

Índice

Presentación	5
Relaciones con el Área y otras asignaturas	7
Enfoque disciplinario y didáctico.....	8
Contribución al perfil del egresado	10
Evaluación.....	11
Concreción en la materia de los principios del Colegio	12
Propósitos generales de la materia	13
Contenidos Temáticos	14

CIBERNÉTICA Y COMPUTACIÓN I

Unidad 1. La cibernética	15
Evaluación	18
Referencias.....	19

Unidad 2. Representación de información en sistemas binario y algebra de Boole Circuitos lógicos

Evaluación	25
Referencias.....	25

Unidad 3. Metodología de solución de problemas e introducción al lenguaje de programación Java	27
Evaluación	34
Referencias.....	35

CIBERNÉTICA Y COMPUTACIÓN II

Unidad 1. Lenguaje Fundamentos de programación orientada a objetos con Java	37
Evaluación	39
Referencias.....	40
Unidad 2. Estructuras de control de secuencia en Java	41
Evaluación	45
Referencias.....	46
Unidad 3. Herencia y Polimorfismo, constructores, colaboración y herencia de clases.....	47
Evaluación	49
Referencias.....	50
Unidad 4. Interfaz gráfica de usuario	51
Evaluación	56
Referencias.....	56

Presentación

Una de las características distintivas del Colegio de Ciencias y Humanidades (CCH) de otros bachilleratos, que lo hacen innovador y de los más adecuados pedagógicamente en México y América Latina, es su modelo educativo, el cual es de cultura básica, propedéutico (esto es, preparará al alumnado estudiante para ingresar a la licenciatura con los conocimientos necesarios para su vida profesional) y está orientado a la formación intelectual ética y social del alumnado, considerados sujetos de la cultura y responsables de su propia educación. Esto significa que, en la institución, la enseñanza dirigida al estudiante fomentará conocimientos, habilidades y actitudes necesarias para que genere pensamientos crítico y computacional para asumir decisiones de forma ética y con valores.¹”

Desde su origen, el CCH adoptó los principios de una educación moderna donde consideraba al alumnado capaz de apropiarse del conocimiento y sus aplicaciones. En este sentido, el trabajo del docente del Colegio consiste en dotarlos de los instrumentos metodológicos necesarios para poseer los principios de una cultura científica–humanística.

El proceso enseñanza-aprendizaje es inseparable, sin embargo, la enseñanza gira entorno del alumnado y el docente que se convierte en un guía. Por lo que, la metodología aplicada persigue que el alumnado aprenda a aprender, que la actividad receptiva y creadora sea continua y que adquiera capacidad auto formativa.

Para lograr el conocimiento auténtico y la formación de actitudes, intercambiando experiencias con sus compañeros de forma colaborativa en diferentes espacios académicos en su beneficio.

El profesorado basado en sus conocimientos, experiencia, habilidad, y didáctica, es el mediador transmisor de los conocimientos y un acompañante responsable del estudiantado al que propone retos de aprendizaje para que ellos adquieran nuevos conocimientos, sean conscientes de su hacer y proceder de forma reflexiva al aplicarlo en su cotidianidad.

Al ser un aprendizaje dinámico el alumnado desarrollará un aprendizaje activo, respetuoso, pensando en la comunidad, en la sustentabilidad, equidad e igualdad en el espacio académico, para la realización de trabajos de investigación y prácticas de laboratorios.”

El Colegio de Ciencias y Humanidades, su modelo de educación vanguardista y el desarrollo vertiginoso de la tecnología resaltan el compromiso de proporcionar al alumnado educación, conocimiento y habilidades que lo orienten vocacionalmente a la elección de su profesión, y a tener antecedentes necesarios para comprender las materias del nivel superior que elijan, mientras que comprenda que la Cibernética y la Computación son ciencias transdisciplinarias, así que buscarán manejar la tecnología de la información y la comunicación de forma eficiente, ética y segura en todos los campos de la actividad humana.

El alumnado, a través de estrategias de enseñanza-aprendizaje, comprenderá la importancia del procesamiento de la información, adquirirá conocimientos y habilidades que podrá aplicar a situaciones cotidianas, percibirá la vinculación de la matemática, la física, la química, la biología entre otras con la cibernética, al estudiar los sistemas naturales y artificiales, el manejo del Algebra de Boole, los circuitos lógicos mediante una metodología dará solución de problemas al elaborar algoritmos, diagramas de flujo y pseudocódigos, que le facilitarán la codificación en un lenguaje de programación.

Las asignaturas de Cibernética y Computación I y II pertenecen al área de Matemáticas, deben cursarse en dos semestres (quinto y sexto), son obligatorias de elección de acuerdo con el área de la profesión de preferencia. Los conocimientos adquiridos con antelación, más los aportados por esta materia, fomentarán al alumnado, la reflexión de los procesos de aprendizaje y la construcción de conocimientos, haciendo énfasis en la investigación y en el desarrollo de habilidades, lo antes mencionado.

El programa contempla tres unidades para la asignatura de Cibernética y Computación I y cuatro unidades para la asignatura de Cibernética y Computación II, con un total de 64 horas por semestre, el tiempo asignado por semana corresponde a dos sesiones dos clases por semana con duración de dos horas cada una.

Las unidades están graduadas para que el alumno logre los aprendizajes, iniciando con los temas más sencillos hasta los más complejos, con la finalidad de que lo aprendido en una unidad sirva de base para su utilización en las subsecuentes, mediante la siguiente estructura:

Cibernética y Computación I:

En la unidad 1. “La cibernética”, se realiza un abordaje histórico subrayando el trabajo de autoras y autores que definieron a la cibernética, al ser una ciencia que se encarga del control y la comunicación entre sistemas y el modelado para su estudio, se ve la importancia de su transdisciplina, sustentabilidad, equidad e igualdad.

En la unidad 2. “Circuitos lógicos Representación de información en sistemas binario y algebra de Boole”, se ve la importancia de los sistemas de numeración binario, octal, y hexadecimal y binario, ya que son la base de la representación de información en una computadora mediante el uso y operación matemática con bits. Posteriormente, el álgebra de Boole, las compuertas y circuitos lógicos y se finaliza con el planteamiento y desarrollo de un proyecto, por parte del alumnado, siguiendo una metodología utilizando circuitos lógicos. Se utiliza una metodología para resolver circuitos lógicos a través del algebra de Bool y el desarrollo de un proyecto.

En la unidad 3. “Metodología de solución de problemas e Introducción a la programación Java”, el alumnado adquiere una metodología que le permite resolver problemas analizando su planteamiento, proponiendo la solución a través del algoritmo, diagramas de flujo, pseudocódigos y su codificación en el lenguaje de programación Java.

Cibernética y Computación II:

En la unidad 1. “Lenguaje Fundamentos de programación orientada a objetos con Java”, se describen las características del lenguaje de programación orientado a objetos Java y su entorno de desarrollo, definiendo clases, atributos y métodos para la implementación de la solución programas de problemas utilizando el lenguaje de programación Java.

En la unidad 2. “Estructuras de control de secuencia en Java”, se utilizan las estructuras de control de secuencia para la solución de problemas a través de la computadora con un lenguaje de programación orientado a objetos Java.

En la unidad 3. “Herencia y polimorfismo”, se elaboran programas en Java utilizando herencia y polimorfismo, como mecanismos de reutilización de código y especialización o personalización de clases.

En la unidad 4. “Interfaz gráfica de usuario”, se desarrollan programas en Java utilizando interfaces gráficas de usuario mediante formularios, conjuntamente con la temática aprendida retomando lo aprendido visto en

las unidades anteriores.

Se recomienda que los programas en Java **que se sugieran** a resolver **sean** **al alcance intelectual** apropiados al nivel de estudios del alumnado, el curso **pretende crear un pensamiento computacional,** **no pretende formar programadores,** **construyendo** **sino sólo darles** las bases para que en sus estudios posteriores puedan profundizar sus conocimientos.

En general, se sugiere que en cada sesión de dos horas se obtenga un aprendizaje, desarrollando la temática y haciendo uso de las estrategias que incluyen actividades de apertura, desarrollo, cierre y trabajo extra clase. Dado que la evaluación es un proceso que permite conocer de manera sistemática el logro de los aprendizajes, las estrategias sugeridas proponen actividades que propician que el alumnado se auto evalúe, y a la vez, le dan información al profesorado sobre el logro de los aprendizajes, **por parte del alumno.** **Dicha información permitirá dar a conocer al alumnado** sus avances, y así **podrá** **ajustar** la planeación de sus propuestas de enseñanza–aprendizaje.

Relaciones con el Área y otras asignaturas

la computación fueron matemáticos o tenían mucha relación con la matemática, en la actualidad para entender el funcionamiento de una computadora se requiere del uso de la matemática, en particular para la enseñanza–aprendizaje de la materia se hace uso de la metodología de resolución de problemas, en este sentido las asignaturas crean vínculos con distintas áreas materias por ejemplo tiene estrecha relación con el área de Matemáticas. Se requiere del conocimiento de los sistemas de numeración y operaciones aritméticas con números binarios. Teoría de conjuntos de las matemáticas y el argumento deductivo de la lógica en filosofía del modus ponens similar al álgebra de Boole, de la física los circuitos integrados para la realización de los circuitos lógicos. En las Ciencias experimentales la resolución de problemas y del uso de algoritmos.

El programa de estudios de la asignatura de Cibernética y Computación I y II tiene una relación directa con: El programa de estudios de Matemáticas I a IV marca que las asignaturas del tronco común le permiten a los alumnos acceder a conocimientos más especializados contenidos, entre otras asignaturas, a Cibernética y Computación; las asignaturas de Inglés de la IV son requeridas principalmente en el estudio del lenguaje de programación ya que el código de lenguaje Java tiene sentencia en Inglés; en Física, Biología y Química el alumnado estudia los temas de sistemas naturales y artificiales, planteando soluciones sustentables y trabajando en comunidad. circuitos, planteamiento y resolución de problemas. La asignatura tiene estrecha relación con las TIC y Taller de Cómputo el alumnado podrá usar un procesador de texto y una hoja de cálculo; harán investigaciones en Internet, en general harán uso de lo aprendido. Con respecto a la asignatura de Taller de Lectura, Redacción e Iniciación a la Investigación Documental I–IV, el alumnado con ayuda de estas asignaturas podrá comprender en el primer semestre artículos de divulgación científica; en el tercer semestre realizar ensayos académicos, en el cuarto semestre realizan y proyectos de investigación. Lo anterior es evidencia de la transdisciplina de la conforma las bases para el estudio de la asignatura Cibernética y Computación I y II.

Con respecto a las materias de quinto y sexto semestres la asignatura se relaciona directamente con Cálculo Diferencial e Integral o con Estadística y Probabilidad, en ambas asignaturas estudiarán la metodología de solución de problemas con mayor o menor profundidad, planteamiento del problema y

uso de algoritmos. La relación con el resto de las asignaturas consiste en el planteamiento de sistemas y el uso de una metodología para resolver problemas. Por lo tanto, esta asignatura le permite al alumnado reflexionar sobre lo que está aprendiendo en otros campos del saber y tomarlo como tema de aplicación en las asignaturas de Cibernética y Computación. los alumnos pueden proponer el diseño de sistemas relacionados con cualquier asignatura

Enfoque disciplinario y didáctico

En este apartado se dará una aproximación específica para abordar la materia dentro del proceso educativo, por lo que, el programa de estudios de Cibernética y Computación busca dar al alumnado una visión general de la Cibernética, la Computación, **así como un enfoque STEAM el que busca de manera integral darle al alumnado los conocimientos y preparación relacionados a los avances tecnológicos, poniendo énfasis en los pilares de la ciencia, la tecnología, la ingeniería, el arte y las matemáticas. Aunada a una metodología que le permita desarrollar propuestas de solución a problemas transdisciplinarios de forma sustentable.**

El enfoque transdisciplinario permite a la disciplina incidir tanto en las ciencias experimentales y sociales, el dominio de las matemáticas y la tecnología buscan sentar las bases para que el **alumnado** pueda dar solución a la gran alteridad de situaciones que le rodean, aunado a lo que menciona “La Gaceta Amarilla”, “la materia debe proporcionar métodos, técnicas y el hábito de aplicarlos en problemas concretos y de adquirir nuevos conocimientos” (CCH, 1971, p.4).

En esta construcción de conocimientos de conceptos y metodologías desde las ciencias de la Cibernética y la Computación tendrá el alumnado los cimientos al retomar aprendizajes de la asignatura de Taller de Cómputo, donde adquirieron conocimientos, habilidades y actitudes, señalados dentro de los primeros siete niveles de la matriz de habilidades digitales (DGTIC, 2014), desde el acceso a la información, seguridad informática, uso y manejo de ofimática y medios digitales, así como, ambientes virtuales de aprendizaje. Mientras que los contenidos temáticos de las propias asignaturas, corresponden al octavo nivel de la matriz de habilidades en TIC, en el momento que el alumnado analiza, diseña e implementa una programación informática, el uso de simuladores. Aunado al álgebra de Boole que en Cibernética y Computación I le permitirá a diseñar circuitos lógicos.

Del área de Matemáticas se aplicará el modelado de la realidad y la solución de problemas ocupando la computadora como herramienta para la realización de cálculos **complejos**, se retoma el concepto de rango y dominio, así como diversos problemas que planteen soluciones algorítmicas novedosas con dificultad incremental.

Además de enseñar la metodología para que la aplique en distintos escenarios como el Método Científico Experimental y el Aprendizaje Basado en Problemas y/o Proyectos.

Para el desarrollo de programas informáticos es necesario darles a conocer herramientas como los algoritmos y diagramas de flujo que permiten describir paso a paso la solución a un problema determinístico o los diagramas de secuencia aplicados en administración; incluso en la aplicación de soluciones optimizadas sustentablemente en el impacto de energía en el medio ambiente.

Es importante resaltar el uso de lenguajes naturales y formales, la aplicación de la semántica, el léxico y la sintaxis en el contexto de los lenguajes de programación; son elementos relacionados con el área de Talleres de Lenguaje y Comunicación y que se abordan de manera más específica para la construcción de oraciones o sentencias en un lenguaje formal (reglas más estrictas que uno natural).

La materia puede aplicarse para resolver problemas en todos los ámbitos profesionales, y es aquí donde radica su potencial de transdisciplinariedad.

Fundamentos teóricos sólidos: el alumnado, mediante el estudio introductorio de la Cibernética y los Sistemas, comprende la importancia de los conocimientos teóricos que sustentan el desarrollo tecnológico para centrar su atención en la solución de problemas computacionales. Para ello, se revisará los antecedentes de la Cibernética y su concepto, los principales personajes que han aportado a su desarrollo hasta la actualidad, los elementos que conforman un sistema, y la conveniencia de su estudio mediante los modelos.

Metodología: El desarrollo de una metodología para abordar problemas no solo ampliará las habilidades analíticas, sintéticas, creativas y reflexivas de los estudiantes, sino que también fomentará un pensamiento lógico, pensamiento crítico que los capacitará para afrontar desafíos de manera sistemática. Este enfoque no solo se aplica al ámbito académico, sino que también beneficia su capacidad para resolver problemas en la vida cotidiana. Además de la algoritmia, conocerá los principios básicos de la programación orientada a objetos que le permitan desarrollar e

integrar elementos de programación utilizando la abstracción, el acceso controlado a la información y la reutilización de código, como parte de las habilidades esenciales del siglo XXI.

