales” dispuesto en la plataforma educativa (<https://www.youtube.com/watch?v=puWMS6qbdjM>) y llegado el momento, selecciona a uno de los estudiantes para que conteste la pregunta.

Los estudiantes: (Grupal)

- Observan los vídeos y contestan las preguntas conforme vayan siendo seleccionados por el profesor.

Cierre

Tiempo estimado: 15 minutos.

Juego Time to Climb (Carrera virtual de avatares)

El profesor (Grupal):

	<p>Crea cuenta en aplicación Time to Climb en https://nearpod.com/time-to-climb</p>  <p>Crea el escenario y carga los ejercicios en la aplicación. Configura la apertura de la sala. Explica la dinámica a los estudiantes y da el banderazo de salida.</p> <p>El alumno (individual):</p> <ul style="list-style-type: none"> • Se registra en nearpod con cuenta de trabajo de Google. • Selecciona y configura su personaje. • Participa contestando y cubriendo los retos. <p>Extra - clase Tiempo estimado: 10 minutos. El alumno (individual):</p>
EVALUACIÓN	<p>Juego Time to Climb (Carrera virtual de avatares)</p> <p>Se vincula la aplicación con la plataforma para que capture la puntuación del juego. Se espera que el profesor coloque 10 ejercicios de expresiones lógicas y relacionales. Cada pregunta equivale a un punto.</p>

IV. REFERENCIAS DE APOYO

<p>BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS</p>	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • Márquez, Verónica (2017). Operadores para algoritmos. Universidad Autónoma del Estado de Hidalgo. • Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw-Hill.
---	---

ESTRATEGIA DIDÁCTICA 6

I. DATOS GENERALES

PROFESOR(A)	Seminario Institucional de Propuesta educativa
ASIGNATURA	Cibernética y Computación
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades. Plantel Sur.
FECHA DE ELABORACIÓN	mayo de 2022
NO DE ALUMNOS	25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Conoce el concepto de algoritmo, diagrama de flujo y pseudocódigo. • Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas secuenciales.
TEMA(S)	<ul style="list-style-type: none"> • Concepto de algoritmo, diagrama de flujo y pseudocódigo. • Elaboración de algoritmos secuenciales. • Representación de algoritmos secuenciales a través de diagramas de flujo y pseudocódigo.

III. ESTRATEGIA

Las actividades de aprendizaje se apoyarán en el uso de la plataforma educativa, se realizará videoconferencia si las circunstancias sanitarias lo exigen y/o en su caso en forma presencial.

El alumno adquiere los conceptos generales de la programación y se enfrenta a problemas sencillos que someterá a la Metodología de solución de problemas y propondrá soluciones algorítmicas concretadas en pseudocódigos y diagramas de flujo.



IV. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	120 minutos. <ul style="list-style-type: none">• Inicio: 20 minutos• Desarrollo: 50 minutos• Cierre: 15 minutos.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none">• Computadora con acceso a internet.• Plataforma educativa.• Presentación electrónica “Expresiones y operadores en algoritmos”.• Vídeo capsula “Estructuras algorítmicas de programación”. (https://www.youtube.com/watch?v=rNY5eWogI18).• Infografía interactiva “Estructura condicional IF” https://view.genial.ly/5e4e3123c66c4d7197700d96/interacti-ve-content-2020-javier-estructuraif
DESARROLLO Y ACTIVIDADES	<p>Actividad previa a la sesión: El Profesor: configura las fechas de apertura y cierre del tema “algoritmo, diagrama de flujo y pseudocódigo” en la plataforma educativa y se asegura que los materiales y recursos estén disponibles para su uso en plataformas seleccionadas.</p> <p>Inicio Tiempo estimado: 20 minutos. El Profesor: (Grupal)</p> <ul style="list-style-type: none">• Da la bienvenida a la clase.• Realiza el encuadre de la sesión.• A continuación, solicita al alumno que observe detenidamente las siguientes imágenes.

Uso de la mascarilla



Lave sus manos con jabón y agua o gel hidroalcohólico



Localice la parte delantera de la mascarilla



Cubra completamente su nariz y boca



Evite tocar la parte frontal mientras la lleva puesta



Tire la mascarilla a la papelera



Lave sus manos con alcohol o gel hidroalcohólico

lifeder.com

COLEGIO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALEO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS

19/07/2022

RECIBIDO

ENSALADA DE ZANAHORIA

Ingredientes:

- 3 piezas de zanahoria.
- 1 limón.
- Sal.

Preparación:

1. Lave y desinfecta la zanahoria y el limón.
2. Quite la cáscara a la zanahoria y rállala.
3. Exprime el limón y agrégalo a la zanahoria.
4. Añade sal al gusto y mezcla todo muy bien.

Recomendaciones

Al preparar esta ensalada puedes combinarla con jicama rallada, o sustituir la zanahoria por pepinos.



INSTRUCTIVO DE INSCRIPCIÓN



Dale click en tu categoría

1

ÍNDICE	
INSTRUCCIONES PARA PERSONAS SIN VISA	PÁGINA 2-11
INSTRUCCIONES PARA PERSONAS CON VISA	PÁGINA 12-17



- Realiza las siguientes preguntas:
 - ¿Te has encontrado con imágenes con estás?
 - ¿Dónde?
 - ¿Qué tienen en común?
 - ¿Para qué sirve?
 - ¿Son útiles estas imágenes para quien las observa?
 - ¿Quién las elabora?
 - ¿Por qué y para qué las elaboran? ¿Cuál es el objetivo?
 - ¿Qué pasaría si no existieran dichas imágenes?
 - ¿Cómo relacionas estas imágenes con la elaboración de programas informáticos?

El alumno: (Grupal)

- A través de participación voluntaria, contesta las preguntas y complementa las respuestas de los compañeros.

Desarrollo

Tiempo estimado: 50 minutos

El algoritmo

El profesor: (Grupal)

Presenta a los estudiantes la infografía ¿Qué es un algoritmo? De <http://www.ed.team/cursos/algoritmos>

INSTITUTO DE CIENCIAS Y HUMANIDADES
PLANTEL AZCAPOTZALEO
JEFATURA DE SECCIÓN
ÁREA DE MATEMÁTICAS
19/07/2022

¿QUÉ ES UN ALGORITMO?

Es la secuencia de pasos que resuelve un problema y es la base de la programación.
Prof. Alexys Lozada

PARTES DE UN ALGORITMO

ENTRADA
Son los datos que se le dan al algoritmo.

PROCESO
Operaciones que se hacen con los datos.

SALIDA
Resultado final que se obtiene de las operaciones, en este caso será 3.

CARACTERÍSTICAS

PRECISO
Tiene que resolver el problema sin errores.

DEFINIDO
Si ejecutas el algoritmo varias veces, los datos de salida serán iguales en cada repetición.

FINITO
Debe tener un inicio y un final.

LEGIBLE
Cualquier persona que vea el algoritmo debe ser capaz de comprenderlo.

Aprende como funcionan los algoritmos en:
ed.team/cursos/algoritmos

EDteam

Proyecta el vídeo ¿Qué es un algoritmo y para qué se usa? (<https://www.youtube.com/watch?v=EkObhToiseo&feature=youtu.be>).

- Proporciona las indicaciones a los estudiantes para la dinámica “Avión de papel”.
 - Solicita a los estudiantes que de manera individual busquen en internet la imagen de un algoritmo o procedimiento para la creación de un avión de papel”.
 - Se integran equipos de trabajo de 5 personas. En caso de requerirse se crearán salas reducidas en plataforma.

	<ul style="list-style-type: none"> ○ Anotarán sus respuestas y observaciones en la “Bitácora de prueba de algoritmos”. (Documento descargado de la plataforma educativa y que será compartido por los integrantes para su llenado de forma colaborativa). ○ Cada integrante, dictará el algoritmo a sus compañeros, quienes con el apoyo en una hoja reciclada realizarán el avión. Tomarán una fotografía de los resultados (aviones de papel) para cada caso y la colocarán en el apartado correspondiente en la bitácora. ○ Compararán con ayuda de la Bitácora los algoritmos y seleccionan el mejor. <ul style="list-style-type: none"> • Orienta, resuelve dudas y acompaña a los diferentes equipos durante la actividad. • Rescata la diversidad de soluciones que puede tener un problema, refuerza el concepto del algoritmo, la identificación de sus partes y la importancia de las características de un algoritmo para llegar a la solución de manera homogénea. <p>Los estudiantes: (Individual – en equipos reducidos)</p> <ul style="list-style-type: none"> • Individual - Buscará y seleccionará una imagen en internet, que refleje el algoritmo o procedimiento para hacer un avión de papel. • Integrará equipos de 5 personas. • Comparan los ejemplos encontrados y a través de un análisis deciden cuál de ellos funciona, cual no y obtienen el mejor. • Dictará el algoritmo de la imagen seleccionada a sus compañeros, quienes deberán elaborar el avión, a partir de las instrucciones y compararán los resultados obtenidos. • Identifican las partes del algoritmo en el ejemplo encontrado y dictaminarán si ¿El algoritmo encontrado cumple con las características del algoritmo? ¿Llegaron al mismo resultado? ¿Qué aspectos ayudaron a alcanzar el producto final? • Seleccionan a un integrante y exponen en plenaria sus resultados • Comentan y reflexionan sobre el trabajo de los otros equipos.
--	---



Herramientas de expresión o representación algorítmica

El profesor: (Grupal)

- A través de una presentación electrónica muestra las herramientas de expresión o representación algorítmica
 - Diagrama de Flujo de Datos
 - Pseudocódigo
- Inicia el programa PSeInt y explica la interfaz.
- Explica el proceso secuencial de lectura y ejecución.
- Con apoyo del programa PSeInt crea un archivo nuevo donde se explica la estructura inicial del Pseudocódigo, a continuación, se escriben mensajes de texto, numéricos, cálculos aritméticos básicos y la unión de textos y números



Cierre

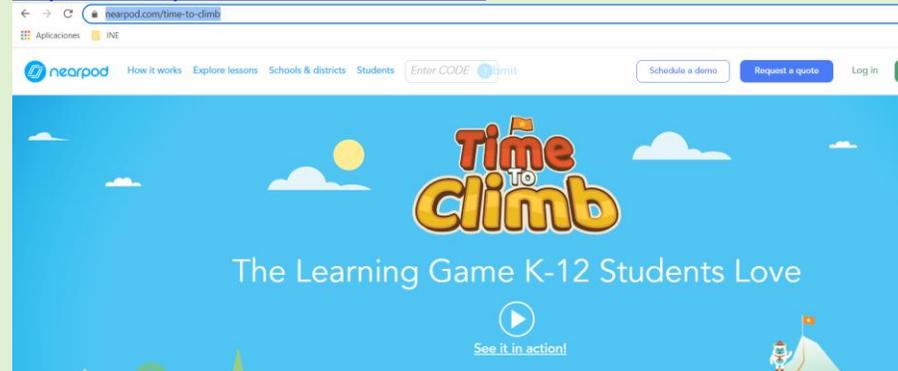
Tiempo estimado: 15 minutos.

Juego Time to Climb (Carrera virtual de avatares)

El profesor (Grupal):

Crea cuenta en aplicación Time to Climb en

<https://nearpod.com/time-to-climb>



Crea el escenario y carga los ejercicios en la aplicación.

Configura la apertura de la sala.

Explica la dinámica a los estudiantes y da el banderazo de salida.

El alumno (individual):

- Se registra en nearpod con cuenta de trabajo de Google.
- Selecciona y configura su personaje.
- Participa contestando y cubriendo los retos.

Extra - clase

Tiempo estimado: 10 minutos.

El alumno (individual):

EVALUACIÓN	<p>Juego Time to Climb (Carrera virtual de avatares)</p> <p>Se vincula la aplicación con la plataforma para que capture la puntuación del juego. Se espera que el profesor coloque ejercicios de algoritmos. Cada pregunta equivale a un punto.</p>
-------------------	--



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • Márquez, Verónica (2017). Operadores para algoritmos. Universidad Autónoma del Estado de Hidalgo. • Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill. • Colegio de Ciencias y Humanidades. (18 de mayo de 2022). Guía de estudio de Cibernética y Computación I y II. https://www.unamenlinea.unam.mx/recurso/83615-guia-para-el-estudiante-de-cibernetica-y-computacion
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • García-Beltrán (2009). Programación en Java. http://ocw.upm.es/pluginfile.php/1037/mod_label/intro/4-operadores.pdf

V. ANEXOS

ESTRATEGIA DIDÁCTICA 7

I. DATOS GENERALES

PROFESOR(A)	Seminario Institucional de Propuesta educativa
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades. Plantel Sur.
FECHA DE ELABORACIÓN	mayo de 2022
NO DE ALUMNOS	25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Realiza programas empleando el método de salida de datos. • Realiza programas empleando la Clase Scanner para la entrada de datos.
TEMA(S)	<ul style="list-style-type: none"> • Pasos para implementar un programa con el lenguaje de programación Java y el entorno de desarrollo. <ul style="list-style-type: none"> ○ Creación de un proyecto. ○ Declaración de la Clase. ○ Método main. ○ Empleo de los métodos System.out. print y System.out.println.

	<ul style="list-style-type: none"> ○ Errores sintácticos y lógicos. ○ Ejecución del programa. ● Introducción de datos desde el teclado. <ul style="list-style-type: none"> ○ La Clase Scanner. ○ Definición del objeto de la Clase Scanner. ○ Método System.in. ○ Tipos de datos primitivos.
--	--



III. ESTRATEGIA

Las actividades de aprendizaje se apoyarán en el uso de la plataforma educativa, se realizará videoconferencia si las circunstancias sanitarias lo exigen y/o en su caso en forma presencial.

El alumno realiza programas básicos en los que despliega contenidos e interactúa con el programa a través de la consola.

IV. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 clases de 120 minutos.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> ● Computadora con acceso a internet. ● Plataforma educativa. ● Presentación electrónica “Salida de datos”. ● Infografía interactiva “Salida de datos por consola”. ● Infografía interactiva “Scanner”
DESARROLLO Y ACTIVIDADES	<p>Actividad previa a la sesión:</p> <p>El Profesor: configura las fechas de apertura y cierre del tema “Realiza programas empleando el método de salida de datos” en plataforma educativa y se asegura que los materiales y recursos estén disponibles para su uso en plataformas seleccionadas.</p> <p>Clase 1 Inicio</p>

Tiempo estimado: 20 minutos.

El Profesor: (Grupal)

- Da la bienvenida a la clase.
- Realiza el encuadre de la sesión.

Desarrollo

Tiempo estimado: 85 minutos.

El Profesor: (Grupal)

- A continuación, comparte su pantalla a los estudiantes y juntos crean un nuevo proyecto de java, utilizando netbeans y/o puede utilizar aplicaciones en línea como https://www.onlinegdb.com/online_java_compiler.
- Retoma conceptos como proyecto, paquetes y clases.
- A través de la presentación digital “Salida de datos por consola”, explica al alumno la importancia de la comunicación entre el usuario y los programas.
- Explica el comando “System.out.print” y “System.out.println” con ayuda de la infografía “Salida de datos por consola”. <https://view.genial.ly/5db950bf0999ad0fd8ff7d81/interactive-content-2020-javier-salidadatosconsola>
- Comparte su pantalla con el alumno para crear proyectos para ejemplificar la impresión o desplegado en consola de: cadenas de textos, números, operaciones aritméticas básicas con números.
- Solicita al alumno realice los siguientes ejercicios, otorga el tiempo necesario para su realización, apoya a los alumnos con dudas y, al final de cada ejercicio, pide que compartan sus ejercicios en plenaria.

El alumno

- Crea un nuevo proyecto llamado “Triangulo” que produzca la siguiente salida en consola:

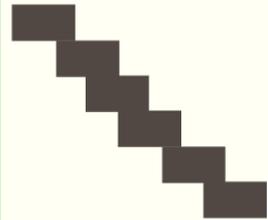


```
**** Programa que presenta tres Triángulos ****
**** Nombre del alumno: paterno materno nombre
**** Grupo: grupo
*
**
***
   *
  **
 ***
     *
    ***
   *****
```



- Elabora un nuevo proyecto llamado “Escalera”, que se apoyará en el código ASCII (<https://www.celeberrima.com/tabla-de-codigo-ascii-completa-para-imprimir/>) para presentar la siguiente salida por consola.

```
**** Programa que presenta una escalera ****
**** Nombre del alumno: paterno materno nombre
**** Grupo: grupo
```



- Elabora un nuevo proyecto llamado “Operaciones_Aritmeticas”, que utiliza dos variables enteras (valores asignados directamente) para presentar como resultado la suma, resta, multiplicación, división y módulo entre ambas variables y presentará los resultados por consola.

**** Programa que presenta las operaciones aritméticas entre dos variables****
**** Nombre del alumno: paterno materno nombre
**** Grupo: grupo
La variable 1 tiene almacenado el valor: 3
La variable 2 tiene almacenado el valor: 2
El resultado de la suma es: 5
El resultado de la resta es: 1
El resultado de la multiplicación es: 6
El resultado de la división es: 1.5
El resultado del módulo es: 1



Cierre 15 minutos

El Profesor: (Grupal)

- Con apoyo de la infografía <https://view.genial.ly/5e4e314bf4570c3ca9b991a4/interactiv-e-content-2020-javier-entradatosescaner> explica a los estudiantes el proceso y codificación de la entrada de datos desde el teclado.
- Retoma el ejemplo de las operaciones aritméticas entre dos variables y realiza las modificaciones necesarias para que los valores de las variables sean asignados por el usuario (Clase Scanner). La salida por consola debe lucir de la siguiente forma:

**** Programa que presenta las operaciones aritméticas entre dos variables****
**** Nombre del alumno: paterno materno nombre
**** Grupo: grupo
Dame el valor de la primera variable: 3
Dame el valor de la segunda variable: 2
La variable 1 tiene almacenado el valor: 3
La variable 2 tiene almacenado el valor: 2
El resultado de la suma es: 5
El resultado de la resta es: 1
El resultado de la multiplicación es: 6
El resultado de la división es: 1.5
El resultado del módulo es: 1

- Con ayuda de la presentación electrónica, explica los tipos de datos que se pueden presentar durante la interacción con el usuario y como almacenarlos.

Clase 2

Desarrollo

El alumno: (Individual) (90 minutos)

- Elabora un programa que solicite al usuario su edad y los presente en la consola de salida.
- Elabora un programa que solicite al usuario, su nombre y apellidos y los presente en la consola de salida.



Cierre

Tiempo estimado: 30 minutos.

Cuestionario Kahoot, Conceptos fundamentales del lenguaje de programación.

El profesor (Grupal):

Crea el cuestionario en Kahoot con preguntas relacionadas al lenguaje de programación Java.
Coloca el enlace en la plataforma educativa.
Explica la dinámica a los estudiantes, lee las preguntas, da lectura de los alumnos ganadores y ánimo al grupo a aparecer en los primeros lugares.
Descarga los resultados de la app y los integra a la evaluación del alumno.

El alumno (individual):

- Contesta las preguntas
- Participa contestando y comentando los resultados.

Extra - clase

Tiempo estimado: 20 minutos.

El alumno (individual)

- Documenta en Word la codificación realizada y las pantallas de salida de la consola para cada ejercicio y las sube a la plataforma educativa.

EVALUACIÓN

Actividad 1. Rúbrica para documento con ejercicios de salidas por consola.

Criterio	Bien (2 pts.)	Regular (1 pts.)	Deficiente (0 pts.)
Título del Programa	Presenta el título del programa de forma clara.	Presenta un título diferente al indicado.	No presenta el título.
Datos del alumno	Presenta Nombre del alumno y grupo.	Sólo presenta nombre o grupo.	No presenta datos del alumno.
Código	El código está completo y ordenado.	El código está completo, pero se observa desorden	Código incompleto.
Ejecución	Se adjunta pantalla de ejecución y realiza lo esperado.	Se adjunta pantalla de ejecución, pero no muestra la salida esperada.	El programa no se ejecuta o no hay evidencia de pantalla

	Documentación interna	Contiene comentarios claros que identifican y explican el funcionamiento del programa.	Contiene comentarios, no son claros o erróneos.	No contiene documentación.	
Actividad 2. Puntuación obtenida en Cuestionario Kahoot.					



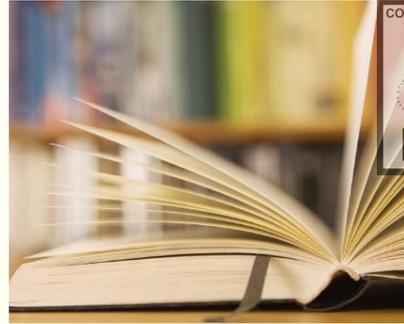
IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • Márquez, Verónica (2017). Operadores para algoritmos. Universidad Autónoma del Estado de Hidalgo. • Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill. • Colegio de Ciencias y Humanidades. (18 de mayo de 2022). Guía de estudio de Cibernética y Computación I y II. https://www.unamenlinea.unam.mx/recurso/83615-guia-para-el-estudiante-de-cibernetica-y-computacion
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<ul style="list-style-type: none"> • Sznajdleder, Pablo (2018). Java a fondo: - estudio del lenguaje y desarrollo de aplicaciones - 2a ed. Buenos Aires: Alfaomega Grupo editor argentino. • García-Beltrán (2009). Programación en Java. http://ocw.upm.es/pluginfile.php/1037/mod_label/intro/4-operadores.pdf

V. ANEXOS

SALIDA DE DATOS POR CONSOLA (PRINT O PRINTLN)

+ç



Java como emisor de información al usuario

Como emisor de datos o información, es necesario enviar mensajes a través de la pantalla de una impresora, de un sensor u otro dispositivo que comunique al usuario, de manera clara, lo que deseamos o lo que esperamos de él o ella en su caso o lo que está sucediendo al interior.

“No sé que le pasó, se quedó allí... pensando”.

“No responde”.

La lectura de la pantalla es fundamental



La colocación de mensajes en pantalla es importante



Por ello se han especializado los perfiles de desarrolladores con la intención de pasar de la interacción a crear una experiencia de usuario:

FrontEnd La parte del sitio web que conecta e interactúa con los usuarios visitantes. (Lado del cliente).

Back End Es la parte que se conecta con la base de datos, el servidor y los niveles de seguridad. (Lado del servidor).

La función Print o Println permite emitir mensajes al usuario



Ejercicios

- Problema 1: Realiza un nuevo proyecto llamado "Triangulo" (observa que no lleva acento y comienza con mayúsculas), que presente la siguiente salida

**** Programa que presenta tres Triángulos ****

**** Nombre del alumno: paterno materno nombre

**** Grupo: grupo

```
*
**
***
   *
  **
 ***
    *
   ***
  *****
```

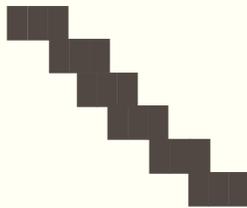
Ejercicios



▪ Problema 2:

Realiza un nuevo proyecto llamado “Escalera” (Recuerda que comienza con mayúsculas). que presente la siguiente salidaPara Obtener el símbolo ocuparás la combinación ASCII (<https://www.celeberrimacom/tabla-de-codigo-ascii-completa-para-imprimir/>)

```
**** Programa que presenta una escalera****
**** Nombre del alumno: paterno materno nombre
**** Grupo: grupo
```



Interacción con el usuario Clase Scanner

```
**** Programa que presenta las operaciones aritméticas entre dos variables****
**** Nombre del alumno: paterno materno nombre
**** Grupo: grupo
Dame el valor de la primera variable: 3
Dame el valor de la segunda variable: 2
La variable 1 tiene almacenado el valor: 3
La variable 2 tiene almacenado el valor: 2
El resultado de la suma es: 5
El resultado de la resta es: 1
El resultado de la multiplicación es: 6
El resultado de la división es: 1.5
El resultado del módulo es: 1
```

Ejercicio 4



- La función `Println` también permite realizar cálculos y operaciones.
- Crea un proyecto en netbeans 8.2. llamado ImpresionCalculos, allí realizaremos procesos de suma, resta, multiplicación, división y módulo.
Operadores aritméticos → +, -, *, /, %
- Presentaremos los datos en pantalla.
- Asimismo, aprenderemos a colocar textos que alternen nuestros resultados para enriquecerlos.
- Al final, en un archivo de Word, colocará él su Nombre, Grupo, Fecha, Nombre del Ejercicio, una descripción de lo que hace el proyecto, el código en texto, las pantallas del texto en netbeans y la salida de la ejecución.

Algunos datos los podemos reutilizar

- Las variables son un espacio de memoria en el que guardamos un determinado valor (o dato).
 - El dato puede existir al principio o puede captarse durante el trayecto del Programa.
 - El dato puede cambiar durante el programa.
 - El dato puede asignarse en varias ocasiones.
 - El dato se puede presentar sólo o se puede agregar texto.
- Puedes guardar números, letras, texto (cadenas de caracteres), valores lógicos, etc. Sin embargo es importante notificarle a la máquina qué es lo que vas a guardar. Para definir una variable seguiremos la estructura:

Tipo de Dato	Guarda	Bits	Datos que se pueden guardar
byte	números	8	(-128 a 127)
short	números	16	[-32.768, 32.767].
int	números	32	-2^{31} a $2^{31}-1$
long	números	64	-2^{63} a $2^{63}-1$

- tipo_dato identificador; **int numero;**
- Los identificadores son secuencias de texto, sensibles a mayúsculas cuya primer carácter solo puede ser una letra, número, símbolo `dolar$` o subrayado `_`.
- Legibles y no acrónimos que no podamos leer.
- Se auto-documenten por sí mismos.
- Nunca podrán coincidir con las palabras reservadas.
- Los identificadores siempre se escriben en minúsculas.
- Si son dos o más palabras, el inicio de cada siguiente palabra se escriba en mayúsculas (penombre `PersonA`).

ESTRATEGIA DIDÁCTICA 8

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	2 de mayo de 2022
NO DE ALUMNOS	25



II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java.
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales. • Construye programas de computadora que resuelvan problemas condicionales.
TEMA(S)	<ul style="list-style-type: none"> • Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas condicionales. • Simple if. • Doble if – else.

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 sesiones de clase, de 120 minutos cada una.	COLEGIO DE CIENCIAS Y HUMANIDADES PLANTEL AZCAPOTZALEO JEFATURA DE SECCIÓN ÁREA DE MATEMÁTICAS 19/07/2022 RECIBIDO				
MATERIALES Y/O RECURSOS	Computadora y servicio de internet Software PSeInt, BlueJ Cuaderno, lápiz, goma, regla, colores, plumas (roja, negra, azul)					
DESARROLLO Y ACTIVIDADES	<p>Sesión 1</p> <p>Inicio (25 minutos)</p> <p>El profesor:</p> <ol style="list-style-type: none"> Plantea el problema: <i>Calcular el índice de masa corporal (IMC) de una persona solicitando el peso en kilogramos y la estatura en metros, así como la interpretación del resultado obtenido con la siguiente fórmula:</i> $IMC = \text{peso} / \text{estatura}^2$ <i>Considerar solamente un peso mayor de 40 kilogramos y una estatura mayor de 1.4 metros.</i> Indica al alumno que investigue en internet la interpretación del IMC y realice un dibujo. <div data-bbox="464 1045 1214 1570" style="text-align: center;"> <p>ÍNDICE DE MASA CORPORAL ($IMC = \text{peso [kg]} / \text{estatura [m}^2\text{]}$)</p> <table border="1" style="width: 100%; text-align: center; margin-top: 10px;"> <tr> <td style="width: 20%;"><18,5 BAJO DE PESO</td> <td style="width: 20%;">18,5-24,9 NORMAL</td> <td style="width: 20%;">25-29,9 SOBREPESO</td> <td style="width: 20%;">30-34,9 OBESO</td> <td style="width: 20%;">35< EXTREMADAMENTE OBESO</td> </tr> </table> </div> <ol style="list-style-type: none"> Pide al alumno conteste en su libreta las siguientes preguntas: <i>Para la construcción del diagrama de flujo y/o algoritmo.</i> ¿Cuál es la condición para que una persona este bajo de peso? ¿Cuál es la condición para que una persona tenga peso normal? ¿Cuál es la condición para que una persona tenga sobrepeso? ¿Cuál es la condición para que una persona tenga obesidad? 		<18,5 BAJO DE PESO	18,5-24,9 NORMAL	25-29,9 SOBREPESO	30-34,9 OBESO
<18,5 BAJO DE PESO	18,5-24,9 NORMAL	25-29,9 SOBREPESO	30-34,9 OBESO	35< EXTREMADAMENTE OBESO		

¿Cuál es la condición para que una persona tenga hiper obesidad?
¿Cuántas estructuras condicionales crees que debes emplear en la solución del presente problema?

Identifique las variables de entrada, proceso y salida.

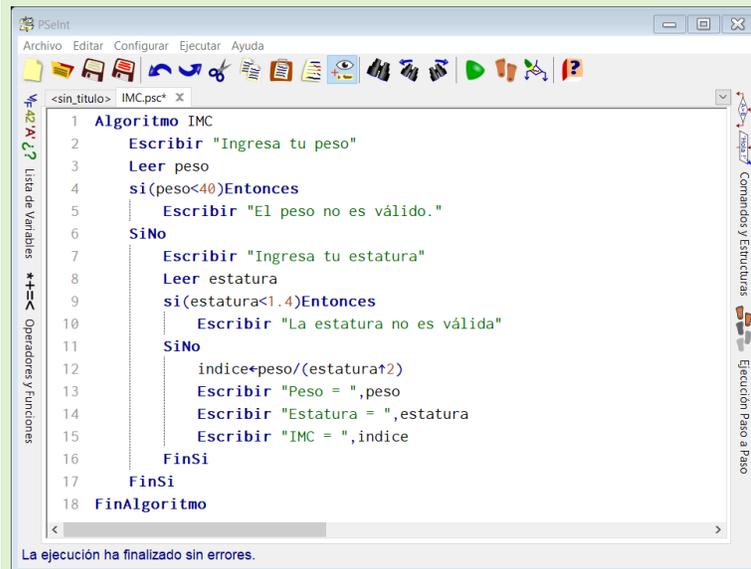
El alumno:

1. Escribe el problema en su cuaderno.
2. Dibuja los elementos del problema.
3. Responde las preguntas planteadas.

Desarrollo (50 minutos)

El profesor:

1. Solicita a 3 alumnos del grupo que expongan el dibujo y las respuestas; aclara dudas; orienta el trabajo realizado por los alumnos del grupo.
2. En forma conjunta con el alumno utiliza la aplicación de PSeInt para realizar la construcción del algoritmo que soluciona el problema.
 - 2.1 Primero se valida el peso y la estatura, y si están dentro del rango especificado se calcula el IMC.



```
1 Algoritmo IMC
2   Escribir "Ingresa tu peso"
3   Leer peso
4   si(peso<40)Entonces
5     Escribir "El peso no es válido."
6   SiNo
7     Escribir "Ingresa tu estatura"
8     Leer estatura
9     si(estatura<1.4)Entonces
10      Escribir "La estatura no es válida"
11     SiNo
12       indice←peso/(estatura^2)
13       Escribir "Peso = ",peso
14       Escribir "Estatura = ",estatura
15       Escribir "IMC = ",indice
16     FinSi
17   FinSi
18 FinAlgoritmo
```

La ejecución ha finalizado sin errores.

2.2 Enseguida se interpreta el resultado del IMC utilizando condiciones para cada caso.



The execution status at the bottom reads: "La ejecución ha finalizado sin errores."


El alumno:

1. Participa respondiendo las preguntas, indica al profesor los pasos del algoritmo y elabora el algoritmo en PSeInt.

Cierre (25 minutos)

El profesor:

1. Solicita al alumno elaborar un documento con el planteamiento del problema (dibujo), análisis (preguntas), diseño (algoritmo y diagrama de flujo de PSeInt), prueba de escritorio (ejecución) y conclusiones del aprendizaje abordado.

El alumno:

1. Elabora un documento con las indicaciones solicitadas por el profesor.

Sesión 2

Inicio (35 minutos)

El profesor:

1. Explica la sintaxis en Java de la estructura de control if-else con algunos ejemplos y solicita al alumno escribir el pseudocódigo del algoritmo del IMC utilizando la sintaxis.

El alumno:

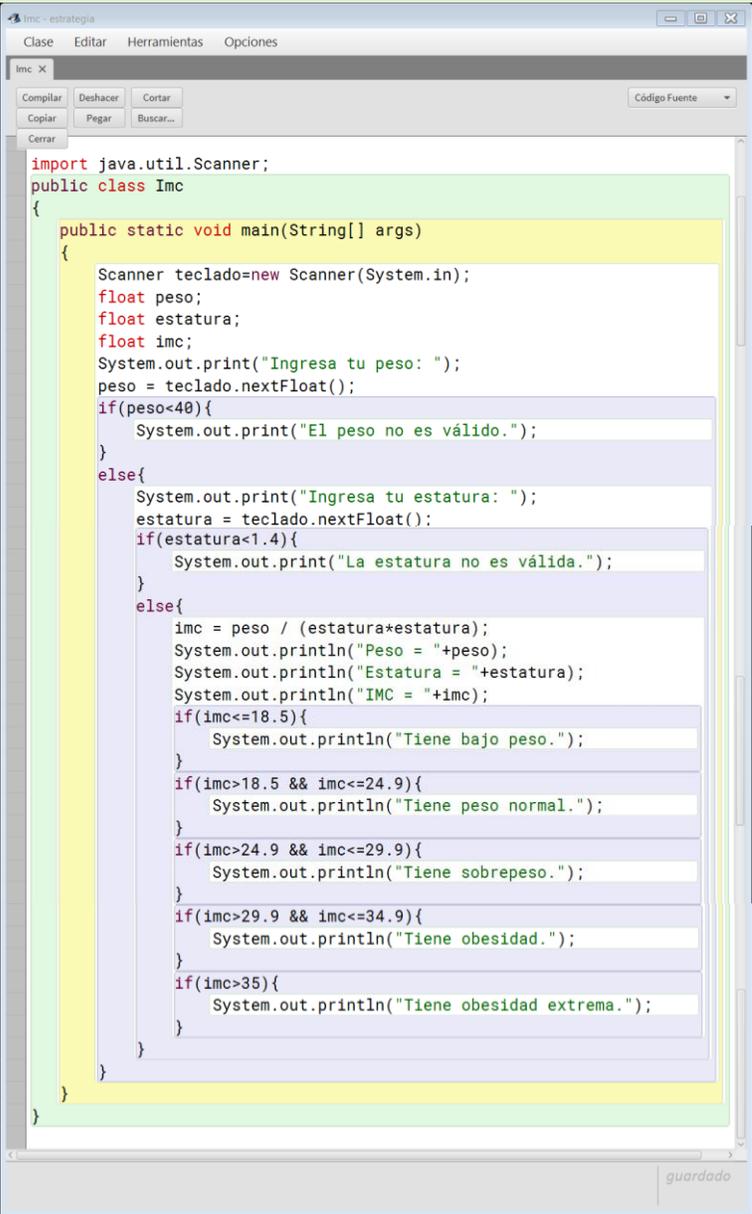
1. Toma nota de la sintaxis de if-else con los ejemplos y escribe el pseudocódigo del IMC.

Desarrollo (50 minutos)

El profesor:

1. En forma junto con el alumno utiliza el programa BlueJ para escribir el programa del problema del IMC utilizando el pseudocódigo descrito.

Código del programa



```
import java.util.Scanner;
public class Imc
{
    public static void main(String[] args)
    {
        Scanner teclado=new Scanner(System.in);
        float peso;
        float estatura;
        float imc;
        System.out.print("Ingresa tu peso: ");
        peso = teclado.nextFloat();
        if(peso<40){
            System.out.print("El peso no es válido.");
        }
        else{
            System.out.print("Ingresa tu estatura: ");
            estatura = teclado.nextFloat();
            if(estatura<1.4){
                System.out.print("La estatura no es válida.");
            }
            else{
                imc = peso / (estatura*estatura);
                System.out.println("Peso = "+peso);
                System.out.println("Estatura = "+estatura);
                System.out.println("IMC = "+imc);
                if(imc<=18.5){
                    System.out.println("Tiene bajo peso.");
                }
                if(imc>18.5 && imc<=24.9){
                    System.out.println("Tiene peso normal.");
                }
                if(imc>24.9 && imc<=29.9){
                    System.out.println("Tiene sobrepeso.");
                }
                if(imc>29.9 && imc<=34.9){
                    System.out.println("Tiene obesidad.");
                }
                if(imc>35){
                    System.out.println("Tiene obesidad extrema.");
                }
            }
        }
    }
}
```



El alumno:
1. Escribe el programa en BlueJ.

Cierre (35 minutos)

El profesor:
1. Solicita al alumno entregar el documento con el planteamiento del problema (dibujo), análisis (preguntas), diseño (algoritmo y diagrama de flujo de PSeInt), prueba de escritorio (ejecución de PSeInt) realizado la sesión anterior y añadir el pseudocódigo, implementación (programa y ejecución en BlueJ), así como las conclusiones del aprendizaje abordado y de acuerdo con la lista de cotejo.

El alumno:
1. Entrega un documento final con las indicaciones del profesor expuestas en la lista de cotejo.



EVALUACIÓN

Evaluación formativa	Evidencia	Porcentaje
Procedimental	Solución del problema	80%
Actitudinal	Participación y conclusiones.	20%

Lista de cotejo para la solución del problema

El alumno	Completa (1 punto)	Parcial (0.5 puntos)	Incompleta (0 puntos)
Elabora el planteamiento del problema con un dibujo.			
Elabora el análisis del problema contestando las preguntas.			
Elabora el algoritmo del problema en PSeInt			
Elabora el diagrama de flujo del problema en PSeInt			
Realiza la prueba de escritorio del problema en PSeInt			
Elabora el pseudocódigo del problema.			

	Elabora el programa que soluciona el problema.			
	Muestra la ejecución del programa que soluciona el problema			



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Cairó, O. (2003). Metodología de la programación. Algoritmos, diagramas de flujo y programas. México: Alfaomega.</p> <p>Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Cairó, O. (2003). Metodología de la programación. Algoritmos, diagramas de flujo y programas. México: Alfaomega.</p> <p>Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill</p>

ESTRATEGIA DIDÁCTICA 9

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación I
SEMESTRE ESCOLAR	Quinto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	2 de abril de 2022
NO DE ALUMNOS	25



II. PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java.
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.
APRENDIZAJE(S)	<ul style="list-style-type: none"> • Elabora el algoritmo, diagrama de flujo y pseudocódigo de problemas de ciclo (cíclicos) que satisfagan una condición. • Construye programas de computadora que involucren la sentencia while.
TEMA(S)	<ul style="list-style-type: none"> • Elaboración de algoritmos de ciclo que satisfagan una condición. • Representación de algoritmos de ciclo a través de diagramas de flujo que satisfagan una condición.

	<ul style="list-style-type: none"> • Representación de algoritmos de ciclo a través de pseudocódigo que satisfagan una condición. • Sentencia while.
--	--



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 sesiones de clase, de 120 minutos cada una.
MATERIALES Y/O RECURSOS	Computadora y servicio de internet Software PSeInt, BlueJ Cuaderno, lápiz, goma, regla, colores, plumas (roja, negra, azul)
DESARROLLO Y ACTIVIDADES	<p>Sesión 1</p> <p>Inicio (25 minutos)</p> <p>El profesor:</p> <p>1. Plantea el problema:</p> <p><i>En un supermercado un cajero captura los precios de los artículos que los clientes compran e indica a cada cliente cual es el monto que debe pagar. Al finalizar su jornada de labores, le indica a su supervisor cuantos clientes atendió y le entrega el importe total que cobró de todos los clientes que pasaron por su caja.</i></p> <p>2. Indica al alumno que dibuje el Supermercado, los clientes, los productos, considere que desarrolla la función del cajero y realice un análisis del problema contestando en su libreta las siguientes preguntas:</p> <p><i>Para la construcción del diagrama de flujo y/o algoritmo.</i></p> <p><i>¿Deberás emplear estructuras de tipo repetitiva (ciclo)? Si la respuesta es afirmativa, explica ¿Cómo lo utilizarás?</i></p> <p><i>Si piensas que deberás emplear estructuras de tipo repetitiva (ciclo), indica:</i></p> <p><i>¿El ciclo se repetirá un número fijo de veces?</i></p> <p><i>¿El ciclo se repetirá hasta que se satisfaga una condición?</i></p> <p><i>¿Cuántas estructuras de tipo repetitiva (ciclos) crees que debes emplear en la solución del presente problema?</i></p> <p><i>¿Deberás emplear variables de tipo contador y/o acumulador? Si la respuesta es afirmativa, indica cuantas variables debes manejar y el objetivo de cada una de ellas.</i></p> <p><i>Identifique las variables de entrada, proceso y salida.</i></p>

El alumno:

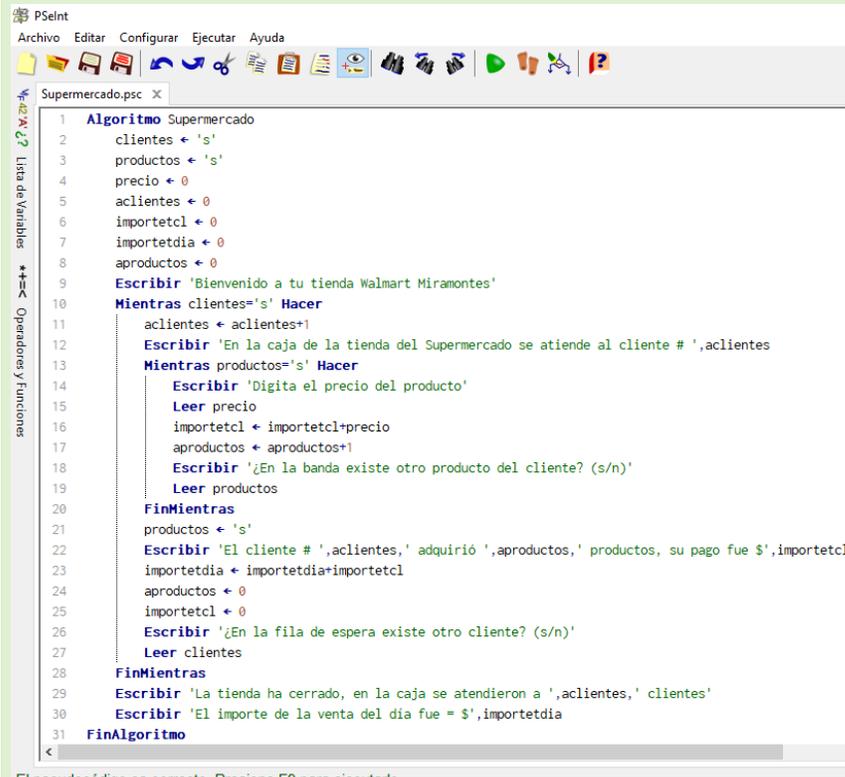
1. Escribe el problema en su cuaderno.
2. Dibuja los elementos del problema y responde las preguntas planteadas.

Desarrollo (50 minutos)

El profesor:

1. Solicita a 3 alumnos del grupo que expongan el dibujo y las respuestas; aclara dudas; orienta el trabajo realizado por los alumnos del grupo y en forma conjunta con el alumno utiliza la aplicación de PSeInt para realizar la construcción del algoritmo, diagrama de flujo y ejecución.

Algoritmo

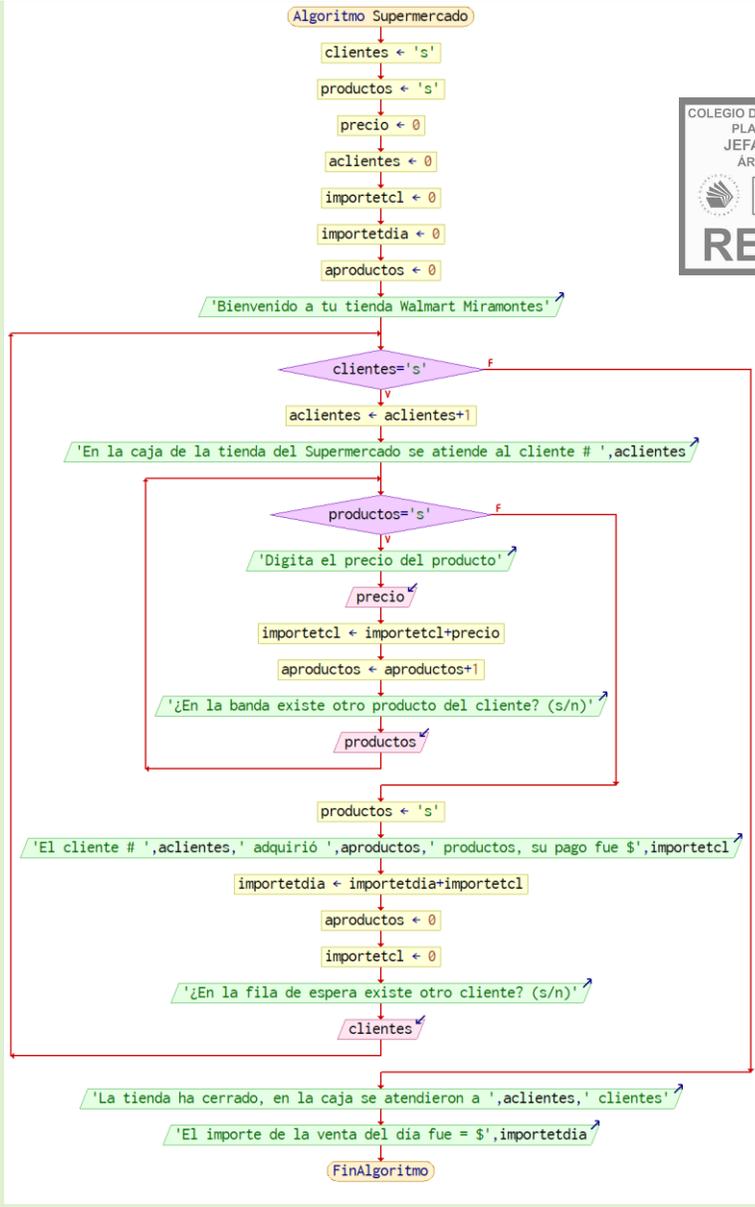


```
1 Algoritmo Supermercado
2 clientes ← 's'
3 productos ← 's'
4 precio ← 0
5 aclientes ← 0
6 importetcl ← 0
7 importetdia ← 0
8 aproductos ← 0
9 Escribir 'Bienvenido a tu tienda Walmart Miramontes'
10 Mientras clientes='s' Hacer
11     aclientes ← aclientes+1
12     Escribir 'En la caja de la tienda del Supermercado se atiende al cliente # ',aclientes
13     Mientras productos='s' Hacer
14         Escribir 'Digita el precio del producto'
15         Leer precio
16         importetcl ← importetcl+precio
17         aproductos ← aproductos+1
18         Escribir '¿En la banda existe otro producto del cliente? (s/n)'
19         Leer productos
20     FinMientras
21     productos ← 's'
22     Escribir 'El cliente # ',aclientes,' adquirió ',aproductos,' productos, su pago fue $',importetcl
23     importetdia ← importetdia+importetcl
24     aproductos ← 0
25     importetcl ← 0
26     Escribir '¿En la fila de espera existe otro cliente? (s/n)'
27     Leer clientes
28 FinMientras
29 Escribir 'La tienda ha cerrado, en la caja se atendieron a ',aclientes,' clientes'
30 Escribir 'El importe de la venta del día fue = $',importetdia
31 FinAlgoritmo
```

El pseudocódigo es correcto. Presione F9 para ejecutarlo.

Diagrama de flujo





Ejecución

```

PSelnt - Ejecutando proceso SUPERMERCADO
Digita el precio del producto
> 15.9
¿En la banda existe otro producto del cliente? (s/n)
> s
Digita el precio del producto
> 32.5
¿En la banda existe otro producto del cliente? (s/n)
> n
El cliente # 1 adquirió 2 productos, su pago fue $48.4
¿En la fila de espera existe otro cliente? (s/n)
> s
En la caja de la tienda del Supermercado se atiende al cliente # 2
Digita el precio del producto
> 11.5
¿En la banda existe otro producto del cliente? (s/n)
> s
Digita el precio del producto
> 100.5
¿En la banda existe otro producto del cliente? (s/n)
> s
Digita el precio del producto
> 65.9
¿En la banda existe otro producto del cliente? (s/n)
> s
Digita el precio del producto
> 45.9
¿En la banda existe otro producto del cliente? (s/n)
> n
El cliente # 2 adquirió 4 productos, su pago fue $223.8
¿En la fila de espera existe otro cliente? (s/n)
> n
La tienda ha cerrado, en la caja se atendieron a 2 clientes
El importe de la venta del día fue = $272.2
*** Ejecución Finalizada. ***

```



El alumno:

1. Participa respondiendo las preguntas, indica al profesor los pasos del algoritmo y elabora el algoritmo en PSeInt.

Cierre (25 minutos)

El profesor:

1. Solicita al alumno elaborar un documento con el planteamiento del problema (dibujo), análisis (preguntas), diseño (algoritmo y diagrama de flujo de PSeInt), prueba de escritorio (ejecución) y conclusiones del aprendizaje abordado.

El alumno:

1. Elabora un documento con las indicaciones solicitadas por el profesor.

Sesión 2

Inicio (35 minutos)

El profesor:

1. Explica la sintaxis en Java de la estructura de control `while` con algunos ejemplos y solicita al alumno escribir el pseudocódigo del algoritmo del supermercado utilizando la sintaxis.

El alumno:

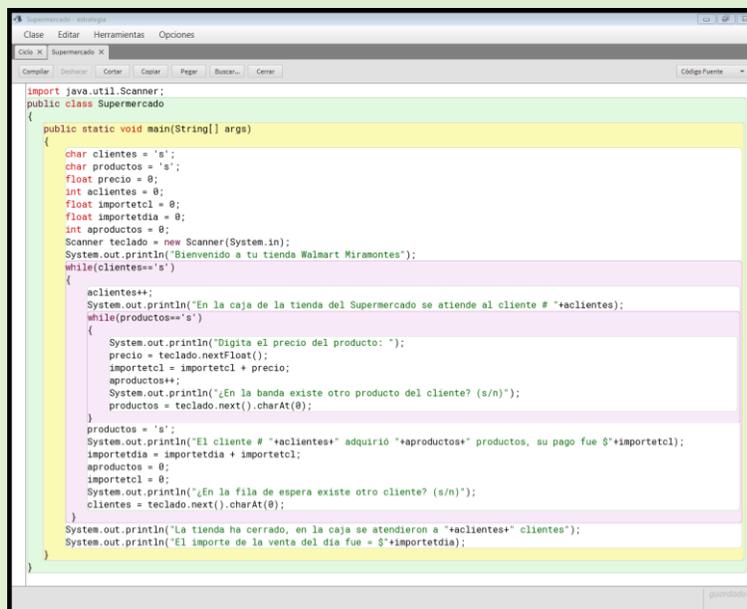
1. Toma nota de la sintaxis de `while` con los ejemplos y escribe el pseudocódigo del supermercado.

Desarrollo (50 minutos)

El profesor:

1. En forma conjunta con el alumno utiliza el programa BlueJ para escribir el programa del problema del supermercado utilizando el pseudocódigo descrito.

Código del programa



```
import java.util.Scanner;
public class Supermercado
{
    public static void main(String[] args)
    {
        char clientes = 's';
        char productos = 's';
        float precio = 0;
        int sclientes = 0;
        float importetcl = 0;
        float importetdia = 0;
        int aproductos = 0;
        Scanner teclado = new Scanner(System.in);
        System.out.println("Bienvenido a tu tienda Walmart Miramontes");
        while(clientes=='s')
        {
            sclientes++;
            System.out.println("En la caja de la tienda del Supermercado se atiende al cliente # "+sclientes);
            while(productos=='s')
            {
                System.out.println("Digita el precio del producto: ");
                precio = teclado.nextFloat();
                importetcl = importetcl + precio;
                aproductos++;
                System.out.println("¿En la banda existe otro producto del cliente? (s/n)");
                productos = teclado.next().charAt(0);
            }
            productos = 's';
            System.out.println("El cliente # "+sclientes+" adquirió "+aproductos+" productos, su pago fue $"+importetcl);
            importetdia = importetdia + importetcl;
            aproductos = 0;
            importetcl = 0;
            System.out.println("¿En la fila de espera existe otro cliente? (s/n)");
            clientes = teclado.next().charAt(0);
        }
        System.out.println("La tienda ha cerrado, en la caja se atendieron a "+sclientes+" clientes");
        System.out.println("El importe de la venta del día fue $"+importetdia);
    }
}
```

El alumno:

1. Escribe el programa en BlueJ.

Cierre (35 minutos)

El profesor:

1. Solicita al alumno entregar el documento con el planteamiento del problema (dibujo), análisis (preguntas), diseño (algoritmo y diagrama de flujo de PSeInt), prueba de escritorio (ejecución de PSeInt) realizado la sesión anterior y añadir el pseudocódigo,



implementación (programa y ejecución en BlueJ), así como las conclusiones del aprendizaje abordado y de acuerdo con la lista de cotejo.

El alumno:

1. Entrega un documento final con las indicaciones del profesor expuestas en la lista de cotejo.



EVALUACIÓN

Evaluación formativa	Evidencia	Porcentaje
Procedimental	Solución del problema	80%
Actitudinal	Participación y conclusiones.	20%

Lista de cotejo para la solución del problema

El alumno	Completa (1 punto)	Parcial (0.5 puntos)	Incompleta (0 puntos)
Elabora el planteamiento del problema con un dibujo.			
Elabora el análisis del problema contestando las preguntas.			
Elabora el algoritmo del problema en PSeInt			
Elabora el diagrama de flujo del problema en PSeInt			
Realiza la prueba de escritorio del problema en PSeInt			
Elabora el pseudocódigo del problema.			
Elabora el programa que soluciona el problema.			

	Muestra la ejecución del programa que soluciona el problema			
--	---	--	--	--



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Cairó, O. (2003). Metodología de la programación. Algoritmos, diagramas de flujo y programas. México: Alfaomega.</p> <p>Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Cairó, O. (2003). Metodología de la programación. Algoritmos, diagramas de flujo y programas. México: Alfaomega.</p> <p>Joyanes, L. (2003). Fundamentos de la programación. Algoritmos, estructura de datos y objetos, México: Mc. Graw–Hill</p>

Cibernética y Computación II

Unidad 1: Lenguaje de programación orientado a objetos.

Propósito: Al finalizar la unidad el alumno:

Conocerá las características del lenguaje de programación orientado a objetos Java y su entorno de desarrollo, definiendo clases, atributos y métodos para la implementación de objetos en programas.



ESTRATEGIA DIDÁCTICA 1

I. DATOS GENERALES

ELABORARON	Seminario institucional
ASIGNATURA	Cibernética y Computación II
SEMESTRE ESCOLAR	Sexto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Abril 2022
NO DE ALUMNOS	25

II.PROGRAMA

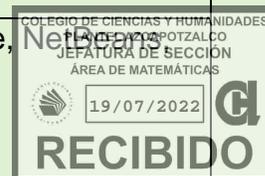
TÍTULO DE LA UNIDAD	Lenguaje de programación orientada a objetos con Java
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Conocerá las características del lenguaje de programación orientado a objetos Java y su entorno de desarrollo, definiendo clases, atributos y métodos para la implementación de objetos en programas
APRENDIZAJE(S)	Conoce los conceptos básicos de la programación orientada a objetos.

	<p>Conoce la organización general de un programa en Java como lenguaje orientado a objetos.</p> <p>Describe los conceptos de Clase y atributo del lenguaje Java</p> <p>Implementa programas utilizando Clases y atributos.</p> <p>Describe los conceptos de métodos del lenguaje java.</p> <p>Conoce cómo instanciar objetos a partir de una Clase.</p> <p>Implementa programas utilizando métodos.</p> <p>Empleará la Clase Scanner para la entrada de datos en la creación de un programa.</p> <p>Reafirma los conceptos adquiridos en la unidad.</p>
<p>TEMA(S)</p>	<p>Lenguaje de programación orientado a objetos</p> <p>Conceptos básicos de la programación orientada a objetos:</p> <ul style="list-style-type: none"> • Identidad, atributos o características, comportamiento, abstracción, encapsulamiento, organización general de un programa en Java, comentarios, uso de bibliotecas, identificadores, palabras reservadas, sentencias, tipos de datos primitivos, bloque de código, operadores, expresiones. <p>Clases:</p> <ul style="list-style-type: none"> • Definición, declaración, atributos, niveles de visibilidad, implementación. <p>Métodos:</p> <ul style="list-style-type: none"> • Definición, declaración, parámetros, métodos Getter y Setter. <p>Objetos:</p> <ul style="list-style-type: none"> • Definición, declaración (instanciación), implementación. <p>La Clase Scanner.</p> <ul style="list-style-type: none"> • Importar la Clase java.util.Scanner, definición del objeto de la Clase Scanner (instanciación), métodos system.in, introducción de datos desde el teclado. <p>Errores sintácticos y lógicos.</p> <p>Ejecución del programa</p>



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	Cinco sesiones de dos horas, total de horas 10.
MATERIALES Y/O RECURSOS	IDE (Interface Development Environment) BlueJ, Eclipse, Onlinegdb, Replit).
DESARROLLO Y ACTIVIDADES	<p>Sesión 1</p> <p>INICIO (40 min)</p> <p>El profesor describe el aprendizaje a lograr durante la sesión, les muestra la presentación (Anexo 1) donde los estudiantes identificarán los siguientes conceptos: Identidad, atributos o características, comportamiento, abstracción, encapsulamiento, organización general de un programa en Java, comentarios, uso de bibliotecas, identificadores, palabras reservadas, sentencias, tipos de datos primitivos, bloque de código, operadores, expresiones. el profesor puede utilizar el siguiente recurso: https://view.genial.ly/626070f6f335560018bf496f/video-presentation-presentapoo</p> <p>DESARROLLO (40 min)</p> <p>Los estudiantes del grupo se dividen en equipos de 4 alumnos para que contesten el crucigrama (Anexo 1)</p> <p>CIERRE (30 min)</p> <p>El grupo utilizando lluvia de ideas expresa ¿Cómo se sintieron en la organización para ponerse a contestar las preguntas del crucigrama? ¿Lograron terminarlo en el tiempo indicado? ¿Creen tener claros los conceptos aprendidos?</p> <p>Sesión 2</p> <p>INICIO (20 min)</p> <p>Describe e implementa Clases con atributos, Los estudiantes describen características y comportamientos de distintos objetos que el profesor sugiere por ejemplo (carro, casa, persona, instrumento musical, etc.).</p>



DESARROLLO (60 min)

El estudiante logró describir distintos objetos, ahora el profesor le muestra como las características descritas se convierten en atributos y como se desarrolla el código desde un IDE (NetBeans, BlueJ, Eclipse, Onlinegdb, Replit) el profesor crea un ejemplo para definir la Clase con sus atributos haciendo uso de distintos tipos de datos primitivos.

El grupo se divide en equipos de 3 integrantes y se les pide que implementen la Clase con sus atributos del objeto designado o elegido diferente al ejemplo.

CIERRE (30 min)

Cada equipo presenta al grupo el código resultante de su Clase con los atributos.

Sesión 3

INICIO (40 min) El Estudiante describe e implementa Clases con sus atributos y métodos.

El profesor muestra la relación que existe con el comportamiento del objeto y el método de una Clase, así como su implementación en el IDE (NetBeans, BlueJ, Eclipse, www.onlinegdb.com, www.replit.com) retomando el ejemplo de la clase anterior agregándole los métodos.

El profesor explica la importancia de los métodos getter y setter y los desarrolla para cada atributo.

Finalmente, el profesor implementa el método main para instanciar el primer objeto

DESARROLLO (60 min) Los equipos se conforman igual que en la clase anterior, utilizan su ejemplo anterior y conjuntamente crean los métodos a la Clase.

CIERRE (20 min)



Cada equipo muestra la Clase completa con sus atributos y métodos a todo el grupo.

Sesión 4

INICIO (10 min)

Empleará la Clase Scanner para la entrada de datos en la creación de un programa. Para la interacción de un programa con el usuario es importante la captura de datos, por ejemplo, un usuario y una contraseña para autenticar la entrada a un correo electrónico, cuando se está haciendo uso de una app para traducir palabras a otro idioma, pide que se introduzca el texto a traducir, de esta forma el profesor explica la importancia del uso de la Clase Scanner y sus métodos.

DESARROLLO (30 min)

El profesor describe la forma de implementar un método de función utilizando los datos introducidos para mostrar resultados (conversiones de medidas, de monedas, fórmulas de áreas, perímetros, etc.).

CIERRE: (80 min)

Les pide de forma individual cada alumno defina un método de función introduciendo datos de entrada desde el teclado a los ejemplos que venían desarrollando, el profesor ira verificando que lo vayan haciendo bien y puedan llegar a la ejecución correcta.

Sesión 5

INICIO (20 min)

Involucrando a todos los estudiantes en conjunto con el profesor reafirman los conceptos adquiridos en la unidad. El profesor explica el planteamiento del problema, para que lo analicen hasta describir la Clase, los atributos y métodos que la constituirán.

DESARROLLO (70 min)

El profesor será el que irá desarrollando el código según las instrucciones que le vayan dando cada estudiante del grupo hasta



completar toda la definición de la Clase con sus atributos, métodos e instancia del objeto. Cada alumno en sus computadoras irá haciendo lo mismo.

CIERRE (30 min)

A manera de recapitulación el profesor indica los conceptos vistos en la unidad 1.



EVALUACIÓN	Sesión 1									
	La elaboración del crucigrama indicará que tan claro quedaron los conceptos básicos									
	Sesión 2									
	<table border="1"> <thead> <tr> <th></th> <th>Puntaje</th> </tr> </thead> <tbody> <tr> <td>Trabajo en equipo</td> <td>2</td> </tr> <tr> <td>Definió la clase</td> <td>4</td> </tr> <tr> <td>Definió los atributos</td> <td>4</td> </tr> <tr> <td>Total</td> <td>10</td> </tr> </tbody> </table>		Puntaje	Trabajo en equipo	2	Definió la clase	4	Definió los atributos	4	Total
	Puntaje									
Trabajo en equipo	2									
Definió la clase	4									
Definió los atributos	4									
Total	10									
	Sesión 3									
	<table border="1"> <thead> <tr> <th></th> <th>Puntaje</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> </tr> </tbody> </table>		Puntaje							
	Puntaje									

	Trabajo en equipo	1
	Definió la clase	1
	Definió los atributos	2
	Definió el o los métodos getter	2
	Definió el o los métodos setter	2
	Instancio los objetos desde la clase Main	2
	Total	10

Sesión 4

	Puntaje
Definió la clase	1
Definió los atributos	1
Definió el o los métodos getter	1
Definió el o los métodos setter	1
Definió el método función	2
Instancio los objetos desde la clase Main e hizo uso de los métodos	2
El programa ejecuto sin errores	2
Total	10

Sesión 5

El profesor hará que todos los estudiantes del grupo participen en el análisis del problema y codificación del programa.



IV. REFERENCIAS DE APOYO

<p>BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS</p>	<p>Ceballos, F. J. (2015). <i>Java: interfaces gráficas y aplicaciones para Internet</i>. Paracuellos de Jarama: Ra–Ma.</p> <p>Eckel, B. (2007). <i>Piensa en Java</i>. Madrid: Pearson–Prentice Hall.</p> <p>Martín Sierra, A. J. (2014). <i>Programador Java certificado práctico</i> (3ª edición). Madrid: Ra-Ma</p> <p>Moisset, D. (s. f.). JavaYa. Recuperado 4 de enero de 2021, de https://tutorialesprogramacionya.com/javaya/</p> <p>Vogel, G. V. L.-. (2015). <i>Using the Eclipse IDE for Java programming - Tutorial</i>. Recuperado 6 de enero de 2021, de https://www.vogella.com/tutorials/Eclipse/article.html</p> <p>Joyanes Aguilar, L. y Zahonero Martínez, I. (2014). <i>Programación en c, c++, Java y uml</i>. México: McGraw–Hill Interamericana</p> <p>Olsson, M. (2013). <i>Java quick syntax reference</i>. Berkeley, California: Apress.</p> <p>Streib, J. T. y Soma, T. (2014). <i>Guide to Java: a concise introduction to programming</i>. London: Springer.</p> <p>Sznajdleder, P. A. (2013). <i>Java a fondo: estudio del lenguaje y desarrollo de aplicaciones</i> (2a. ed.). México: Alfaomega</p> <p>Weiss, M. A. & Weiss, M. A. (2013). <i>Estructura de datos en Java</i>. Madrid: Pearson Educación.</p>
<p>BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR</p>	<p>Arnow, D. y Weiss, G. (2000). <i>Introducción a la programación con Java</i>. Madrid: Addison–Wesley</p> <p>Villalobos Salcedo, J. A. (2008). <i>Introducción a las estructuras de datos : aprendizaje activo basado en casos : un enfoque moderno usando Java, UML, Objetos y Eclipse</i>. México: Pearson Educación.</p> <p>Malik, D. S. (2013). <i>Programación Java: del análisis de problemas al diseño de programas</i>. Recuperado 29 de febrero de 2016 en http://site.ebrary.com.pbidi.unam.mx:8080/lib/bibliodgbsp/deta il.action?docID=11087780</p>



	<p>Moisset, D. (s. f.). <i>Java Ya</i>. Recuperado 4 de enero de 2021, de https://www.tutorialesprogramacionya.com/javaya/</p> <p>Rodríguez, A. (s. f.) <i>Curso Aprender programación Java desde cero</i>. APRENDERAPROGRAMAR.COM Recuperado 5 de enero de 2021, de http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188</p>
--	---



Anexos

Anexo 1.

Crucigrama que evalúa que el alumno conozca los conceptos básicos de la programación orientada a objetos

Abstracción	Extrae las características esenciales y comportamiento de un objeto.
Encapsular	Unir en una clase las características y comportamientos para crear una unidad.
Atributo	Características que identifican a un objeto.
Método	Es la forma en la que un objeto realiza ciertas acciones o funciones o tareas.
Identidad	Propiedad del objeto que los distingue diferenciándose de los demás.
Objeto	Es la unidad dentro de un programa informático instanciado desde una clase.
Comentarios	Notas para el programador que el compilador no toma como parte de la sintaxis del código.
Sentencia	Orden que se le da al programa para realizar una tarea específica con el propósito de resolver un problema, termina con ;
Operadores	Llevan a cabo operaciones sobre los datos o variables devolviendo un resultado.
Clase	Plantilla a partir de la cual se crea el objeto.
Identificador	Conjunto de caracteres alfanuméricos que sirven para nombrar a Clases, atributos o métodos.



Acceso al crucigrama

www.educima.com/crosswords/programacion_orientada_a_objet-2705625df5de6aee13bfa3bd96f34c58

Acceso a una autoevaluación con preguntas y respuestas:

<https://view.genial.ly/6260619ceb9663001a7b771c/interactive-content-conoce-mas-poo>

Presentación en Genially de conceptos básicos de POO:

<https://view.genial.ly/626070f6f335560018bf496f/video-presentation-presentapoo>

Unidad 2: Estructuras de control de secuencia en Java

Propósito: Al finalizar la unidad el alumno:

Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java.



ESTRATEGIA DIDÁCTICA 1

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación II
SEMESTRE ESCOLAR	Sexto Semestre
PLANTELE	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	11 de abril del 2022

II. PROGRAMA

UNIDAD TEMÁTICA	Unidad 2. Estructuras de control de secuencia en Java
PROPÓSITO(S) DE LA UNIDAD	Propósitos Indicativos: Al finalizar la unidad el alumno: Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java
APRENDIZAJE(S)	Aprendizaje indicativo: Desarrolla programas que involucren las estructuras condicionales simples, compuestas y anidadas en los métodos de una Clase.
DURACIÓN	Tiempo estimado actividades presenciales: 240 minutos Tiempo estimado actividades no presenciales: 60 minutos
APRENDIZAJES ESPERADOS	Al finalizar el tema, el estudiante Desarrollará programas que involucren las estructuras condicionales simples, compuestas y anidadas en los métodos de una Clase.