Desarrollo de habilidades para la convivencia profesional: Además de las habilidades cognitivas y técnicas, se busca fomentar el desarrollo de habilidades de comunicación, trabajo en equipo, colaboración y oralidad, preparando al alumnado para que continúe con éxito su preparación académica o profesional; que cuente con la habilidad para comunicarse de forma asertiva con equipos de desarrollo de software como usuario, administrador de proyecto o desarrollador y así, construir conocimiento desde la indagación, transformación y divulgación del mismo.

La materia se fundamenta en los siguientes conceptos básicos, los cuales se agrupan en unidades de estudio.

Cibernética y Computación I

La cibernética

En esta unidad se estudian la definición, antecedentes y personajes que contribuyeron a la formalización de la Cibernética, el concepto de sistema, sus tipos y elementos; enfatizando la importancia del modelado de los sistemas para su estudio, ya que son conceptos asociados a cualquier rama del conocimiento humano se vuelve transdisciplinario. Además, se busca formar conciencia en la responsabilidad de mejorar a través de la sustentabilidad el ambiente físico, psicológico y social, poniendo énfasis en la equidad e igualdad.

Circuitos lógicos Representación de información en sistema binario y álgebra de Boole.

Se usan los sistemas de numeración binario, octal y hexadecimal, y se reconocen como base del manejo y la representación de información en la memoria de la computadora. para comprender de la información en la memoria de la computadora. El alumnado explora los conceptos de tipos de datos, rango, precisión, uso de números enteros con y sin signo, variables booleanas, tipos alfanuméricos; en general tipos de datos primitivos y su codificación en bytes. Posteriormente, se estudia el álgebra de Boole, tablas de verdad, diagramas de compuertas para el diseño y armado de circuitos lógicos; siendo un antecedente a la toma de decisiones con operadores lógicos en estructuras de control de un lenguaje de programación de alto nivel.

Metodología de solución de problemas e introducción al lenguaje de programación Java.

El manejo de métodos que le permitan poner en práctica el análisis, el razonamiento en el desarrollo de algoritmos, el uso de diagramas de flujo y pseudocódigos, para el apoyo de la codificación en el lenguaje de programación Java, que permita sistematizar necesidades planteadas, donde el alumnado analiza, propone y construye la solución de problemas de temas encontrados en algunas de las asignaturas que el alumno cursa, verificando que los resultados obtenidos sean correctos, de forma transdisciplinaria, sustentable, con actitudes para la convivencia contemplando la equidad y la igualdad.

Cibernética y Computación II:

Lenguaje Fundamentos de programación orientada a objetos con Java

Reconocimiento de las características del paradigma de la Programación Orientada a Objetos POO y las del lenguaje de programación "Java", conociendo e implementando la abstracción y encapsulamiento de clases, con sus atributos, métodos e interfaces para poder crear objetos y así obtener programas sencillos con los principios básicos de la POO.

Retomando lo aprendido en la unidad 3 de CyCI, y profundizar en los principios de la programación: abstracción y encapsulamiento, elementos de una clase, instancia de objetos, constructores, uso de la consola para pruebas y depuración; en otras palabras, la formalización de los elementos del lenguaje de programación Java.

Inclusión del Lenguaje de Modelado Unificado UML de las herramientas de la ingeniería de software que combinan los aspectos del proceso de desarrollo (como fases definidas, técnicas, y prácticas) con otros componentes de desarrollo (como documentos, modelos, manuales, código fuente, etc.), llamado Proceso Racional Unificado RUP. se identificarían las fases de desarrollo de programas desde el punto de vista de modelado utilizando algunos diagramas de UML que permitirán la separación de elementos de lógica y de presentación en el diseño de programas. Se modelarán diversas clases bajo los principios de la POO, que se irán perfeccionando conforme avancen los contenidos de las siguientes unidades.

Estructuras de control de secuencia en Java

Aplicación de la programación orientada a objetos en la solución de

problemas en los que se tomen decisiones simples, combinadas y múltiples; el alumno utilizará estas estructuras para incorporar en sus clases la protección de los atributos mediante la validación previa a su asignación. Implementará programas que den solución a problemas que involucren estructuras de control iterativas, aplicará las estructuras de datos: arreglos unidimensionales, bidimensionales, ArrayList para el manejo de arreglos dinámicos, la estructura foreach (para el recorrido de colecciones), operador ternario, operadores de incremento (++, --) y acumulación (+=, -=, *=, /=, ...).

Herencia y Polimorfismo, constructores, colaboración y herencias de clases

Se profundiza en los conceptos de programación orientada a objetos como herencia y polimorfismo, constructores, interacción y comunicación entre clases, como temas que refuerzan las características de la programación orientada a objetos con Java. Permite entender la reutilización y especialización de código y las jerarquías de clases que se encontraran en un lenguaje de programación orientado a objetos, identificando elementos comunes y que necesitan modificarse o adicionarse a una clase. Además, el alumnado aprenden a integrar elementos de software desarrollados por otros programadores en proyectos propios y a generar elementos que pueden ser utilizados al administrar proyectos en equipos trabajando de forma colaborativa.

Interfaz gráfica de usuario

Desarrollo de objetos del paquete javax.swing para crear la Interfaz Gráfica que permitan mostrar la solución a problemas que engloben los aprendizajes adquiridos. Debe considerarse los enfoques basados en problemas o en proyectos.

Enfoque didáctico

Este programa de estudios de las asignaturas de Cibernética y Computación I y II busca su propósito con este enfoque didáctico, centrado fundamentalmente en la enseñanza-aprendizaje constructivista, para que guíe el proceso de enseñanza del profesorado y el de aprendizaje del alumnado.

La propuesta educativa busca promover acciones didácticas para formar en el alumnado una autogestión del conocimiento, la cual genera el desarrollo de hábitos que le permiten apropiarse de estrategias cognitivas y metacognitivas, para favorecer el dominio de las habilidades de pensamiento de orden superior.

El alumnado es el centro del proceso educativo, por lo que se busca su aprendizaje activo mediante la metodología basada en problemas o proyectos, cuidando de contar con espacios áulicos donde exista respeto, seguridad y empatía, así el estudiantado se sienta motivado de comentar sus dudas o sugerir ideas dentro del aula. El proceso de enseñanza aprendizaje de las asignaturas debe orientarse para que el alumno logre los aprendizajes y propósitos de la materia a través del desarrollo de los contenidos temáticos. Por la naturaleza de la materia, el uso de las Tecnologías de la Información y la Comunicación TIC, es intrínseca, por lo que se sugiere utilizar estrategias didácticas que incluyan el uso ético de herramientas para la búsqueda de información, comunicación y colaboración en línea, seguridad informática, plataformas educativas, simuladores, almacenamiento en la nube e inteligencia artificial; con el fin de que el alumnado adquiera, mejor y desarrolle habilidades tecnológicas aplicadas a la solución de problemas para todos los ámbitos de manera sustentable, transdisciplinaria con equidad e igualdad.

Utilizar el aprendizaje basado en problemas y el aprendizaje basado en proyectos ABP, para generar en el alumnado un pensamiento computacional, para que, partiendo, del análisis de un sistema construya el algoritmo de solución que representa el proceso o funcionamiento del mismo, el cual se puede llevar a la computadora convertido en una aplicación o programa informático, a través de un lenguaje de programación (Java). El profesorado debe tener la consigna de enfocar los problemas o proyectos en situaciones reales, proporcionando un contexto motivador que interese a el alumnado a lo largo de su desarrollo. Esta orientación asegura que el conocimiento adquirido sea aplicable en un contexto de su formación como cultura general, lo que garantiza su durabilidad y utilidad a largo plazo.

Los temas de sustentabilidad, desarrollo sostenible del medio en el que vivimos, equidad y perspectiva de género en la educación, son elementos importantes a

incluir durante el ABP, siempre buscando de manera conjunta, la toma de conciencia del daño que se está generando en nuestro ambiente a nivel físico, económico y social, por lo que es imprescindible fomentar la conciencia social al alumnado, del gran impacto que tendrán sus acciones, de ahí que sea deseable contar con la formación integral del alumnado, cuidando el equilibrio de los conocimientos su aplicación y las actitudes a través del trabajo en equipo de forma colaborativa, practicando la tolerancia, la solidaridad y la responsabilidad, la comunicación efectiva y afectiva, la toma de decisiones, la honestidad, creatividad para la solución a problemas a través de alternativas menos centradas en los datos de conocimiento y más ajustadas a los desafíos culturales y contextuales del alumnado. De igual forma, generando oportunidades para incluirlos, creando ambientes de aprendizaje que sean de utilidad para su diversidad en pensamientos, costumbres, hábitos, etc. Para lograr este tipo de trabajo, se debe de considerar por igual a los participantes, sin importar su condición social, ideología, raza o preferencia sexual.

Por lo anterior, el programa de estudios está diseñado para que el alumnado logre los aprendizajes para obtener los propósitos, y el profesorado planea sobre cómo explicar los contenidos temáticos, a través de actividades que le permitan al mismo alumnado apropiarse de esos conocimientos y aplicarlos, para esta integración, el profesorado debe considerar secuencias didácticas en sesiones de dos horas con la siguiente estructura: Apertura, momento didáctico que corresponde a la bienvenida y presentación del encuadre, si es el inicio del semestre incluirá la posibilidad de acordar las reglas de convivencia dentro del aula, la forma de organizar las actividades y de evaluarlas, el docente realizará una evaluación diagnóstica para valorar los conocimientos previos del alumnado, asimismo, que el profesorado evalúe y a la clase siguiente entregue los resultados de la misma, realizando la retroalimentación pertinente. Para las sesiones siguientes, la apertura tendrá la introducción al tema del día y/o retroalimentación a partir de una pregunta detonadora o el planteamiento de un problema para dar introducción al desarrollo, este momento didáctico, permite abordar los contenidos mediante los materiales o recursos didácticos para la apropiación y aplicación del conocimiento, por medio de la exposición, lecturas, indagación, análisis, reflexión, diálogo grupal, consenso de opiniones o diversas técnicas y recursos didácticos con empleo de las herramientas tecnológicas. Para finalizar con el cierre, momento didáctico que corresponde a la consolidación de los contenidos abordados, al refuerzo de aprendizajes, al resumen de lo realizado y a la revisión del objetivo, es decir, verificar si se logró el o los aprendizajes de los estudiantes. Dicha valoración se logra a partir de los distintos tipos de evaluación: diagnóstica, formativa y sumativa, donde el rol del evaluador puede ser: el mismo estudiante, entre pares, o el profesor en conjunto con el estudiante. En este mismo sentido, se sugiere que las secuencias didácticas tengan un enfoque gradual, es decir, es importante que en cada sesión se construya sobre el conocimiento previo, para que el alumnado tenga una

comprensión sólida y progresiva.

Como apoyo para la indagación de la información que se requiere para la resolución de problemas o proyectos, al final de cada unidad se sugiere fuentes de consulta, su importancia radica en el hecho que tanto el alumnado y el profesorado no invertirán tanto tiempo para localizar la información relacionada con la temática a estudiar. Al final de cada unidad aparece una lista con fuentes de consulta de libros y direcciones electrónicas separadas en básica y complementaria, para ambos. Debido a que las fuentes de consulta incluyen varios temas del programa estas se pueden repetir en las distintas unidades.

El uso de fuentes de consulta por parte del alumnado es indispensable, porque le ayuda a aprender, a desarrollar su capacidad de autoaprendizaje a aprender no sólo en su estancia en el Colegio, sino en toda su vida académica, las fuentes de consulta le permiten desarrollar su capacidad de autoaprendizaje. Sin embargo, la guía y supervisión del profesorado es fundamental.

Aunado a lo anterior, es conveniente que el profesorado considere al aprendizaje colaborativo en el desarrollo de sus estrategias ya que, fomenta el trabajo en equipo, la colaboración, el intercambio de ideas, la retroalimentación entre pares, el desarrollo de habilidades comunicativas y el fortalecimiento de valores y actitudes. Además, el trabajo colaborativo permite la consolidación de la identidad de un grupo, genera espacios de aprendizaje, fomenta la organización diferenciada para compartir, reflexionar y construir conocimientos, con ello, se invita a que los estudiantes sumen esfuerzos para conseguir un mismo objetivo en comunidad y para la comunidad.

En resumen, este enfoque didáctico pretende generar las condiciones para que el estudiantado y el profesorado construyan conocimiento que de fe del aprendizaje del alumnado y aplicación bajo una metodología activa donde el asuma una postura consiente, responsable, en comunidad para la comunidad y así enfrentar los desafíos que aquejan a su medio ambiente y a la sociedad.

logren los aprendizajes y propósitos a través del desarrollo de los contenidos temáticos, acompañados de estrategias de enseñanza-aprendizaje, que contienen actividades a realizar en cada sesión, las sugerencias se basan en usar estrategias que impulsen a los alumnos a hacer investigaciones relacionadas con el tema, se busca crear una motivación que despierte la curiosidad y el entusiasmo del alumno.

En la primera clase el profesor deberá presentarse, ofrecerle al alumno una visión general del programa de estudio, las formas de evaluación y la

información que permita al alumno conocer la manera en que se llevará a cabo el curso.

Se recomienda que el tiempo de cada clase se distribuya en **una Apertura**, consistente en una investigación o su equivalente dirigida por el profesor que le permita al alumno adquirir por su cuenta conocimiento; **un desarrollo**, que sirva para que el alumno comparta el conocimiento adquirido y, a la vez, adquiera nuevos conocimientos de parte de sus compañeros; **un cierre**, donde cada alumno, en forma individual o en grupo, proponga aplicaciones que le permita tanto al alumno como al profesor conocer si se logró el aprendizaje marcado. Todas las investigaciones y trabajos extra-clase deben estar delimitados para que los alumnos tengan claridad en el resultado a presentar, lo cual se puede hacer diseñando una rúbrica.

En las estrategias sugeridas se propone que sea el alumno el que investigue, el que comparta la información con sus compañeros y el que proponga aplicaciones, todo esto sólo se puede lograr bajo la dirección y supervisión del profesor, a través del diseño de una rúbrica que le informe al alumno lo que se espera de él a cada momento. Aunque en las estrategias, casi, no se menciona la figura del profesor, los alumnos no podrán realizar con éxito el trabajo si él no está presente. La intención de remarcar el trabajo que debe realizar el alumno es con la intención de lograr que el alumno aprenda a aprender, aprenda a ser y aprenda a hacer.

Al final de cada unidad se sugiere una bibliografía, su importancia radica en el hecho que el alumno y el profesor no invertirán tanto tiempo para localizar la información relacionada con la temática a estudiar. Al final de cada unidad aparece una lista con información de libros y direcciones electrónicas separadas en básica y complementaria, tanto para el alumno como para el profesor. Debido a que las referencias bibliográficas incluyen varios temas del programa estas se pueden repetir en las distintas unidades.

El uso de la bibliografía por parte del alumno es indispensable porque ella le ayudará a aprender a aprender no sólo en su estancia en el colegio, sino en toda su vida académica, la bibliografía lo volverá independiente, en este sentido, la dirección y supervisión del profesor es fundamental.