III. ESTRATEGIA

El desarrollo de esta estrategia será básicamente **modalidad presencial**, ya que se propone una actividad en **modalidad no presencial**. El docente compartirá a los alumnos, hojas impresas que contienen información sobre las

estructuras condicionales **if y switch**, el propósito de cada una, la sintaxis en java, ejemplos y la representación de algunos ejemplos mediante diagrama de flujo. Explicará de manera grupal la información impresa compartida y solicitará la realización de algunas actividades con la finalidad de analizar su uso.



III. ORGANIZACIÓN DE ACTIVIDADES

<p>MATERIALES Y/O RECURSOS</p>	<ul style="list-style-type: none"> a. Hojas impresas que contienen información b. Aplicación de un compilador Java para poder crear los programas c. Laboratorio de cómputo d. Computadora personal e. Pizarrón y plumones
<p>DESARROLLO Y ACTIVIDADES</p>	<p>Inicio</p> <p>Tiempo estimado actividades presenciales: 60 minutos Tiempo estimado actividades no presenciales: 0 minutos</p> <p>Actividad 1. Presencial / Grupal. Tiempo estimado 60 minutos. El profesor comparte información impresa en hojas, la cual explicará de manera grupal. La información que explicará es la siguiente:</p> <p>Estructuras condicionales</p> <p>Las estructuras condicionales son sentencias que dirigen la ejecución de un programa dependiendo del resultado de una expresión lógica.</p> <p>Estructura if</p> <p>Sintaxis de la estructura if:</p> <p>La sintaxis de la sentencia es:</p>

```
if (condición)
{
    [sentencias]
}
else if (condición)
{
    [sentencias]
}
else if (condición)
{
    [sentencias]
}
else
{
    [sentencias]
}
```



Considera que:

- ✓ los bloques *else if* y *else* son optativos.

Esta sentencia ejecutará el conjunto de sentencias que se encuentran dentro de las llaves cuando la expresión lógica sea verdadera. Si es falsa ese conjunto de sentencias jamás se ejecutará y se procederá a evaluar la expresión lógica de cada bloque *else if*. De no evaluarse ninguna expresión lógica como verdadera, se ejecutarán las sentencias del bloque *else*.

```
if (x==0)
{
    System.out.println("El número es par");
}
else
{
    System.out.println("El número es impar");
}
```

Ejemplo 1:

Si x es igual a 0 entonces se mostrará el mensaje “El número es par”, sino se mostrará el mensaje “El número es impar”.

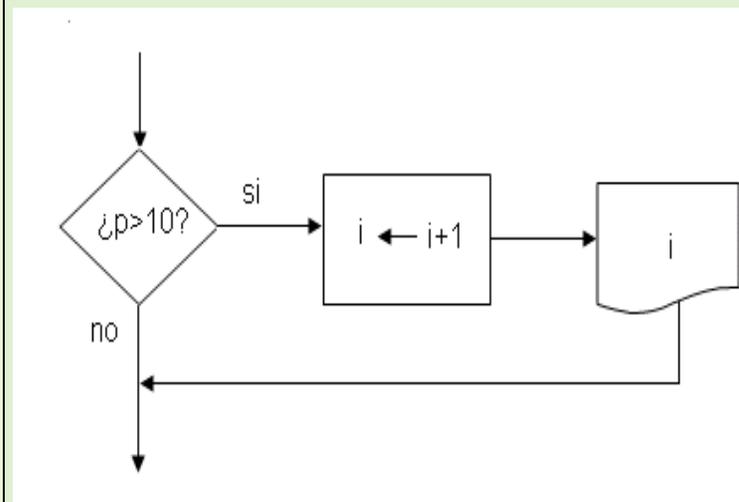
```
if (p>10)
{
    i = i++;
    System.out.println (i);
}
```



Ejemplo 2:

Si p es mayor a 10 entonces se incrementa la variable i en uno y se muestra el valor de i .

El bloque de diagrama de flujo que representa este ejemplo es:

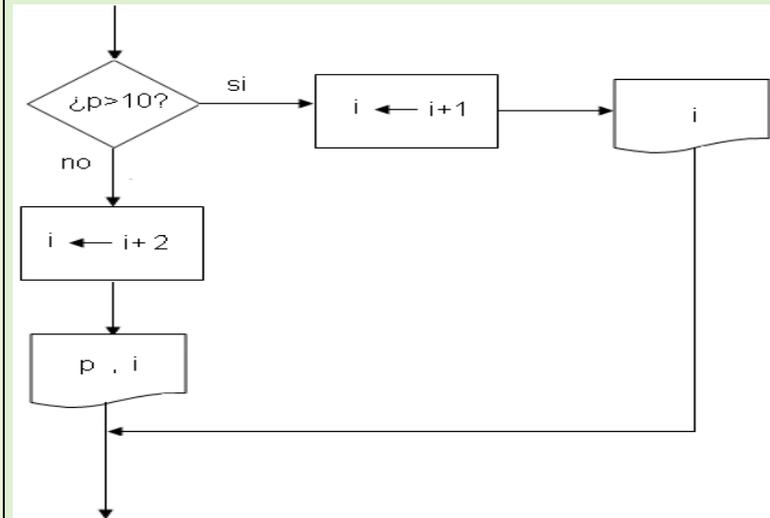


Ejemplo 3:

```
if (p>10)
{
    i=i++;
    System.out.println (i);
}
else
{
    i=i-2.
}
```

Si p es mayor a 10 entonces se incrementa el valor de i en uno y se muestra el valor de i , sino se incrementa el valor de i en dos y se muestra el valor de p y el de i .

El bloque de diagrama de flujo que representa este ejemplo es:



Estructura switch

Esta estructura se utiliza cuando se requiere comparar una variable de **tipo entero** contra una lista de posibles valores.

La sintaxis de la estructura switch es:

```
switch(variable)
{
  case valor1:
    [sentencias]
    break;
  case valor2:
    [sentencias]
    break;
  ....
  case valorN:
    [sentencias]
```



Ejemplo. En la siguiente estructura **switch**, se compara la variable **dia** que es de tipo entero y la cual representa el número de día de la semana:

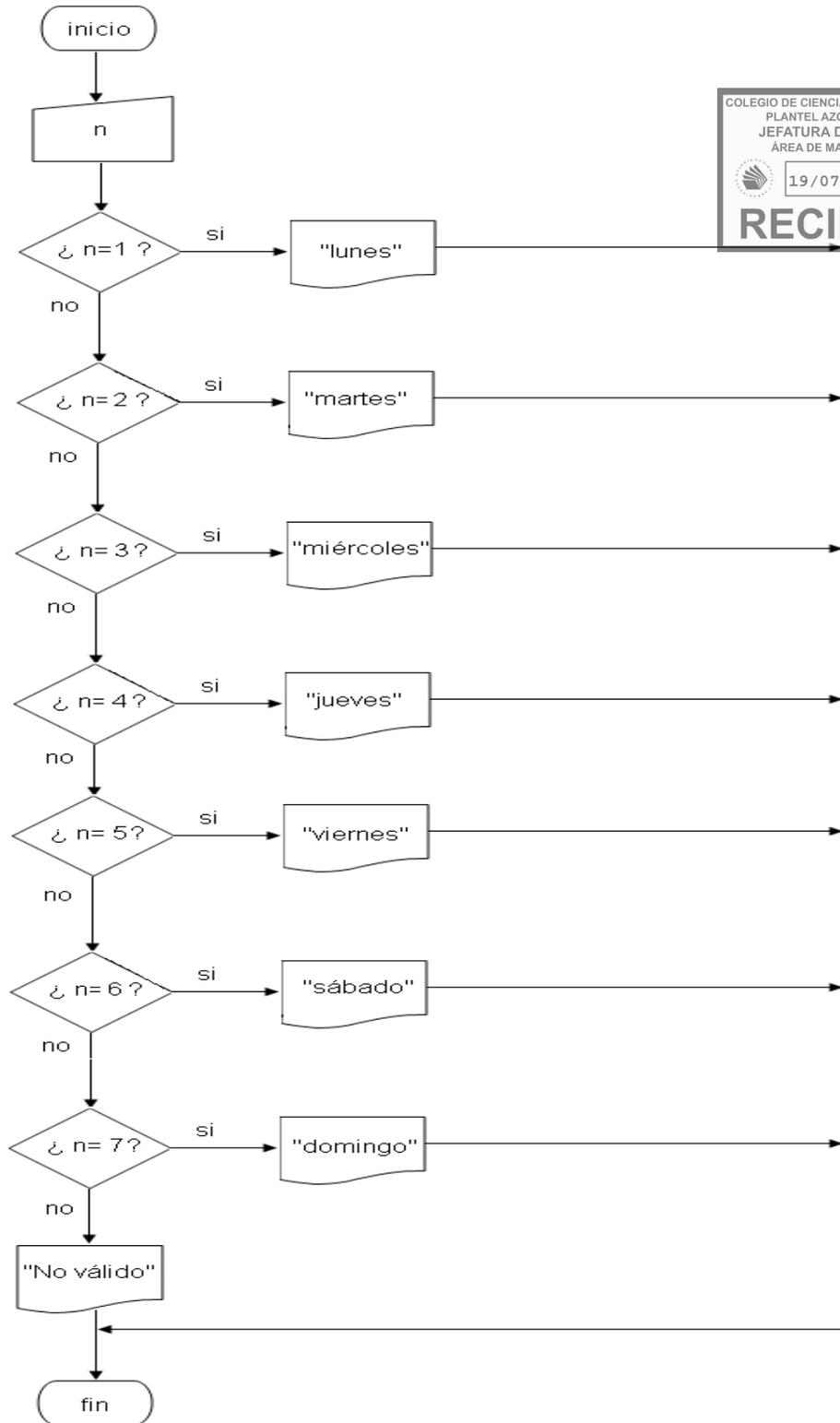
```
switch (n)
{
    case 1:
        System.out.println ("Lunes")
        break;
    case 2:
        System.out.println ("Martes")
        break;
    case 3:
        System.out.println ("Miércoles")
        break;
    case 4:
        System.out.println ("Jueves")
        break;
    case 5:
        System.out.println ("Viernes")
        break;
    case 6:
        System.out.println ("Sábado")
        break;
    case 7:
        System.out.println ("Domingo")
        break;
    default:
        System.out.println ("No válido")
}
```



	}
--	---

El bloque de diagrama de flujo que representa este ejemplo es:





Desarrollo

Tiempo estimado actividades presenciales: 60 minutos

Tiempo estimado actividades no presenciales: 0 minutos



Actividad 2. Presencial / Individual. Tiempo estimado 60 minutos.

En cada una de las siguientes situaciones selecciona la estructura condicional que consideres más conveniente o adecuada para representarla. Usa variables de acuerdo con el contexto. De ser necesario utiliza los bloques else if, else y default. Nota: No se solicita realices el programa.

Situación 1.

En el sistema de pago de un restaurante, si la variable cuenta es mayor de 400, se realiza un descuento del 10%.

Representación con código Java:

A large, empty rectangular box with a light blue background and a thin black border, intended for the student to write their Java code for Situation 1.

Situación 2.

En el registro de la base de datos correspondiente a cada empleado de una empresa, si la variable años es mayor a 5, la empresa aumenta a la variable salario, el 2% por el valor de la variable años.

Representación con código Java:

A large, empty rectangular box with a light blue background and a thin black border, intended for the student to write their Java code for Situation 2.



Situación 3.

“Estoy muy preocupado y debo estudiar mucho, ya que, si obtengo un promedio inferior a 7 en Cibernética, me descontarán \$100 de la cantidad de dinero que me dan por semana”

Representación con código Java:

Situación 4.

“Me motiva pensar que si obtengo un promedio mínimo de 8 me darán \$100 más por semana, pero me estresa pensar que si no lo obtengo me darán \$50 menos.

Representación con código Java:

Situación 5.

En el trabajo de Karla, si el número de ventas por semana es por lo menos de 20 le aumentarán el sueldo 20%, sino solamente le dirán que le “eche más ganas”.

Representación con código Java:



Situación 6.

Si la edad de los empleados de la empresa Bachoco es **mínimo 20 y** máximo 25 años la empresa les da un 10% adicional a su salario.

Representación con código Java:

Situación 7.

Si obtengo un par al lanzar un dado perderé \$20 de cierta cantidad, sino ganaré \$50.

Representación con código Java:

Situación 8.

Si el valor de la variable d es 0 mostrar el mensaje “raíces iguales”, si el valor de la variable d es negativo mostrar el mensaje “raíces imaginarias, en cualquier otro caso mostrar el mensaje “raíces reales”.

Representación con código Java:



Situación 9.

Si tu estatura es mayor a 1.50 metros y no pesas más de 55 kilos podrás ingresar como edecán y tu sueldo base será de \$9000 mensuales, sino podrán ingresar como vendedora y tu sueldo base será de \$8000.

Representación con código Java:

A large, empty rectangular box with a light blue background and a thin black border, intended for the student to write their Java code.

Situación 10.

Si en tu examen obtuviste una calificación superior a 8 o el número de errores no excede a 3 tendrán un punto adicional en el examen.

Representación con código Java:

A large, empty rectangular box with a light blue background and a thin black border, intended for the student to write their Java code.

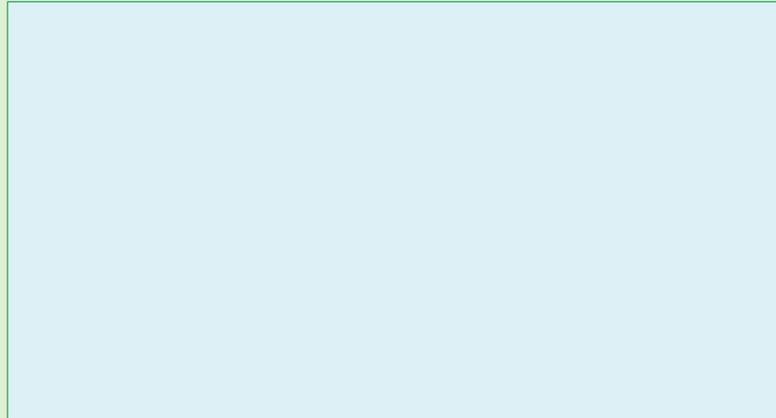
Situación 11.

Si tu calificación es menor a 6 mostrar el mensaje “Reprobado”, si tu calificación es mayor o igual a 6 pero menor a 7.5 mostrar el mensaje

“Suficiente”, si tu calificación es mayor o igual a 7.5 pero menor a 9.0 mostrar el mensaje “Bien” y si tu calificación es mayor o igual a 9.0 y menor o igual a 10 mostrar el mensaje “Muy bien”. Si tu calificación no se encuentra en ninguno de los rangos anteriores mostrar el mensaje “Inválida”.



Representación con código Java:



Situación 12.

Si el valor de la variable num_sem es 1 se mostrará el mensaje “primero”, si es valor de la variable num_sem es 2 se mostrará el mensaje “segundo”, así hasta si el valor de la variable num_sem es 6 se mostrará el mensaje “sexto”. En caso de que la variable num_sem no tome alguno de los anteriores valores, mostrar el mensaje “inválido”.

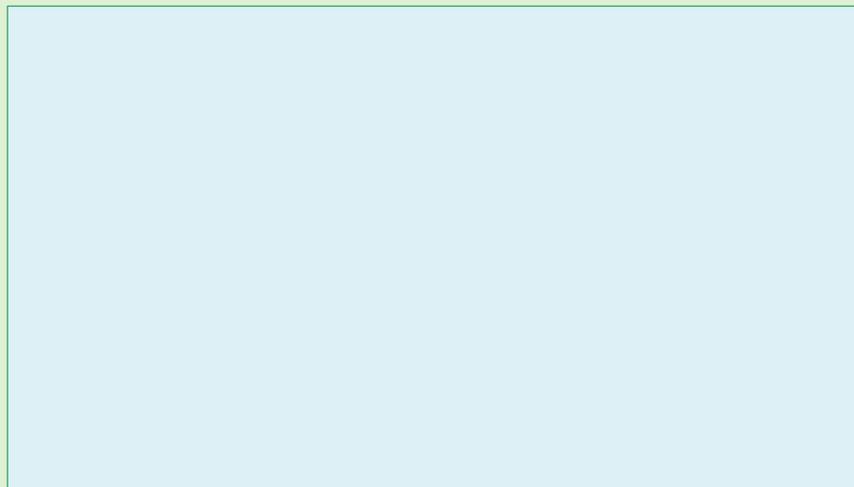
Representación con código Java:



Situación 13.

Considerando el valor de las variables a , b y c , mostrar el mensaje “tercia” si las tres variables tienen el mismo valor, mostrar el mensaje “par” si sólo un par de variables tienen el mismo valor y mostrar el mensaje “nada” si los valores de las tres variables son diferentes.

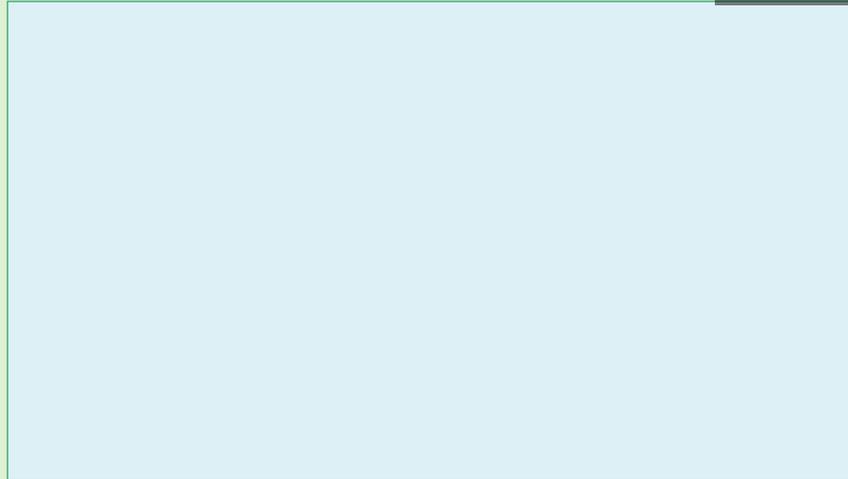
Representación con código Java:



Situación 14.

Mostrar con letra el valor de un número entero entre 20 y 50. Si el número no está dentro del rango, mandar el mensaje “Número no permitido”

Representación con código Java:



Cierre

Tiempo estimado actividades presenciales: 120 minutos

Tiempo estimado actividades no presenciales: 60 minutos

Actividad 3. No presencial / Individual. Tiempo estimado 60 minutos. El profesor solicita el código Java de los siguientes programas:

- i. Programa que obtenga el mayor de tres números.
- ii. Programa que obtenga la raíz cuadrada de un número n . Incluso si n es negativo indique el resultado como número imaginario.

Actividad 4. Presencial / Grupal. Tiempo estimado 30 minutos.

	<p>Se solicita la participación voluntaria de los alumnos para escribir en el pizarrón el código Java de la representación de las 14 situaciones de la Actividad 2, con la finalidad de que en plenaria se discuta sobre lo expuesto en el pizarrón, se resuelvan dudas y se evalúe la Actividad 2.</p> <p>Actividad 5. Presencial / Grupal. Tiempo estimado 20 minutos.</p> <p>Se solicita la participación voluntaria de los alumnos para escribir en el pizarrón el código Java de los programas solicitados en la Actividad 3, con la finalidad de que en plenaria se discuta sobre lo expuesto en el pizarrón, se resuelvan dudas y se evalúe la Actividad 3.</p> <p>Actividad 6. Presencial / Individual. Tiempo estimado 40 minutos. El profesor solicita desarrollar usando el compilador de Java, los siguientes programas:</p> <ul style="list-style-type: none"> a) Programa para mostrar el mensaje de “REPROBADO” si el promedio de 3 calificaciones es menor a 6 o si alguna de las calificaciones es menor a 6, en caso contrario mostrar el mensaje “APROBADO”. b) Programa que muestre y ordene de mayor a menor 4 números cualesquiera. <p>Actividad 7. Presencial / Grupal. Tiempo estimado 30 minutos.</p> <p>Se solicita la participación voluntaria de los alumnos para escribir en el pizarrón el código Java de los programas solicitados en la Actividad 6, con la finalidad de que en plenaria se discuta sobre lo expuesto en el pizarrón, se resuelvan dudas y se evalúe la Actividad 6.</p>
<p>ORGANIZACIÓN</p>	<p>La estrategia está diseñada para grupos de 45 estudiantes. Los estudiantes participarán de manera individual y grupal.</p>



EVALUACIÓN

Mientras el desarrollo de la estrategia, el profesor estará realizando evaluación formativa por medio del monitoreo, guiando y resolviendo las dudas que pudieran surgir.

Las Actividad 3 y la Actividad 6 se entregarán de manera escrita, serán evaluadas por el profesor y consideradas para su calificación. Se sugiere utilizar la siguiente Rúbrica para evaluar:



Entrega	Se entrega con los 3 siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o aclaren la estructura del programa. (2 puntos)	Se entrega con 2 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o aclaren la estructura del programa. (1 punto)	Se entrega con sólo 1 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o aclaren la estructura del programa. (0.5 punto)	Se entrega sin ninguno de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o aclaren la estructura del programa. (0 punto)
Resolución	Utiliza todos los elementos solicitados para resolver cada programa (4 puntos)	Utiliza algunos elementos solicitados para resolver cada programa (2 puntos)	No utiliza los elementos solicitados para resolver cada programa (1 punto)	No resuelve programa. (0 punto)
Funcionamiento	El funcionamiento del programa es completo (4 puntos)	El funcionamiento del programa no es completo, aunque tiene fallas sin importancia. (2 puntos)	El funcionamiento del programa tiene fallas importantes. (1 punto)	No funciona el programa. (0 punto)

IV. REFERENCIAS DE APOYO

<p>BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS</p>	<p>Ceballos, F. (2015). <i>Java: interfaces gráficas y aplicaciones para Internet</i>. Paracuellos de Jarama: Ra–Ma.</p> <p>Jiménez, J. (2014). <i>Fundamentos de programación, diagramas de flujo, diagramas n–s, pseudocódigo y Java</i>. México: Alfaomega.</p> <p>Olsson, M. (2013). <i>Java quick syntax reference</i>. Berkeley, California: Apress.</p> <p>Sierra, M. (2014). <i>Programador Java certificado: curso práctico</i>. Madrid: Ra–Ma.</p> <p>Sznajdleder. (2012). <i>Java a fondo, estudio del lenguaje y desarrollo de aplicaciones</i>. México: Alfaomega.</p>
<p>BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR</p>	<p>Deitel, P. y Deitel, H. (2012). <i>Cómo programar en Java</i> (9ª edición). México: Pearson.</p> <p>Discoduroderoer. (2013). <i>Bucles anidados en Java</i>. Recuperado de https://www.discoduroderoer.es/bucles-anidados-en-java/</p> <p>Gaddis, T. (2012). <i>Starting out with Java from control structures through objects</i>. United States: Pearson.</p> <p>López, L. (2013). <i>Metodología de la programación orientada a objetos</i>. México: Alfaomega.</p> <p>Lowe, D. (2014). <i>Java for dummies</i>. United States: John Wiley & Sons, Inc.</p> <p>Compartiendo conocimientos (2020). <i>Estructuras de control repetitivas</i>. Recuperado de http://elaprenderdiadia.blogspot.com/2020/10/sentencias-repetitivas-en-java.html</p>



ESTRATEGIA DIDÁCTICA 2

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación II
SEMESTRE ESCOLAR	Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	28 de febrero del 2022



II. PROGRAMA

UNIDAD TEMÁTICA	Unidad 2. Estructuras de control de secuencia en Java
PROPÓSITO(S) DE LA UNIDAD	Propósitos Indicativos: Al finalizar la unidad el alumno: Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java
APRENDIZAJE(S)	Aprendizaje indicativo: Desarrolla para resolver problemas que involucren la estructura repetitiva for, while, y do while en los métodos de una Clase.
DURACIÓN	Tiempo estimado actividades presenciales: 120 minutos
APRENDIZAJES ESPERADOS	Al finalizar el tema, el estudiante realizará programas que involucren las Estructuras de control Repetitivas: for, while y do while mediante prácticas dirigidas para el manejo de la solución de problemas donde se requiera hacer iteraciones, bucles, etc.

III. ESTRATEGIA

El desarrollo de esta estrategia será en modalidad presencial, el docente revisará las estructuras repetitivas **for**, **while** y **while-do** el propósito de cada una, la sintaxis en java y ejemplos mediante una presentación, ejemplificando el uso de cada una de las estructuras de control repetitivas con algún problema elaborado en lenguaje java.

III. ORGANIZACIÓN DE ACTIVIDADES

<p>MATERIALES Y/O RECURSOS</p>	<p>a. Proyector de multimedios b. Computadora Personal c. Presentación realizada en Google Slides de Estructuras Repetitivas. d. Aplicación NetBeans y/o BlueJ para poder crear los programas en java.</p>	<p>COLEGIO DE CIENCIAS Y HUMANIDADES PLANTEL AZCAPOTZALEO JEFATURA DE SECCIÓN ÁREA DE MATEMÁTICAS</p> <p>19/07/2022</p> <p>RECIBIDO</p>
<p>DESARROLLO Y ACTIVIDADES</p>	<p>Inicio (Duración: 30 minutos).</p> <p>1. Llevando como guía la presentación electrónica, el profesor elabora la siguiente pregunta:</p> <p style="text-align: center;">¿Qué es un contador y un acumulador?</p> <p>Conceptos relevantes que se estarán utilizando con estas estructuras repetitivas.</p> <p>contador: Variable que inicia con un valor numérico, generalmente 0 o 1, y que va aumentando o disminuyendo a condición del programa. Por ejemplo: i++, significa que la variable «i» aumentará su valor de 1 en 1.</p> <p>acumulador: También es una variable, como su nombre indica, va acumulando valores diferentes. Por ejemplo:</p> <p style="text-align: center;"><i>SumadeCalificaciones = SumadeCalificaciones + Calificación</i></p> <p>La diferencia entre un contador y un acumulador es su incremento, en el contador el incremento es una constante y en el acumulador es una variable.</p> <p>Presentación por parte del docente donde se exponen las diferentes estructuras de control iterativas y a través de la ejemplificación de pequeños ejercicios donde el alumno los irá entendiendo y apropiando; se darán los elementos necesarios para el empleo de las estructuras de control repetitivas for, while y while-do.</p> <p>Desarrollo (55 minutos)</p>	

Los estudiantes realizarán tres prácticas aplicando la estructura de **for**, **while** y **while-do**. Usarán la aplicación NetBeans y/o BlueJ para crear los programas y se solicitará que los estudiantes escriban el código de sus programas para posteriormente entregarla al profesor.



Practica #1. Utilizando la estructura while “mientras.... hacer”

El bucle “mientras.... “, es una estructura que se repite mientras una condición sea verdadera. La condición, en forma de expresión lógica, se escribe en la cabecera del bucle, y a continuación aparecen las acciones que se repiten.

Ejercicio: Escribir un algoritmo que muestre en la pantalla los números enteros entre el 1 y 100.

```
public class while100
{
    public static void main (String[]args)
    {
        int cont=0;
        while(cont<100) {
            cont=cont+1;
            System.out.println(cont);
        }
    }
}
```

Practica #2. Utilizando la estructura do while “hacer....mientras”

Esta estructura es muy similar al bucle while con la salvedad de que la condición se evalúa al final del ciclo y no al principio. Se debe mencionar que la diferencia fundamental es que las acciones del cuerpo del do while se ejecutan al menos una vez, cuando en el while es posible que no se ejecute ninguna (si la condición es falsa desde el principio).

Ejercicio. Escribir un algoritmo que muestre en la pantalla los números enteros entre el 1 y 100. Utilizando el ciclo do while.

```

public class dowhile100
{
    public static void main (String[]args)
    {
        int cont=0;
        do{
            cont=cont+1;
            System.out.println(cont);
        }while (cont<100)
    }
}

```



Practica #3. Utilizando la estructura FOR “para”

En muchas ocasiones se conoce de antemano el número de veces que se desean ejecutar las acciones del cuerpo del bucle. Cuando el número de repeticiones es fijo, lo más cómodo es usar un bucle “para” aunque puede ser perfectamente sustituido por uno “mientras”.

Ejercicio. Escribir un algoritmo que muestre en la pantalla los números enteros entre el 1 y 100. Utilizando el ciclo FOR.

```

public class dowhile100
{
    public static void main (String[]args){
        for (int cont=1; cont<=100; cont++){
            System.out.println(cont);
        }
    }
}

```

Cierre. (30 minutos)

Se solicitará al estudiante realizar la siguiente Práctica #3 Crear un programa que solicite al usuario **n** calificaciones, donde **n** es también proporcionado por el usuario, obtenga el promedio de las **n** calificaciones y muestre la leyenda “APROBADO, ¡FELICIDADES!” si es que el promedio es mayor o igual a 6, en caso contrario muestre la leyenda “REPROBADO, CONTINÚA PREPARÁNDOTE”. Usarán la

	<p>aplicación NetBeans y/o BlueJ y se solicitará que el estudiante escriba el código de su programa en una hoja para entregarla al profesor.</p>
--	--



ORGANIZACIÓN	<p>La estrategia está diseñada para grupos de 45 estudiantes. Los estudiantes participarán de manera individual y grupal.</p>
EVALUACIÓN	<p>Mientras el desarrollo de esta actividad, el profesor estará realizando evaluación formativa por medio del monitoreo, guiando y resolviendo las dudas que pudieran surgir.</p> <p>Se indican las evidencias de aprendizaje de los alumnos y los instrumentos con los que se realizará su evaluación.</p>

IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama: Ra–Ma.</p> <p>Jiménez, J. (2014). Fundamentos de programación, diagramas de flujo, diagramas n–s, pseudocódigo y Java. México: Alfaomega.</p> <p>Olsson, M. (2013). Java quick syntax reference. Berkeley, California: Apress.</p> <p>Sierra, M. (2014). Programador Java certificado: curso práctico. Madrid: Ra–Ma.</p> <p>Sznajdleder. (2012). Java a fondo, estudio del lenguaje y desarrollo de aplicaciones. México: Alfaomega.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Deitel, P. y Deitel, H. (2012). <i>Cómo programar en Java</i> (9ª edición). México: Pearson.</p> <p>Discoduroderoer. (2013). <i>Bucles anidados en Java</i>. Recuperado de https://www.discoduroderoer.es/bucles-anidados-en-java/</p> <p>Gaddis, T. (2012). <i>Starting out with Java from control structures through objects</i>. United States: Pearson.</p> <p>López, L. (2013). <i>Metodología de la programación orientada a objetos</i>. México: Alfaomega.</p> <p>Lowe, D. (2014). <i>Java for dummies</i>. United States: John Wiley & Sons, Inc.</p> <p>Compartiendo conocimientos (2020). <i>Estructuras de control repetitivas</i>. Recuperado de http://elaprenderdiadia.blogspot.com/2020/10/sentencias-repetitivas-en-java.html</p>

ESTRATEGIA DIDÁCTICA 3

I. DATOS GENERALES



PROFESOR(A)	Seminario Institucional
ASIGNATURA	Cibernética y Computación II
SEMESTRE ESCOLAR	Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	25 noviembre del 2021

II. PROGRAMA

UNIDAD TEMÁTICA	Unidad II. Estructuras de control de secuencia en Java
PROPÓSITO(S) DE LA UNIDAD	<p>Propósitos indicativos:</p> <p>Al finalizar la unidad el estudiante: Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java</p>
APRENDIZAJE(S)	<p>Aprendizaje indicativo:</p> <p>Resuelve problemas que involucren el uso de los arreglos unidimensionales en los métodos de una Clase.</p>
CONTENIDO TEMÁTICO	<p>Arreglos unidimensionales:</p> <ul style="list-style-type: none"> • Concepto. • Declaración. • Tipos. • Uso.
DURACIÓN	<p>Tiempo estimado actividades presenciales: 120 minutos</p> <p>Tiempo estimado actividades no presenciales: 120 minutos</p>

APRENDIZAJES ESPERADOS	Al finalizar el tema, el estudiante realizará programas que involucren arreglos unidimensionales mediante prácticas dirigidas para el manejo de la solución de problemas con arreglos.
MATERIALES Y RECURSOS DE APOYO	<ul style="list-style-type: none"> • Computadora Personal y/o teléfono móvil • Plataforma Moodle • Presentación realizada en Google Slides de Arreglos • Tutorial de Arreglos ubicado en YouTube. • Aplicación NetBeans y/o BlueJ para poder crear los programas en java.



III. ESTRATEGIA

El desarrollo de esta estrategia será en modalidad presencial y no presencial. EL profesor revisará arreglos unidimensionales definición y uso de vectores, recorrido, búsqueda e inserción mediante una presentación localizada en el aula virtual de MOODLE y ejemplificará el uso de la estructura de los arreglos por medio de programas ya elaborados. Además, que los estudiantes elaborarán algunos ejercicios propuestos por el profesor.

IV. SECUENCIA

DESARROLLO Y ACTIVIDADES	<p>Inicio</p> <p>Tiempo estimado actividades presenciales: 0 minutos</p> <p>Tiempo estimado actividades no presenciales: 30 minutos</p> <ol style="list-style-type: none"> 1. Actividad no presencial/Individual. Los estudiantes, a través de la presentación de arreglos ubicada en la plataforma virtual de Moodle, se apropiarán de los conceptos de arreglos unidimensionales mediante ejemplos. En la misma presentación, se explica la sintaxis para declarar o inicializar a un arreglo. (Anexo A).
---------------------------------	--

2. Actividad no presencial/Individual. Los estudiantes revisarán el tutorial sobre la estructura de datos Array
<https://youtu.be/azLb3M3k9pc>

Desarrollo

Tiempo estimado actividades presenciales: 90 minutos

Tiempo estimado actividades no presenciales: 0 minutos

1. Actividad presencial / En parejas. Los estudiantes realizarán dos prácticas aplicando la estructura de arreglos unidimensionales. Usarán la aplicación NetBeans y/o BlueJ para crear los programas y se solicitará que los estudiantes escriban el código de sus programas en una hoja para posteriormente entregarla al profesor.

Practica #1. Hacer un programa en Java que permita al usuario introducir n números dentro de un arreglo, donde n es proporcionado por el usuario, después mostrarlos en el orden introducido.

Practica #2. Hacer un programa en Java que solicite 10 números al usuario y obtenga y muestre el mayor de esos números.