Contribución al perfil del egresado

Las asignaturas contribuyen fundamentalmente

con el perfil del egresado, mediante el desarrollo

de conocimientos, habilidades y destrezas intelectuales, la evolución de su formado pensamiento y la adquisición de valores, actitudes y normas, las asignaturas de Cibernética y Computación I y II propician entre otros:

- La valoración de las Tecnologías de la información y Comunicación su conocimiento y aplicación para la solución de problemas de forma sustentable, transdisciplinaria con equidad e igualdad en distintas problemáticas.
- La apropiación del conocimiento científico, tecnológico y humanístico.
- Las habilidades permiten aplicar los conocimientos para desarrollar propuestas de solución a sistemas que resuelven necesidades transdisciplinarias de forma sustentable a través de un dispositivo de cómputo
- El interés por la lectura y comprensión de textos diversos, particularmente científicos y de divulgación.
- El saber documentarse por su cuenta por medio de trabajos de investigación. Aprender a seleccionar y organizar información mediante trabajos de investigación para reflexionar sobre el conocimiento y ser autogestivo.
- La habilidad de realizar investigaciones documentales.
- El respeto a los derechos de autor de las obras consultadas
- La habilidad de proponer aplicaciones de la temática aprendida.
- La habilidad de plantear y resolver problemas aplicando la metodología aprendida. El desarrollo continuo de la habilidad comunicativa y reflexiva para presentar propuestas de aplicaciones de las temáticas sugeridas o de nuevas temáticas de interés personal, escolar, social o profesional.
- El conocimiento para desarrollar software de aplicación que permita resolver necesidades de sus distintas materias.
- El compartir sus conocimientos con sus compañeros. Aprender a trabajar con otros, compartir sus ideas y conocimientos y respetar las de otros; trabajando colaborativamente, buscando el bien común.

- El estar preparado para abordar nuevos conocimientos.

Las asignaturas de Cibernética y Computación I y II propician sustancialmente en el perfil del egresado, una formación integral de conocimientos, habilidades y actitudes, para su desarrollo cognitivo, mejora en sus destrezas en el análisis, diseño y solución a necesidades planteada para su aprendizaje basado en problemas y el fomento de valores que motivan a actitudes encaminadas a un bienestar y convivencia:

- En cuanto a los conocimientos, giran en torno a conceptos como la cibernética, sistemas, modelos, sistemas de numeración (decimal, binario, octal y hexadecimal) para que reconozca la importancia del carácter digital de una computadora, así como la forma de almacenar y comunicar la información en ella. Se continua con el aprendizaje de una metodología para que resuelva problemas y los sistematice, usando el lenguaje de programación Java, su sintaxis, semántica, operadores, tipos de datos, estructuras de control, clases, paquetes, entornos de trabajo en modo consola y gráfico.
- Fomenta el pensamiento analítico y científico para la toma de decisiones que le permitan desarrollar propuestas de solución a sistemas que resuelven necesidades transdisciplinarias de forma sustentable a través de algoritmos, simuladores y el lenguaje de programación Java desde dispositivo de cómputo.

- La habilidad de manejar las TIC, como herramientas para sus actividades académicas, entretenimiento, interacción social, entre otras. Para hacer uso de las TIC, estas propician habilidades métodos para obtener de forma segura y adecuada información desde el internet, que logre comunicarla y colaborar por redes sociales o mensajería, plataformas educativas de forma responsable; cuidado de su integridad.

Evaluación

La evaluación es un proceso continuo desde inicio a fin de cada asignatura, cuya finalidad es determinar en qué medida se lograron los propósitos y aprendizajes marcados en el programa de estudios.

En ella se valora la evolución cognitiva del alumnado, observando el avance en cada una de las actividades que realiza, donde conjuntamente alumnado y profesorado el cual debe ser visto como el constructor de su propio aprendizaje van construyendo los conocimientos necesarios para poderlos aplicar. Además de considerar las actitudes y valores dentro del salón de clases Todas las actividades serán analizadas y tomadas en cuenta por el profesorado.

Se plantean tres tipos momentos de evaluaciones: diagnóstica, formativa y sumativa con la posibilidad de utilizar los tipos de evaluación: autoevaluación, coevaluación y/o heteroevaluación.

La evaluación diagnóstica es la que se realiza al inicio del curso, para recoger datos personales y académicos en la situación de partida, su finalidad es que el profesor inicie el proceso educativo con un conocimiento real de las características de cada alumno o alumna, lo que permitirá diseñar estrategias didácticas y acoplar su práctica docente a la realidad del grupo y de sus singularidades individuales. Esta evaluación diagnóstica se puede contemplar al inicio de las unidades temáticas que requieran la concretización del conocimiento previo.

Se recomienda llevar a cabo una evaluación diagnóstica al inicio del curso. Puede Estas herramientas pueden ser un cuestionario escrito o simplemente en plantear un tema de discusión y dejar que el alumnado opine. Las actividades extra-clase también permiten diagnosticar si en general los alumnos están adquiriendo los aprendizajes a través del análisis de las respuestas dadas, e incluso la elaboración de los trabajos extra clase. Permitirá al profesorado darse

- Las actitudinales buscan en el perfil del alumnado: que sea consciente y responsable de su diálogo, pensamiento y acción, a través de su comunicación asertiva para relacionarse con su entorno, desde la libertad, empatía, y congruencia; prevaleciendo la equidad, igualdad, trabajo en y para la comunidad, pensando en un bienestar común y sustentable.

cuenta de las características personales en lo cognitivo, socioeconómico, actitudinal y valores al ser factores que influyen en su rendimiento académico.

La evaluación formativa es una valoración continua y sistemática, a través de la observación, retroalimentación de actividades o tareas para su análisis del aprendizaje grupal y de cada alumno y alumna, durante funcionamiento el curso, también sirve como estrategia para mejorar, ajustar y regular sobre la marcha los procesos educativos de cada alumno y del grupo.

Con respecto a la evaluación formativa la recomendación es permitir que el alumnado recopile información en forma individual o en equipos y que la discutan primero en sus equipos y después en plenaria, con la finalidad de que sea el mismo alumno el que obtenga conclusiones de forma consensuada y enriquecida por el grupo adquiera el aprendizaje. Es claro que el alumnado puede llevar a cabo las prácticas, proyectos e investigaciones con éxito siempre y cuando el profesorado le dé lineamientos detallados y que en cada momento lo esté asesorando, para ello existen varias herramientas, como: rúbricas, lista de verificación, reportes de realización de proyectos, participación individual, portafolio de evidencias, incluyendo la misma observación del Trabajo en equipo, respeto, responsabilidad y comportamiento de cada alumno y alumna.

La evaluación sumativa consiste en la recolección y ponderación de los datos obtenidos de las dos evaluaciones anteriores al finalizar el curso escolar. Sealimenta de las evaluaciones diagnóstica y formativa, sirve para contrastar los resultados iniciales con respecto a los propósitos y aprendizajes obtenidos durante y al final del curso. Permitiendo reflexionar sobre el trabajo del profesor, de cada alumno, y del grupo. Puede ser el punto de arranque de la evaluación diagnóstica del siguiente periodo escolar.

Concreción en la materia de los principios del Colegio

Aprender a aprender

Las asignaturas permiten que el alumno investigue la temática a estudiar a través de la identificación y selección de fuentes de información confiables, las cuales analizará y empleará en forma eficaz, situación que le permitirá desarrollar habilidades para adquirir nuevos conocimientos para adquirir y construir nuevos conocimientos de forma responsable y lo motive al autoaprendizaje por cuenta propia.

Aprender a hacer

Durante el desarrollo de ambas asignaturas se elaboran prácticas de muy diversa índole que se asimilan conocimientos que permitirán elaborar actividades de diversa índole para desarrollar habilidades procedimentales, a través del aprendizaje basado en problemas o proyectos, para que el alumnado aborde situaciones transdisciplinarias, La solución de problemas permite que el alumno aborde diferentes situaciones, las analice y observe las relaciones entre los elementos del problema y el mismo o en equipo de forma colaborativa e implementar con el empleo de las herramientas de desarrollo de software aplicaciones concretas. de apoyo, que proponga aplicaciones de la temática aprendida y que desarrolle proyectos interdisciplinarios.

Aprender a ser

Durante el desarrollo de actividades se acepta la diversidad del pensamiento y se fomenta el espíritu crítico a través de la discusión abierta, propiciando en todo momento la tolerancia, el respeto a él mismo y para con los demás. Los alumnos integran responsablemente en su manera de ser, hacer y pensar los conocimientos y habilidades que lo llevan a mejorar su propia interpretación del mundo y a adquirir mayor madurez intelectual. Aprender a ser. Durante el proceso de formación se fomentará el trabajo en y para la comunidad, la pertenencia e identidad, promoviendo el autoconocimiento de sus habilidades, destrezas, aptitudes y actitudes, a través de valores que le permitan ser una persona capaz de escuchar, dialogar de forma reflexiva, expresar sus experiencias asertivamente, propiciando en todo momento la empatía, tolerancia, respeto a él, los demás y a su medio, responsablemente, con libertad, justicia, honestidad; actitudes que le permitirán al alumnado una formación integral en su pensar, hacer y ser, para emplear los conocimientos y habilidades aprendidos, procurando la sustentabilidad, equidad e igualdad.

Propósitos generales de la materia

Al finalizar, el alumnado:

- Aplicará los conceptos de la cibernética a lo largo del curso para comprender y mejorar los sistemas mediante la tecnología, para de forma que se pueda aumentar el rendimiento y eficiencia de los procesos al menor costo económico y ecológico.
- La transdisciplina multidisciplinaria para entender la interacción entre los sistemas existentes en cualquier área, ya sea: biología, matemática, física, informática, medicina, etc.
- Utilizará algunos elementos de los sistemas de numeración y el álgebra de Boole para resolver problemas a través de la construcción de circuitos lógicos y, a criterio del profesor, los representará utilizando un simulador o un protoboard, y posteriormente se aplicará en la construcción de expresiones para la toma de decisiones en un lenguaje de programación de alto nivel.
- Obtendrá una metodología, con el apoyo de los algoritmos, diagramas de flujo, pseudocódigo y el lenguaje de programación Java para la solución de problemas a través de la computadora.
- Conocerá las características de la programación orientada a objetos, así como las del lenguaje de programación Java y su entorno de desarrollo para la solución de problemas de diferentes áreas del conocimiento.
- Utilizará las estructuras de control de secuencia: incondicionales, condicionales y de ciclo, a través del uso del lenguaje de programación orientado a objetos Java para resolver problemas.
- Implementará características de la programación orientada a objetos, como la herencia y el polimorfismo, la comunicación entre clases para la especialización y reutilización del código.
- Utilizará los objetos para instrumentar propiedades de la programación orientada a objetos para realizar interfaces gráficas retomando los conceptos vistos durante el curso.
- Realizará interfaces gráficas para reutilizar los objetos del paquete Swing, retomando los aprendizajes obtenidos durante el curso.
- A continuación, se presentan las unidades de cada asignatura y el tiempo requerido. Las horas asignadas a cada unidad dependen de la cantidad de los aprendizajes y contenidos que se deben alcanzar, a su grado de dificultad y su importancia.

Contenidos Temáticos Cibernética y Computación I

Unidad	Nombre de la unidad	Horas
1	La Cibernética	14
2	Circuitos lógicos, Representación de información en sistemas binario y algebra de Boole	18
3	Metodología de solución de problemas e introducción al lenguaje de programación Java	32
	Total	64

Contenidos Temáticos Cibernética y Computación II

Unidad	Nombre de la unidad	Horas
1	Lenguaje Fundamentos de programación orientada a objetos con Java	10
2	Estructuras de control de secuencia en Java	24
3	Herencia y polimorfismo, constructores, colaboración y herencia de clases	14
4	Interfaz gráfica de usuario	16
	Total	64

CIBERNÉTICA Y COMPUTACIÓN I

Unidad 1. La cibernética

En esta unidad como introducción a la asignatura de Cibernética y computación se pretende que el alumnado se adentre a través de los trabajos de investigación de distintos autores, así como la historia de los grandes descubrimientos alrededor del control y comunicación de los sistemas, para encontrar los elementos que lo conforman y que le permitirán analizarlos mediante su modelamiento, permitiendo dar paso a la era de las computadoras y que aún en nuestros días sigue en constante innovación.

Sin dejar de lado la forma en que permea tanto la cibernética como la computación en todas las disciplinas y que permite predecir y tomar decisiones en todas ellas de manera sustentable.

Propósito:

Al finalizar la unidad el **alumnado**:

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, **aplicando los principios de sustentabilidad**

Tiempo:
14 horas

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumno: Identifica a la cibernética como una ciencia transdisciplinaria</p>	<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 y la transdisciplina. 	<p>Apertura El alumnado, en equipo, investigan los temas relativos a la cibernética, concepto, antecedentes históricos, la relación y aplicación con otras disciplinas.</p> <p>Desarrollo</p> <ul style="list-style-type: none"> El profesorado presenta a los alumnos materiales didácticos donde se muestren aplicaciones de la cibernética, por ejemplo, la creación y diseño de prótesis, robótica, procesos productivos y administrativos, entre otros. El profesorado propicia un intercambio de ideas en donde los alumnos exponen su punto de vista relativo al material didáctico observado y los conceptos investigados. <p>Cierre El alumnado contesta un cuestionario de los conceptos relativos a la cibernética y sus aplicaciones.</p> <p>Extraclase El alumnado busca un video sobre la cibernética y sus aplicaciones y escriben una reseña.</p>

<p>Describe el trabajo científico sobre la cibernética de Norbert Wiener, Arturo Rosenblueth, Claude Shannon, entre otros.</p>	<p>Obra de distintos autores en trabajos científicos sobre la cibernética:</p> <ul style="list-style-type: none"> • Norbert Wiener. • Arturo Rosenblueth. • Claude Shannon. • Ada Lovelace • Jean Jennings Bartik 	<p>Apertura</p> <ul style="list-style-type: none"> • El alumnado, en equipo, investigan las aportaciones para el desarrollo de la cibernética de Norbert Wiener, Arturo Rosenblueth y Claude Shannon, Ada Lovelace y Jean Jennings Bartik, entre otros.
		<p>Desarrollo</p> <ul style="list-style-type: none"> • El profesorado propicia un intercambio de ideas en donde el alumnado expone sus puntos de vista relativo a las aportaciones a la cibernética de los autores y autoras investigados • El alumnado obtiene conclusiones. <p>Cierre</p> <p>El alumnado elabora una línea de tiempo con fechas, autores y aportaciones.</p> <p>Extraclase</p> <p>El alumnado investiga el concepto, clasificación y ejemplos de sistema.</p>
<p>Describe los componentes de un sistema.</p>	<p>Sistemas:</p> <ul style="list-style-type: none"> • Concepto. • Elementos. • Ambiente. • Clasificación. 	<p>Apertura</p> <p>En equipo, El alumnado, expone el tema relativo a los sistemas: concepto, elementos, ambiente y clasificación.</p> <p>Desarrollo</p> <p>El alumnado unifica conceptos y obtiene conclusiones.</p> <p>Cierre</p> <p>El alumnado elabora un mapa conceptual sobre la clasificación de los sistemas.</p> <p>Extraclase</p> <p>El alumnado investiga y hace un reporte sobre qué es un sistema de control, lazoabierto, lazo cerrado y retroalimentación.</p>
<p>Explica el funcionamiento de los sistemas y el control de estos</p>	<p>Funcionamiento de un sistema</p> <p>Sistemas de control:</p> <ul style="list-style-type: none"> • Lazo abierto. • Lazo cerrado. • Retroalimentación. 	<p>Apertura</p> <p>Los alumnos en equipo exponen su reporte.</p> <p>Desarrollo</p> <p>El alumnado, en plenaria, debaten y unifican criterios sobre los conceptos y obtienen conclusiones.</p> <p>Cierre</p> <p>En la plenaria el alumnado obtiene conclusiones y unifican criterios sobre los conceptos. El alumno contesta un cuestionario de los conceptos relativos a los sistemas.</p> <p>Extraclase</p> <p>El alumando, en equipo, investigan sobre el concepto de sistemas socio ecológicos, así como de riesgos o crisis socioambientales que estén ocurriendo</p>