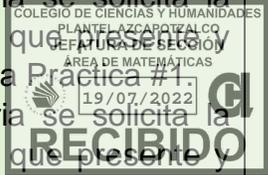
Mientras el desarrollo de esta actividad, el profesor estará realizando evaluación formativa por medio del monitoreo, guiando y resolviendo las dudas que pudieran surgir.

Cierre

Tiempo estimado actividades presenciales: 30 minutos



	<p>Tiempo estimado actividades no presenciales: 30 minutos</p> <ol style="list-style-type: none"> Actividad presencial / Grupal. En plenaria se solicita la participación voluntaria de una pareja para que presente y explique su programa elaborado respecto a la Práctica #1. Actividad presencial / Grupal. En plenaria se solicita la participación voluntaria de una pareja para que presente y explique su programa elaborado respecto a la Práctica #2. Actividad no presencial / Individual. Se solicitará al estudiante realizar la siguiente <u>Práctica #3</u> Crear un programa usando arreglos que solicite al usuario n calificaciones, donde n es también proporcionado por el usuario, obtenga el promedio de las n calificaciones y muestre la leyenda "APROBADO, ¡FELICIDADES!" si es que el promedio es mayor o igual a 6, en caso contrario muestre la leyenda "REPROBADO, CONTINÚA PREPARÁNDOSE". Usarán la aplicación NetBeans y/o BlueJ y se solicitará que el estudiante escriba el código de su programa en una hoja para entregarla al profesor. 					
<p>ORGANIZACIÓN</p>	<p>La estrategia está diseñada para grupos de 45 estudiantes.</p> <p>Se conforma de actividades presenciales y no presenciales.</p> <p>Los estudiantes participarán de manera individual, en pareja y grupal.</p>					
<p>EVALUACIÓN</p>	<p>Las prácticas #1, #2 y #3 serán evaluadas por el profesor y consideradas para su calificación. Se utilizará la siguiente Rúbrica para evaluar cada una de las prácticas:</p> <table border="1" data-bbox="516 1352 1474 1879"> <tr> <th data-bbox="516 1352 711 1879">Entrega</th> <td data-bbox="711 1352 906 1879">Se entrega con los 3 siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o</td> <td data-bbox="906 1352 1101 1879">Se entrega con 2 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o</td> <td data-bbox="1101 1352 1295 1879">Se entrega con sólo 1 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o</td> <td data-bbox="1295 1352 1474 1879">Se entrega sin ninguno de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios</td> </tr> </table>	Entrega	Se entrega con los 3 siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o	Se entrega con 2 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o	Se entrega con sólo 1 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o	Se entrega sin ninguno de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios
Entrega	Se entrega con los 3 siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o	Se entrega con 2 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o	Se entrega con sólo 1 de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios que definan o	Se entrega sin ninguno de los siguientes elementos: 1. código escrito en hojas de papel. 2. en los tiempos establecidos 3. con letra legible y con los espacios necesarios		



	aclaren la estructura del programa. (2 puntos)	aclaren la estructura del programa. (1 punto)	aclaren la estructura del programa. (0.5 punto)	que definan o aclaren la estructura del programa. (0 punto)
Resolución	Utiliza todos los elementos solicitados para resolver el ejercicio. (4 puntos)	Utiliza algunos elementos solicitados para resolver el ejercicio. (2 puntos)	No utiliza los elementos solicitados para resolver el ejercicio. (1 punto)	No resuelve el ejercicio. (0 punto)
Funcionamiento	El funcionamiento del programa es completo (4 puntos)	El funcionamiento del programa no es completo, aunque tiene fallas sin importancia. (2 puntos)	El funcionamiento del programa tiene fallas importantes. (1 punto)	No funciona el programa. (0 punto)



V. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ESTUDIANTES.	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama: Ra–Ma.</p> <p>Jiménez, J. (2014). Fundamentos de programación, diagramas de flujo, diagramas n–s, pseudocódigo y Java. México: Alfaomega.</p> <p>Olsson, M. (2013). Java quick syntax reference. Berkeley, California: Apress.</p> <p>Tutorial sobre la estructura de datos Array. Youtube, (2015) https://youtu.be/azLb3M3k9pc</p>
---	---

	<p>Sierra, M. (2014). Programador Java certificado: curso práctico. Madrid: Ra–Ma.</p> <p>Sznajdleder. (2012). Java a fondo, estudio del lenguaje y desarrollo de aplicaciones. México: Alfaomega.</p>
<p>BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR</p>	<p>Aguilar, L. J. (1999). <i>Programación orientada a objetos</i>. McGraw-Hill Interamericana.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>García, L., et al. (2003). Construcción lógica de programas. Teoría y problemas resueltos. México: Alfaomega.</p> <p>Jackson, T. (2021). <i>Lenguaje Java para principiantes: Una guía paso a paso en el lenguaje de programación Java</i>. Independently Published.</p> <p>Wu, C. (2008). Programación en Java. México: McGraw Hill</p>
<p>COMENTARIOS ADICIONALES</p>	





PRESENTACIÓN

 <p>Arreglos Matrices, Arrays</p>	<p>ARREGLOS (ARRAYS) Vectores o Listas</p> <p>Es una colección de datos, organizados de un modo particular, los cuales permiten representar, almacenar, organizar y procesar datos simples y datos más complejos o compuestos.</p> <p>Sirven para almacenar valores que normalmente tienen una relación entre sí.</p>
<p>ARREGLOS (ARRAYS) Vectores o Listas</p> <p>Existen dos tipos de arreglos:</p> <ul style="list-style-type: none"> • Unidimensionales • Bidimensionales <p>El tamaño de los arreglos se declaran en un primer momento y no puede cambiar luego durante la ejecución del programa.</p>	<p>ARREGLOS</p> <p>Un arreglo es una estructura de datos en la que se almacena un conjunto de datos del mismo tipo (por ejemplo: los salarios de los empleados de una empresa).</p> <p>Un arreglo es una lista de un número finito de "N" elementos del mismo que se caracteriza por:</p> <ol style="list-style-type: none"> 1. Almacenar los elementos del arreglo en posiciones de memoria continua. 2. Tener un único nombre de variables que representa a todos los elementos y estos a su vez, se diferencian por un índice o subíndice. 3. Acceso directo ó aleatorio a los elementos individuales.

Arreglos UNIDIMENSIONALES: (Vectores o Listas).

Es un tipo de dato estructurado compuesto de un número de elementos finitos, tamaño fijo y elementos homogéneos

Finito Siempre existe un último elemento.

Tamaño El Tamaño del arreglo debe ser conocido en tiempo de compilación.

Homogéneo Todos los elementos son del mismo tipo.

Los elementos del arreglo se almacenan en posiciones contiguas de memoria, a cada una de las cuales se puede acceder directamente.

Arreglos UNIDIMENSIONALES: (Vectores o Listas).

Se desea conocer las calificaciones de 20 estudiantes de Ciberfísica II. Para almacenar las calificaciones simultáneamente se necesitan reservar 20 posiciones en memoria.

	Calificaciones
Calificaciones [1]	9.35
Calificaciones [2]	8.37
Calificaciones [3]	6.54
.	
.	
.	
Calificaciones [20]	9.54

En este caso el nombre del vector se llama **CALIFICACIONES** el cual se divide en **20 localidades de memoria**.

Nombre de vector:
CALIFICACIONES
Subíndice: [1..[20]
Contenido Calificaciones[2]= 8.37



Arreglos UNIDIMENSIONALES: (Vectores o Listas).

Se desea almacenar simultáneamente 100 diferentes sueldos de los empleados de una empresa.

	SUELDOS
Sueldos [1]	100.50
Sueldos [2]	2,500.00
Sueldos [3]	6,000.50
Sueldos [4]	1,570.50
.	
.	
Sueldos [100]	3,587.76

En este caso el nombre del vector se llama **SUELDOS** el cual se divide en **100 localidades de memoria**.

Nombre de vector: SUELDOS
Subíndice: [1..[100]
Contenido Sueldos[4]= 1,570.50

Arreglos UNIDIMENSIONALES: (Vectores o Listas).

Ejemplos de arreglos

`int edad[] = new int [4];`

`long [] edad = new long [4];`

`float [] estatura = new float [3];`

`double [] estatura = new double [3];`

`boolean [] estado = new boolean [5];`

`char [] sexo = new char [2];`

`String [] nombre = new String [2];`

Unidad 3: Polimorfismo, constructores, colaboración y herencia de clases.

Propósito: Al finalizar la unidad el alumno:

Implementará programas en Java utilizando polimorfismo, constructores, colaboración y herencia de Clases para aprovechar las bondades de la programación orientada a objetos.



ESTRATEGIA DIDÁCTICA 1

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Cibernética y Computación II
SEMESTRE ESCOLAR	Sexto semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Abril 2022
NO DE ALUMNOS	25-30

II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad 3. Polimorfismo, constructores, colaboración y herencia de clases
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Implementará programas en Java utilizando polimorfismo, constructores, colaboración y herencia de Clases para aprovechar las bondades de la programación orientada a objetos.
APRENDIZAJE(S)	<p>Conoce el concepto de polimorfismo y constructor.</p> <p>Desarrolla programas que involucren polimorfismo y constructores.</p> <p>Comprende la colaboración de Clases para la resolución de problemas</p> <p>Desarrolla programas que involucren la colaboración de Clases</p> <p>Comprende el concepto de herencia en la resolución de un problema</p> <p>Desarrolla programas que involucren la herencia de Clases.</p>
TEMA(S)	Herencia simple, colaboración entre clases, polimorfismo y clases abstractas

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	Tres sesiones de dos horas, total de horas 6.
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar en equipo
DESARROLLO Y ACTIVIDADES	<p>Sesión 1</p> <p>INICIO (20 min)</p> <p>El profesor describe la sobrecarga de métodos explicando a los alumnos los siguientes conceptos:</p> <p>Constructores</p> <p>Los constructores son métodos especiales que se usan para inicializar los valores de los atributos de un objeto.</p> <p>Las características que debe cumplir un método constructor son:</p> <ul style="list-style-type: none"> • Mismo nombre de la clase. • No tiene tipo de retorno. <p>Existen 3 tipos de constructores</p> <ul style="list-style-type: none"> • Por omisión • Por parámetros • Por copia <p>1. Un constructor por omisión se utiliza para inicializar los miembros dato (variables) de la clase con un determinado valor.</p> <p>Puede ser escrito o no ya que Java implícitamente lo declara en la creación de cada una de sus clases. Se puede declarar igual con o sin elementos.</p>



```

public class Cuenta
{
    long int numero;
    double saldo;
    String titular;

    public Cuenta()
    {
        numero = 1234567;
        saldo = 1500.0;
    }
}

```



2. Un constructor por parámetros inicializa los miembros dato (variables) de la clase con la lista de valores que recibe.

```

public class Cuenta
{
    long int numero;
    double saldo;
    String titular;

    public Cuenta(long int num, double s, String tit)
    {
        numero = num;
        saldo = s;
        titular = tit;
    }
}

```

3. Este constructor recibe una referencia a un objeto para copiar los valores de este

```

public class Cuenta
{
    long int numero;
    double saldo;
    String titular;

    public Cuenta(Cuenta c)
    {
        numero = c.getNumero();
        saldo = c.getSaldo();
        titular = c.getTitular();
    }
}

```

En programación orientada a objetos es posible definir dos o más métodos, dentro de la misma clase, que compartan el mismo nombre, pero las declaraciones de sus parámetros deben de ser diferentes, esto es a lo que se le conoce como Sobrecarga de Métodos.

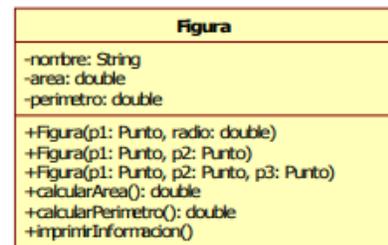
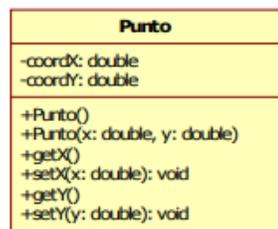
Reglas para sobrecargar métodos:

- Los métodos sobrecargados deben cambiar la lista de argumentos.
- Pueden cambiar el tipo de retorno.
- Pueden declarar nuevas o más amplias excepciones.
- Un método puede ser sobrecargado en la misma clase.

El concepto de sobrecarga se utiliza en polimorfismo.

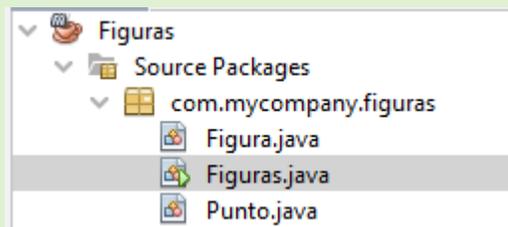
DESARROLLO (20 min)

El profesor interpreta y codifica los siguientes diagramas de clases:



Con ayuda de un IDE codificaremos las siguientes clases en lenguaje java.

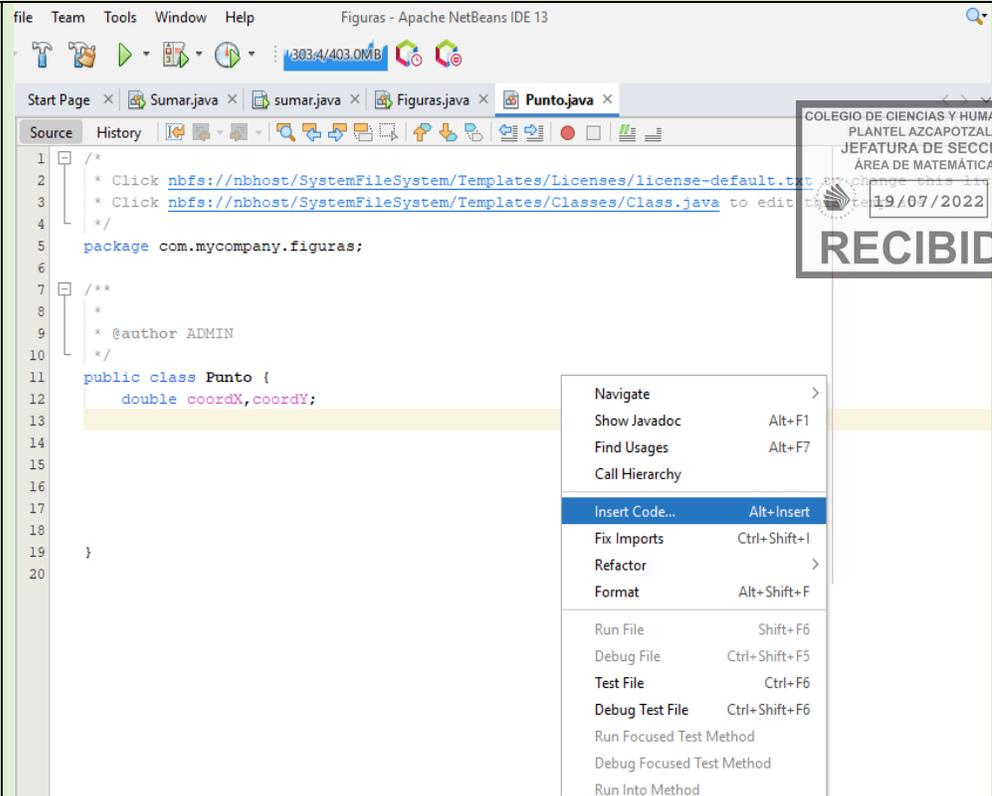
Una vez creado nuestro proyecto procederemos a agregar las siguientes clases:



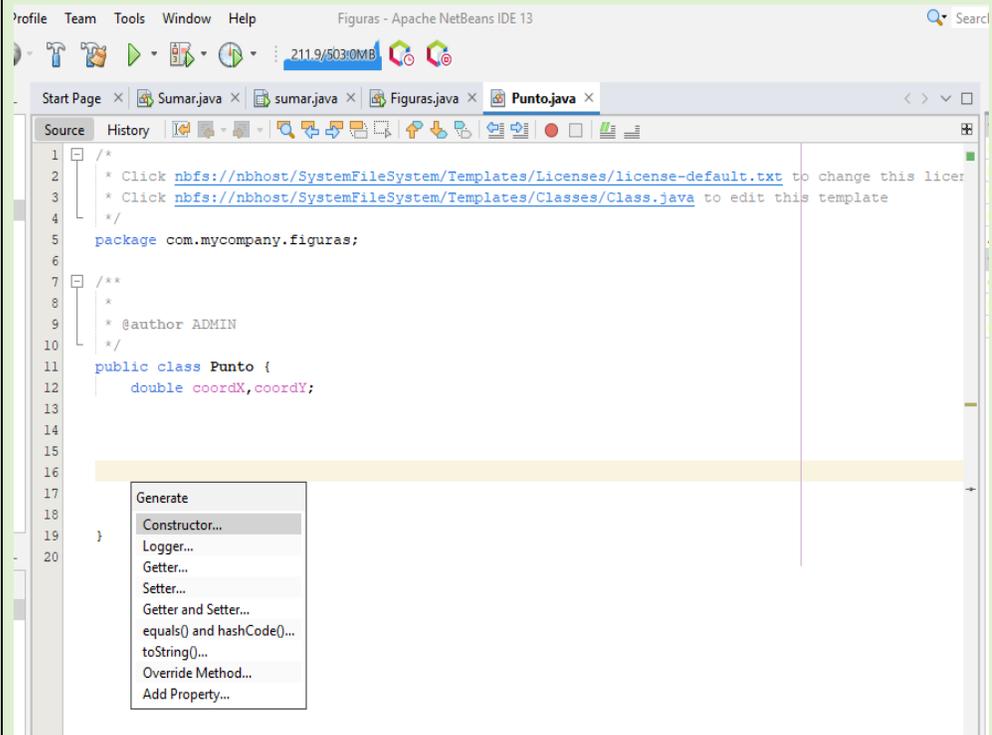
Clase Punto:

Declaremos dos variables de acceso privado y tipo doble para almacenar valores numéricos reales. Usemos la automatización del IDE para completar el código de nuestra clase punto, dando click derecho en la clase como muestra la figura aparecerán las siguientes opciones:

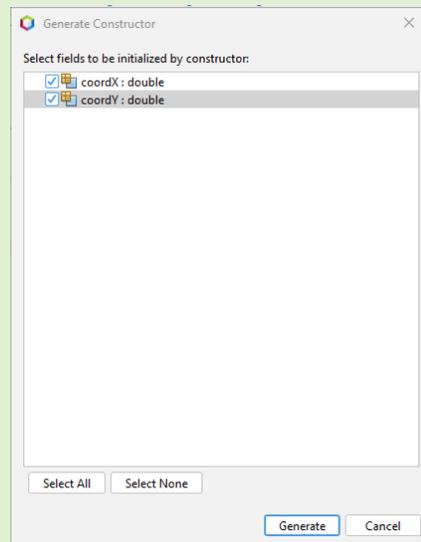




Seleccionando la opción Insert Code aparecerá la siguiente pantalla de opciones:

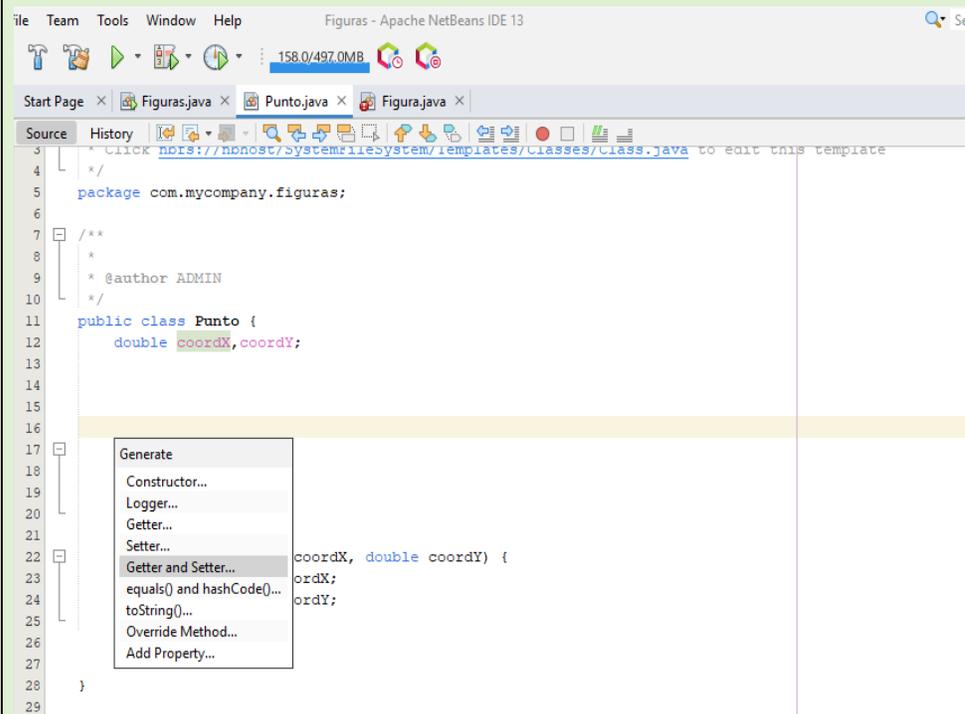


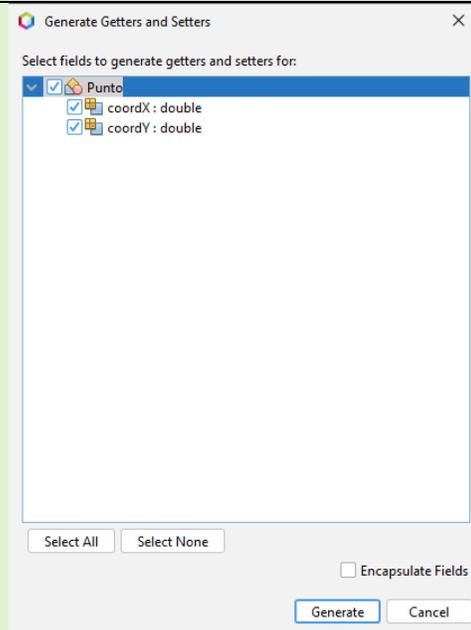
Si seleccionamos la opción Constructor autocompletará el constructor por parámetros seleccionando las variables de tipo privado que tiene la clase apareciendo la siguiente ventana:



```
public Punto(double coordX, double coordY) {  
    this.coordX = coordX;  
    this.coordY = coordY;  
}
```

Usemos la misma capacidad del IDE de autocompletar código para crear los setters y getters, dando click derecho en la clase y seleccionando la opción de Getters y Setters:





Quedando finalmente la clase Punto de la siguiente manera:

```
public class Punto {
    double coordX, coordY;

    public Punto(){
        coordX = 10;
        coordY = 10;
    }

    public Punto(double coordX, double coordY) {
        this.coordX = coordX;
        this.coordY = coordY;
    }

    public double getCoordX() {
        return coordX;
    }

    public void setCoordX(double coordX) {
        this.coordX = coordX;
    }

    public double getCoordY() {
        return coordY;
    }
}
```

```

public void setCoordY(double coordY) {
    this.coordY = coordY;
}
}

```

Nótese que se ha agregado un constructor por default que usaremos para el cálculo de las figuras geométricas:

```

public Punto() {
    coordX = 10;
    coordY = 10;
}

```

A continuación, se mostrará la aplicación de la sobrecarga de métodos constructores, recordando que el constructor es el método que inicializa los atributos de una clase, y al ser un método puede ser sobrecargado.

Clase Figura:

```

public class Figura {

    private String nombre;
    private double area;
    private double perimetro;

    public Figura(){
        nombre = "Cuadrado";
        area = 100;
        perimetro = 40;
    }

    public Figura(Punto p1, double radio){
        nombre = "Circulo con centro en (" + p1.getCoordX() + ", " + p1.getCoordY() + ")";
        area = Math.PI * (Math.pow(radio, 2));
        perimetro = 2 * Math.PI * radio;
    }

    public Figura(Punto p1, Punto p2){

        nombre = "Rectangulo";
        double x1, x2, y1, y2;

```





```
}  
  
public Figura(Punto p1, Punto p2, Punto p3){  
  
    nombre="Triangulo";  
    double x1,x2,x3,y1,y2,y3,d1,d2,d3,p;  
    x1=p1.getCoordX();  
    x2=p2.getCoordX();  
    x3=p3.getCoordX();  
    y1=p1.getCoordY();  
    y2=p2.getCoordY();  
    y3=p3.getCoordY();  
    d1=Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));  
    d2=Math.sqrt((x3-x1)*(x3-x1)+(y3-y1)*(y3-y1));  
    d3=Math.sqrt((x3-x2)*(x3-x2)+(y3-y2)*(y3-y2));  
    perimetro=d1+d2+d3;  
    p=(d1+d2+d3)/2.0;  
    area=Math.sqrt(p*(p-d1)*(p-d2)*(p-d3));  
}  
  
public void imprimirInformacion(){  
  
    System.out.println("-----Datos de la figura geometrica-----");  
    System.out.println("nombre="+nombre);  
    System.out.println("area="+area);  
    System.out.println("perimetro="+perimetro);  
}  
} //fin de la clase Figura
```

En la clase Figura la sobrecarga del método constructor servirá para calcular el área y el perímetro de una figura geométrica dadas las coordenadas de sus puntos y fórmulas de geometría.

En la clase principal dependiendo de la invocación del método sobrecargado serán los datos de la figura geométrica que se mostrará en pantalla.

A continuación, se deberá crear una nueva Clase llamada Figuras la cual contendrá el método main.

```

Clase Figuras(main):
public class Figuras {

    public static void main(String[] args) {
        Punto p1,p2,p3,p4,p5;
        p1=new Punto();
        p2=new Punto(5.6,1.6);
        p3=new Punto(8.9,7.2);
        p4=new Punto(6.0,4.1);
        p5=new Punto(12.6, 14.8);
        Figura figs[]=new Figura[4];
        figs[0]=new Figura();
        figs[1]=new Figura(p1,18.5);
        figs[2]=new Figura(p1,p3);
        figs[3]=new Figura(p4,p5,p1);
        for(int i=0; i<figs.length;i++){
            figs[i].imprimirInformacion();
        }

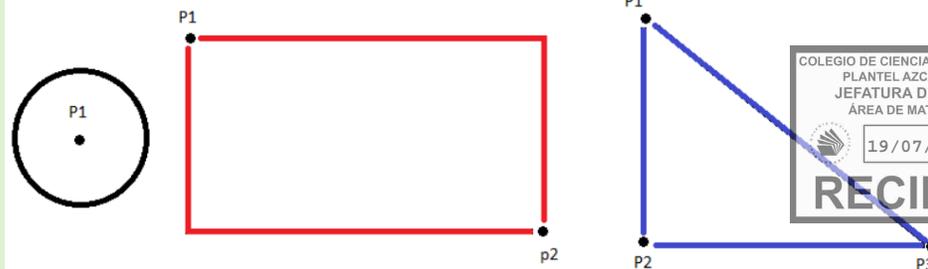
    }
}

```



En la clase principal se declara un arreglo de objetos de tipo Figura mandando a sus constructores sobrecargados en cada objeto del arreglo un conjunto de puntos y los parámetros específicos en cada constructor para aplicar la sobrecarga y saber distinguir que método se encarga de mostrar lo datos de la figura geométrica dependiendo del constructor invocado.

El constructor por default calcula los datos de un cuadrado, el constructor que tiene por parámetros un objeto punto y valor doble calculará los datos de un círculo, el constructor que acepta dos puntos por parámetros mostrará el área y perímetro de un rectángulo y el constructor que solicita 3 puntos como parámetros calcula los datos de un triángulo.



Cierre (80 min)

Los alumnos desarrollan el código mostrado para observar la aplicación de la sobrecarga de métodos en la programación orientada a objetos y completan el código del método sobrecargado para el cálculo de los datos del rectángulo:

```
public Figura(Punto p1, Punto p2){

    nombre="Rectangulo";
    double x1,x2,y1,y2;
}
```

Sesión 2

El profesor describe la herencia entre clases explicando a los alumnos los siguientes conceptos:

Herencia (Especialización/Generalización): 

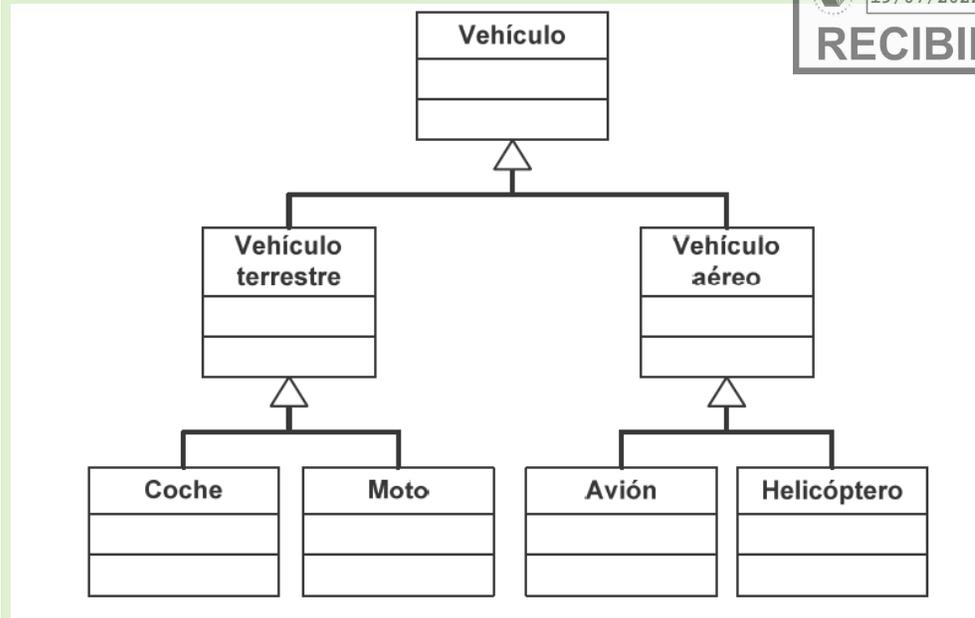
Indica que una subclase (clase hija) hereda los métodos y atributos especificados por una súper clase (clase padre), por ende, la subclase además de poseer sus propios métodos y atributos también poseerá las características y atributos visibles de la súper clase (public y protected).

Herencia simple:

La herencia es el mecanismo por el que se crean nuevos objetos definidos en términos de objetos ya existentes. Por ejemplo, si se tiene la clase Ave, se puede crear la subclase Pato, que es una especialización de la primera.

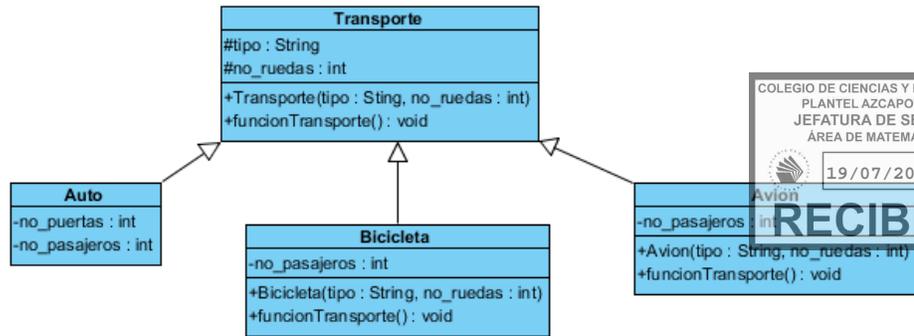
Es la facilidad mediante la cual una clase hereda las propiedades y métodos públicos de otra clase. La herencia puede ser simple o

múltiple. Es simple cuando una clase hereda solamente de otra clase superior (se convierte una relación subclase – superclase) mientras que la herencia múltiple es cuando la clase hereda de 2 o más clases superiores. En Java y .NET sólo existe la herencia simple.



DESARROLLO (20 min)

Los estudiantes del grupo se dividen en equipos de 4 alumnos, con base en el diagrama de clases previamente explicado por el profesor, se solicita a los equipos de trabajo que elaboren un programa en java en donde aplicarán el concepto de herencia simple, dependiendo de la cantidad de pasajeros, cantidad de ruedas y el número de puertas que tenga un vehículo, la salida del programa deberá de indicar si se trata de un auto, una bicicleta o un avión. Como sugerencia se muestra el diagrama de clases en donde se observa gráficamente el problema planteado.

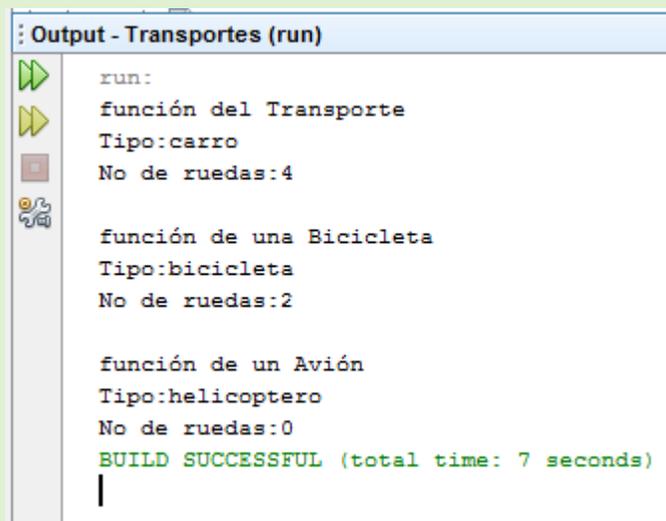


El diagrama de clases muestra una super clase o clase padre **Transporte** que heredará sus atributos a las 3 clases hijas **Auto**, **Bicicleta** y **Avión** y un método a solo una clase hija, ilustrando en este ejemplo que con la herencia de métodos se puede realizar la misma acción ya sea desde la clase padre o en las clases hijas. Recordar siempre usar la palabra reservada `extends` en las clases hijas para poder aplicar la herencia en Java

```
public class Auto extends Transporte {
```

En la clase principal(Figuras.java) al volver los objetos de tipo Transporte pero haciendo la invocación con los constructores de las clases hijas estamos aplicando la herencia simple y sabiendo de que tipo de transporte se trata.

```
Transporte carro = new Auto("carro",4);
Transporte bici=new Bicicleta("bicicleta",2);
Transporte helicoptero=new Avion("helicoptero",0);
```



Clase Transporte (clase padre)

```
public class Transporte {
    _____ String tipo;
    _____ int no_ruedas;

    public Transporte(String tipo, int no_ruedas){
        this.tipo = tipo;
        this.no_ruedas = no_ruedas;
    }

    public void funcionTransporte(){
        System.out.println("función del Transporte");
        System.out.println("Tipo:"+tipo);
        System.out.println("No de ruedas:"+no_ruedas);
    }
}
```

Clase Auto(clase hija)

```
public class Auto _____ Transporte {
    private int no_puertas;
    private int no_pasajeros;

    public Auto(String tipo, int no_ruedas) {
        _____(tipo, no_ruedas);
    }
}
```

Clase avión (clase hija)

```
public class Avion _____ Transporte {

    private int no_pasajeros;

    public Avion(String tipo, int no_ruedas) {
        super(tipo, no_ruedas);
    }
}
```





```
@Override
public void funcionTransporte(){
    System.out.println("función de un Avión");
    System.out.println("Tipo:"+tipo);
    System.out.println("No de ruedas:"+no_ruedas);
}

}

Clase Bicicleta( clase hija)

public class Bicicleta _____ Transporte{
    private int no_pasajeros;

    public Bicicleta(String tipo, int no_ruedas) {
        super(tipo, no_ruedas);
    }

    @Override
    public void funcionTransporte(){
        System.out.println("función de una Bicicleta");
        System.out.println("Tipo:"+tipo);
        System.out.println("No de ruedas:"+no_ruedas);
    }
}

Clase transportes (clase principal)
public class Transportes {

    public static void main(String[] args) {
        _____ carro = new Auto("carro",4);
        _____ bici=new Bicicleta("bicicleta",2);
        Transporte helicoptero=new Avion("helicoptero",0);
    }
}
```

```
carro.funcionTransporte();
System.out.println();
bici.funcionTransporte();
System.out.println();
helicoptero.funcionTransporte();
}
}
```

Cierre (80 min)

Los alumnos desarrollan (en el IDE que se utilice en las sesiones) y completan los espacios en blanco del código mostrado para observar la aplicación de la herencia simple.