		<p>en México y en el mundo. Así como el concepto de desarrollo sustentable.</p>
<p>Asocia el concepto de Sustentabilidad dentro de los sistemas</p>	<p>Sustentabilidad:</p> <ul style="list-style-type: none"> • Concepto • Ejemplos de Sistemas Socioambientales <ul style="list-style-type: none"> - Medio ambiente - Sistemas de Producción - Bienestar Social - Consumo Responsable • Crisis Socioambientales y prácticas encaminadas a mitigarlas 	<p>Desarrollo</p> <p>La Unión Europea sufrió de la peor ola de calor desde el renacimiento, en el verano del 2022, con un incremento sustancial en la cantidad de muertes por esa razón, en diversos países. Alemania, solamente, registró 3000 muertes adicionales durante la semana del 18 de julio, en comparación de los últimos 5 años.</p> <p>En promedio, de 6 a 8 millones de personas han muerto a nivel global debido a la contaminación.</p> <p>Lo anterior muestra claramente que necesitamos sustentabilidad ambiental, no solo para mejorar nuestra salud, sino para eliminar una amenaza en nuestras vidas</p> <p>El profesor propicia un intercambio de ideas en donde los alumnos exponen su punto de vista relativo al material didáctico observado y los conceptos investigados. Entre los puntos fundamentales a analizar son el origen de estas crisis socioambientales, consecuencias a largo plazo y la forma de mitigarlas.</p> <p>Cierre</p> <p>Se genera una lluvia de ideas para analizar las posibles soluciones sustentables a largo plazo para disminuir el calentamiento global y disminuir sus efectos, lo que significará que ha comprendido los problemas socio ecológicos. Debe hacerse énfasis en el uso racional de los recursos naturales y en las formas de lograr una sociedad equitativa.</p> <p>Extra-clase</p> <p>El alumnado investiga que es un modelo y sus tipos.</p>

<p>Identifica el concepto y la importancia del modelo.</p>	<p>Modelos:</p> <ul style="list-style-type: none"> • Concepto. • Tipos. <ul style="list-style-type: none"> - Naturales y artificiales. - Analógicos y digitales. - Matemáticos. - Conceptuales. • Relación. 	<p>Apertura El alumnado, en equipo, investigan el tema relativo a modelos: concepto, tipos y relación.</p> <p>Desarrollo El alumnado debate y unifica criterios sobre el concepto, tipos y relación de los modelos.</p> <p>Cierre En equipo, El alumnado modela, por medio de una representación gráfica, una maqueta, etcétera, un sistema sencillo, propuesto por el profesor o por iniciativa propia.</p> <p>Extraclase El alumnado propone un modelo de un sistema de alguna disciplina de su interés.</p>
<p>Explica las características de un modelo del sistema, las partes que lo conforman y su funcionamiento.</p>	<p>Elementos para modelar un sistema:</p> <ul style="list-style-type: none"> • Entrada y salida. • Proceso. <p>Desarrolla un modelo de un sistema</p>	<p>Apertura En equipos, el alumnado desarrolla la modelación de un sistema a través de una representación gráfica, una maqueta, entre otros.</p> <p>El alumnado analiza los elementos que lo conforman.</p> <p>Desarrollo El alumnado prepara una exposición del modelo del sistema para explicarlo en clase</p> <p>Cierre El alumnado explica el funcionamiento del modelo del sistema desarrollado.</p>

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra a la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes instrumentos:

Diagnóstica

A través del análisis de las respuestas dadas en los trabajos extraclase y de preguntas formuladas al inicio de cada clase.

Formativa

- **Listas de verificación:** para evaluar los trabajos extraclase.
- **Pruebas de ejecución:** ensamblado del modelo de un sistema.
- **Observación** de las actividades realizadas en el salón de clases
- **Rúbrica:** para evaluar las investigaciones realizadas.

- **Exámenes.**
- **Cuestionarios.**

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios.

- **Nota:** La ponderación de cada actividad será asignada por el profesor

Referencias

Para el alumnado

Básica

CCH Portal académico (2017). Cibernética y computación I.
<https://e1.portalacademico.cch.unam.mx/alumno/cibernetica1/unidad1/laCibernetica/introduccion>

Maturana, H. y von Foerster, H. (). Historia de la Cibernética.

<https://www.asc-cybernetics.org/foundations/timeline.htm>

TV UNAM (2023) Historia gloriosa y casi secreta de la cibernética en México, con Adrián Santuario.

<https://youtu.be/rftoHABQQP4?si=SS8aLLb94u1xm5ff>

Complementaria

Wiener, N. (1988). *Cibernética y sociedad*. Buenos Aires: Editorial Sudamericana.

Infoamerica (1988) *Arturo Rosenblueth y Norbert Wiener: dos científicos en la historiografía de la educación contemporánea*,
https://www.infoamerica.org/documentos_pdf/wiener1.pdf

Infoamerica (2013). *Claude Elwood Shannon (1916-2001)*.
<http://www.infoamerica.org/teoria/shannon1.htm>.

Infoamerica (s/a) *Norbert Wiener y el origen de la cibernética desde 1939*.
https://www.infoamerica.org/documentos_pdf/wiener2.pdf

Para el profesor

Básica

Distefano, J. *et al.* (1975). *Retroalimentación y sistemas de control*. México: McGraw-Hill.

Gifreu, A. (2014). *Pioneros de la tecnología digital. Ideas visionarias del mundo tecnológico actual*. España: UOC

Jramoi, V. *et al.* (1971). *Introducción e historia de la cibernética*. México: Grijalbo.

Rosenblueth, A. (1981). *Mente y cerebro. Una filosofía de la ciencia*. México: Siglo xxi.

Wiener, N. (1988). *Cibernética y sociedad*. Buenos Aires: Editorial Sudamericana.

Complementaria

Ogata, K. (1980). *Ingeniería de control moderna*. México: Prentice Hall.

Raymound, R. (1984). *La cibernética y el origen de la información*. México: Fondo de Cultura Económica.

CIBERNÉTICA Y COMPUTACIÓN I

Unidad 2. Representación de información en sistema binario y Álgebra de Boole

En esta unidad se busca que el alumnado comprenda cual es lenguaje interno de la computadora, que todo lo que realiza son operaciones o cálculos en nanosegundos y que esos datos están representados por bits, la unidad mínima de información, que son almacenados en espacios de memoria con direcciones en hexadecimal y que hasta este momento la arquitectura de la computadora sigue manufacturándose a través de circuitos lógicos diseñados con la teoría del algebra de Boole. En otras palabras, aprenderá a sistematizar problemas de diferentes disciplinas mediante circuitos lógicos.

<p>Propósito:</p> <p>Al finalizar la unidad el alumnado: realizará operaciones aritméticas básicas y conversiones de base entre los sistemas de numeración: binario, octal y hexadecimal, para comprender la representación de la información en la memoria de una computadora, así como sus diferentes aplicaciones en el campo de la computación.</p> <p>Al finalizar la unidad el alumno utilizará el álgebra de Boole para diseñar, construir o simular circuitos lógicos utilizando una protoboard o un simulador, así como una introducción a la toma de decisiones en estructuras de control en un lenguaje de programación.</p>	<p>Tiempo: 14 horas</p>
--	------------------------------------

Aprendizajes	Temática a	Estrategias sugeridas
<p>El alumno: Convierte números entre los sistemas de numeración binario, octal, decimal y hexadecimal.</p> <p>El alumnado Comprende las características de los sistemas de numeración decimal y binario, así como sus equivalencias en octal y hexadecimal, para poder representar información en la computadora como tipos de datos primitivos en un lenguaje de programación.</p>	<p>Sistemas de numeración.</p> <ul style="list-style-type: none"> • Binario, octal, decimal y hexadecimal. • Conversiones numéricas entre los sistemas: binario, octal, decimal y hexadecimal. • Representaciones octal y hexadecimal como simplificaciones del sistema binario. • Tipos de datos primitivos: enteros con y sin signo, números de punto flotante, caracteres y booleanos. • Tipos de datos con representaciones en memoria, rango y precisión. 	<p>Apertura El alumno investiga la importancia, utilidad y características de los sistemas de numeración: binario, octal, decimal y hexadecimal en el campo de la computación. En plenaria, los alumnos obtienen conclusiones con la supervisión del profesor.</p> <p>Desarrollo Los alumnos, supervisados por el profesor, realizan conversiones numéricas entre los distintos sistemas de numeración investigados.</p> <p>Cierre Los alumnos resuelven ejercicios de conversión entre los distintos sistemas de numeración y comprueban sus resultados utilizando la calculadora.</p> <p>Extraclase Los alumnos investigan cómo se realizan las operaciones aritméticas básicas en el sistema de numeración binario.</p>

		<p>Apertura: Docente: inicia el tema con preguntas reflexivas o la descripción de hechos interesantes asociados con los sistemas de numeración.</p> <p>Docente: Explica la importancia de los sistemas de numeración en la vida cotidiana y en la historia de las matemáticas, así como su aplicación en el campo de la computación.</p> <p>Docente: Revisar algunos videos para comparar los sistemas de numeración acerca de cuáles son posicionales y la importancia del uso del cero y símbolos únicos para los dígitos del sistema.1</p> <p>Desarrollo: Estudiantes: En equipos o plenaria realiza una tabla indicando cuales de los sistemas de numeración se consideran posicionales y cuáles son sus características.</p> <p>Estudiantes: Realiza conversiones de enteros sin signo desde decimal a binario, octal y hexadecimal, y viceversa.</p> <p>Cierre: Docente y estudiantes: Ubican los sistema decimal y binario como sistemas de numeración posicionales.</p> <p>Apertura: Se realiza la comprobación de las conversiones de la actividad extraclase previa.</p> <p>Desarrollo: El profesor pregunta qué es un número entero, un número racional, un número irracional. Los alumnos deben investigar en Internet si es que no saben las respuestas.</p> <p>Estudiantes: reflexionan si el número pi se puede representar con todos sus decimales en las estructuras de almacenamiento de la computadora.</p> <p>Estudiantes: debe dar relaciones como $3 < 4$ y determinar qué valor lógico asignarle para que vea la necesidad de asignarle un valor de verdadero o falso para una relación.</p> <p>Estudiantes: investiga la representación de un dato tipo entero, punto flotante, carácter y booleano, en la memoria.</p> <p>Cierre: Estudiantes: argumenta, para una cantidad fija de bits para almacenar información, que rango de números enteros puede</p>
--	--	--

		<p>representarse, qué precisión se puede tener para números con punto flotante.</p> <p>Docente y estudiantes: realizan un cuadro con la clasificación de los tipos de datos primitivos.</p> <p>Extraclase:</p> <p>Estudiantes: investigan la forma de realizar las operaciones de suma, resta, multiplicación y división en binario.</p>
<p>Realiza operaciones aritméticas con el sistema de numeración binario.</p>	<p>Aritmética del sistema de numeración binario.</p> <ul style="list-style-type: none"> Operaciones de adición, sustracción, multiplicación y división. 	<p>Apertura</p> <p>En plenaria los alumnos obtienen conclusiones de la investigación extraclase, posteriormente el profesor explica el desarrollo de las operaciones aritméticas con números binarios.</p> <p>Desarrollo</p> <p>El alumno, orientado por el profesor, realiza operaciones aritméticas básicas en el sistema de numeración binario.</p> <p>Cierre</p> <p>Los alumnos resuelven ejercicios, propuestos por el profesor, de operaciones aritméticas básicas en el sistema de numeración binario y comprueban sus resultados utilizando la calculadora.</p> <p>Extraclase</p> <p>Los alumnos investigan las operaciones booleanas básicas de conjunción, disyunción y negación y entregan un reporte</p> <p>Apertura:</p> <p>Realice una revisión de las conversiones entre decimal y binario: con y sin signo, ubique que las operaciones se limitan a un espacio en memoria (8, 16, 32 o 64 bits).</p> <p>https://portalacademico.cch.unam.mx/cibernetica1/sistemas-de-numeracion</p> <p>Desarrollo:</p> <p>Docente y estudiante: Ubican que la resta puede expresarse como la suma de un número positivo con uno negativo, la multiplicación como varias sumas, y la división como varias restas. Realice diversos ejemplos de sumas, restas y multiplicaciones.</p> <p>Cierre:</p> <p>Realizar diversos ejercicios de repaso</p>

Reconoce los elementos del álgebra de Boole para alternar entre las distintas representaciones y disciplinas que ocupan variables booleanas	Álgebra de Boole Características de una variable booleana. Operaciones básicas del álgebra de Boole. Concepto de función booleana. Representación de funciones booleanas como expresiones algebraicas. Representa funciones booleanas mediante tablas de verdad.	
Obtiene funciones booleanas a partir de sistemas digitales y los reducirá a una expresión equivalente	Equivalencia de las tablas de verdad, funciones booleanas en su forma algebraica y su representación en circuito lógico.	Apertura: Docente: Recapitula el tema con una breve revisión acerca de lo que los estudiantes saben sobre las operaciones booleanas (AND, OR, NOT, XOR). Desarrollo: El profesor realiza la explicación teórica acerca de las tablas de verdad y cómo estas representan las salidas de funciones booleanas para todas las combinaciones posibles de entradas, además, muestra cómo la tabla de verdad y la forma algebraica están relacionadas. Estudiantes: Desarrollan ejemplos prácticos de funciones booleanas mediante la creación de tablas de verdad y analiza cómo estas tablas pueden ser utilizadas para derivar funciones; examinan ejemplos concretos de cómo construir funciones booleanas a partir de las tablas de verdad correspondiente. Docente: explica los símbolos lógicos utilizados para representar las diferentes compuertas y su relación con las funciones booleanas. Docente y estudiantes: Relacionan tablas de verdad y