Sesión 3

INICIO (20 min)

El profesor describe el polimorfismo y clases abstractas explicando a los alumnos los siguientes conceptos:

Polimorfismo

Se refiere a la capacidad de los objetos de comportarse de acuerdo a la funcionalidad requerida. Es decir, establecer diferentes comportamientos para los métodos, por ejemplo, del objeto, que implementan diversas funcionalidades de acuerdo con parámetros recibidos a través de los mismos. Por ejemplo, se puede realizar una suma de enteros que devuelva enteros y reciba parámetros enteros, o una suma de valores flotantes donde se reciban parámetros flotantes y devuelva un valor flotante. Ambos métodos pueden llamarse igual, pero se diferencian por los parámetros que reciben.

Clases Abstractas

Una clase abstracta implementa la estructura genérica de la funcionalidad que deberá ser implementada en una clase derivada. Una clase abstracta en realidad no puede usarse, es decir, no se puede instanciar en un objeto para ser utilizada. Más bien se deriva en otra clase y sobre ésta es que se crea la instanciación.

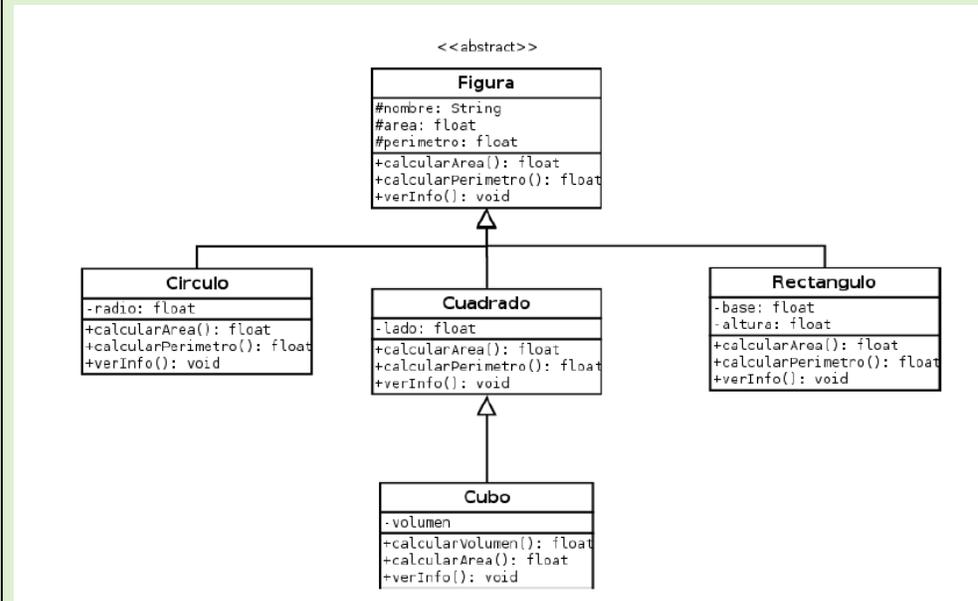
Una clase abstracta se puede crear o definir cuando se necesita englobar objetos de tipo diferente y se quiere implementar polimorfismo. Supóngase que se requiere crear un sistema que sea



lo suficientemente genérico (parametrizable) pero que establezca el cliente los métodos que obligatoriamente deben realizarse.

Desarrollo y cierre (100 min)

El profesor explica el siguiente diagrama de clases y los alumnos con los conocimientos previamente obtenidos en esta sesión, así como en las unidades anteriores lo interpretan y codifican.



En esta actividad además de implementar el polimorfismo se ilustra la herencia por niveles como lo muestra el diagrama de clases, recordar que en el polimorfismo las clases abstractas son las clases que carecen de implementación en sus métodos ya que éstos serán implementados en otras clases ya sea en una relación de herencia, como en este caso, o en otras relaciones entre clases. El diagrama de clases de este ejemplo muestra que mediante la herencia y el polimorfismo se pueden describir las características de figuras geométricas y cuerpos geométricos al repetirse los métodos en las clases hijas, pero con diferente implementación.

Output - HerenciaMultiple (run)

```
run:
Tipo de figura:Cuadrado
Lado:4.0
Area:16.0
Perimetro:16.0

Tipo de figura:Circulo
Radio:3.0
Area:28.274334
Perimetro:18.849556

Tipo de figura:Rectangulo
Base:3.0, Altura:5.0
Area:7.5
Perimetro:16.0

Tipo de figura:Cubo
Lado:3.0
Area:54.0
Perimetro:144.0
Volumen:27.0
BUILD SUCCESSFUL (total time: 4 seconds)
```



Clase Figura(Clase padre abstracta)

```
public abstract class Figura {

    protected String nombre;
    protected float area,perimetro;

    public Figura(String nombre){

        this.nombre= nombre;
    }

    public abstract float calcularArea();
    public abstract float calcularPerimetro();

    public void verInfo() {

        System.out.println("Tipo de figura:" +nombre);
    }
}
```

Clase Cuadrado (clase hija)

```

public class Cuadrado extends Figura {

    protected float lado;

    public Cuadrado(float lado) {
        super("Cuadrado");
        this.lado = lado;
        area = calcularArea();
        perimetro = calcularPerimetro();
    }

    @Override
    public float calcularArea() {
        //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
        return lado*lado;
    }

    @Override
    public float calcularPerimetro() {
        //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
        return lado*4.0f;
    }

    public void verInfo(){
        super.verInfo();
        System.out.println("Lado:"+lado);
        System.out.println("Area:"+area);
        System.out.println("Perimetro:"+perimetro);
    }
}

```

Clase Rectángulo (clase hija)

```

public class Rectangulo extends Figura {

    private float base,altura;

    public Rectangulo(float base, float altura){
        super("Rectangulo");
    }
}

```



```
this.base = base;
this.altura = altura;
area = calcularArea();
perimetro = calcularPerimetro();
}
```

```
@Override
public float calcularArea() {
    //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    return base*altura/2;
}
```

```
@Override
public float calcularPerimetro() {
    //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    return (2*base)+(2*altura);
}
```

```
public void verInfo(){
    super.verInfo();
    System.out.println("Base:"+base+", Altura:"+altura);
    System.out.println("Area:"+area);
    System.out.println("Perimetro:"+perimetro);
}
}
```

Clase Círculo (clase hija)

```
public class Circulo extends Figura {

    private float radio;

    public Circulo(float radio){
        super("Circulo");
        this.radio = radio;
        area = calcularArea();
        perimetro = calcularPerimetro();
    }
}
```



```

@Override
public float calcularArea() {
    //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    return (float) (Math.PI*radio*radio);
}

@Override
public float calcularPerimetro() {
    //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    return (float) (Math.PI*2.0f*radio);
}

public void verInfo(){
    super.verInfo();
    System.out.println("Radio:"+radio);
    System.out.println("Area:"+area);
    System.out.println("Perimetro:"+perimetro);
}
}

```

Clase Cubo (clase hija)

```

public class Cubo extends Cuadrado{

    private float volumen;

    public Cubo(float lado) {
        super(lado);
        this.nombre="Cubo";
        this.lado=lado;
        perimetro = calcularPerimetro();
        area = calcularArea();
        volumen=calcularVolumen();
    }

    public float calcularArea(){
        return super.calcularArea()*6.0f;
    }
}

```



```

public float calcularPerimetro() {
    //throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    return super.calcularPerimetro()*12.0f;
}

public float calcularVolumen(){
    return super.calcularArea()*lado;
}

public void verInfo(){
    super.verInfo();
    System.out.println("Volumen:"+volumen);
}
}

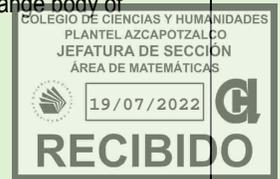
```

Clase Main (clase principal)

```

public class Main {
    public static void main(String[] args) {
        Cuadrado obj = new Cuadrado(4);
        obj.verInfo();
        System.out.println();
        Figura cir = new Circulo(3);
        cir.verInfo();
        System.out.println();
        Rectangulo rect = new Rectangulo(3,5);
        rect.verInfo();
        System.out.println();
        Figura cb = new Cubo(3);
        cb.verInfo();
    } //fin del método main
} //fin de la Clase Main

```



EVALUACIÓN	Sesión 1	
		Puntaje
	Definió las clases	2
	Definió y desarrolló la	2

		sobrecarga de constructores	
		Implementó correctamente el programa en java	6
		Total	10
	Sesión 2		
			Puntaje
		Definió las clases	2
		Definió y desarrolló la herencia simple de clases	2
		Implementó correctamente el programa en java	6
		Total	10
	Sesión 3		
			Puntaje
		Interpretó correctamente el diagrama de clases	4
		Implementó correctamente el programa en java	6
		Total	10



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Joyanes, L. & Zahonero, M. (2014). Programación en c, c++, Java y uml. México: McGraw–Hill Interamericana.</p> <p>Jiménez, J. (2014). Fundamentos de programación, diagramas de flujo, diagramas n–s, pseudocódigo y Java. México: Alfaomega.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Ceballos, F. (2006). Java: curso de programación. México: Alfaomega.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson Educación.</p> <p>Wu, C. (2008). Programación en Java. México: McGraw Hill.</p> <p>Belmonte, O. Introducción al lenguaje de programación Java.</p> <p>Joyanes, L. & Zahonero, M. (2014). Programación en c, c ++, Java y uml. México: McGraw–Hill Interamericana.</p> <p>Wu, C. (2008). Programación en Java. México: McGraw Hill.</p>



V. ANEXOS

Unidad 4: Interfaz gráfica de usuario

Propósito: Al finalizar la unidad el alumno:

Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos.



Estrategia didáctica 1

DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre
PLANTELE	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 28 de marzo de 2022
NO DE ALUMNOS	Número de alumnos: 25

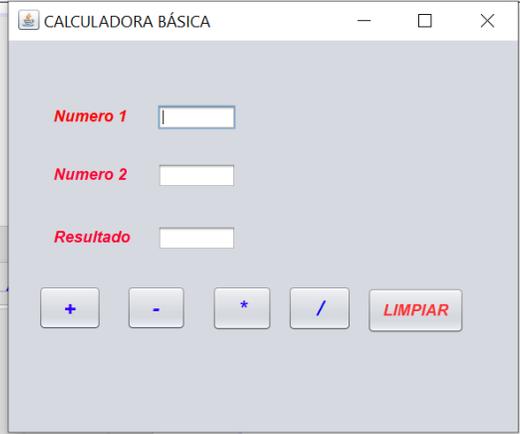
II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Elabora programas con una interfaz gráfica de usuario, aplicando las Clases: JFrame, JLabel, JTextField y JButton
TEMA(S)	Clase Swing: <ul style="list-style-type: none">• JFrame

	<ul style="list-style-type: none"> • JLabel • JTextField • JButton
--	---



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual
DESARROLLO Y ACTIVIDADES	<p>Inicio (20 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> • Comenta a los alumnos del grupo que empleando la interfaz gráfica de usuario se desea construir una calculadora básica, para ello muestra lo siguiente. <div style="text-align: center;">  </div> <p style="text-align: center;">Figura 1</p> <ul style="list-style-type: none"> • Comenta a los alumnos del grupo qué para el desarrollo de la Calculadora, deberán realizar lo siguiente.

- ✓ Utilizando el paquete **Swing** y empleando la clase **JFrame**, crear una ventana llamada **Calculadora**.
- ✓ De la ventana creada, seleccionar sus propiedades con la propiedad **title** colocar en el título de la ventana **CALCULADORA BÁSICA**, el tamaño y tipo de letra podrá ser la que el alumno elija.
- ✓ De la ventana creada, seleccionar sus propiedades, en la propiedad **preferredSize** definir las dimensiones de la ventana ancho = 600 y alto = 500.
- ✓ De la ventana **Swing Controls** (Controles Swing) selecciona y arrastrar a la ventana **Calculadora** los siguientes componentes:
 - ❖ 3 Etiquetas (Label).
 - ❖ 3 Campos de texto (Text Field)
 - ❖ 5 Botones (Button)
- ✓ Los elementos seleccionados y arrastrados a la ventana **Calculadora** deberán ser ubicados en una forma similar a lo que se observa en la **Figura 1**.

Desarrollo (80 minutos)

Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

1. Diseño de la ventana (20 minutos).
De la ventana **Swing Controls** (Controles Swing), selecciona los componentes (3 etiquetas, 3 campos de texto y 5 botones), los arrastra a la ventana **Calculadora**.
- ✓ Selecciona los componentes etiquetas, empleando su propiedad (**Edit Text** editar texto), modifica su contenido a **numero1**, **numero2** y **resultado**, empleando

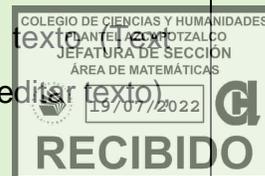


sus propiedades **font** y **foreground** modifica tamaño, tipo de letra y color a la que desees.

- ✓ Selecciona los componentes campos de texto (Text Field), empleando su propiedad (**Edit Text** editar texto), limpia su contenido.
- ✓ Selecciona los componentes campos de texto (Text Field), empleando su propiedad (**Change Variable Name** cambia nombre de la variable) a **num1**, **num2** y **res**.
- ✓ Selecciona los componentes Button (botones), empleando su propiedad (**Edit Text** editar texto), modifica su contenido a (+, -, *, /, **LIMPIAR**), empleando sus propiedades **font** y **foreground** modifica tamaño, tipo de letra y color que desee.
- ✓ Selecciona los componentes Button (botones), empleando su propiedad (**Change Variable Name** cambia nombre de la variable) a **Suma**, **Resta**, **Multiplica**, **Divide**, **Limpiar**.
- ✓ Con el propósito de observar en pantalla el funcionamiento de su ventana **Calculadora**, ejecuta su aplicación, en el caso de ser necesario, realiza los ajustes pertinentes.

2. Programación de los botones Suma, Resta, Multiplicas, Limpiar (40 minutos).

- ✓ Con el propósito de dar vida a la aplicación, el profesor comenta que es necesario llevar a cabo la programación de los botones.



- ❖ Colocado en la pestaña de diseño posicionar el mouse sobre el botón **+**, dar doble click apareciendo el método llamado **SumaActionPerformed** donde se realizan las acciones pertinentes para llevar a cabo la suma de 2 números.

```
private void SumaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void SumaActionPerformed(java.awt.event.ActionEvent evt) {  
    int numero1 = 0, numero2 = 0, suma=0;  
    numero1 = Integer.parseInt(num1.getText());  
    numero2 = Integer.parseInt(num2.getText());  
    suma = numero1 + numero2;  
    String salida = String.valueOf(suma);  
    res.setText(salida);  
}
```

- ❖ Explicación de la programación del botón **Suma**.

- Se definen 3 variables numéricas de tipo entero numero1, numero2 y suma.
- En la variable de tipo entero numero1 se almacenará el contenido del campo de texto **num1** cuyo contenido es una cadena de caracteres, para convertirlo a un número entero se emplea la función **Integer.parseInt** **Integer.parseInt(nombre del campo de texto.getText())**.
- En la variable de tipo entero numero2 se almacenará el contenido del campo de texto **num2** cuyo contenido es una cadena de caracteres, para convertirlo a un número entero se emplea la función **Integer.parseInt** **Integer.parseInt(nombre del campo de texto.getText())**.
- Se realiza la operación correspondiente, el resultado se almacena en la variable **suma=numero1+numero2**.





- Para observar el resultado de la operación **suma** es necesario cargar su contenido en el campo de texto **res**, para lograrlo, se define una variable de tipo **String** (cadena de caracteres) **salida** en la cual se almacenará el contenido de la variable **suma**, para realizarlo debemos emplear la función **String.valueOf** que permite convertir un número entero a una cadena de caracteres.

```
String salida = String.valueOf(suma);
```

- Por último con la ayuda de la propiedad **setText()** se muestra el resultado de la **suma** en el campo de texto **res**.

```
res.setText(salida);
```

- ❖ Colocado en la pestaña de diseño posicionar el mouse sobre el botón **Limpiar**, dar doble click apareciendo el método llamado

```
private void LimpiarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void LimpiarActionPerformed(java.awt.event.ActionEvent evt) {  
    num1.setText("");  
    num2.setText("");  
    res.setText("");  
}
```

- ❖ Explicación de la programación del botón **Limpiar**.
 - ✓ El objetivo del botón **Limpiar** es limpiar el contenido de los campos de texto **num1**, **num2** y **res**, para lograrlo se emplea la propiedad **setText** dentro de los paréntesis se colocan ("") lo que indica que el campo de texto se limpia.

Como parte de la evaluación del alumno, se solicita al estudiante que construya la programación relativa al funcionamiento de los botones **“Resta”** y **“Multiplica”**.

- ❖ El alumno escribe la programación del botón **Resta**

```
private void RestaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

```
private void MultiplicaActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

3. Explicación del funcionamiento del botón **Divide (20 minutos)**.

- ❖ En el caso de la división, el programa debe verificar que el denominador sea diferente de cero, para ello:

- ✓ Verificar que el contenido del campo de texto **num2**, sea diferente de cero.
- ✓ En el caso que el contenido del campo de texto **num2** sea igual a cero, en el campo de texto **resul** deberá aparecer el siguiente mensaje

“División entre cero no permitida, verifica el contenido del campo de texto num2”

- ❖ Bajo la supervisión del profesor el alumno construye el código del botón **Divide**.

```
private void DivideActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

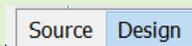


Cierre (20 minutos)

El profesor, en reunión plenaria solicita a dos alumnos del grupo que muestran el funcionamiento de su programa.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma responsable y respetuosa expresa comentarios y aclara dudas, comparan su programa contra el expuesto por sus compañeros.
- ✓ Obtiene conclusiones relativas al manejo de la interfaz gráfica de usuario (JFrame), en relación a:
 - La creación del JFrame que en forma automática genera una clase que hereda los métodos y propiedades del paquete Swing.
 - Observa que al trabajar en el ambiente gráfico se tiene 2 pestañas.



- ❖ La pestaña **Source** es la parte donde se genera el código respectivo de la Clase (programación), y es la parte donde se definen variables y se construyen métodos.
- ❖ La pestaña **Design** es la parte gráfica donde el usuario (alumno) inserta en un lienzo (JFrame) los componentes gráficos que le permite estructurar la solución del problema.
- Comprende como modificar el método constructor en la creación del objeto de la clase.
- Comprende que los botones que inserta en el **JFrame** en la fase de diseño (**Design**), posteriormente, en la pestaña fuente (**Source**), el usuario (alumno) construye los métodos pertinentes que manipulan la información capturada en los campos de texto y proporciona los resultados esperados.



EVALUACIÓN	Lista de cotejo		
	Concepto	Si	No

	El alumno escribe el código pertinente en el método Resta		
	El alumno presenta los resultados esperados al ejecutar el método Resta.		
	El alumno escribe el código pertinente en el método Multiplica		
	El alumno presenta los resultados esperados al ejecutar el método Multiplica.		
	Al dar click al botón LIMPIAR se limpia el contenido de los campos de texto pertinentes.		
	Total		



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama, Ra–Ma.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>Martín, A. (2014). Programador Java certificado: curso práctico.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Arnaw, D. & Weiss, G. (2000). Introducción a la programación con Java. Madrid: Addison–Wesley</p> <p>Villalobos J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, uml, Objetos y Eclipse. México: Pearson Educación.</p>

Estrategia didáctica 2

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 30 de abril 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Propone un proyecto que utilice las Clases: JFrame, JLabel, JButton, JTextField y JComboBox.
TEMA(S)	Clase Swing: <ul style="list-style-type: none"> • JFrame • JLabel • JTextField • JButton • JTextField • JComboBox

III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual
DESARROLLO Y ACTIVIDADES	<p>Inicio (20 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> • Comenta a los alumnos del grupo que empleando la interfaz gráfica de usuario se desea construir proyecto relativo a “Cálculo del área de figuras Geométricas”, para ello muestra lo siguiente. <div data-bbox="516 1104 1349 1507" data-label="Image"> </div> <p style="text-align: center;">Figura 1</p> <ul style="list-style-type: none"> • Comenta a los alumnos del grupo qué para la elaboración del citado proyecto, deberán realizar lo siguiente. <ul style="list-style-type: none"> ✓ Utilizando el paquete NetBeans deberá crear un JFrame (ventana) cuyo nombre será FigurasGeométricas.



✓ De la ventana creada (**JFrame**), seleccionar sus propiedades, en la propiedad **title** colocar en el título de la ventana: **CALCULO DE ÁREAS DE GEOMÉTRICAS**, el tamaño y tipo de letra podrá ser la que el alumno elija.

✓ De la ventana creada, seleccionar sus propiedades, en la propiedad **preferredSize** definir las dimensiones de la ventana ancho = 700 y alto = 700.

✓ De la ventana **Swing Controls** (Controles Swing) selecciona y arrastrar a la ventana **FigurasGeométricas** los siguientes componentes gráficos:

- ❖ 11 Etiquetas (Label).
- ❖ 7 Campos de texto (Text Field)
- ❖ 2 Botones (Button)
- ❖ 1 ComboBox

✓ Los elementos seleccionados y arrastrados a la ventana **FigurasGeométricas** deberán ser ubicados en forma similar a lo que se observa en la **Figura 1**.

- Envío previamente al correo institucional del estudiante las imágenes correspondientes a **cuadrado.png**, **rectángulo.png**, **triangulo.png** y **circulo.png**.
- Comenta a los alumnos que empleando NetBeans creen un paquete llamado “**Imágenes**”, en donde deberán copiar y colocar los archivos: **cuadrado.png**, **rectángulo.png**, **triangulo.png** y **circulo.png**.

Desarrollo (80 minutos)

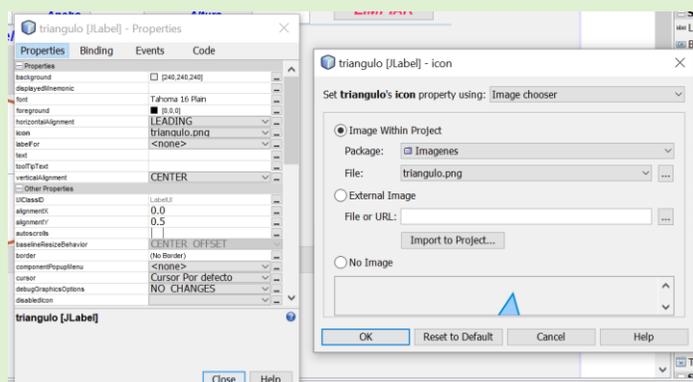
Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

Diseño de la ventana (30 minutos).



De la ventana **Swing Controls** (Controles Swing), selecciona los componentes (11 etiquetas, 7 campos de texto, 2 botones y 1 ComboBox), los arrastra a la ventana.

- ✓ Selecciona los componentes etiquetas que se encuentran en la parte superior de la ventana, empleando su propiedad (**Edit Text** editar texto), modifica su contenido colocando, **lado, radio, largo, ancho, base, altura** y **resultado**, empleando sus propiedades **font** y **foreground** modifica tamaño, tipo de letra y color que desee.
- ✓ Selecciona los componentes etiquetas que se encuentran en la parte inferior de la ventana, empleando su propiedad (**Edit Text** editar texto), modifica su contenido por blancos.
- ✓ Selecciona los componentes etiquetas que se encuentran en la parte inferior de la venta, empleando su propiedad (**Change Variable Name** cambia nombre de la variable) a **cuadrado, rectángulo, triángulo** y **circulo**.
- ✓ Selecciona los componentes etiquetas que se encuentran en la parte inferior de la venta, empleando su propiedad **icon**, selecciona la carpeta **Imágenes**, e inserta en la etiqueta correspondiente la imagen de **cuadrado, rectángulo, triángulo** y **circulo**.



- ✓ Selecciona los componentes campos de texto (Text Field), empleando su propiedad (**Edit Text** editar texto) limpia su contenido.
- ✓ Selecciona los componentes campos de texto (Text Field), empleando su propiedad (**Change Variable Name** cambia nombre de la variable) a **lado**, **radio**, **largo**, **ancho**, **base**, **altura** y **resultado**.
- ✓ Selecciona el componente **ComboBox**, empleando su propiedad **model** inserta el contenido del Combo.



- ✓ Selecciona la propiedad **font** y **foreground** y elige el tipo, tamaño de letra y el color que desee.
- ✓ Selecciona los componentes Button (botones), empleando su propiedad (**Edit Text** editar texto), modifica su contenido colocando (**CALCULAR**, **LIMPIAR**), empleando sus propiedades **font** y **foreground** modifica tamaño, tipo de letra y color que desee.
- ✓ Selecciona los componentes Button (botones), empleando su propiedad (**Change Variable Name** cambia nombre de la variable) a **calcular**, **limpiar**.
- ✓ Con el propósito de observar en pantalla el funcionamiento de la ventana **FigurasGeométricas**,



ejecuta su aplicación, en el caso de ser necesario, realiza los ajustes pertinentes.

Programación de los métodos: constructor, calcular y limpiar. (50 minutos).



- ✓ Con el propósito de complementar el programa, el profesor indica que es necesario llevar a cabo lo siguiente:

Modificación del método constructor

```
public FigurasGeometricas() { //método constructor
    initComponents();
    setLocationRelativeTo(null); //centra la ventana
    this.getContentPane().setBackground(Color.ORANGE); //elige color del fondo de la ventana
    rectangulo.setVisible(false); //en tiempo de ejecución oculta el contenido de las etiquetas
    cuadrado.setVisible(false); //rectangulo, cuadrado, triangulo, circulo.
    triangulo.setVisible(false);
    circulo.setVisible(false);
}
```

Comenta que en el método constructor se debe de colocar el nombre de la etiqueta **rectangulo**, **cuadrado**, **triangulo** y **circulo** con su propiedad **setVisible(false)** que provoca que al momento de construir el objeto las imágenes de las figuras geométricas se oculten.

Programación de los botones calcular y limpiar.

- ✓ Colocado en la pestaña de diseño posicionar el mouse sobre el botón **Calcular** dar doble click apareciendo el método **calcularActionPerformed**, en donde se realizan las acciones pertinentes para llevar a cabo el cálculo del área de la figura correspondiente.

```

private void calcularActionPerformed(java.awt.event.ActionEvent evt) {
    String figgeo = figura.getSelectedItem().toString(); //recupera el termino

    int posicion = figura.getSelectedIndex(); // recupero el indice del elemento seleccionado del
    //if (figgeo.equals ("Cuadrado"))
    //{
    // cuadrado.setVisible(true);
    //}
    int areafge = 0;
    float areafgd = 0;
    String salida;
    JOptionPane.showMessageDialog(null, "El valor de posicion = " + posicion);
    imagen.setIcon(new ImageIcon("C:/Documentos/NetBeansProjects/CiberneticaII/src/Imagenes/triangulo.png"));
    switch (posicion)
    {
        case 1:
            int ladoc = 0;
            ladoc = Integer.parseInt(lado.getText());
            areafge = ladoc * ladoc;
            salida = String.valueOf(areafge);
            resultado.setText(salida);
            // cuadrado.setVisible(true);
            imagen.setIcon(new ImageIcon("C:/Documentos/NetBeansProjects/CiberneticaII/src/Imagenes/triangulo.png"));
            break;
        case 2:
            int largor=0, anchor=0;
            largor = Integer.parseInt(largo.getText());
            anchor = Integer.parseInt(ancho.getText());
            areafge = largor * anchor;
            salida = String.valueOf(areafge);
            resultado.setText(salida);
            rectangulo.setVisible(true);
            break;
    }
}

```



- ✓ En la variable de tipo entero posición = `figura.getSelectedIndex()`; se recupera el índice del elemento seleccionado del **Combo**.
- Se emplea la instrucción `switch (posición)` para analizar el contenido de la variable posición.
- **case 1**: desarrolla el procedimiento que permite calcular el área de la figura geométrica **Cuadrado**, su resultado es almacenado en la variable **areafge**, posteriormente, se almacena en el campo de texto **resultado** y se muestra en pantalla, también se muestra la imagen de la figura geométrica **cuadrado**.
- **case 2**: desarrolla el procedimiento que permite calcular el área de la figura geométrica **Rectangulo**, su resultado es almacenado en la variable **areafge**,

posteriormente, se almacena en el campo de texto

resultado y se muestra en pantalla,

muestra la imagen de la figura

rectangulo.



- ✓ Como parte de la evaluación del estudiante, el alumno continúa con la programación del método **calcular** relativa al cálculo de las figuras geométricas **triángulo y círculo.**
- ✓ Por último, el estudiante construye el código del botón **LIMPIAR**, cuyo objetivo es el limpiar todos los campos de texto, así como ocultar las etiquetas correspondientes a las imágenes de las figuras geométricas.

Cierre (20 minutos)

El profesor en reunión plenaria elige a dos alumnos del grupo los cuales muestran el funcionamiento de su programa.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma respetuosa y responsable expresa comentarios y aclara dudas, compara su programa contra el expuesto por sus compañeros.
- ✓ Obtiene conclusiones relativas al manejo de la interfaz gráfica de usuario (JFrame), tales como:
 - Manejo del componente gráfico ComboBox
 - Programación relativa al componente gráfico ComboBox.
- ✓ Comprende que en la programación del botón **Calcular** debe escribir las acciones pertinentes que le permitan calcular el área de las diversas figuras geométricas, para ello debe hacer uso de una variable de tipo número entero en la cual

se almacena el "Item" seleccionado por el usuario del componente gráfico ComboBox.



EVALUACIÓN	Lista de cotejo		
	Concepto	Si	No
	El alumno escribe el código pertinente en el método calcular para obtener el área de la figura geométrica triángulo.		
	Muestra en pantalla el resultado correcto del área del triángulo.		
	Al momento de mostrar en la pantalla el resultado del área de la figura geométrica triángulo también se muestra la imagen correspondiente.		
	Al momento de dar click al botón LIMPIAR se limpia el contenido de los campos de texto pertinentes y se ocultan las imágenes respectivas.		
	El alumno escribe el código pertinente en el método calcular para obtener el área de la figura geométrica círculo.		

	Muestra en pantalla el resultado correcto del área del círculo		
	Al momento de mostrar en la pantalla el resultado del área de la figura geométrica círculo también se muestra la imagen correspondiente.		
	Total		



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama, Ra–Ma.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>Martín, A. (2014). Programador Java certificado: curso práctico.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Arnold, D. & Weiss, G. (2000). Introducción a la programación con Java. Madrid: Addison–Wesley</p> <p>Villalobos J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, uml, Objetos y Eclipse. México: Pearson Educación.</p>

V. ANEXOS TRABAJO DESARROLLADO POR LOS ALUMNOS

Estrategia didáctica 3

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 4 de abril 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Propone un proyecto que utilice las Clases: JFrame, JLabel, JButton, JTextField , JComboBox, JText Area.
TEMA(S)	Clase Swing: <ul style="list-style-type: none"> • JFrame • JLabel • JTextField • JButton • JTextField • JComboBox

	<ul style="list-style-type: none"> • JText Area • JPanel
--	--



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual

	<p>Inicio (25 minutos)</p> <p>El profesor:</p> <p>Comenta a los alumnos qué empleando el ambiente gráfico de Java, se desea elaborar un programa que realice lo siguiente.</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="color: red; font-weight: bold; margin: 0;">EJEMPLOS VARIOS</p> <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> <p style="font-size: small; margin: 0;">Elige Opción</p> <p style="font-size: small; margin: 0;">Elige Opción</p> <p style="margin: 0;">Tabla de Multiplicar</p> <p style="margin: 0;">Factorial de un Número</p> <p style="margin: 0;">Sumatoria del 1 hasta un número</p> </div> </div> <p>Ejemplo.</p> <p style="color: red; font-weight: bold; margin: 0;">Tabla de multiplicar</p> <p>Número: 7</p> <p>7 x 1 = 7</p> <p>7 x 2 = 14</p> <p>7 X 3 = 21</p>
--	---



$$7 \times 4 = 28$$

$$7 \times 5 = 35$$

$$7 \times 6 = 42$$

$$7 \times 7 = 49$$

$$7 \times 8 = 56$$

$$7 \times 9 = 63$$

$$7 \times 10 = 70$$

Factorial de un número

Número: 6

$$1 \times 1 = 1$$

$$1 \times 2 = 2$$

$$2 \times 3 = 6$$

$$6 \times 4 = 24$$

$$24 \times 5 = 120$$

$$120 \times 6 = 720$$

El factorial del número $6! = 720$

Sumatoria del 1 hasta el número Número: 7

$$0 + 1 = 1$$

$$1 + 2 = 3$$

$$3 + 3 = 6$$

$$6 + 4 = 10$$

$$10 + 5 = 15$$

$$15 + 6 = 21$$

$$21 + 7 = 28$$

La sumatoria del número 1 al 7 = 28

El profesor muestra a los alumnos del grupo lo siguiente.

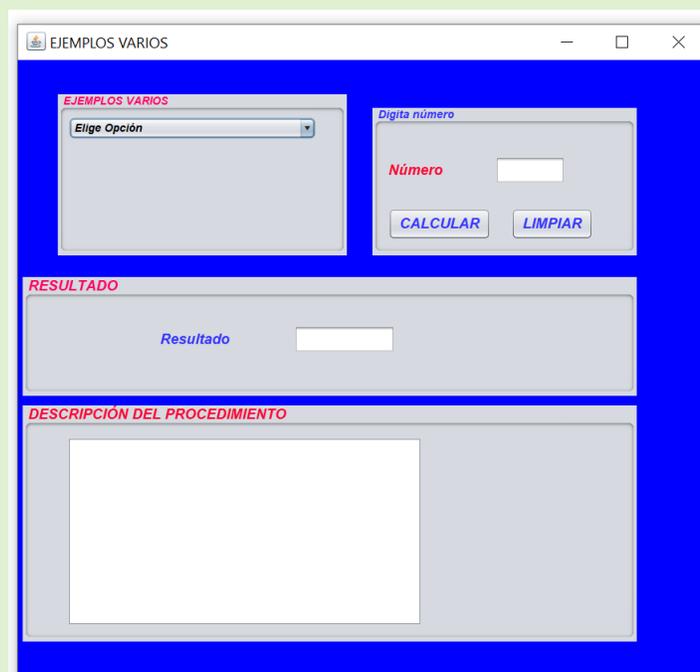
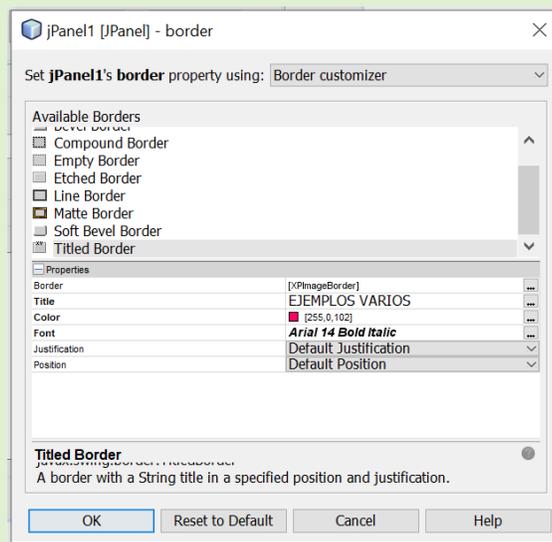
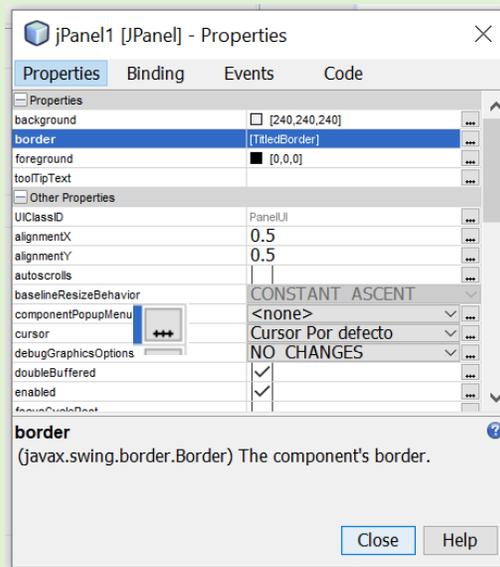


Figura 1

Comenta a los alumnos del grupo que, en la elaboración de este proyecto, adicional al elemento gráfico **Text Area** (área de texto), se empleará un nuevo elemento gráfico el contenedor **Panel**, componente que permitirá realizar agrupaciones de componentes gráficos el cual insertaremos en nuestra ventana (JFrame). Cometa que para la elaboración del citado proyecto, deberán realizar lo siguiente.

- Crear un JFrame (ventana) llamada **EjemplosVarios**.
- En la ventana creada, seleccionar sus propiedades, en la propiedad **title** colocar en el título de la ventana: **Ejemplos Varios**, el tamaño y tipo de letra podrá ser la que el alumno elija.

- En la ventana creada, seleccionar sus propiedades, en la propiedad **preferredSize** definir las dimensiones de la ventana ancho = 700 y alto = 700.
- De **Swing Containers** (Contenedores Swing) seleccionar **Panel**, arrastrarlos a la ventana de acuerdo a lo que se muestra en la **Figura 1**.
- Para colocar el nombre a un Panel, deberás seleccionar el Panel, dar clic derecho, y seleccionar la propiedad **border**.



A la derecha de la propiedad **Title**, colocar el nombre del título del Panel (EJEMPLOS VARIOS), posteriormente, empleando las propiedades **Color** y **Font** seleccionar el tipo de letra y el color que desees.



A continuación, arrastra a la ventana **EjemplosVarios** los siguientes componentes gráficos:

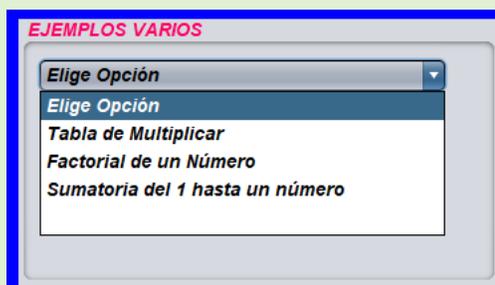
- 2 Etiquetas (Label).
- 2 Campos de texto (Text Field)
- 2 Botones (Button)
- 1 ComboBox
- 1 Área de texto

Los elementos gráficos arrastrados los deberás colocar en el Panel correspondiente de acuerdo como se muestra en la Figura 1.

Desarrollo (80 minutos)

Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

- Diseño de la ventana (30 minutos).
 - ✓ Modifica el contenido del componente gráfico **ComboBox** insertado en el **Panel Ejemplos Varios**, empleando su propiedad **model** coloca.



- ✓ Del **Panel Digita número** selecciona el componente etiqueta empleando su propiedad (**Edit Text** editar texto), modifica su contenido colocando, **Número**, también selecciona el componente etiqueta que se encuentra situado en el **Panel Resultado**, modifica su contenido por **Resultado**, empleando sus propiedades **font** y

foreground modifica tamaño, tipo de letra y color a la que desees.

✓ Selecciona los componentes campos de texto (**Text Field**) que se encuentran situados en los Panel **desarrollar** anteriormente, empleando su propiedad (**Edit Text** editar texto), limpia su contenido, posteriormente, empleando su propiedad (**Change Variable Name** cambia nombre de la variable) a **numero** y **resultado**, empleando las propiedades **font** y **foreground** elige el tipo, tamaño de letra y el color que desee.

✓ Del **Panel Digita número** selecciona los componentes Button (botones), empleando su propiedad (**Edit Text** editar texto), modifica su contenido colocando (**CALCULAR**, **LIMPIAR**), empleando sus propiedades **font** y **foreground** modifica tamaño, tipo de letra y color que desees, empleando su propiedad (**Change Variable Name**) cambia su contenido a **calcular** y **limpiar**.

✓ Del **Panel DESCRIPCIÓN DEL PROCEDIMIENTO** selecciona el componente Text Area (área de texto) empleando su propiedad (**Edit Text** editar texto), modifica su contenido por espacios, empleando su propiedad (**change variable name**) modifica su contenido por **areat**.

✓ Con el propósito de observar en pantalla el funcionamiento de la ventana **EjemplosVarios**, ejecútala, en el caso de ser necesario, realizar los ajustes pertinentes.

- **Programación de los métodos: constructor, calcular y limpiar.** (50 minutos).

Programación de los botones calcular y limpiar.

✓ Colocado en la pestaña de diseño posicionar el mouse sobre el botón **CALCULAR** dar doble click apareciendo el

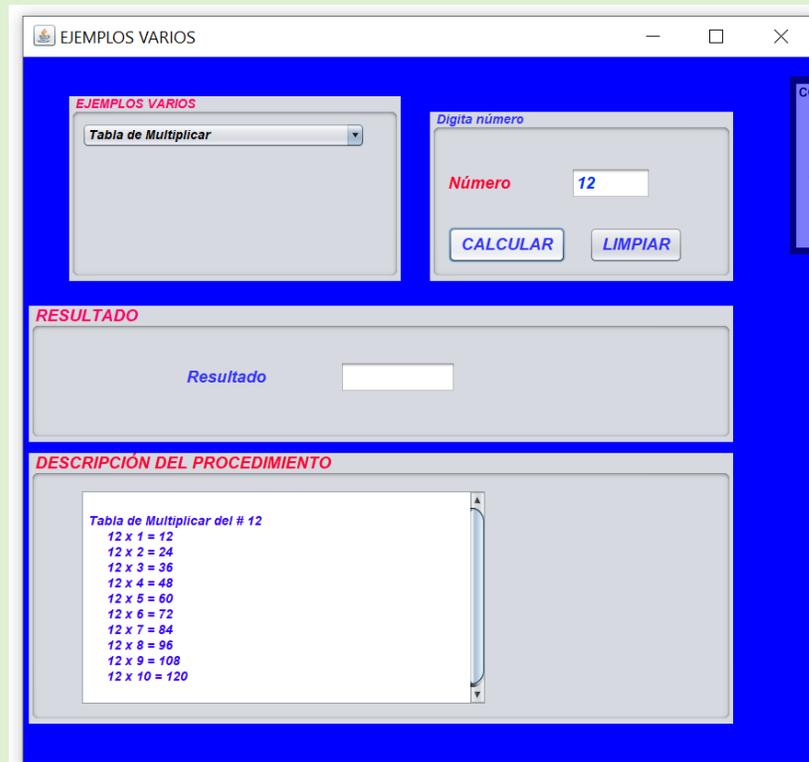


método **calcularActionPerformed**, en donde se realizan las acciones siguientes.

```
private void
calcularActionPerformed(java.awt.event.ActionEvent evt)
    int posicion = opcion.getSelectedIndex();
//recupera el índice del combo
    int indice=0, tabla=0, resul=0, nume=0;
    String salida;
    switch (posicion)
    {
        case 1: //procedimiento para la
tabla de multiplicar
            areat.append(" " + "\n");
//escribiendo en el area de texto      append escribe 1
línea en el área de texto  "\n" salto de línea
            nume =
Integer.parseInt(numero.getText()); //Integer.parseInt
función que convierte una cadena de caracteres a un número
entero
            areat.append("Tabla de
Multiplicar del # " + nume + "\n");
            for (indice = 1 ; indice <= 10 ;
indice++)
            {
                resul = nume * indice;
                areat.append("      " + nume
+ " x " + indice + " = " + resul + "\n");
//escribiendo en el área de texto
            }
            break;
```

El **profesor** comenta que el elemento gráfico **Text Area** (área de texto), es un componente gráfico que permite escribir líneas (cadenas de caracteres) en forma indefinida, que para insertar una línea en el **área de texto** debemos emplear la propiedad **append**, el carácter **"\n"** provoca un salto de línea. Para observar el funcionamiento de la aplicación, se ejecuta, del Componente **ComboBox** se elige **Tabla de Multiplicar**, en el campo de texto **Número** se digita 12 y el resultado se observa en el panel **DESCRIPCIÓN DEL PROCEDIMIENTO**.





- El profesor solicita al estudiante que continúe desarrollando el método **calcular** relativo a el cálculo del **factorial de un número** y **la sumatoria del número 1 hasta un número proporcionado**.
- ✓ Como parte de la evaluación del estudiante, el alumno continúa con la programación del método **calcular**, relativo al **Factorial de un número** y la **Sumatoria del 1 hasta un número proporcionado**, así como el método **limpiar**.

Cierre (20 minutos)

El profesor en reunión plenaria elige a dos alumnos los cuales muestran el funcionamiento de su programa.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma responsable, respetuosa y ordenada expresa comentarios y

aclara dudas, comparan su programa contra el expuesto por sus compañeros.

✓ Obtiene conclusiones relativas al manejo de gráfica de usuario (JFrame), en torno a los gráficos.

- Manejo del componente gráfico TextArea (área de texto).
- Programación relativa al componente gráfico TextArea.
- Programación relativa al componente gráfico JComboBox.
- Comprende que el componente gráfico permite agrupar otros componentes gráficos el cual se debe de insertar dentro del cuerpo del JFrame.



EVALUACIÓN

Lista de cotejo

Concepto	Si	No
En el método calcular el alumno escribe el código pertinente que le permite calcular el Factorial de un número.		
En el cálculo del Factorial de un número, el alumno emplea de manera adecuada las estructuras de control.		
En la pantalla de la computadora en el “área de texto” se observa la impresión del factorial en forma desglosada.		
En el método calcular el alumno escribe el código pertinente que le permite obtener la Sumatoria del número 1 hasta un el número proporcionado.		
En el cálculo de la Sumatoria del número 1 hasta un número proporcionado, el alumno emplea de manera adecuada las estructuras de control.		

	En la pantalla de la computadora en el “área de texto”, se observa la impresión del número 1 hasta un número proporcionado, en forma desglosada.		
	El código escrito en el método limpiar borra el contenido de los campos de texto.		
	El código escrito en el método limpiar borra el contenido del área de texto.		
	Total		



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama, Ra–Ma.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>Martín, A. (2014). Programador Java certificado: curso práctico.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Arnold, D. & Weiss, G. (2000). Introducción a la programación con Java. Madrid: Addison–Wesley</p> <p>Villalobos J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, uml, Objetos y Eclipse. México: Pearson Educación.</p>

V. ANEXOS

Estrategia didáctica 4

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional	
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II	
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre	
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades	
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 6 de abril 2022	
NO DE ALUMNOS	Número de alumnos: 25	

II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Propone un proyecto que utilice las Clases: JFrame, JLabel, JButton, JTextField , JComboBox, JTable, JRadio Button, JButton Group, JPanel
TEMA(S)	Clase Swing: <ul style="list-style-type: none"> • JFrame • JLabel • JTextField • JButton • JTable • JComboBox

	<ul style="list-style-type: none"> • JRadio Button • JButton Group • JTable • JPanel
--	--



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual
	<p>Inicio (25 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> ✓ Comenta a los alumnos del grupo que empleando la interfaz gráfica de usuario se desea construir proyecto relativo a “Depósito Bancario”, para ello muestra lo siguiente.

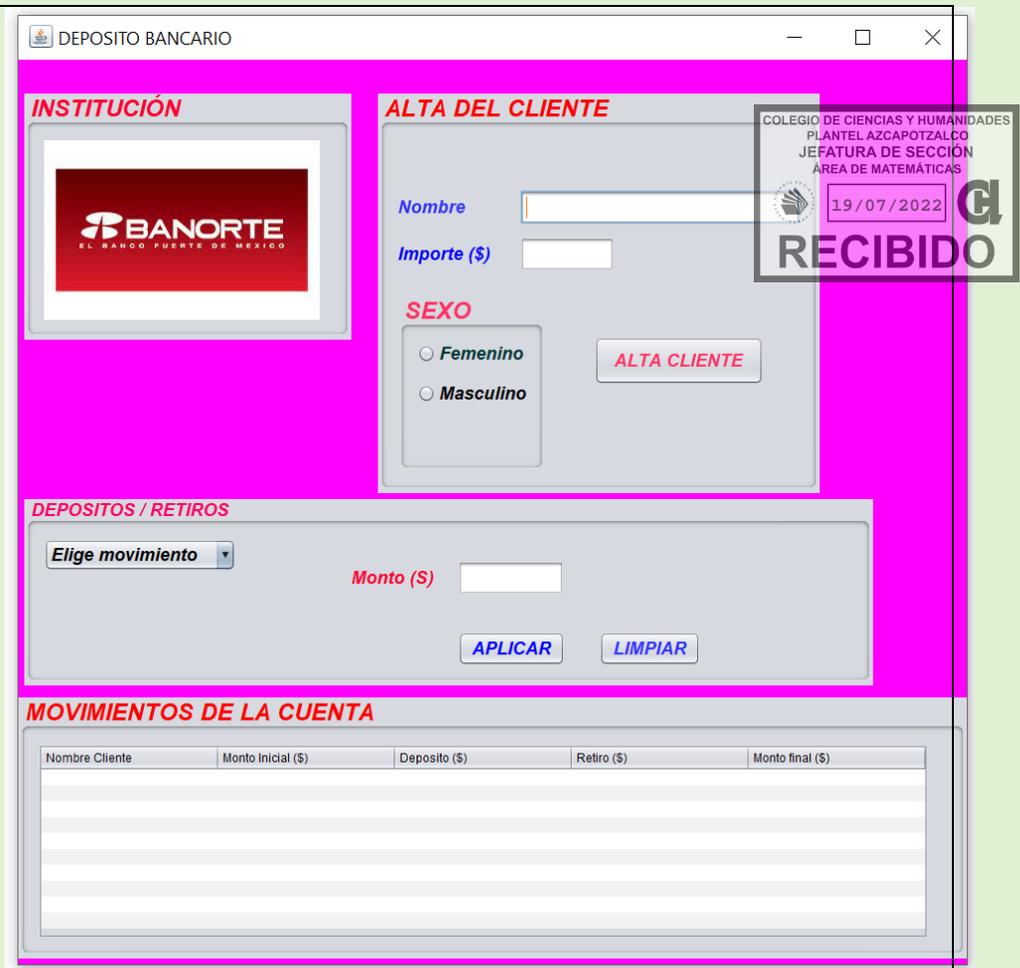


Figura 1

- ✓ Comenta a los alumnos del grupo que en el desarrollo de este proyecto (**DEPÓSITO BANCARIO**); adicional a los elementos gráficos ya manejados, se emplearán nuevos elementos tales como: **JRadio Button** (botón de radio), **JButton Group** (botón de grupo) y **JTable** (tabla).

Indica que los botones de radio son elementos gráficos que le permiten al usuario elegir entre un conjunto de opciones, que para trabajar adecuadamente con estos botones de radio se deben de asociar a un botón de grupo. Comenta también, que se empleará un nuevo elemento gráfico el **JTable (tabla)**, cuyo funcionamiento es similar a un arreglo bidimensional, con la ventaja de que en el componente gráfico **JTable** a cada una

de las columnas de la **tabla** se le puede asociar un tipo de dato diferente: String (cadena de caracteres), Integer (número entero), Float o Double (número decimal), etc.



A continuación, el profesor comenta a los alumnos del grupo el funcionamiento de la ventana:

- ✓ **Panel INSTITUCIÓN** contiene un elemento gráfico JLabel (etiqueta), dentro de la cual se debe insertar una imagen relativa a la institución bancaria de tu preferencia.
- ✓ **Panel ALTA DEL CLIENTE** contiene 2 elementos gráficos JLabel (etiquetas) **Nombre** e **Importe (\$)**, así como sus respectivos Text Field (campos de texto) en donde el usuario deberá de capturar el nombre del cliente y el importe con el que el cliente causa Alta en la base de datos de la institución bancaria. Observa que, dentro del **Panel ALTA DEL CLIENTE**, se inserta otro **Panel SEXO** dentro del cual se colocan 2 Radio Button cuyo contenido es **Femenino** y **Masculino**. Dentro del **Panel ALTA DEL CLIENTE**, se sitúa un botón “**ALTA DEL CLIENTE**” cuyo propósito es dar de ALTA al cliente en la base de datos de la institución bancaria, situación que se visualiza en el elemento gráfico **tabla**.

ALTA DEL CLIENTE

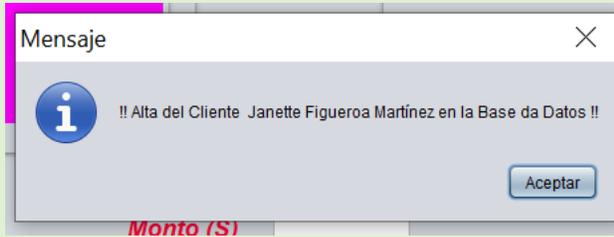
Nombre

Importe (\$)

SEXO

Femenino

Masculino



MOVIMIENTOS DE LA CUENTA

Nombre Cliente	Monto Inicial (\$)	Deposito (\$)	Retiro (\$)	Monto final (\$)
Janette Figueroa Martí...		45000		45000

- ✓ **Panel DEPOSITOS / RETIROS** contiene 1 elemento gráfico Jlabel (etiqueta) **Monto (\$)**, 1 cuadro de texto, 1 Combo Box y 2 botones **APLICAR** y **LIMPIAR**. El contenido del Combo Box es el siguiente.



Depósito

Al elegir del Combo Box la opción **Deposito**, es necesario capturar el importe correspondiente a **Monto (\$)**, observa.

DEPOSITOS / RETIROS

Deposito

Monto (\$) 5000

APLICAR LIMPIAR

Al dar click en el botón **APLICAR**

el movimiento se refleja en el componente gráfico. **Depósito** se observa en la columna correspondiente, también se actualiza el importe del Monto Final en la columna respectiva.



DEPOSITOS / RETIROS

Deposito Monto (\$) **5000**

APLICAR **LIMPIAR**

MOVIMIENTOS DE LA CUENTA

Nombre Cliente	Monto Inicial (\$)	Deposito (\$)	Retiro (\$)	Monto final (\$)
Jeanette Figueroa Martinez	45000		5000	45000
				50000

Nombre Cliente | Monto Inicial (\$) | Deposito (\$) | Retiro (\$) | Monto final (\$)

Retiro

Al elegir del Combo Box la opción **Retiro**, es necesario capturar el importe correspondiente a **Monto (\$)**, observa.

DEPOSITOS / RETIROS

Retiro Monto (\$) **6000**

APLICAR **LIMPIAR**

Al dar click en botón **APLICAR**

el movimiento se refleja en el componente gráfico jTable, el importe del **Retiro** se refleja en la columna correspondiente, también se actualiza la columna de Monto Final.



DEPOSITOS / RETIROS

Retiro Monto (\$)

MOVIMIENTOS DE LA CUENTA

Nombre Cliente	Monto Inicial (\$)	Deposito (\$)	Retiro (\$)	Monto final (\$)
Jeanette Figueroa Martínez	45000		5000	45000
			6000	44000

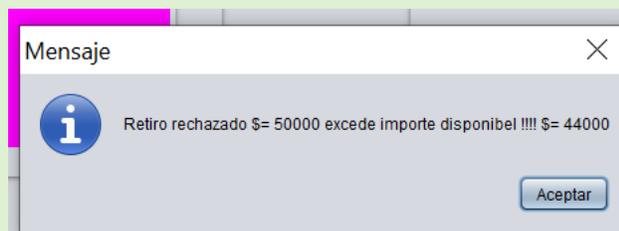
Retiro improcedente

En el caso de que el usuario solicite el **Retiro** de un importe mayor al disponible que se tenga en la cuenta, se tendrá lo siguiente, observa.

DEPOSITOS / RETIROS

Retiro Monto (\$)

Al dar click en botón



Total

Al elegir del Combo Box la opción **Total** y dar clic en botón

APLICAR



en el componente gráfico JTable se muestra un resumen de los movimientos realizados, observa lo siguiente.

DEPOSITOS / RETIROS

Total

MOVIMIENTOS DE LA CUENTA

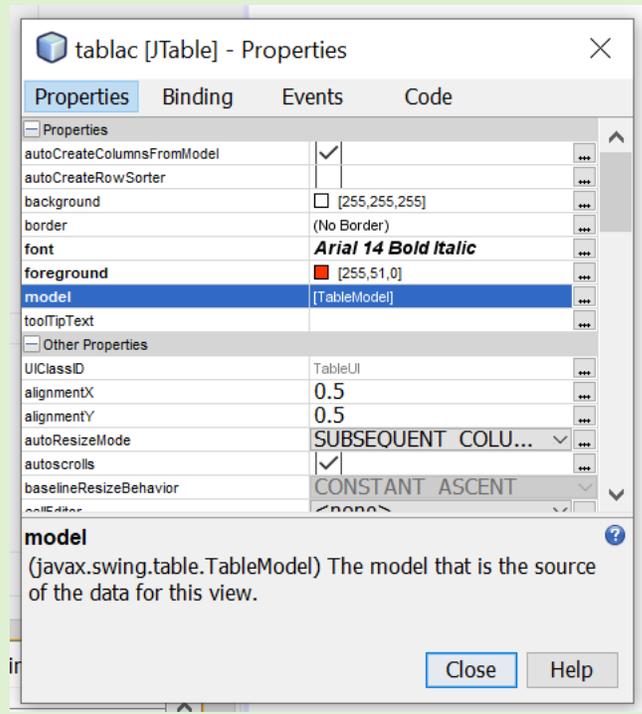
Nombre Cliente	Monto Inicial (\$)	Deposito (\$)	Retiro (\$)	Monto final (\$)
Jeanette Figueroa Martinez	45000		5000	45000
			6000	44000
RESUMEN		5000	6000	44000

Desarrollo (80 minutos)

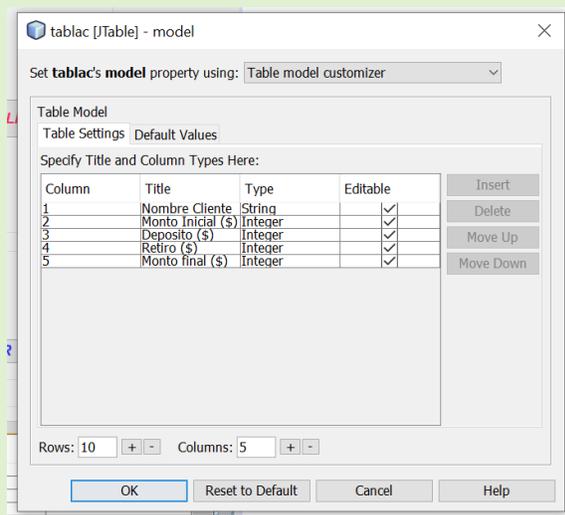
Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

- ✓ **Diseño de la ventana** (30 minutos).
- ✓ Crear una ventana (JFrame) con el nombre de DepositoBancario.
- ✓ Colocar en los paneles correspondientes los elementos gráficos de acuerdo a los estipulado (Figura 1).
- ✓ El profesor explica el funcionamiento de los botones de radio, indicando que es necesario asociarlos a un botón de grupo (Buttom Group) y muestra la forma de realizarlo.

- ✓ El profesor explica el funcionamiento del componente gráfico `Jtable`, comentando que es necesario modificar la propiedad **model** (modelo de la tabla) de acuerdo a las características de nuestro problema y muestra lo siguiente.



Al dar click en , aparece.



Modificando el contenido de las columnas título (**Title**) y **tipo (Type)** de acuerdo a las necesidades de nuestro problema.



- ✓ El alumno continúa trabajando en la construcción de la ventana, colocando los elementos gráficos en los paneles correspondientes.

Programación de los métodos: constructor, ALTA CLIENTE, APLICAR y LIMPIAR. (50 minutos).

- ✓ El profesor muestra el código del botón **ALTA CLIENTE** en donde explica la forma de trabajar del componente gráfico JTable.