		<p>expresiones algebraicas y el diseño del circuito lógico.</p> <p>Cierre:</p> <p>Estudiantes: resuelven problemas que involucren la conversión entre tablas de verdad, expresiones algebraicas y circuitos lógicos.</p> <p>Fomenta el trabajo colaborativo para la solución de problemas utilizando plataformas tecnológicas o herramientas convencionales no tecnológicas.</p> <p>Extraclase:</p> <p>Estudiantes: Realizan otros ejercicios para reforzar como tarea. Revisan una investigación sobre la metodología de solución de problemas aplicado a circuitos a partir del modelo de caja negra.</p>
	<p>Metodología de solución de problemas a partir del modelo de caja negra.</p>	<p>Apertura:</p> <p>Docente y estudiantes: revisan los resultados de la investigación, recuerdan que las equivalencias de las funciones booleanas (tablas de verdad, expresión algebraica y circuito) y concluyen que los circuitos equivalentes más reducidos tienen ventajas (simplicidad, menor uso de materiales, menor consumo de energía, etc.)</p> <p>Desarrollo:</p> <p>Docente y estudiantes: Desarrollan un ejemplo que no requiera simplificación, pero que realice el proceso completo. Por ejemplo, una compuerta OR que "pueda verse a simple vista" (planteamiento, análisis de la caja negra, tabla de verdad, obtención de una expresión algebraica equivalente, obtención de un circuito e implementación en un circuito digital, pruebas, correcciones y documentación)</p> <p>Cierre:</p> <p>Docente y estudiantes: Concluyen que la aplicación metodológica es aplicable a diversos sistemas y a la construcción de circuitos digitales.</p> <p>Extraclase:</p> <p>Estudiantes: los teoremas y postulados del Álgebra de Boole, identificar los comportamientos de las compuertas AND (sólo enciende una combinación) y OR (encienden varias</p>

		combinaciones).
	Funciones booleanas a partir de la tabla de verdad empleando suma de productos (mintérminos).	<p>Apertura: Docente: introduce los "mintérminos" como productos que involucran a todas las variables.</p> <p>Desarrollo: Docente y alumnos: plantean problemas como suma de productos.</p> <p>Cierre: Docente: Plantea un problema de "mediana dificultad" como un display de siete segmentos para el despliegue de información (Sistema de numeración octal o hexadecimal) desarrollarlo con el grupo.</p> <p>Extraclase: Estudiantes: Identifican patrones a partir de la distributividad para poder simplificar suma de mintérminos.</p>
	Simplificación de funciones booleanas equivalentes a partir de postulados y teoremas básicos a partir de la suma de productos de una tabla de verdad.	<p>Apertura: Docentes: plantea que existen diversas formas de aplicar la distributividad (factorización).</p> <p>Desarrollo: Docente: Identifica patrones de reducción de circuitos, identifica términos que puede "clonarse" para agruparlos de diversas formas mediante la aplicación de postulados y teoremas básicos.</p>

		<p>Cierre: Docente y estudiantes: aplican los patrones de reducción de circuitos para obtener un equivalente simplificado.</p> <p>Extraclase: Estudiantes: completan el ejercicio aplicando estos patrones.</p>
	<p>Aplicación de mapas de Karnaugh para la obtención de circuitos lógicos equivalentes.</p>	<p>Apertura: Docente y estudiantes: Recapitulan y revisan las soluciones obtenidas.</p> <p>Desarrollo: Docente: Se plantea el uso de mapas de Karnaugh como una forma de identificar los patrones de simplificación de circuitos digitales.</p> <p>Cierre: Estudiantes: completan la simplificación del ejercicio.</p> <p>Extraclase: Revisa los materiales del portal académico para la construcción de circuitos digitales.</p>

<p>Construye tablas de verdad de funciones booleanas.</p>	<p>Elementos del álgebra de Boole:</p> <ul style="list-style-type: none"> • Variable booleana. • Operaciones básicas: conjunción, disyunción y negación. <p>Función booleana:</p> <ul style="list-style-type: none"> • Concepto. • Expresiones booleanas. <p>Tablas de verdad.</p>	<p>Apertura El profesor muestra ejemplos de proposiciones que involucren las operaciones de conjunción, disyunción y negación, para que el alumno identifique las variables y operaciones booleanas.</p> <p>Desarrollo Con base en las proposiciones propuestas, el alumno identifica el número de variables que emplea, las asocia como entradas y define el concepto de función booleana.</p> <p>Cierre El profesor supervisa a los alumnos para que construyan la tabla de verdad, de una función dada, tomando en cuenta que el número de variables define la cantidad de combinaciones de entradas a analizar en la tabla. Se recomienda utilizar una columna por operación para facilitar la comprensión de la construcción de la función booleana.</p> <p>Extraclase El profesor proporciona una serie de ejercicios relativos a funciones booleanas para que los alumnos obtengan las tablas de verdad.</p>
<p>Simplifica funciones booleanas utilizando postulados y teoremas básicos.</p>	<p>Simplificación de funciones booleanas.</p> <ul style="list-style-type: none"> • Postulados. • Teoremas básicos. 	<p>Apertura Los alumnos investigan y escriben una lista de postulados y teoremas básicos del álgebra de Boole.</p> <p>El profesor, en plenaria, explica el significado y el uso de los postulados y teoremas básicos.</p>

		<p>Desarrollo En equipo, utilizando los postulados y teoremas básicos, los alumnos simplifican algunas funciones booleanas.</p> <p>Construyen y comparan las tablas de verdad de la función booleana original con la simplificada.</p> <p>Cierre En plenaria, cada equipo, explica cómo simplificaron las funciones booleanas.</p> <p>Extraclase Los alumnos, en equipo, investiga los conceptos de interruptor, circuito eléctrico, compuertas lógicas y circuito lógico</p>
<ul style="list-style-type: none"> Describe los conceptos de interruptor, circuito eléctrico, compuerta lógica y circuito lógico. Aprende a utilizar el protoboard o un simulador 	<p>Interruptor y circuito eléctrico. Compuertas y circuitos lógicos:</p> <ul style="list-style-type: none"> Compuerta: <i>And, Or y Not.</i> <p>Uso del protoboard o un simulador.</p>	<p>Apertura El profesor solicita a los equipos que expliquen los conceptos de interruptor, circuito eléctrico, compuertas lógicas y circuito lógico para, después, obtener conclusiones.</p> <p>Los alumnos, en equipo, elaboran el diagrama y construyen y simulan un interruptor y un circuito eléctrico que sirva, por ejemplo, para encender y apagar las lámparas de su casa.</p> <p>El profesor explica el funcionamiento de los circuitos integrados y el uso del protoboard o de un simulador.</p> <p>Desarrollo En equipo, los alumnos relacionan la tabla de verdad AND, OR y NOT, con los diagramas de las compuertas lógicas y construyen y simulan los circuitos lógicos correspondientes.</p> <p>Cierre En equipo, los alumnos representan por medio de un diagrama lógico una función booleana y viceversa.</p> <p>Extraclase El alumno construye o simula funciones booleanas del tipo: $AB', ABC', A'+B$</p>

Construye la función booleana a partir de la tabla de verdad, empleando suma de productos.

Describe los conceptos de interruptor, circuito eléctrico, compuerta lógica y circuito lógico.

Aprende a utilizar el protoboard o un simulador

Construye la función booleana a partir de la tabla de verdad, empleando suma de productos

Construye un semisumador.

Construye un sumador completo

Diseña circuitos lógicos a partir de un problema cotidiano usando la metodología aprendida

Aplica una metodología para la implementación de circuitos lógicos

Obtención de la función booleana.

- Suma de productos.

Elementos de implementación de circuitos digitales:

- fuentes de poder,
- interruptor,
- circuito eléctrico,
- compuerta lógica,
- cables, resistencias y LEDs,
- circuito lógico.

Apertura

Los alumnos investigan el concepto de suma de productos (minterminos) para la obtención de una función booleana a partir de la tabla de verdad correspondiente.

Desarrollo

El profesor muestra tablas de verdad y los alumnos, utilizando el método de suma de productos encuentran la función booleana correspondiente.

Cierre

Los alumnos, en equipo, simplifican las funciones booleanas encontradas, emplean un protoboard o un simulador para implementar la función booleana.

Extraclase

El alumno investiga qué es un semisumador.

Apertura:

Docente y estudiantes: identifican en la tableta protoboard como se implementan los distintos elementos del diagrama lógico.

Desarrollo:

Docente y estudiantes: Mediante un simulador o físicamente, de forma gradual implementa:

- interruptores y verifica su correcta polarización, con el uso de un LED:

- comprueba el funcionamiento de compuertas TTL básicas (AND, OR, XOR de dos entradas) y ve que hay varias en cada una, sustituye el circuito integrado para ver cómo funciona cada una.

Cierre:

Estudiantes: implementa un circuito que involucre dos circuitos integrados interconectados incluyendo interruptores y LEDs .
Comparan un simulador del circuito digital (p.ej. iCircuit) verificar la salida con la tableta protoboar o un simulador (p.ej. TinkerCAD).

Extraclase:

Estudiantes: Implementan otros ejemplos, pueden basarse en los provistos en el Portal Académico.

	<p>Simulación en tabletas <i>protoboard</i> (físicas y simuladores) para implementar un circuito digital.</p> <ul style="list-style-type: none"> • Planteamiento del problema (caja negra). • Obtención de la tabla de verdad. • Obtención de la función booleana. • Simplificación. 	<p>Apertura: El docente a través de un ejemplo y utilizando un simulador (TinkerCAD) explica cómo se usa la Tableta protoboard para implementar circuitos digitales.</p> <p>Desarrollo: Los alumnos elaboran un diagrama de un circuito digital y lo implementan utilizando CI en la tableta de experimentación protoboard y obtienen su tabla de verdad.</p> <p>Los alumnos a partir de diagrama del circuito digital obtienen su función, su tabla de verdad y verifican su comportamiento.</p> <p>Cierre: Los alumnos de manera grupal muestran sus trabajos.</p> <p>Extraclase: Revisa los materiales del portal académico de Tableta protoboard</p>
	<p>Construcción o simulación del circuito lógico. Implementa circuitos lógicos en tabletas protoboard como semisumadores, sumadores completos,</p>	<p>Apertura: Los alumnos, con ayuda del docente, plantean el problema de sumar dos dígitos binarios (semisumador) enfatizando el concepto de suma y acarreo y obtienen la tabla de verdad, la función booleana y el diagrama del diseño lógico.</p> <p>Desarrollo: En equipo los alumnos implementan el circuito lógico del semisumador, empleando un protoboard o un simulador y comprueban su funcionamiento.</p> <p>Cierre: Los alumnos exponen el circuito lógico construido.</p> <p>Extraclase: Los alumnos investigan y plantean el problema del sumador completo.</p>

	<p>Circuitos complejos a partir de circuitos simples (semisumador a sumador completo o restador; sumador y restador de un bit a sumador o restador de dos o más bits, etc.)</p>	<p>Apertura: Estudiantes muestran la solución obtenida de su investigación o del desarrollo de su solución.</p> <p>Desarrollo: Estudiantes: Implementan la solución en una tableta protoboard o en un simulador (TinkerCAD), realizan pruebas y documentan todo el proceso.</p> <p>Cierre: Estudiantes: exponen la solución y sus conclusiones. Docente: plantea un circuito con Arduino (TinkerCAD) que utilice un sumador de dos bits haciendo énfasis en que la salida de uno es la entrada del siguiente.</p> <p>Extraclase: Estudiantes: analizan el circuito planteado</p>
<p>Construye un sumador completo</p>	<p>Sumador completo:</p> <ul style="list-style-type: none"> • Planteamiento del problema (caja negra). • Obtención de la tabla de verdad. • Obtención de la función booleana. • Simplificación. • Diagrama del diseño lógico. • Construcción o simulación del circuito lógico. 	<p>Apertura Los alumnos, con ayuda del profesor, plantean el problema de sumar tres dígitos binarios (dos bits de entrada, más el acarreo) enfatizando el concepto de suma y acarreo de salida en el sumador completo, obtienen la tabla de verdad, la función booleana y el diagrama del diseño lógico.</p> <p>Desarrollo En equipo, los alumnos implementan el circuito lógico del sumador completo, empleando un protoboard o un simulador comprueban su funcionamiento.</p> <p>Cierre Los alumnos exponen el circuito lógico construido.</p> <p>Extraclase Los alumnos plantean y diseñan un problema de su interés que se pueda resolver empleando circuitos lógicos.</p>

<p>Diseña circuitos lógicos a partir de un problema cotidiano usando la metodología aprendida.</p>	<p>Construcción del circuito lógico.</p>	<p>Apertura En equipo los alumnos retoman el problema diseñado en extra-clase y lo implementan empleando un protoboard o un simulador y comprueban su funcionamiento.</p> <p>Desarrollo Los alumnos, en equipo, presentan sus trabajos explicando los detalles.</p> <p>Cierre En plenaria, el profesor, pide la opinión de los alumnos sobre los proyectos presentados, resaltando si éstos cumplen con todas las características del diseño de circuitos lógicos.</p> <p>Extraclase Los alumnos resuelven un cuestionario sobre los temas correspondientes a la unidad.</p>
--	---	--

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

La unidad 3 da continuidad a una programación de bajo nivel con unos y ceros que entiende la computadora, para transitar a un lenguaje de alto nivel muy similar al lenguaje en inglés. El alumnado podrá apropiarse de una metodología que lo conduzca a la solución de un problema mediante el desarrollo de un programa informático, utilizando el lenguaje de alto nivel Java.

<p>Propósito:</p> <p>Al finalizar la unidad el alumnado:</p> <p>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 de un problema utilizando la computadora.</p>	<p>Tiempo: 32 horas</p>
---	------------------------------------

Aprendizaje	Temática	Estrategias sugeridas
<p>El alumno:</p> <ul style="list-style-type: none"> Define el concepto de problema. Identifica los elementos de un problema. Identifica los diferentes tipos de problemas: determinísticos, probabilísticos, secuenciales, condicionales y cíclicos. 	<p>Definiciones y conceptos generales de un problema.</p> <p>Elementos y relaciones del problema:</p> <ul style="list-style-type: none"> Entrada. Proceso. Salida. <p>Tipos de problema:</p> <ul style="list-style-type: none"> Determinísticos. Probabilísticos. Secuenciales. Condicionales. Ciclos 	<p>Apertura El alumno orientado por el profesor propone problemas de la vida cotidiana y sus posibles soluciones.</p> <p>Desarrollo El alumno, en equipo, investiga la definición y algunos elementos de un problema, así como sus relaciones.</p> <p>Cierre Los alumnos contestan un cuestionario sobre la temática, resaltando los elementos que tienen en común los problemas.</p> <p>Extraclase Los alumnos plantean un problema resaltando la entrada, proceso y salida.</p> <p>Apertura En equipos, los alumnos revisan los objetos de aprendizaje relacionados a la temática que se encuentra en el portal Académico del CCH y realizan las actividades propuestas. El profesor posteriormente presentar ejemplos de problemas abordados, los alumnos determinan el tipo de problema e identifican su entrada y salida mediante Diagramas EPS.</p> <p>Desarrollo Los alumnos en equipos proponen un tipo de problema de su interés.</p>

		<p>identifican las causas, insumos o entradas; operaciones o tratamiento efectuados; así como los efectos, productos o salidas para cada problema, el profesor resuelve dudas y realiza sugerencias.</p> <p>Cierre Los alumnos en equipos exponen su actividad y en plenaria se debate, así como unifican criterios sobre los conceptos abordados y se obtienen conclusiones.</p> <p>Extraclase El profesor propone una serie de problemas y el alumno selecciona solo uno. Determinando el tipo de problema, así como menciona su entrada y salida mediante Diagramas EPS</p>
<p>Describe la diferencia entre problemas determinísticos, probabi-lísticos, secuenciales y condicio-nales.</p>	<p>Tipos de problema:</p> <ul style="list-style-type: none"> • Determinísticos. • Probabilísticos. • Secuenciales. • Condicionales. 	<p>Apertura En equipo los alumnos investigan la temática y hacen un resumen.</p> <p>Desarrollo El profesor les presenta ejemplos de los cuatro tipos de problemas y los alumnosdeterminan el tipo de problema, mencionando su entrada y salida.</p> <p>Cierre En equipo, los alumnos plantean un problema de cada tipo.</p> <p>Extraclase Loa alumnos contestan un cuestionario que destaque las diferencias entre loscuatro tipos de problemas.</p>

<p>Conoce las etapas de la metodología de solución de problemas.</p>	<p>Etapas de la metodología de solución de problemas:</p> <ul style="list-style-type: none"> • Planteamiento del problema. • Análisis del problema. • Diseño de la solución del problema: • Elaboración de algoritmos. • Representación del algoritmo através de pseudocódigo y diagrama de flujo. • Prueba de escritorio. 	<p>Apertura En equipo, los alumnos, investigan el concepto y las etapas de la metodología de solución de problemas.</p> <p>Desarrollo En plenaria, los equipos junto con el profesor, formulan un problema y lo re-suelven a través de todas las etapas de la metodología de solución de problemas.</p> <p>Cierre El profesor realiza un resumen de las etapas de la metodología de solución de problemas.</p> <p>Extraclase Los alumnos proponen un problema, desarrollan una posible solución y entregan una reseña.</p> <p>Inicio El profesor con ayuda de una presentación digital expone el concepto y las etapas de la metodología de solución de problemas.</p> <p>Desarrollo El profesor formula un problema secuencial y lo resuelve en conjunto con los alumnos a través de todas las etapas de la metodología de solución de problemas, propiciando un intercambio de ideas en donde los alumnos exponen su punto de vista relativo a la solución.</p> <p>Cierre El profesor realiza un resumen de las etapas de la metodología de solución de problemas, en plenaria los alumnos unifican conceptos y obtienen conclusiones.</p>
--	---	---

<p>Conoce el concepto de algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas secuenciales.</p>	<p>Concepto de algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Elaboración de algoritmos secuenciales.</p> <p>Representación de algoritmos secuenciales a través de diagramas de flujo y pseudocódigo</p>	<p>Apertura Los alumnos, investigan el concepto de algoritmo, diagrama de flujo y pseudo-código.</p> <p>Desarrollo En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas secuenciales, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, pseudocódigo y diagrama de flujo.</p> <p>Cierre En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema secuencial propuesto por el profesor.</p> <p>Extraclase Los alumnos proponen y construyen la solución de un problema secuencial. En-tregan un reporte que incluya las etapas de la metodología de solución de problemas.</p> <p>Apertura: El profesor comienza la clase planteando la importancia de comprender los conceptos fundamentales en programación, los alumnos con ayuda de una computadora y utilizando fuentes confiables definirán los siguientes conceptos: Algoritmo, Diagrama de flujo, Pseudocódigo, Prueba de escritorio.</p> <p>Desarrollo: El profesor propone un ejercicio y en conjunto con el grupo se elabora el algoritmo, el diagrama de flujo, el pseudocódigo y la prueba de escritorio, posteriormente el profesor organiza a los alumnos por equipos y les asigna diferentes problemas, cada equipo deberá desarrollar el algoritmo, diagrama de flujo, pseudocódigo y la prueba de escritorio al problema asignado.</p> <p>Cierre: Cada equipo comparte la solución propuesta de su problema y el diagrama de flujo, para finalizar el profesor destaca la importancia de estos conceptos como base para el desarrollo de habilidades de programación y resolución de problemas en computación.</p>
---	---	--

<p>Construye expresiones lógicas utilizando operadores relacionales y lógicos.</p>	<p>Expresiones y operadores relacionales y lógicos:</p> <ul style="list-style-type: none"> • Operadores relacionales. • Operadores lógicos. • Jerarquía de operadores lógicos. • Evaluación de expresiones lógicas. 	<p>Apertura En equipo los alumnos investigan la temática.</p> <p>Desarrollo El profesor organiza a los alumnos en equipos para que propongan ejemplos y expongan los temas.</p> <p>Cierre En equipo, los alumnos, resuelven ejercicios sobre expresiones lógicas.</p> <p>Extraclase Los alumnos proponen expresiones lógicas, para evaluarlas y analizar porque unas son verdaderas y otras falsas.</p> <p>Inicio: Plantear problemas que responden a preguntas cuyas respuestas son falso o verdadero. Por ejemplo: realizar una búsqueda en internet de trámites como la credencial del INE.</p> <p>Desarrollo Plantear la necesidad de operadores lógicos.</p> <p>Inicio: Identificar en problemas matemáticos rangos de valores, plantear algunos ejemplos donde ubique que los valores eventualmente no satisfacen criterios de valores de entrada (excepciones).</p> <p>Desarrollo: Plantear la necesidad del uso de operadores relacionales</p> <p>Cierre: Plantear ejemplos que combinen ambos operadores.</p>
--	--	---

Analiza el resultado de expresiones aritméticas utilizando la jerarquía de las operaciones.

Construye expresiones aritméticas utilizando la jerarquía de las operaciones y los operadores apropiados en el lenguaje de programación Java.

Expresiones y operadores aritméticos:

- Asignación.
- Operadores aritméticos.
- Jerarquía de operadores aritméticos.
- Evaluación de expresiones aritméticas.
- Funciones aritméticas básicas de la clase Math.

Apertura

En equipo los alumnos investigan los conceptos correspondientes a expresiones y operadores aritméticos.

Desarrollo

Cada equipo, en plenaria, expone un subtema de la temática y el grupo obtiene conclusiones.

Cierre

Los alumnos, resuelven ejercicios sobre expresiones aritméticas y comprueban el resultado a través del uso de la calculadora.

Extraclase

Los alumnos encuentran el resultado de una lista de ejercicios sobre la precedencia de las operaciones.

Apertura:

El profesor presenta los diferentes operadores aritméticos, así como la jerarquía de operadores, el uso correcto de paréntesis y las funciones básicas de la biblioteca Math.

Desarrollo:

El profesor organiza equipos, a cada equipo se les facilitará un problemario donde tendrán que evaluar un conjunto de expresiones aritméticas. Al finalizar los equipos seleccionan un problema y tendrán que codificarlo en el lenguaje de programación Java, se contribuirán sus respuestas con los diferentes equipos.

Cierre:

Con la guía y la orientación del profesor, participan en la construcción de fórmulas matemáticas para la construcción de expresiones utilizando solo los operadores aritméticos en el lenguaje de programación Java

<p>Conoce el concepto de algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas secuenciales.</p>	<p>Concepto de algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Elaboración de algoritmos secuenciales.</p> <p>Representación de algoritmos secuenciales a través de diagramas de flujo y pseudocódigo</p>	<p>Apertura</p> <p>Los alumnos, investigan el concepto de algoritmo, diagrama de flujo y pseudo-código.</p> <p>Desarrollo</p> <p>En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas secuenciales, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, pseudocódigo y diagrama de flujo.</p> <p>Cierre</p> <p>En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema secuencial propuesto por el profesor.</p> <p>Extraclase</p> <p>Los alumnos proponen y construyen la solución de un problema secuencial. En-tregan un reporte que incluya las etapas de la metodología de solución de problemas.</p> <p>Apertura:</p> <p>El profesor comienza la clase planteando la importancia de comprender los conceptos fundamentales en programación, los alumnos con ayuda de una computadora y utilizando fuentes confiables definirán los siguientes conceptos: Algoritmo, Diagrama de flujo, Pseudocódigo, Prueba de escritorio.</p> <p>Desarrollo:</p> <p>El profesor propone un ejercicio y en conjunto con el grupo se elabora el algoritmo, el diagrama de flujo, el pseudocódigo y la prueba de escritorio, posteriormente el profesor organiza a los alumnos por equipos y les asigna diferentes problemas, cada equipo deberá desarrollar el algoritmo, diagrama de flujo, pseudocódigo y la prueba de escritorio al problema asignado.</p> <p>Cierre:</p> <p>Cada equipo comparte la solución propuesta de su problema y el diagrama de flujo, para finalizar el profesor destaca la importancia de estos conceptos como base para el desarrollo de habilidades de programación y resolución de problemas en computación.</p>
---	---	---

		<p>Cierre En equipo, los alumnos, resuelven ejercicios sobre expresiones lógicas.</p> <p>Extraclase Los alumnos proponen expresiones lógicas, para evaluarlas y analizar porque unas son verdaderas y otras falsas.</p> <p>Inicio: Plantear problemas que responden a preguntas cuyas respuestas son falso o verdadero. Por ejemplo: realizar una búsqueda en internet de trámites como la credencial del INE.</p> <p>Desarrollo Plantear la necesidad de operadores lógicos.</p> <p>Inicio: Identificar en problemas matemáticos rangos de valores, plantear algunos ejemplos donde ubique que los valores eventualmente no satisfacen criterios de valores de entrada (excepciones).</p> <p>Desarrollo: Plantear la necesidad del uso de operadores relacionales</p> <p>Cierre: Plantear ejemplos que combinen ambos operadores.</p>
<p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales</p>	<p>Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas condicionales simples</p>	<p>Inicio: Docente: propone problemas que requieren la evaluación de sentencias condicionales, que involucren estructuras de control de condicionales: ¿el dato es válido?, ¿hacer acciones si se cumple algo?, ¿qué hacer si se decide entre dos opciones?, comparar valores en tablas, manejo de menús.</p> <p>Desarrollo: Docente: Plantear problemas que requieren el uso de la estructura condicional: por ejemplo raíces de una ecuación de segundo, el describir el IMC de una personal, identificar qué día descansa un auto con placas de distinta terminación, registro de documentos para trámite de la credencial del INE. Estudiantes en equipo: discuten los problemas y proponen condiciones. Docente: formaliza el trazado de las rutas de instrucciones en pseudocódigo correspondientes a cada valor de falso y verdadero. Introducir el uso del símbolo de toma de decisiones (rombo en diagrama de flujo o el uso de las palabras Si-Entonces-FinSi) y resalte la importancia de la indentación. Estudiantes: Identifican las combinaciones de los lazos involucrados en las expresiones booleanas de las sentencias condicionales que se requieren para</p>

		<p>plantear un plan de pruebas.</p> <p>Optimice los algoritmos entre el uso de simple, doble (ecuación de 2° grado), anidada (interpretación de tablas).</p> <p>Para situaciones donde los valores son atómicos presente la estructura condicional múltiple (días que circula un auto, presionar teclas en un juego, seleccionar opciones de menú).</p> <p>Extraclase:</p> <p>Estudiante: realiza ejercicios de tarea y una redacción sobre el aprendizaje.</p>
(Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales múltiples)	<p>Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas:</p> <ul style="list-style-type: none"> • Estructuras condicionales anidadas. • Estructuras condicionales múltiples. 	
Elabora el algoritmo, diagrama de flujo y pseudocódigo para resolver problemas cíclicos	<p>Estructuras de control de secuencia repetitivas:</p> <ul style="list-style-type: none"> • Componentes básicos de una estructura repetitiva: Sentencia de asignación del valor o condición inicial, evaluación de la expresión booleana, sentencia de modificación de estado, sentencias (que se repiten). • Concepto de bandera. • Concepto de contador. • Concepto de acumulador. • Elaboración de algoritmos de ciclo. • Representación de algoritmos de ciclo a través de diagramas de flujo. • Representación de algoritmos de ciclo a través de pseudocódigo 	<p>Inicio:</p> <p>El profesor presenta un problema que requiera el uso de un ciclo de repetición y en plenaria con los alumnos, se buscarán diversas formas de solucionarlo.</p> <p>Desarrollo:</p> <p>Se presentarán los diferentes conceptos importantes sobre los ciclos de repetición y la forma de representarlos en un algoritmo, diagrama de flujo y pseudocódigo. El profesor asignará a los alumnos en equipos y les facilitará un problema que requiera el uso de ciclos. Cada equipo tendrá que elaborar el algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Cierre:</p> <p>El profesor resalta las diferencias entre los diferentes tipos de ciclo que existen y la manera de controlarlos.</p> <p>Extraclase:</p> <p>Cada alumno identificará un problema que requiera el uso de un ciclo de repetición y elaborará su algoritmo, diagrama de flujo y pseudocódigo.</p>

<ul style="list-style-type: none"> • Conoce las generalidades de los lenguajes de programación más utilizados • Conoce la historia del lenguaje de programación Java. • Conoce las características básicas del lenguaje de programación Java. • Conoce el entorno de desarrollo para el lenguaje de programación Java. 	<p>Lenguaje de programación:</p> <ul style="list-style-type: none"> • Lenguajes de programación más utilizados. • Paradigmas de programación. • Requerimientos funcionales y no funcionales. • Conjunto de reglas de léxico y de sintaxis. • Código fuente. • Código máquina y objeto. • Compiladores, interpretes. <p>Lenguaje de programación Java:</p> <ul style="list-style-type: none"> • Historia del lenguaje. • Características: <ul style="list-style-type: none"> - Tipos de aplicaciones. - Arquitectura neutral. - Lenguaje orientado a objetos. - Disponibilidad de un amplio conjunto de bibliotecas. - Interpretado. - Robusto. - Distribuido. 	<p>Apertura El profesor organiza a los alumnos en equipos y en forma aleatoria les asigna alguno de los siguientes temas a investigar: historia, características y entorno de desarrollo del lenguaje de programación Java.</p> <p>Desarrollo En plenaria los equipos exponen la investigación realizada y obtienen conclusiones.</p> <p>Cierre El profesor proporciona a los alumnos, una guía donde se muestran los pasos que deberá seguir para realizar la instalación de las herramientas para el entorno.</p> <p>Cierre El profesor proporciona a los alumnos, una guía donde se muestran los pasos que deberán de seguir para realizar la instalación de las herramientas del entorno de desarrollo del lenguaje de programación Java (Eclipse, Netbeans, JCreator, etc.).</p> <p>Extraclase Los alumnos instalan el JDK e IDE en su computadora personal o aplicaciones para el desarrollo de aplicaciones Java en su smartphone o tableta (Jvroid, Dcoder, etc).</p>
	<p>Entorno de desarrollo del lenguaje de programación:</p> <ul style="list-style-type: none"> - JDK (kit de desarrollo de Java). - JVM(Máquina virtual de Java) - IDE (interfaz de entorno de desarrollo). 	<p>de desarrollo del lenguaje de programación Java (Eclipse, Netbeans, JCreator, JBuilder, entre otros).</p> <p>Los alumnos realizan un programa de prueba en Java propuesto por el profesor.</p> <p>Extraclase. Contestan un cuestionario sobre la temática vista.</p>
<ul style="list-style-type: none"> • Realiza programas empleando el método de salida de datos. 	<p>Pasos para implementar un programa con el lenguaje de programación Java y el entorno de desarrollo.</p> <ul style="list-style-type: none"> • Creación de un proyecto. • Declaración de la Clase. • Método <i>main</i>. • Empleo de los métodos <i>System.out.print</i> y <i>System.out.println</i>. 	<p>Apertura</p> <ul style="list-style-type: none"> • Los alumnos escriben un programa propuesto por el profesor, que utilice el método de salida de datos, emplea el IDE, en donde observan la creación de: un proyecto, una Clase y el método <i>main</i>. • Los alumnos ejecutan el programa observando, el resultado. <p>Desarrollo</p> <ul style="list-style-type: none"> • En equipo, los alumnos investigan los conceptos relativos a la Clase <i>Scanner</i>, definición del objeto de la Clase <i>Scanner</i> y el método <i>System.in</i>.