```
private void altaclActionPerformed(java.awt.event.ActionEvent evt) {  
    String dnombre= nombre.getText(); //get recupero el contenido del campo de texto  
    importe = Integer.parseInt(importe.getText());  
    tablac.setValueAt(dnombre, fila,0); //set cambia el contenido del campo  
    tablac.setValueAt(importe, fila, 1);  
    tablac.setValueAt(importe, fila, 4);  
    altacl.setVisible(false);  
    JOptionPane.showMessageDialog(null, "!! Alta del Cliente " + dnombre + " en la Base da Datos !!");  
}
```

Cierre (15 minutos)

El profesor en reunión plenaria elige a dos alumnos los cuales muestran el funcionamiento de su programa.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma responsable, ordenada y respetuosa expresa comentarios y aclara dudas, comparan su programa contra el expuesto por sus compañeros.
- ✓ Obtiene conclusiones relativas al manejo de la interfaz gráfica de usuario (JFrame), tales como:
 - Manejo de botones de radio y botones de grupo.

	<ul style="list-style-type: none"> ▪ Programación de botones de radio y botones de grupo. ▪ Creación y manejo del componente gráfico JTable (tabla). ▪ Programación inherente al componente gráfico JTable (tabla). ▪ Observa el beneficio de manejar Jtable (tablas) en lugar de arreglos bidimensionales.
--	---



EVALUACIÓN	Lista de cotejo		
	Concepto	Si	No
	Al elegir del elemento gráfico ComboBox el Item Depósito, en el método aplicar el alumno escribe las acciones pertinentes que permiten llevar a cabo el movimiento seleccionado.		
	El Depósito efectuado se refleja en el elemento gráfico JTable en la columna correspondiente.		
	Al elegir del elemento gráfico ComboBox el Item Retiro, en el método APLICAR el alumno escribe las acciones pertinentes que permiten llevar a cabo el movimiento seleccionado.		
	El Retiro efectuado se refleja en el elemento gráfico JTable en la columna correspondiente.		
	Al elegir del elemento gráfico ComboBox el Item Retiro, en el método APLICAR el alumno escribe las acciones pertinentes que permiten rechazar el importe de un retiro mayor al dinero disponible en la cuenta.		
	Al elegir del elemento gráfico ComboBox el Item Total, en el método APLICAR el alumno escribe las acciones pertinentes que permiten llevar a cabo un desglose del		

	número e importe de depósitos y retiros realizados.		
	El Total de movimientos y dinero efectuados relativos a depósitos y retiros se refleja de manera correcta en el elemento gráfico JTable en las columnas correspondientes.		
	El manejo del componente gráfico JTable es adecuado.		
	Al dar click en el botón Limpiar se borra el contenido de los campos de texto.		
	Total		



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama, Ra–Ma.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>Martín, A. (2014). Programador Java certificado: curso práctico.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Arnou, D. & Weiss, G. (2000). Introducción a la programación con Java. Madrid: Addison–Wesley</p> <p>Villalobos J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, uml, Objetos y Eclipse. México: Pearson Educación.</p>

V. ANEXOS

Estrategia didáctica 5

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 20 de abril 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Propone un proyecto que utilice las Clases: JFrame, JLabel, JButton, JTextField , JComboBox, JTable, JRadio Button , JButton Group, JPanel, JMenu Bar y JMenuItem
TEMA(S)	Clase Swing: <ul style="list-style-type: none"> • JFrame • JLabel • JTextField • JButton • JTextField • JComboBox

	<ul style="list-style-type: none"> • JRadio Button • JButton Group • JTable • Pane • JMenu Bar • JMenu Item
--	---



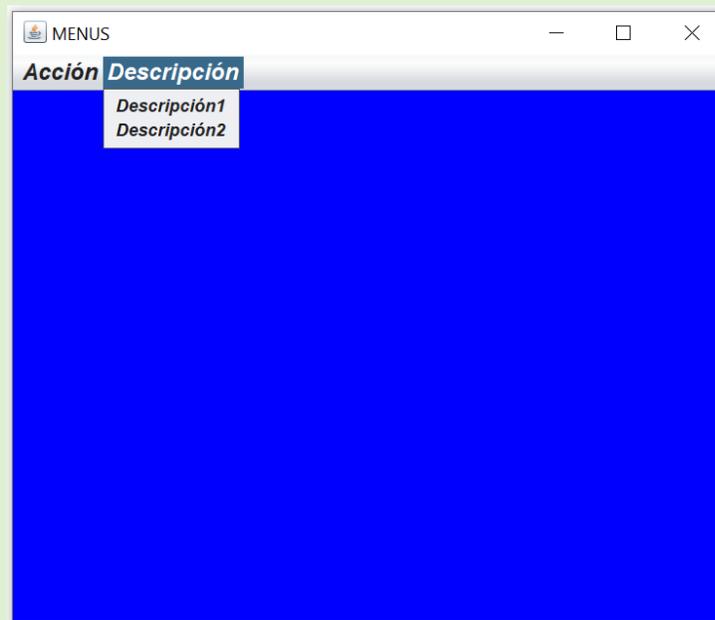
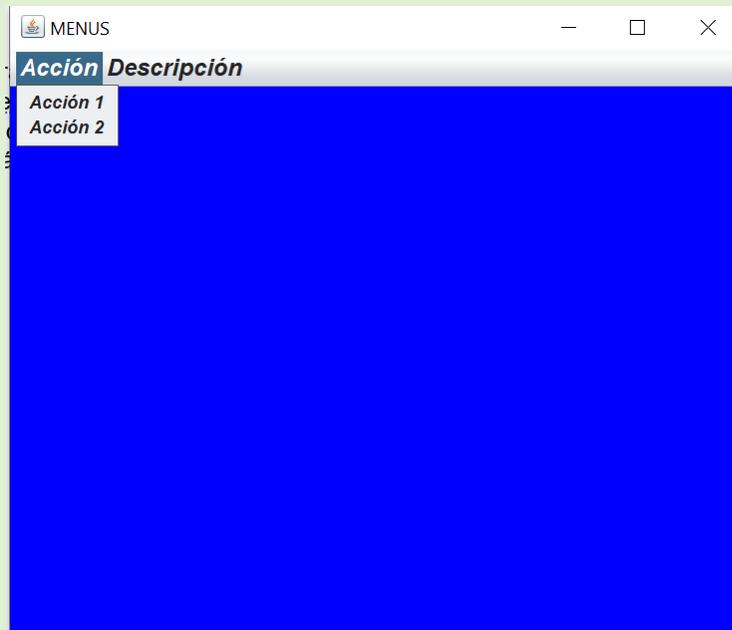
III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual

Inicio (30 minutos)

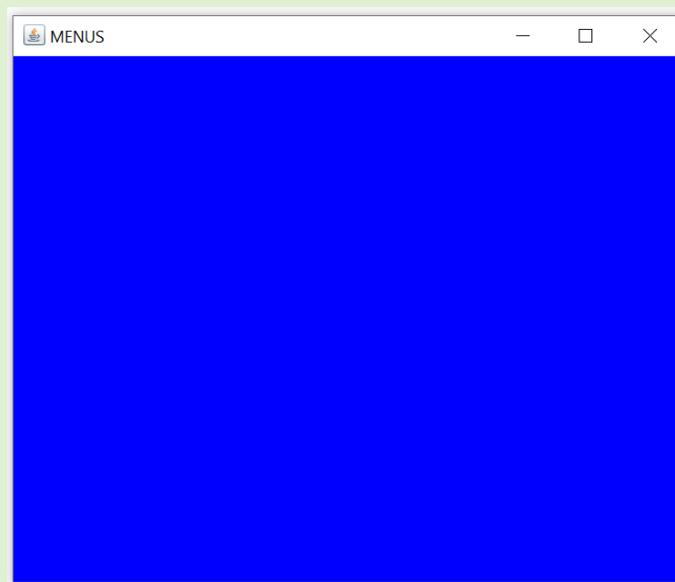
El profesor:

- Comenta a los alumnos del grupo que empleando la interfaz gráfica de usuario y empleando los componentes **JMenu Bar** y **JMenu Item**, se desea construir proyecto relativo a “**Crear un Menu**”, para ello muestra lo siguiente.



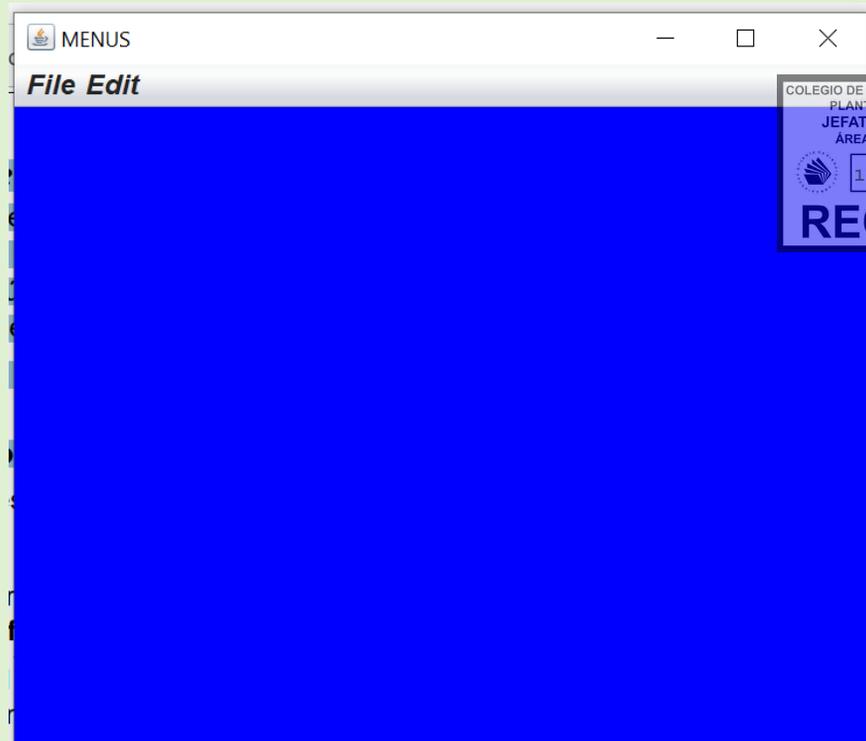
- Comenta a los alumnos del grupo que en el desarrollo de este proyecto (**CREACIÓN MENÚS**); se manejarán los elementos gráficos **JMenu Bar** (barras de menú) y **JMenuItem** (Submenús), indica que la barra de menús y submenús son componentes gráficos que le permiten al usuario navegar entre diferentes ventanas o JFrames, a continuación, el profesor solicita a los alumnos del grupo que realicen lo siguiente.

- ✓ Empleando Netbeans crear un **JFrame** llamado **Menu**.
- ✓ Empleando las propiedades del **JFrame**, colocar en la propiedad **title** de la ventana la palabra **MENUS**.
- ✓ Modificar el método constructor de la Clase **Menu**, colocando las acciones pertinentes que permitan centrar la ventana, y el color del fondo de tu elección, observa.



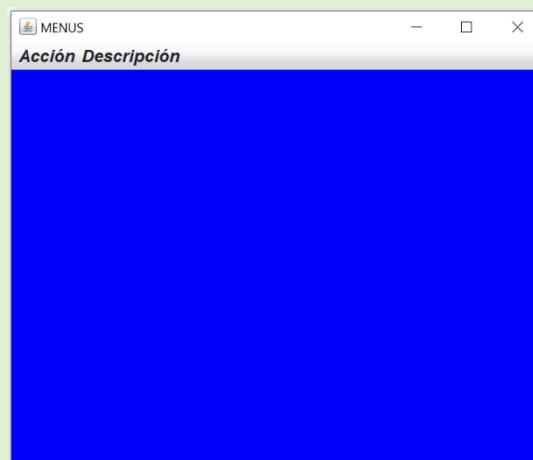
- ✓ Del panel de componentes, ubicar la categoría **Swing Menus** y arrastrar al **JFrame** el componente **Menu Bar**, el **JFrame** deberá verse como sigue.



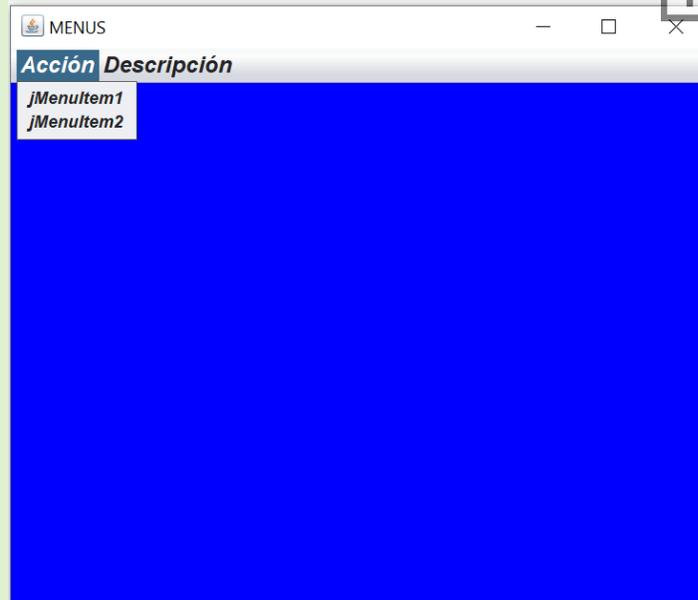


observa que por default se colocan 2 menús con nombres de **File** y **Edit**.

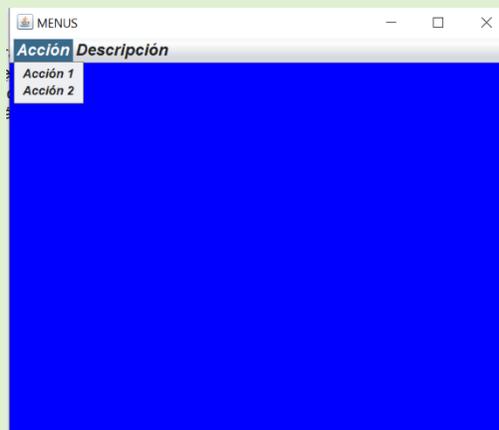
- ✓ Modificar el contenido de **File** y **Edit**; haciendo uso de la propiedad **Edit Text**, **File** lo modificamos por **Acción**, y **Edit** lo modificamos por **Descripción**, el **JFrame** deberá de verse como sigue.



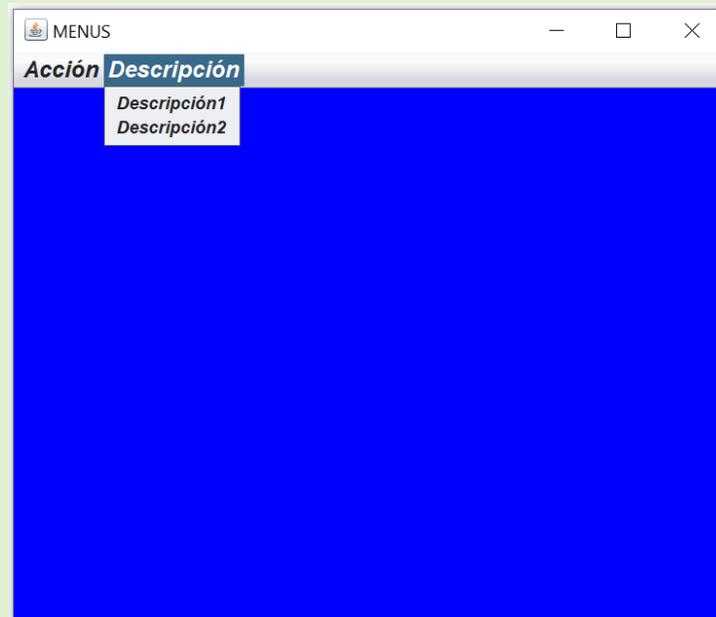
- ✓ A continuación, se insertarán 2 **Submenús** en los Menús **Acción** y **Descripción**, del panel **Swing Components** ubicar el componente **Menu Item**, arrastrarlo y colocarlo en el **Menú Acción**, el **Jframe** deberá de verse como sigue.



- ✓ Modificar el contenido de la palabra **JMenuItem1** y **JMenuItem2**, con la ayuda de la propiedad **Edit Text**, cambia su contenido por **Acción1** y **Acción2**, el **JFrame** deberá de verse como sigue.



- ✓ Insertar **Submenús** en el Menú **Descripción**, del panel **Swing Components** ubicar el componente **Menu Item**, arrastrarlo y colocarlo en el Menú **Descripción**, inserta 2 **Submenús**, modifica su contenido por **Descripción1** y **Descripción2**, El **JFrame** deberá de verse como sigue.



- ✓ Solicita a los alumnos del grupo que ejecuten el **JFrame Menu** que observen su funcionamiento, en el caso de ser necesario realicen las adecuaciones pertinentes.
- ✓ Comenta que los Submenús funcionan en forma similar a los botones (JButton), en virtud de que al momento de dar click con el mouse en el Submenú respectivo, el control del programa nos enviará a un método.
- ✓ Solicita que elijan los Submenús **Acción1**, **Acción2**, y que empleando la propiedad **Change Variable Name** coloquen el



nombre que deseen, posteriormente colocados sobre el submenú al dar doble click visualizan el método respectivo.

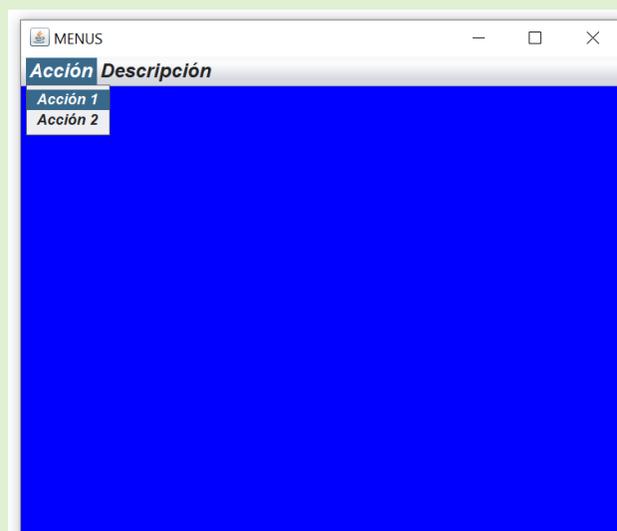
```
private void accion2ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}  
  
private void accion1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

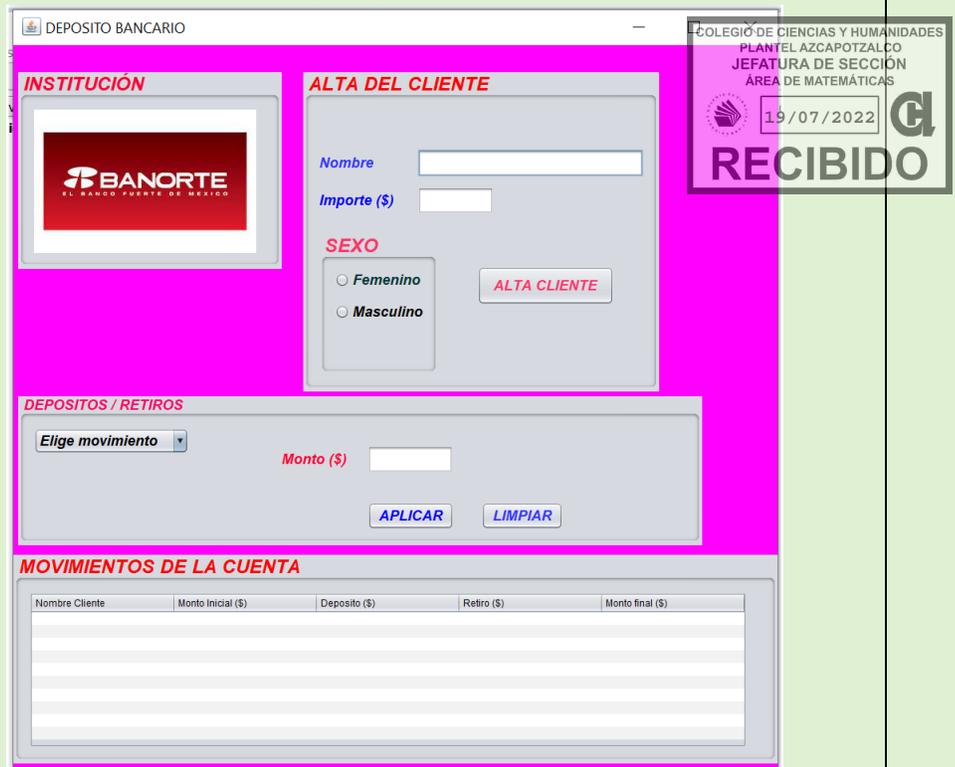


- ✓ Indica que para poder interactuar con otra ventana (**JFrame**), en el método **accion1** se deberá crear un objeto de la Clase (**JFrame**) que se desea visualizar y con base en el objeto creado haciendo uso de su propiedad **setVisible(true)** se podrá visualizar la ventana.

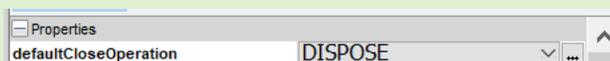
```
private void accion1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Tabla inversion = new Tabla();  
    inversion.setVisible(true);  
    // Seguridad candado = new Seguridad();  
    // candado.setVisible(true);  
}
```

Comenta que en el método **accion1** se creó un objeto llamado **inversion** de la Clase **Tabla**, al ejecutarse la ventana **Menu** y elegir el submenú **Acción1** se observará lo siguiente.





- ✓ Comenta que para que la ventana invocada trabaje en forma adecuada y que al momento de cerrarla retorne a la ventana que la llamo, es necesario que la ventana invocada **JFrame (Tabla)** en su propiedad **defaultCloseOperation** se elija la opción **DISPOSE**, acción que permitirá retornar a la ventana **Menu (JFrame)** inicial.



Desarrollo (70 minutos)

Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

- **Diseño de la ventana** (20 minutos).

- ✓ El alumno realiza la construcción de la ventana (JFrame) **Menu** empleando los componentes gráficos **MenuBar** y **MenuItem**

- Actividad a desarrollar (50 minutos).

Con el propósito de consolidar el aprendizaje de los alumnos y que éstos desarrollen una nueva aplicación, el profesor solicita a los alumnos del grupo que construyan un nuevo JFrame con el nombre que deseen, dentro del cual deberán de insertar un **MenuBar** y **MenuItem** que deberá contener los submenús necesarios que les permita interactuar con todos los proyectos gráficos que previamente han construido, recuerda que dichos proyectos son:

- ✓ Calculadora básica.
- ✓ Cálculo del área de figuras geométricas.
- ✓ Tabla de Multiplicar, Factorial de un número y sumatoria del número 1 hasta el número proporcionado.
- ✓ Depósito Bancario
- ✓ Cálculo del índice de masa corporal.

Cierre (20 minutos)

El profesor en reunión plenaria elige a dos alumnos los cuales muestran el funcionamiento de su programa.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma responsable, ordenada y respetuosa expresa comentarios y aclara dudas, comparan su programa contra el expuesto por sus compañeros.
- ✓ Obtiene conclusiones relativas al manejo de la interfaz gráfica de usuario (JFrame), tales como:



	<ul style="list-style-type: none"> ▪ Manejo de Menus y Menu Items. ▪ Programación relativa a Menu y Menu Items ▪ Comprende la manera en que deberá cerrar la ventana invocada para que el control del programa retorne a la ventana original que la llamo. ▪ Comprende la forma de interactuar entre diversas ventanas, aprendizaje fundamental que le permite adquirir destrezas y habilidades que aplicará en el diseño y desarrollo de su proyecto final.
--	--



EVALUACIÓN	Lista de cotejo		
	Concepto	Si	No
	El alumno inserta y maneja en el JFrame en forma adecuada el componente gráfico Menu Bar.		
	El alumno inserta y maneja en el JFrame en forma adecuada el componente gráfico Menu Item y lo coloca en el Menu Bar correspondiente.		
	El alumno escribe el código pertinente en el método del Menu Item correspondiente que le permite interactuar con otra ventana.		
	El alumno modifica en forma correcta las propiedades de la ventana invocada para que al momento de cerrarla esta retorne a la ventana que la invoco.		
	La aplicación desarrollada por el estudiante relaciona al menos 4 diferentes ventanas		

	La aplicación desarrollada se ejecuta sin contratiempos y el control entre ventanas funciona correctamente.		
	Total		



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama, Ra–Ma.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>Martín, A. (2014). Programador Java certificado: curso práctico.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Arnaw, D. & Weiss, G. (2000). Introducción a la programación con Java. Madrid: Addison–Wesley</p> <p>Villalobos J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, uml, Objetos y Eclipse. México: Pearson Educación.</p>

V. ANEXOS

Se incluyen todos los textos, instrumentos, imágenes, entre otros, en apoyo a la estrategia/secuencia.

Estrategia didáctica 6

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 2 de mayo 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Propone un proyecto que utilice las Clases: JFrame, JLabel, JButton, JTextField, JPasswordField, JComboBox, JTable, JRadio Button, JButton Group, JPanel, JMenu Bar y JMenuItem.
TEMA(S)	Clase Swing: <ul style="list-style-type: none"> • JFrame • JLabel • JTextField • JButton • JPasswordField

	<ul style="list-style-type: none"> • JComboBox • JRadio Button • JButton Group • JTable • JPanel • JMenu Bar • JMenu Item
--	--



III. ORGANIZACIÓN DE ACTIVIDADES

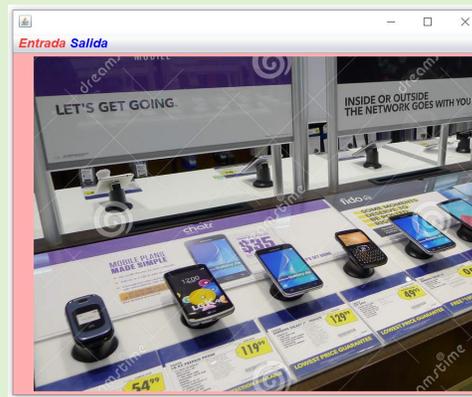
TIEMPO	2 sesiones (240 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual

1era. Sesión.
Inicio (40 minutos)

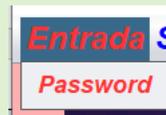
El profesor:

- Comenta a los alumnos del grupo que empleando la interfaz gráfica de usuario y utilizando los componentes gráficos de la clase **Swing**, construir proyecto que englobe todos los aprendizajes vistos de la clase Swing, relativo a:

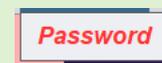
- a) Crear ventana (JFrame) que contendrá una etiqueta en la cual se insertará imagen a su elección como fondo de la ventana, un Menú de barras y Submenú, observa.



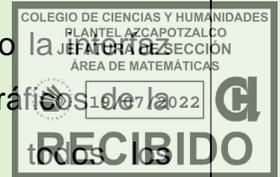
- b) Al abrir el Menú **Entrada** aparece lo siguiente.

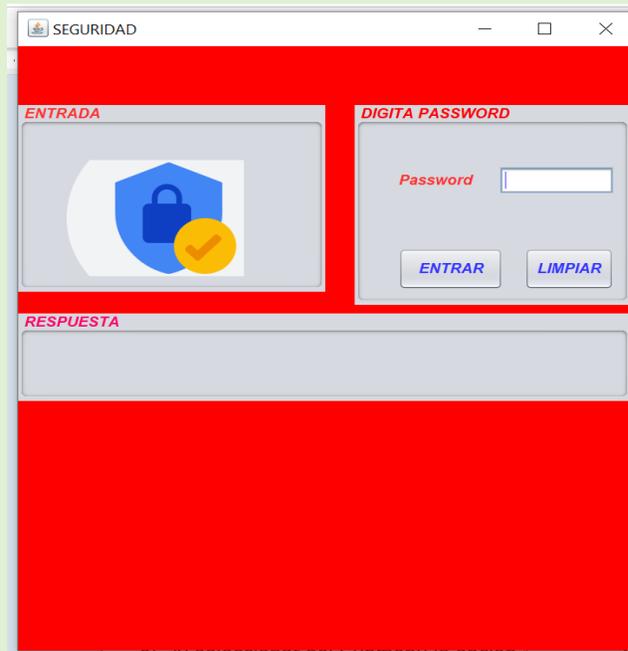


- c) Al seleccionar del Submenú la opción



Aparece ventana.





La Ventana **SEGURIDAD** (JFrame) mostrada, contiene 3 Panel, que se describen a continuación:

- Panel **ENTRADA** contiene una etiqueta dentro del cual se insertará imagen a tu elección.
- Panel **DIGITA PASSWORD** contiene etiqueta con la palabra **Password**, y un campo de texto de tipo “Password Field” y 2 botones

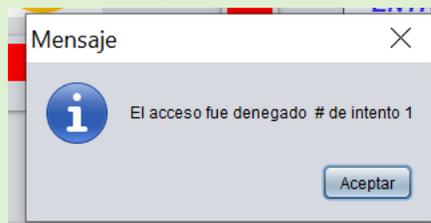


- Botón

Verifica que el contenido del “Password Field” capturado sea el adecuado.

Password incorrecto.

En el caso de que el contenido del “Password Field” sea incorrecto aparece el mensaje.



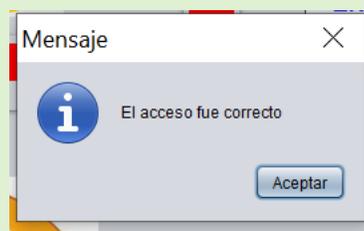
En el caso de 3 intentos incorrectos el programa se desactiva.

Botón **LIMPIAR**

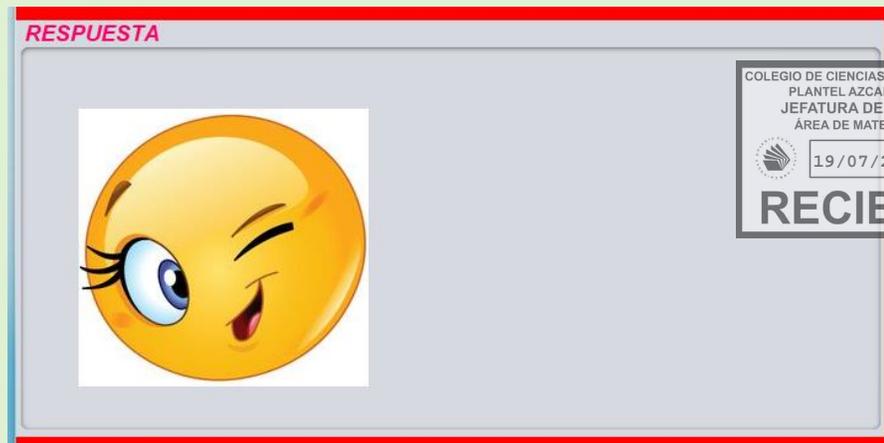
limpia el contenido del campo de texto “Password Field”.

Password correcto.

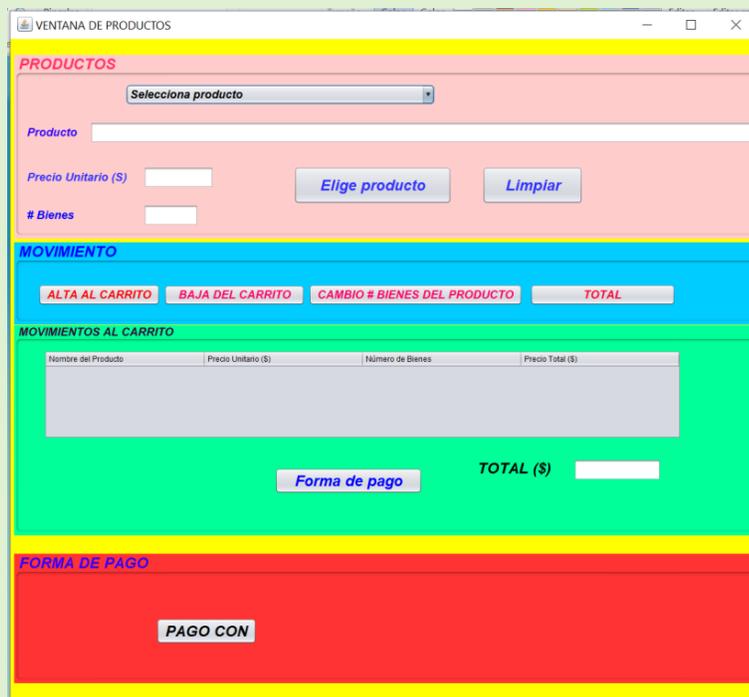
En el caso de que el password tecleado sea correcto, aparece en pantalla



Posteriormente, en el panel **RESPUESTA** aparece etiqueta con imagen de tu elección.



A continuación, aparece la siguiente ventana (JFrame).



La Ventana (JFrame) VENTANA DE PRODUCTOS, contiene 3 Panel, los cuales se describen a continuación:

- Panel **PRODUCTOS** contiene componente gráfico Combo Box en donde se insertará un catálogo de productos (los productos

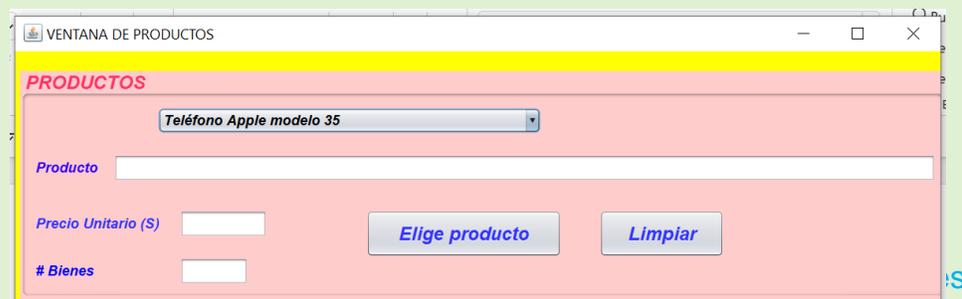
pueden ser los que tu eligas), 3 etiquetas (**Producto**, **Precio Unitario (\$)**, **# Bienes**), 3 campos de texto, y 2 botones



✓ Botón



del componente gráfico Combo Box se selecciona un producto. al dar click al botón **Elige producto** el producto seleccionado se muestra en el campo de texto así como su precio unitario, observa.



✓ Botón **Limpiar** limpia el contenido de los campos de texto.

- Panel **MOVIMIENTO**
Contiene 4 botones.

ALTA AL CARRITO BAJA DEL CARRITO CAMBIO # BIENES DEL PRODUCTO TOTAL



✓ Al dar click al Botón **ALTA AL CARRITO**, el producto seleccionado se muestra en el componente gráfico JTable que se muestra en el Panel **MOVIMIENTOS AL CARRITO**, observa.

Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
Teléfono Apple modelo 35	6800	2	13600

Considerando que el cliente ha seleccionado varios productos que han sido dados de alta en el carrito, se tiene lo siguiente observa.

Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
Teléfono Apple modelo 35	6800	2	13600
Microcomputadora Huawei procesador 33	7900	3	23700
Microcomputadora HP procesador 18	9500	2	19000
Refrigerador Samsung 14	10700	3	32100

✓ **Baja del carrito**

Para dar de baja un producto, con la ayuda del mouse nos debemos de colocar en el componente gráfico JTable y seleccionar el producto.



MOVIMIENTOS AL CARRITO

Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
Teléfono Apple modelo 35	6800	2	13600
Microcomputadora Huawei procesador 33	7900	3	23700
Microcomputadora HP procesador 18	9500	2	19000
Refrigerador Samsung 14	10700	3	32100

Forma de pago TOTAL (\$)

Microcomputadora HP procesador 18 causará baja, al dar click sobre el botón



observa que el producto **Microcomputadora HP procesador 18** se ha eliminado de la Tabla.

MOVIMIENTOS AL CARRITO

Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
Teléfono Apple modelo 35	6800	2	13600
Microcomputadora Huawei procesador 33	7900	3	23700
Refrigerador Samsung 14	10700	3	32100

Forma de pago TOTAL (\$)

✓ **Cambio de bienes del carrito**

Para dar de un Cambio de Bienes del carrito, con la ayuda del mouse nos debemos de colocar en el componente gráfico JTable y seleccionar el producto.

Al dar click al botón



MOVIMIENTOS AL CARRITO

Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
Teléfono Apple modelo 35	6800	2	13600
Microcomputadora Huawei procesador 33	7900	3	23700
Refrigerador Samsung 14	10700	3	32100

Forma de pago TOTAL (\$)



Aparece el nombre del producto y su precio unitario.

Producto

Precio Unitario (\$) Elige producto Limpiar

Bienes

MOVIMIENTO

ALTA AL CARRITO
BAJA DEL CARRITO
CAMBIO # BIENES DEL PRODUCTO
TOTAL

APLICA CAMBIO

El usuario deberá capturar el # Bienes, en forma adicional aparece en el Panel **MOVIMIENTO**, botón APLICA CAMBIO

Al dar click a dicho botón en el componente gráfico JTable se actualiza el contenido de #Bienes, observa que el #Bienes del producto **Microcomputadora Huawei procesador 33** se modifico de 3 a 1.

MOVIMIENTOS AL CARRITO

Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
Teléfono Apple modelo 35	6800	2	13600
Microcomputadora Huawei procesador 33	7900	1	7900
Refrigerador Samsung 14	10700	3	32100

Forma de pago TOTAL (\$)

✓ **TOTAL**

El botón **TOTAL**

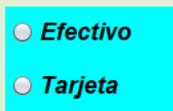
nos permite conocer el costo total de los productos adquiridos, mismo que se muestra en el campo de texto ubicado en el Panel **MOVIMIENTOS AL CARRITO**, al dar click a dicho botón se tiene.



Panel **FORMA DE PAGO** contiene 1 botón



Al dar click a dicho botón aparece. agrupa 2 botones de radio



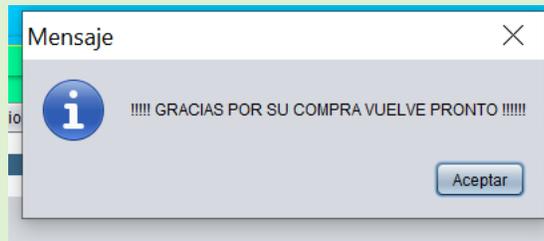
Al seleccionar el botón de radio Efectivo

Aparece el botón **REALIZA COBRO**

Al dar click en dicho botón se tiene.

Recibo Dinero (\$)

y el mensaje.

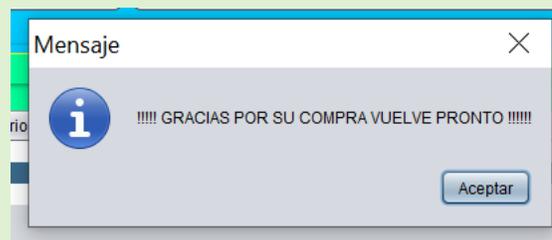


✓ Al seleccionar el botón de radio **Tarjeta**

Aparece el botón **REALIZA COBRO**

Al dar click en dicho botón se tiene. **Inserte Tarjeta**

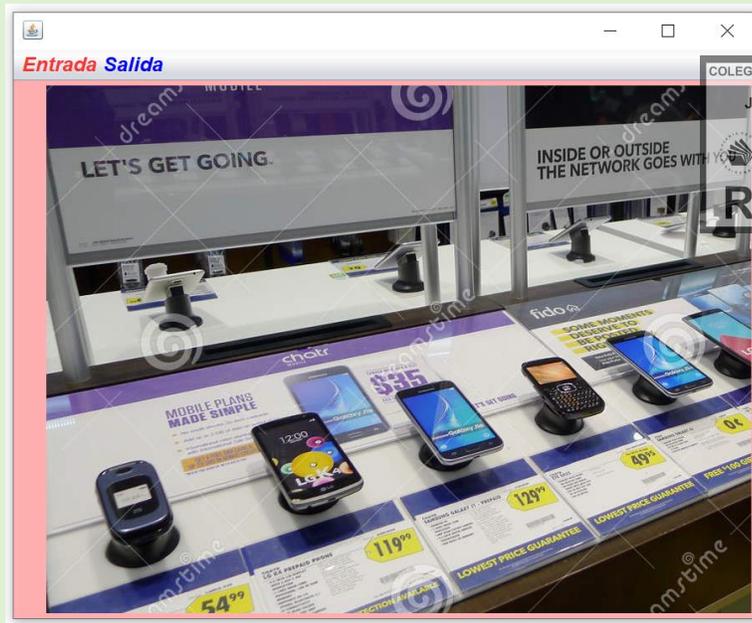
Apareciendo el mensaje



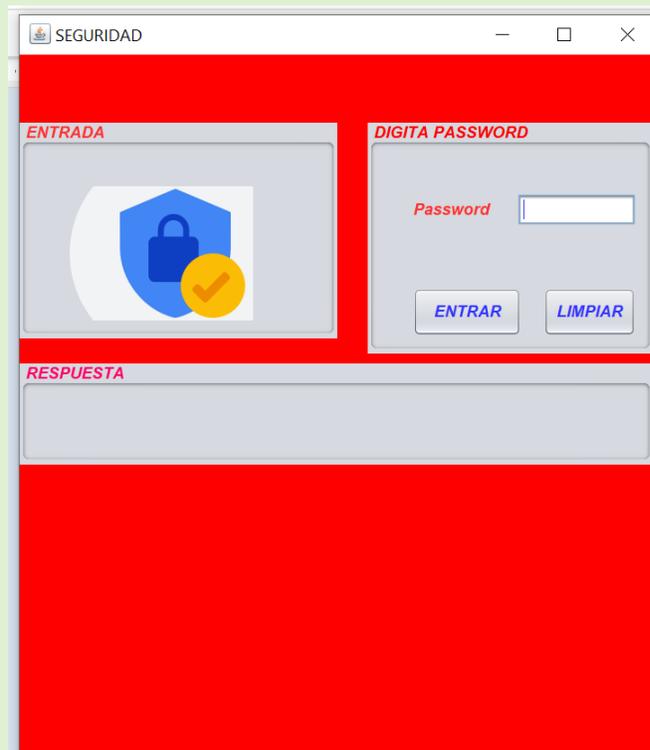
Desarrollo (80 minutos)

Bajo la guía y supervisión del profesor, el alumno realiza lo siguiente:

- **Diseño de la ventana** (JFrame) relativa al Menú (20 minutos).
- ✓ El alumno realiza la construcción de la ventana (JFrame) **Menu** empleando los componentes gráficos **MenuBar** y **MenuItem**.