<ul style="list-style-type: none"> Realiza programas empleando la Clase <i>Scanner</i> para la entrada de datos. Codifica algoritmos secuenciales en el lenguaje de programación Java empleando la interfaz de consola 	<ul style="list-style-type: none"> Errores sintácticos y lógicos. Ejecución del programa. <p>Introducción de datos desde el teclado.</p> <ul style="list-style-type: none"> La Clase <i>Scanner</i> y sus principales métodos. Definición del objeto de la Clase <i>Scanner</i>. Método <i>System.in</i>. Instanciación del objeto de la clase <i>Scanner</i>. Entrada estándar de datos por consola (<i>System.in</i>) Tipos de datos primitivos. <ul style="list-style-type: none"> Enteros: <i>byte, short, int, long</i>. Reales: <i>float, double</i>. Booleanos: <i>boolean</i>. Caracteres: <i>char</i>. 	<ul style="list-style-type: none"> El profesor da un ejemplo sobre el uso de la Clase <i>Scanner</i>. <p>Cierre El profesor plantea un problema que emplee lectura y escritura de datos, los alumnos auxiliados por el profesor escriben el código del programa correspondiente y comprueban su funcionamiento.</p> <p>Extraclase. El profesor propone una serie de problemas de tipo secuencial, los alumnos construyen el código del programa respectivo y entregan un reporte con las evidencias de la ejecución.</p> <p>Apertura: Los alumnos con ayuda del profesor realizan un primer programa de prueba en Java. El profesor borrará elementos del programa inicial para provocar errores de sintaxis que los alumnos deberán de corregir en plenaria.</p> <p>Desarrollo: El profesor mediante una aplicación de Java presentará a la clase el uso de los diferentes tipos de datos primitivos del lenguaje. Por ejemplo, un programa para calcular el área de un triángulo en donde los valores para la base y la altura sean declarados e inicializados desde un inicio. Los alumnos codificarán en Java sus primeros algoritmos secuenciales haciendo uso de los tipos de datos primitivos de Java.</p> <p>Cierre: Los alumnos con ayuda del profesor optimizarán el uso de la memoria que hace la aplicación, dependiendo de los valores que tomen las variables de sus programas. Por ejemplo, si las variables toman valores positivos menores a 255 se recomendará usar el tipo <i>byte</i></p>
<p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales.</p> <p>Codifica algoritmos condicionales en el lenguaje de programación java</p>	<p>Estructuras condicionales.</p> <ul style="list-style-type: none"> Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas condicionales. <p>Sentencias condicionales:</p> <ul style="list-style-type: none"> Simple: <i>if</i>. Doble: <i>if-else</i>. Anidadas: <i>if-else-if</i>. Múltiples: <i>switch</i>. 	<p>Apertura En equipo, los alumnos, investigan el concepto de estructura condicional.</p> <p>Desarrollo En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas condicionales, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Cierre En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema condicional propuesto por el profesor.</p> <p>Extraclase</p>

		<p>Los alumnos proponen y construyen la solución de un problema condicional. Entregan un reporte que incluya las etapas de la metodología de solución de problemas.</p> <p>Apertura El profesor presenta las sentencias condicionales de Java y las relaciona con sus respectivos diagramas de flujo y pseudocódigos. Revisar los operadores lógicos y relacionales. Completando la tabla de operadores relacionales en Java</p> <p>Desarrollo: El profesor muestra un ejemplo de codificación de algoritmos secuenciales en Java. El profesor y los estudiantes codifican los algoritmos condicionales elaborados previamente. Identificar sentencias nulas, simples y múltiples (bloques) y cómo opera en estos casos el terminador de sentencia “;”.</p> <p>Cierre: El alumno traduce las sentencias lógicas condicionales de sus diagramas de flujo en sentencias de Java identificando la más conveniente para cada tipo de problema: if, if-else, if-else-if o switch.</p> <p>Extraclase: Proponer nuevos problemas para codificar directamente en Java y, de forma inversa, obtener sus diagramas de flujo y pseudocódigos.</p>
<p>Construye programas de computadora que resuelvan problemas condicionales.</p>	<p>Tipo de dato primitivo: Lógico.</p> <p>Sentencias condicionales:</p> <ul style="list-style-type: none"> • Simple <i>if</i>. • Doble <i>if-else</i>. 	<p>Apertura Los alumnos, investigan el concepto de Sentencia Condicional</p> <p>Desarrollo</p> <ul style="list-style-type: none"> • En plenaria, los equipos, obtienen conclusiones. • A partir de la lista de problemas condicionales, de la sección anterior, los alumnos escriben, auxiliados por el profesor, el código del programa correspondiente y comprueban su funcionamiento. <p>Cierre En equipo, los alumnos, escriben el código de un programa relativo a un problema condicional propuesto por el profesor y comprueban su funcionamiento.</p> <p>Extraclase Los alumnos proponen y construyen el código de un programa relativo a un problema condicional. Entregan un reporte que incluya el código y evidencias de la ejecución.</p>

<p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales múltiples.</p>	<p>Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas:</p> <ul style="list-style-type: none"> • Sentencias condicionales anidadas. • Condicionales múltiples. 	<p>Apertura En equipo, los alumnos investigan el concepto de estructura condicional anidada y múltiple.</p> <p>Desarrollo En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas condicionales anidados y múltiples, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Cierre En equipo, los alumnos construyen el algoritmo, diagrama de flujo y pseudocódigo de problemas condicionales anidados y múltiples propuestos por el profesor.</p> <p>Extraclase Los alumnos proponen y construyen la solución de un problema condicional múltiple. Entregan un reporte que incluya las etapas de la metodología de solución de problemas.</p>
<p>Construye programas de computadora que resuelvan problemas que involucren la toma de decisiones múltiples.</p>	<p>Sentencias condicionales anidadas.</p> <p>Sentencias condicionales múltiples.</p> <ul style="list-style-type: none"> • Switch. 	<p>Apertura Los alumnos investigan el concepto de sentencia condicional anidada y múltiple.</p> <p>Desarrollo En plenaria, los equipos obtienen conclusiones. A partir de la lista de problemas condicionales anidados y múltiples, de la sección anterior, los alumnos escriben, auxiliados por el profesor, el código del programa correspondiente y comprueban su funcionamiento.</p> <p>Cierre En equipo, los alumnos escriben el código de un programa relativo a un problema condicional múltiple propuesto por el profesor y comprueban su funcionamiento.</p> <p>Extraclase Los alumnos proponen y construyen el código de un programa relativo a un problema condicional múltiple. Entregan un reporte que incluya el código y evidencias de la ejecución.</p>

<p>Elabora el algoritmo, diagrama de flujo y pseudocódigo para resolver problemas de estructura de ciclo.</p>	<p>Estructuras de ciclo.</p> <ul style="list-style-type: none"> • Elaboración de algoritmos de ciclo. • Representación de algoritmos de ciclo a través de diagramas de flujo. • Representación de algoritmos de ciclo a través de pseudocódigo. • Concepto de contador. • Concepto de acumulador. 	<p>Apertura En equipo, los alumnos investigan el concepto de estructuras de ciclo.</p> <p>Desarrollo En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas de estructuras de ciclo, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Cierre En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema de estructura de ciclo propuesto por el profesor.</p> <p>Extraclase Los alumnos proponen y construyen la solución de un problema de estructura de ciclo. Entregan un reporte que incluya las etapas de la metodología de solución de problemas.</p>
<p>Construye programas de computadora que empleen la sentencia <i>for</i>.</p>	<p>Estructuras de control de ciclo.</p> <ul style="list-style-type: none"> • Sentencia <i>for</i>. 	<p>Apertura Los alumnos, investigan el concepto de estructuras de control de ciclo <i>for</i>.</p> <p>Desarrollo En plenaria, los equipos obtienen conclusiones.</p> <p>A partir de la lista de problemas de estructuras de control de ciclo <i>for</i>, de la sección anterior, los alumnos escriben, auxiliados por el profesor, el código del programa correspondiente y comprueban su funcionamiento.</p> <p>Cierre En equipo, los alumnos escriben el código de un programa relativo a un problema de estructuras de control de ciclo <i>for</i> propuesto por el profesor y comprueban su funcionamiento.</p> <p>Extraclase Los alumnos proponen y construyen el código de un programa relativo a un problema de estructuras de control de ciclo <i>for</i>. Entregan un reporte que incluya el código y evidencias de la ejecución.</p>

<p>Elabora el algoritmo, diagrama de flujo y pseudocódigo de problemas de ciclo (cíclicos) que satisfagan una condición.</p>	<p>Estructuras de ciclo.</p> <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. • Representación de algoritmos de ciclo a través de pseudocódigo que satisfagan una condición. 	<p>Apertura En equipo, los alumnos investigan el concepto de ciclo que satisfaga una condición.</p> <p>Desarrollo En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas de ciclo que satisfagan una condición, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, diagrama de flujo y pseudocódigo.</p> <p>Cierre En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema de ciclo que satisfaga una condición propuesto por el profesor.</p> <p>Extraclase Los alumnos proponen y construyen la solución de un problema de ciclo que satisfaga una condición. Entregan un reporte que incluya las etapas de la metodología de solución de problemas.</p>
<p>Construye programas de computadora que involucren la sentencia <i>while</i>.</p>	<p>Estructura de control de ciclo.</p> <ul style="list-style-type: none"> • Sentencia <i>while</i>. 	<p>Apertura Los alumnos, investigan el concepto de Estructuras de control de ciclo <i>while</i>.</p> <p>Desarrollo En plenaria, los equipos obtienen conclusiones.</p> <p>A partir de la lista de problemas de estructuras de control de ciclo <i>while</i>, de la sección anterior, los alumnos escriben, auxiliados por el profesor, el código del programa correspondiente y comprueban su funcionamiento.</p> <p>Cierre En equipo, los alumnos escriben el código de un programa relativo a un problema de estructuras de control de ciclo <i>while</i> propuesto por el profesor y comprueban su funcionamiento.</p> <p>Extraclase Los alumnos proponen y construyen el código de un programa relativo a un problema que utilice estructuras de control de ciclo <i>while</i>. Entregan un reporte que incluya el código y evidencias de la ejecución.</p>

CIBERNÉTICA Y COMPUTACIÓN II

Unidad 1. Lenguaje Fundamentos de programación orientada a objetos con Java

En la unidad se podrá identificar el paradigma de la programación orientada a objetos POO, sus pilares fundamentales: encapsulamiento, abstracción, herencia y polimorfismo, de forma detallada los primeros dos, para que el alumnado pueda diseñar bajo el Lenguaje de Modelado Unificado UML e implementar dichos conceptos dentro de un código en lenguaje java y así crear sus programas orientados a objetos

<p>Propósito:</p> <p>Al finalizar la unidad el alumnado:</p> <p>Conocerá las características de la lenguaje programación orientado a objetos Java y su entorno de desarrollo, definiendo clases, atributos y métodos para la implementación de objetos en programas, desde el modelado de las clases, instanciación de objetos para la implementación de objetos en la resolución de problemas</p>	<p>Tiempo: 10 horas</p>
---	------------------------------------

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Conoce los conceptos básicos de la programación orientada a objetos</p> <p>Conoce la organización general de un programa en Java como lenguaje orientado a objetos.</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"> • Paquete. • Clase. <ul style="list-style-type: none"> ○ Atributos o características. ○ Métodos o comportamiento. • Objeto. <ul style="list-style-type: none"> ○ Identidad. ○ Atributos. ○ Métodos. <p>Características de la programación orientada a objetos.</p> <ul style="list-style-type: none"> • Abstracción. • Encapsulamiento. • Herencia. • Polimorfismo. <p>Organización general de un programa</p>	<p>Apertura</p> <p>El profesor organiza a los alumnos en equipos, y en forma aleatoria, les asigna alguno de los siguientes temas a investigar sobre la programación orientada a objetos: identidad, atributos o características, comportamiento, abstracción y encapsulamiento.</p> <p>Desarrollo</p> <p>En plenaria los equipos exponen la investigación realizada y obtienen conclusiones.</p> <p>Cierre</p> <p>El profesor utilizando un programa ejemplo muestra a los alumnos la organización general de un programa Java.</p> <p>Extraclase</p> <p>Los alumnos investigan la definición de Clase, atributo y método, así como su implementación en el lenguaje Java.</p> <p>Apertura</p> <p>El docente arma equipos de trabajo y solicita que realicen un organizador gráfico a su gusto sobre los conceptos a abordar. El estudiantado presenta sus organizadores gráficos y llegan a conclusiones sobre los conceptos.</p> <p>Desarrollo</p>

	<p>ma en Java</p> <ul style="list-style-type: none"> • Comentarios. • Uso de bibliotecas. • Identificadores. • Palabras reservadas. • Sentencias. • Tipos de datos primitivos. • Bloque de código. • Operadores. • Expresiones. 	<p>El docente solicita que los alumnos de manera individual identifiquen los atributos (características) y métodos (acciones) comunes a una clase de objetos, como pueden ser: Celulares, tabletas, computadoras, consolas de videojuegos, televisores, etc.</p> <p>En plenaria, el estudiantado presenta su trabajo y con base en dichos ejemplos, el docente generaliza la información relevante de las diferentes clases de los objetos expuestos, mediante diagramas de clase y de objetos (Abstracción).</p> <p>Cierre</p> <p>El docente junto con el estudiantado, concluyen y retroalimentan lo aprendido en clase. Se sugiere realizar una actividad interactiva con algún software o documento impreso (consultar las referencias).</p> <p>Actividad extra clase</p> <p>El alumnado elabora de forma similar a la trabajada en clase los diagramas para los siguientes casos: Alumnos, trabajadores, figuras geométricas, datos de un paciente, etc.</p>
<ul style="list-style-type: none"> • Describe los conceptos de Clase y atributo del lenguaje Java. • Implementa programas utilizando Clases y atributos. 	<p>Clases:</p> <ul style="list-style-type: none"> • Definición. • Declaración. 	<p>Apertura</p> <p>En plenaria, el profesor propicia el intercambio de opiniones para definir qué es una Clase y sus atributos.</p>
	<ul style="list-style-type: none"> • Atributos. <ul style="list-style-type: none"> - Definición. - Declaración. - Niveles de visibilidad. • Implementación. 	<p>Desarrollo</p> <p>Los alumnos escriben programas, utilizando el IDE, en donde codifiquen Clases y sus atributos.</p> <p>Cierre</p> <p>Los alumnos contestan un cuestionario.</p> <p>Extraclase</p> <p>Los alumnos investigan qué es un método e instanciación de objetos.</p>