- Diseño de la ventana **SEGURIDAD**



Programación relativa a los botones



2da sesión (90 minutos)

- Diseño de la ventana JFrame (**VENTANA DE PRODUCTOS**) y programación respectiva.



Nombre del Producto	Precio Unitario (\$)	Número de Bienes	Precio Total (\$)
---------------------	----------------------	------------------	-------------------

Cierre (30 minutos).

La presente estrategia fue desarrollada con los alumnos del grupo 601, a través de esta aplicación, el estudiante reafirmó aprendizajes como estructuras de control, arreglos unidimensionales y bidimensionales que estudio en unidades anteriores, y consolido aprendizajes recientes como herencia y los componentes gráficos del paquete Swing, el desarrollo y aplicación de la presente estrategia le permitió al alumno adquirir habilidad y experiencia en el manejo del lenguaje de programación Java, la cual empezó a trabajar en primera instancia en forma consola y posteriormente en el ambiente gráfico.

Con el propósito de incentivar la curiosidad y el talento de nuestros jóvenes, en forma adicional, como parte de su evaluación se solicitó que **“propusiera”**, **“diseñara”** y **“construyera”** proyecto relacionado con algún tema de su interés en donde englobará la

	<p>mayoría de los aprendizajes que se estudiaron durante el curso incluyendo el paquete Swing, en la solución de dicho proyecto deberá manejar varias ventanas (JFrames).</p> <p>En reunión plenaria.</p> <p>El Alumno:</p> <ul style="list-style-type: none"> ✓ Observo que el lenguaje Java en general y particularmente su ambiente gráfico es una herramienta muy poderosa que le brindará aprendizajes significativos que le permitirá desarrollar aplicaciones de diversa índole que podrá aplicar en su vida actual o futura escolar o personal.
EVALUACIÓN	Se proporciona rúbrica, misma que se presenta a continuación cuyo propósito fue guiar el trabajo del alumno.



Estrategia didáctica 7

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 4 de mayo 2022
NO DE ALUMNOS	Número de alumnos: 25



II.PROGRAMA

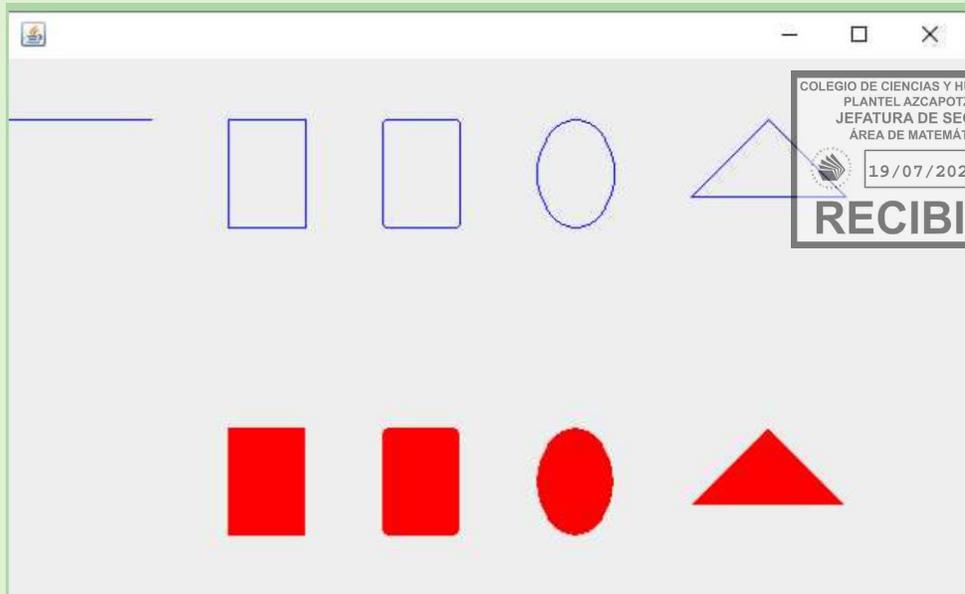
TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Elabora programas con interfaz gráfica de usuario aplicando las Clases: setColor, drawLine, drawRect, drawRoundRect, drawOval, drawPolygon, fillRect, fillRoundRect, fillOval, fillPolygon.
TEMA(S)	Clase Graphics: <ul style="list-style-type: none"> • setColor. • drawLine. • drawRect. • drawRoundRect. • drawOval. • drawPolygon.

	<ul style="list-style-type: none"> • fillRect. • fillRoundRect. • fillOval. • fillPolygon.
--	--

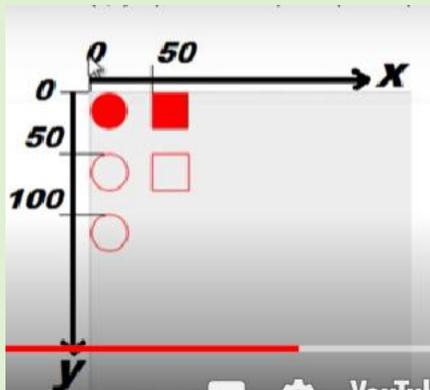


III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	1 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual
	<p>Inicio (50 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> • Comenta a los alumnos del grupo que el lenguaje de programación Java proporciona la clase Graphics que nos permite dibujar líneas, cuadrados, rectángulos, óvalos, círculos, polígonos, mostrar texto y muchos otros métodos de dibujo, que para poder pintar, un programa necesita un contexto gráfico válido, representado por una instancia de la clase Graphics, pero esta clase no se puede instanciar directamente; así que debemos crear un componente y pasarlo al programa como un argumento al método paint(). • Con el propósito de utilizar los métodos que provee la clase Graphics, se propone crear la siguiente aplicación.



- Comenta lo siguiente:
 - ✓ La ventana del JFrame es un lienzo donde se podrá dibujar, para dibujar algo dentro del lienzo debemos de indicar en que posición empezaremos a pintar, para esto cada punto del lienzo tiene una posición (x,y) asociada, siendo el punto (0,0) el punto de la esquina superior izquierda, observa lo siguiente.



- ✓ Para dibujar una línea se emplea el método `drawLine(x1, y1, x2, y2)`, donde $(x1, y1)$ y $(x2, y2)$, serán las

coordenadas del punto inicial y final donde se dibujara la línea.

✓ Para dibujar un rectángulo se emplea el método `g.drawRect(x1,y1,ancho,alto)`, donde $(x1, y1)$ son las coordenadas del punto inicial a partir cual se dibujara el rectángulo.

✓ Para dibujar un rectángulo con puntas redondeadas se emplea el método `drawRoundRect(x1,y1,ancho,alto,anchop,altop)`, donde $(x1, y1)$ son las coordenadas del punto inicial a partir cual se dibujará el rectángulo y anchop y altop son la parte redondeada en las esquinas.

✓ Para dibujar un óvalo se emplea el método `g.drawOval(x1,y1,ancho,alto)`, donde $(x1, y1)$ son las coordenadas del punto inicial a partir cual se dibujará el óvalo.

✓ Para dibujar un polígono debemos especificar el número de puntos que conforma el polígono, si deseamos dibujar un triángulo debemos de indicar las coordenadas de 3 puntos (vértices del triángulo), para poder especificar las coordenadas se emplean 2 arreglos de tipo unidimensional, uno para las coordenadas del **eje x** y el otro para las coordenadas del **eje y**.

```
int cejex [ ] = {500,600,400}
```

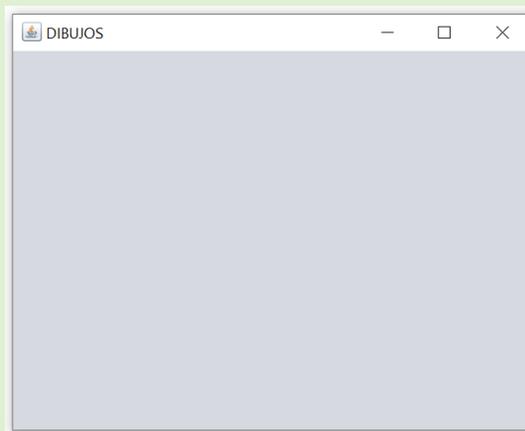
```
int cejey [ ] = {80,150,150}
```

para dibujar el polígono escribimos.

```
g.drawPolygon(cejex,cejey,3)
```

- A continuación, se crea una Ventana (JFrame) denominado **Dibujos**.





Seleccionar la pestaña

Source

Colocarnos después del método **main** y crear un método público llamado **paint**, observa.

```
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

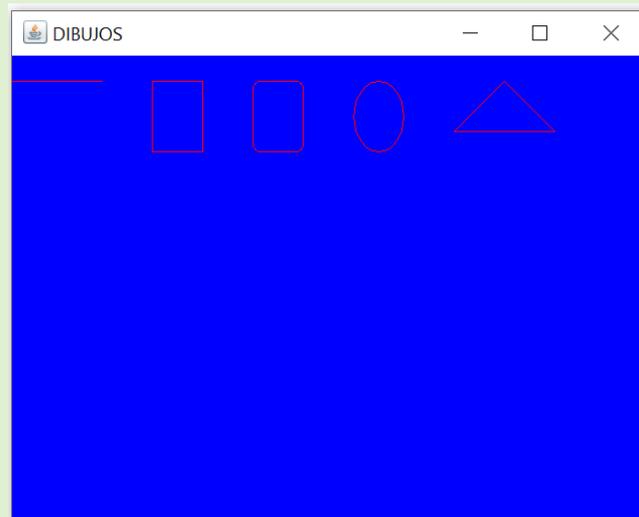
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Dibujos().setVisible(true);
        }
    });
} //Fin del método main

//Insertar el método paint
```

✓ En el método **paint** escribe.

```
public void paint(Graphics g){
    super.paint(g); //se llama al paint de la clase Padre JFrame
    g.setColor(Color.red);
    g.drawLine(0,70,100,70); //dibuja una línea
    g.drawRect(150,70, 50,70); //dibuja un rectángulo
    g.drawRoundRect(250,70, 50,70,15,15); //dibuja un rectángulo redondeado
    g.drawOval(350,70,50,70); //dibuja un ovalo
    int [] ex = {500,550,450}; //define las coordenadas del eje x
    int [] ey = {70,120,120}; //define las coordenadas del eje y
    g.drawPolygon(ex, ey, 3); // construye un triángulo
} // fin del método paint
```

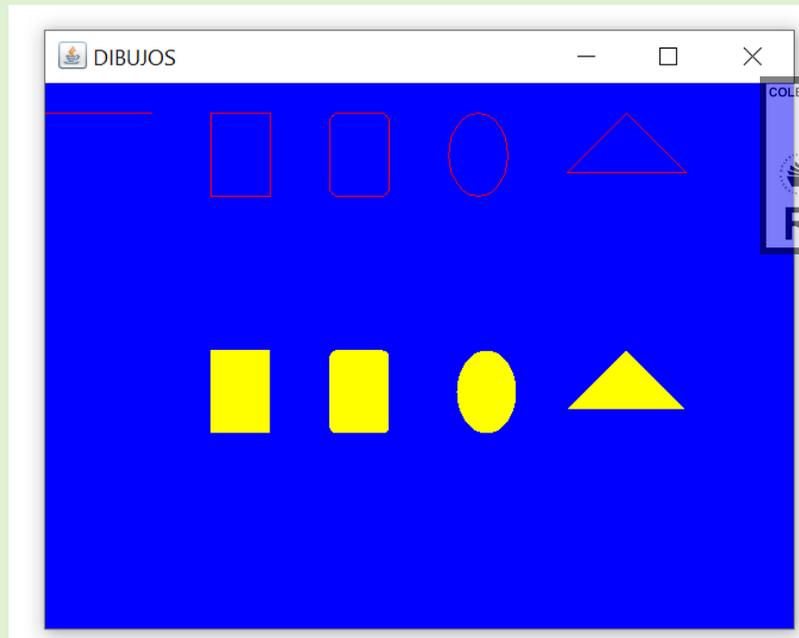
✓ Ejecuta el JFrame creado y observa.



- Comenta que la clase Graphics también cuenta con los métodos **fillRect()** rellena un rectángulo, **fillRoundRect()** rellena un rectángulo con puntas redondeadas, **fillOval()** rellena un óvalo, **fillPolygon()** rellena un polígono.
- Solicita modificar la ventana (JFrame) **Dibujos** para que en forma adicional se construya y se muestre las figuras geométricas con relleno.
 - ✓ Para ello solicita a los alumnos del grupo que adicionen en el método paint lo siguiente.

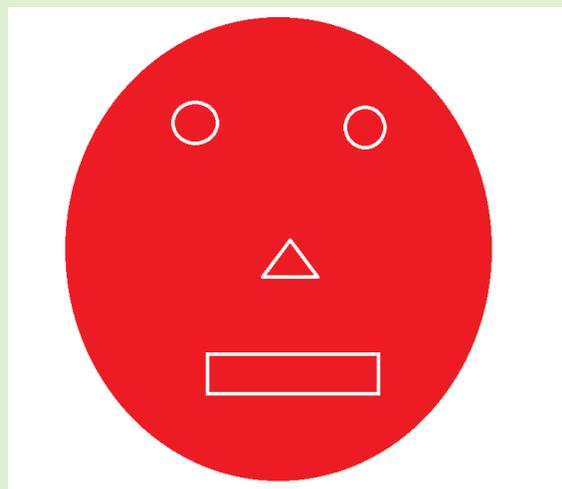
```
g.setColor(Color.YELLOW);  
g.fillRect(150,270, 50, 70); //rellena rectángulo  
g.fillRoundRect(250,270,50,70,10,10); //rellena rectángulo con puntas redondeadas  
g.fillOval(357,270,50,70); //rellena ovalo  
int [] vx = {500,550,450}; //coordenadas del eje x  
int [] vy = {270,320,320}; //coordenadas del eje y  
g.fillPolygon(vx, vy,3); //rellena polígono  
}  
// fin del método paint
```

✓ Ejecuta el JFrame creado y observa.



Desarrollo (50 minutos)

Con el propósito de evaluar el aprendizaje alcanzado por los alumnos, el profesor solicita a los alumnos del grupo que empleando los métodos de la clase Graphics estudiados, creen una ventana (JFrame) con el nombre que ellos decidan en donde en el lienzo del JFrame deberán de construir y mostrar la siguiente imagen.



Cierre (20 minutos)

En reunión plenaria tres alumnos exponen el desarrollado.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma responsable, ordenada y respetuosa expresan comentarios y aclaran dudas, comparan su programa contra el expuesto por sus compañeros.
- ✓ Obtiene conclusiones relativas al manejo de la interfaz gráfica de usuario (JFrame), tales como:
 - Construcción de líneas y figuras geométricas.
 - Construcción de cuadrados, rectángulos, óvalos o círculos a través del método drawRect y drawOval.
- ✓ Concluye observa que las clases Swing y Graphics le proporcionan aprendizajes útiles que podrán llegar a ser significativos en la medida que el estudiante los aplique en su vida cotidiana o futura.



EVALUACIÓN

Lista de cotejo

Concepto	Si	No
El alumno crea en forma correcta el método paint que le permite emplear los elementos gráficos de la clase Graphics.		
En el programa fuente inserta el método Paint en el lugar adecuado		
Los elementos gráficos empleados de la clase Graphics son adecuados para la actividad solicitada		
El alumno crea en forma adecuada la imagen solicitada		
Los colores de fondo empleados son adecuados y permiten que la imagen solicitada sea agradable a la vista.		
Total		

Estrategia didáctica 8

I. DATOS GENERALES

PROFESOR(A)	Seminario institucional	
ASIGNATURA	Nombre de la Asignatura: Cibernética y Computación II	
SEMESTRE ESCOLAR	Semestre escolar: Sexto Semestre	
PLANTEL	Escuela Nacional Colegio de Ciencias y Humanidades	
FECHA DE ELABORACIÓN	Fecha de elaboración de la estrategia: 5 de mayo 2022	
NO DE ALUMNOS	Número de alumnos: 25	

II.PROGRAMA

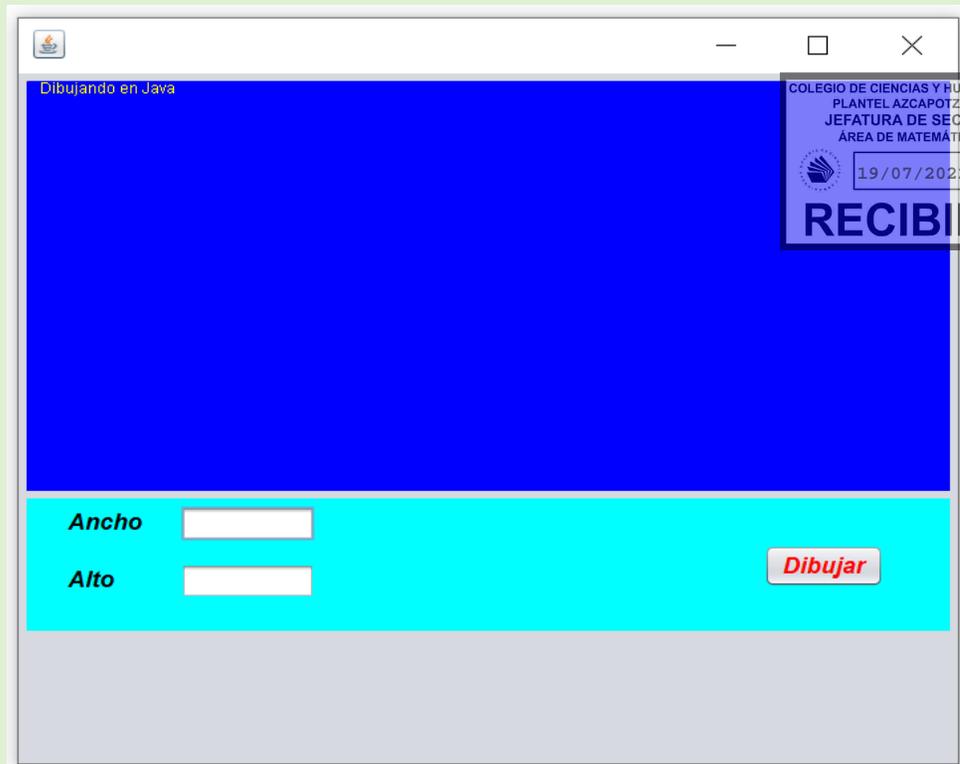
TÍTULO DE LA UNIDAD	Unidad IV Interfaz gráfica de usuario
PROPÓSITO(S) DE LA UNIDAD	Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos
APRENDIZAJE(S)	Elabora programas con interfaz gráfica de usuario aplicando las Clases: setColor, drawLine, drawRect, drawRoundRect, drawOval, drawPolygon, fillRect, fillRoundRect, fillOval, fillPolygon.
TEMA(S)	Clase Graphics: <ul style="list-style-type: none"> • setColor. • drawLine. • drawRect. • drawRoundRect. • drawOval. • drawPolygon.

	<ul style="list-style-type: none"> • fillRect. • fillRoundRect. • fillOval. • fillPolygon.
--	--



III. ORGANIZACIÓN DE ACTIVIDADES

TIEMPO	2 sesión (120 minutos).
MATERIALES Y/O RECURSOS	<ul style="list-style-type: none"> • Aula en el laboratorio de cómputo • Aula en condiciones adecuadas de ventilación y limpieza • Espacio entre computadoras de al menos 1 metro • Pizarrón • Plumones • Computadora con acceso a internet • Software IDE de NetBeans • Forma de trabajar individual
	<p>Inicio (40 minutos)</p> <p>El profesor:</p> <ul style="list-style-type: none"> • Comenta que, en equipos de trabajo de 4 alumnos, empleando la interfaz gráfica de usuario (clases Swing y Graphics) desarrollará proyecto, para ello propone 3 alternativas, los equipos de trabajo deberán elegir 1. <p>a) Movimiento de figura geométrica (Rectángulo o cuadrado) en una ventana, observa.</p>

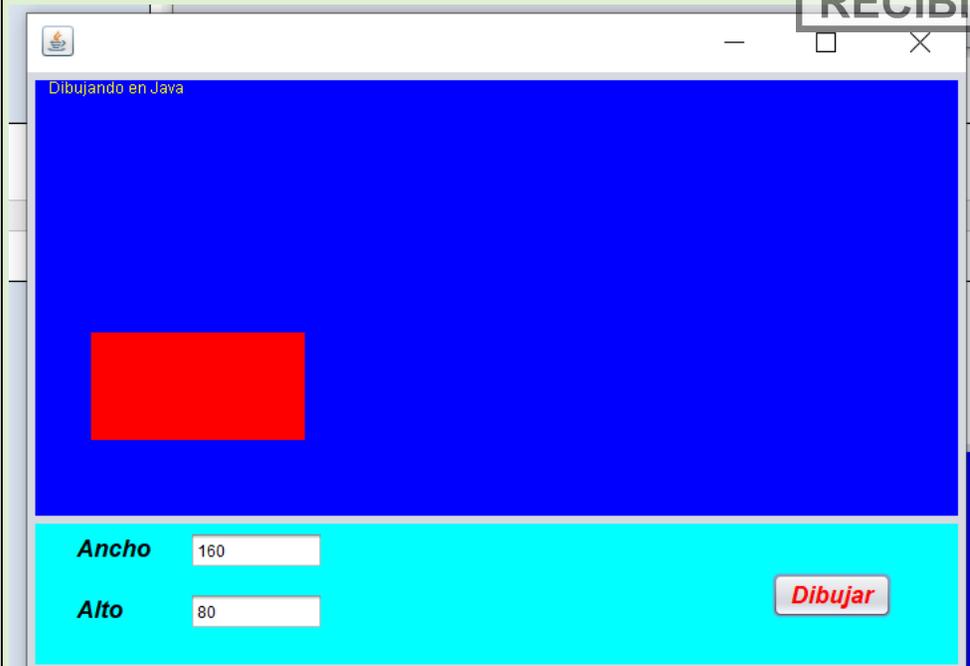


- ✓ Captura el **Ancho** y **Alto** de la figura geométrica.
- ✓ Con ayuda del mouse desplázate en el panel “dibujado en Java” y da click en la posición que deseess, lo que te permitirá capturar las
- ✓ coordenadas iniciales (x,y), a partir de donde quieres que se dibuje la figura geométrica, posteriormente da click en el botón.



Dibujar

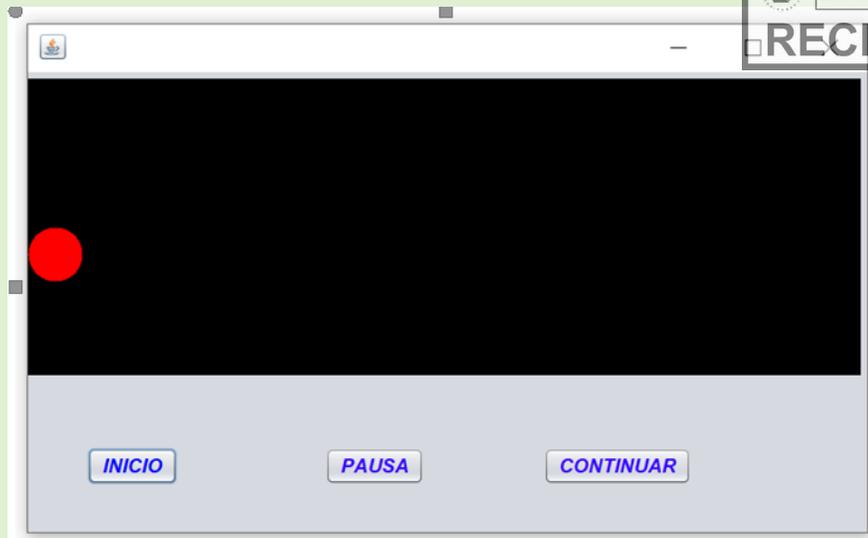
Observa.



- ✓ Con la ayuda del mouse selecciona otro punto a partir de donde desees construir la figura geométrica, da click en el botón **Dibujar** y observa.



b) Movimiento horizontal de figura geométrica (Círculo o Óvalo) en una ventana, observa.



Da click al botón 

observa el movimiento de la figura en el panel.



Al momento de dar click al botón

PAUSA

El movimiento de la figura geométrica se detiene.

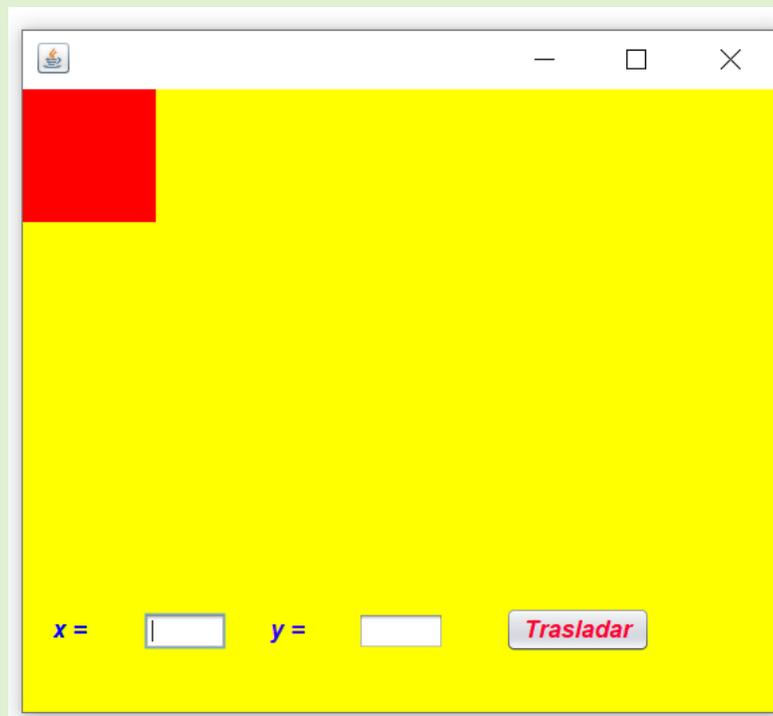
Al momento de dar click al botón

CONTINUAR

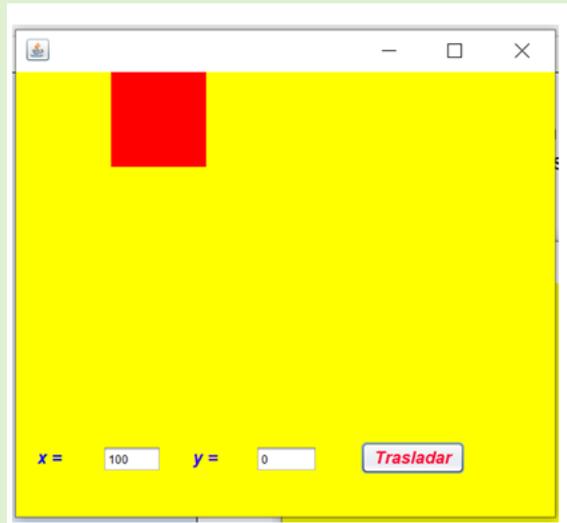
El movimiento de la figura geométrica continua.



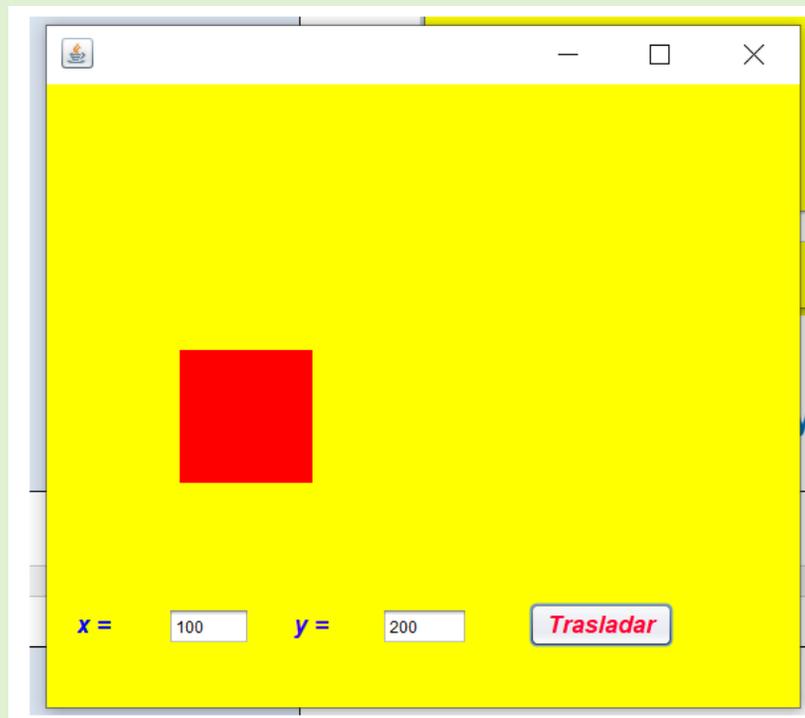
c) Movimiento de traslación de una figura geométrica



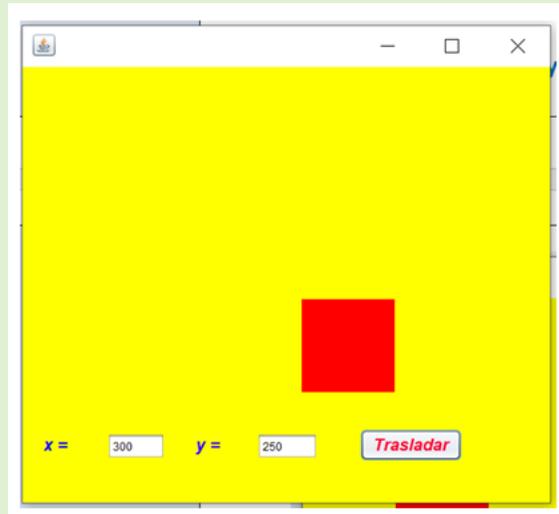
- ✓ Digitando valores $x = 100$, $y = 0$ y dando click al botón **Trasladar** la figura geométrica se trasladará hacia la derecha.



- ✓ Digitando valores $x = 100$, $y = 200$ y dando click al botón **Trasladar** la figura geométrica se trasladará hacia abajo.



- ✓ Digitando valores $x = 300$, $y = 250$ la figura geométrica se trasladará hacia la derecha y abajo.



Desarrollo (80 minutos) de la primera clase y (100 minutos) de la segunda clase

- Los equipos de trabajo seleccionan 1 de las 3 alternativas propuestas.
- ✓ Bajo la guía y asesoramiento del profesor los equipos de trabajo realizan el trabajo.

Cierre (20 minutos)

En reunión plenaria tres equipos exponen su trabajo, en donde cada equipo expone una diferente alternativa.

Alumno:

- ✓ Observa el trabajo de sus compañeros, en forma respetuosa, ordenada y responsable expresa comentarios y aclara dudas, compara su trabajo contra el expuesto por los equipos de trabajo.

✓ Durante el desarrollo de los diferentes proyectos se retomaron y consolidaron diversos aprendizajes que se trabajaron a lo largo del curso de Cibernética y Computación II, ejemplo de ello: **herencia, polimorfismo, modificación de métodos constructores, e interfaz gráfica de usuario (clases Swing y Graphics)**, situación que beneficio al estudiante, los proyectos propuestos y desarrollados le permitieron incrementar su capacidad cognoscitiva en relación al lenguaje, así como, adquirir y habilidad y destreza en su manejo. Comprende que la **interfaz gráfica de usuario (clases Swing y Graphics)**, le proporcionaron valiosas y poderosas herramientas que podrá emplear tanto en su vida escolar, personal, actual o futura.



Lista de cotejo

Concepto	Si	No
El alumno analiza reconoce e interpreta perfectamente los datos, identificando con certeza lo que se busca y demostrando una absoluta comprensión del problema.		
Emplea estrategia de solución heurística efectiva y eficiente, con base a los datos previamente identificados propone modelo de solución sencillo.		
Emplea de manera adecuada los componentes de la interfaz gráfica de usuario del lenguaje de programación Java, y en forma óptima los aplica al modelo de solución propuesto, relaciona y aplica en gran medida conceptos matemáticos y de la programación orientada objetos.		
Desarrolla el modelo propuesto empleando en forma adecuada y rigurosa el lenguaje matemático y de la programación orientada a objetos, realiza cálculos correctos y proporciona la totalidad de los resultados.		

No tiene errores de compilación, y ejecuta correctamente obteniendo el 100% de los requerimientos planteados en el problema.			
Total	Total		



IV. REFERENCIAS DE APOYO

BIBLIOGRAFÍA DE CONSULTA PARA LOS ALUMNOS	<p>Ceballos, F. (2015). Java: interfaces gráficas y aplicaciones para Internet. Paracuellos de Jarama, Ra–Ma.</p> <p>Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall.</p> <p>Martín, A. (2014). Programador Java certificado: curso práctico.</p>
BIBLIOGRAFÍA DE CONSULTA PARA EL PROFESOR	<p>Arnaw, D. & Weiss, G. (2000). Introducción a la programación con Java. Madrid: Addison–Wesley</p> <p>Villalobos J. A. (2008). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, uml, Objetos y Eclipse. México: Pearson Educación.</p>

V. ANEXOS