<p>Conoce los estándares del Lenguaje Unificado de Modelado (UML) para la elaboración de un diagrama de clase y de objeto.</p>	<p>Estructura del diagrama de objetos:</p> <ul style="list-style-type: none"> • Identificación. • Datos. • Métodos. <p>Estructura del diagrama de clase:</p> <ul style="list-style-type: none"> • Nombre de la clase. • Atributos. • Métodos. <p>Modificadores de acceso (Niveles de visibilidad):</p> <ul style="list-style-type: none"> • De los atributos: <ul style="list-style-type: none"> ○ Privados. ○ Públicos. ○ Protegidos. ○ Estáticos. • De los métodos: <ul style="list-style-type: none"> ○ Privados. ○ Públicos. ○ Protegidos. ○ Estáticos. ○ Abstractos. 	<p>Apertura</p> <p>El alumnado, usando de preferencia las fuentes de consulta de este documento, realiza una investigación documental sobre los estándares del Lenguaje Unificado de Modelado (UML) para la representación gráfica de una clase de objetos.</p> <p>Desarrollo</p> <p>Con base en los primeros diagramas elaborados por los alumnos extra clase de la sesión anterior (Alumnos, trabajadores, figuras geométricas, datos de un paciente, etc.), el docente junto con el estudiantado, aplican el Lenguaje Unificado de Modelado (UML) para la elaboración de las representaciones gráficas de las clases y de los objetos de uno de los casos.</p> <p>El docente ejemplifica el uso de los modificadores de acceso para manipular los niveles de visibilidad de los atributos y métodos de una clase. El alumnado integra a sus diagramas de clase los símbolos relacionados a cada modificador de acceso empleado en cada atributo y método.</p>
<p>Elabora los diagramas de clase y de objetos usando el Lenguaje Unificado de Modelado (UML).</p>		<p>Cierre</p> <p>De forma individual el alumnado elabora los diagramas de clase y de objetos usando el Lenguaje Unificado de Modelado (UML) del resto de los ejercicios extra clase.</p> <p>Actividad extra clase.</p> <p>Los alumnos realizarán una investigación documental sobre los métodos constructores, getters y setters de una clase.</p>

<ul style="list-style-type: none"> Describe los conceptos de métodos del lenguaje java. Conoce cómo instanciar objetos a partir de una Clase. Implementa programas utilizando métodos. <p>Implementa la codificación en Java del modelo de clase, utilizando el diagrama de clase (atributos y los 3 métodos constructor, setter y getter).</p>	<p>Métodos:</p> <ul style="list-style-type: none"> Definición. Declaración. Parámetros. <p>Métodos:</p> <ul style="list-style-type: none"> Getter. Setter. <p>Objetos:</p> <ul style="list-style-type: none"> Definición. Declaración (instanciación). <p>Implementación.</p> <ul style="list-style-type: none"> Declaración de la clase. Declaración de atributos. Declaración e implementación de los métodos <ul style="list-style-type: none"> constructor. setters y getters. 	<p>Apertura</p> <p>En plenaria con base en la investigación documental desarrollada por el estudiantado extraclase, se obtienen conclusiones sobre los métodos constructores, getter y setters de una clase.</p> <p>Desarrollo</p> <p>Con base en los diagramas de clase desarrollados por el estudiantado, el docente ejemplifica la codificación de una de esas clases en Java, especificando: declaración de la clase, declaración de atributos e implementación de los métodos getter, setter y constructores.</p> <p>Cierre</p> <p>El estudiantado elabora de forma individual la codificación en Java de las clases de los ejercicios restantes.</p> <p>Actividad extra clase</p> <p>El estudiantado realiza una investigación documental sobre la arquitectura modelo-vista-controlador.</p> <ul style="list-style-type: none">
<p>Empleará la Clase <i>Scanner</i> para la entrada de datos en la creación de un programa.</p>	<p>La Clase <i>Scanner</i>.</p> <ul style="list-style-type: none"> Importar la Clase <code>java.util.Scanner</code>. Definición del objeto de la Clase <i>Scanner</i> (instanciación). Método <code>system.in</code> Introducción de datos desde el teclado. <p>Errores sintácticos y lógicos.</p> <p>Ejecución del programa.</p>	<p>Apertura</p> <p>El profesor da un ejemplo sobre el uso de la Clase <i>Scanner</i> y de cómo se instancia un objeto de ésta Clase para poder introducir datos desde el teclado.</p> <p>Desarrollo</p> <p>El profesor propone un ejemplo de entrada y salida de datos, haciendo énfasis en cómo se instancia un objeto de la Clase <i>Scanner</i>.</p>

		<p>Cierre El profesor plantea problemas que empleen lectura y escritura de datos y los alumnos elaboran el programa correspondiente, insistiendo en la importancia de la instanciación del objeto de la Clase <i>Scanner</i>.</p> <p>Extraclase Los alumnos proponen un problema y elaboran el programa que considere todos los conceptos vistos en la unidad.</p>
<p>El alumnado: Emplea la arquitectura modelo-vista-controlador para instanciar una clase.</p> <p>Escribe el código del controlador para la instanciación de la clase.</p>	<p>Arquitectura modelo-vista-controlador:</p> <ul style="list-style-type: none"> • Modelo. • Vista. • Controlador. <p>Diagrama Clase-Objeto.</p> <p>Instanciación.</p> <ul style="list-style-type: none"> • Crear objetos. • Hacer llamada a los métodos (notación punto). 	<p>Apertura Según la investigación documental del estudiantado, el docente mostrará a los estudiantes un proyecto orientado a objetos en Java. Utilizando el IDE de su preferencia el docente creará un proyecto, el paquete, y los archivos para la clase de los objetos y la clase principal o controladora. El estudiantado de forma similar a la mostrada en clase creará el proyecto para uno de los casos previamente trabajados (Alumnos, trabajadores, figuras geométricas, pacientes, etc.).</p> <p>Desarrollo Con base en el proyecto ejemplo, el docente mostrará la instanciación de objetos desde la clase controladora usando el operador new y los constructores de la clase. El estudiantado de forma individual codificará la instanciación de al menos cinco objetos desde la clase controladora.</p> <p>Cierre Utilizando el programa ejemplo, el docente utilizará los métodos getters y setters de la clase para manipular la información de cada uno de los objetos instanciados (encapsulamiento). El estudiantado modificará el código de la clase controladora para que desde ella se pueda actualizar la información de sus objetos.</p> <p>Extra clase Los estudiantes realizan una investigación documental sobre la declaración en Java de los métodos de clase y de instancia.</p>

<p>Reafirma los conceptos adquiridos en la unidad.</p> <p>Elabora la solución a un problema definiendo la clase e instanciando el objeto.</p>	<p>Implementa una aplicación.</p> <ul style="list-style-type: none"> • Métodos de clase y de instancia. • Componentes de un método de clase y de instancia: <ul style="list-style-type: none"> ▪ Modificador de acceso. ▪ Tipo de dato de retorno. ▪ Identificador. ▪ Parámetros. ▪ Bloque de sentencias. 	<p>Apertura</p> <p>El profesor organiza a los alumnos para que presenten sus trabajos.</p> <p>Desarrollo</p> <p>Los alumnos exponen sus trabajos.</p> <p>Cierre</p> <p>Los alumnos obtienen conclusiones.</p> <p>Extraclase</p> <p>Los alumnos resuelven un cuestionario acerca de los conceptos vistos en la unidad.</p> <p>Apertura</p> <p>Con base en la investigación documental elaborada previamente por el estudiantado y los proyectos de las sesiones previas. El docente y el estudiantado generalizan la declaración de un método en Java (Modificador de acceso, tipo de dato de retorno, identificador, parámetros y bloque de sentencias).</p> <p>Desarrollo</p> <p>Con base en el proyecto ejemplo que se haya elegido en sesiones previas, el docente ejemplificará la codificación en Java de un método de clase y de instancia. Por ejemplo, si se ha estado trabajando con el caso de los Alumnos, el creará un método de clase para obtener el promedio de clase y un método de instancia para el promedio de un alumno; si se ha trabajado con el caso de los trabajadores, el docente creará un método de instancia para el cálculo del salario bruto y uno de instancia para el cálculo de la nómina.</p> <p>Cierre</p> <p>El docente determina y solicita al estudiantado los métodos de clase y de instancia que deberá cada uno de implementar en su proyecto.</p> <p>Extra clase</p> <p>El estudiantado elabora y documenta un proyecto entregable orientado a objetos en Java aplicando todos los aprendizajes de la unidad.</p>
---	--	--

Fuentes de consulta

Alumnado

Básica

Jiménez, J., Jiménez E. M. y Alvarado L. N: (2014). Fundamentos de programación, diagramas de flujo, diagramas, diagramas N-S, pseudocódigo y Java. México: Alfaomega. <https://content.e-bookshelf.de/media/reading/L-18448289-5aeff52fbc.pdf>

Joyanes, L. y Zahonero, M. (2014). Programación en c, c++, Java. México: McGraw–Hill Interamericana.

<https://idoc.pub/documents/programacion-en-c-c-java-y-uml-2da-edicion-luis-joyanes-aguilar-librosvirtualcom-2nv59vr03rlk>

Joyanes, L. y Zahonero, M. (2011). Programación en Java: algoritmos, programación orientada a objetos e interfaz gráfica de usuarios. México: McGraw–Hill Interamericana RUA

Sznajdleder. P. (2012). Java a fondo, estudio del lenguaje y desarrollo de aplicaciones. México: Alfaomega.

Moisset, D. (2016). Curso de programación Java [en línea]. <https://tutorialesprogramacionya.com/javaya/>

Pérez G. (2008), Aprendiendo Java y poo. [en línea]. <https://www.cartagena99.com/recursos/programacion/apuntes/AprendiendoJava.pdf>

Heno, C. (2013). Conceptos Básicos de Programación Orientada a Objetos. CoDejaVu: Conceptos Básicos de Programación Orientada a Objetos

Complementaria

Bernal, J. (2012). Programación orientada a objetos con java. Universidad Politécnica de Madrid. Master Universitario en Ingeniería Web.

https://www.etsisi.upm.es/sites/default/files/curso_2013_14/MASTER/MIW.JEE.POOJ.pdf

Ceballos, F. (2015). Java. Interfaces gráficas y aplicaciones para Internet. España: Ra-ma

Martín, A. (2014). Programador Java certificado: curso práctico. Madrid: Ra–Ma.

*Wu, C. (2008). Programación en Java. México: McGraw Hill.

Olsson, M. Java quick syntax reference [en línea]. Recuperado 29 de febrero de 2016 en <http://ebooks.mpdl.mpg.de/ebooks/Record/EB001085057/Details>>

Programación básica en Java [en línea]. Recuperado el 1 de febrero de 2016 en: <<http://bit.ly/1P1PaCU>>.

Deitel P. y Deitel H. (2016). Como programar en Java. 10ma edición. Pearson Education. México.

Ossio, S. W., Prudencio, A. M., Salgado I. y Valda, F. S. (2007). Programación II Guía de Java + ejercicios. Bolivia; Universidad Católica Boliviana “San Pablo”.

Profesorado

Básica

Cairó, O. (1995). *Metodología de la programación, algoritmos, diagrama de flujo y programas*. México: Computec.

Cardona, S. (2008), *Introducción a la programación en java*, México: Editorial Elizcom.

Casanova, A. *Empezar a programar usando Java* [en línea]. Recuperado 29 de febrero de 2016 en <<http://site.ebrary.com.pbidi.unam.mx:8080/lib/bibliodgbsp/detail.action?docID=10831789>>

Deitel P. y Deitel H. (2016). Como programar en Java. 10ma edición. Pearson Education. México.

Joyanes, L. & Zahonero, M. (2014). *Programación en c, c ++, Java y uml*. México: McGraw–Hill Interamericana.

López, L. (2013). Metodología de la programación orientada a objetos. 2da. Edición. Edit. Alfaomega.

Moreno, J. (2015). Programación orientada a objetos. Ra-Ma.

Wu, C. (2008). *Programación en Java*. México: McGraw Hill.

Blasco, F. (2019). Programación orientada a objetos en Java Colombia: Ediciones de la U. BIDI

Malik, D.S. (2013). Programación jaca del análisis de problemas al diseño de programas 5a. Ed. (traducción; León, J.). México: Cengage learning Editores. BIDI

Complementaria

*Ceballos, F. (2006). *Java 2: curso de programación*.

México: Alfaomega. Eckel, B. (2007). *Piensa en Java*.

Madrid: Pearson Educación.

*Wu, C. (2008). *Programación en Java*. México: McGraw Hill.

Belmonte, O. *Introducción al lenguaje de programación Java. Una guía básica* [en línea].

<https://docplayer.es/1441515-Introduccion-al-lenguaje-de-programacion-java.html>

Software interactivo:

Kahoot!: App. Preguntas para repaso y/o generar estadísticas para diagnóstico.

Educaplay: Actividades diversos tipos como crucigramas, memoramas, relación de columnas, etc.

<https://es.educaplay.com/>

Brainscape: App. Diseño de cartas para repasar aprendizajes.

Flippity: Actividades diversas. <https://www.flippity.net/>

Edpuzzle: videos interactivos. <https://edpuzzle.com/>

Wordwall: Recursos didácticos imprimibles. <https://wordwall.net/es>

IDE programación en línea.

Replit: <https://replit.com/>

Onlinegdb: www.onlinegdb.com

Software para el desarrollo de la POO

BlueJ.

Replit (replit.com).

Eclipse.

NetBeans.

Evaluación

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e

instrumentos de evaluación:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- preguntas formuladas al inicio de la unidad y de cada clase
- análisis de las respuestas dadas en los trabajos extraclase.
- Cuestionarios o actividades interactivas del portal académico.

Formativa.

Evaluación para la reorientación para el logro de los aprendizajes.

- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento de los alumnos.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios.

- Proyecto entregable: Documento desde el diagrama de clase, codificación en Java y evidencias de la salida del programa.
- Portafolio de evidencias. Actividades realizadas en cada sesión o por aprendizaje.
- Listas de verificación: para evaluar los trabajos extraclase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Nota: La ponderación de cada actividad será asignada por el profesor.

Unidad 2. Estructuras de control de secuencia en Java

Dentro de un lenguaje de programación, se podría ir haciendo instrucción tras instrucción, pero al conocer las estructuras de control el alumnado tiene la posibilidad de alterar, controlar o modificar el orden del flujo en el que se ejecutan las instrucciones, al tomar decisiones, creando ciclos de repetición, guardando datos en arreglos, dando mayor versatilidad a la forma de resolver problemáticas.

<p>Propósito:</p> <p>Al finalizar la unidad el alumnado:</p> <p>Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java.</p>	<p>Tiempo:</p> <p>22 24 horas</p>
--	---

Aprendizajes	Temática	Estrategias sugeridas
<p>El alumnado:</p> <p>Desarrolla programas que involucren las estructuras condicionales simples, compuestas, anidadas y múltiples en los métodos de una Clase.</p>	<p>Estructuras condicionales:</p> <ul style="list-style-type: none"> • Simples: if. • Compuestas: if-else • Anidadas • Operador ternario • Implementación. • Estructura condicional múltiple: • Switch. • Implementación. 	<p>Apertura</p> <ul style="list-style-type: none"> • En equipo, los alumnos investigan los conceptos relativos a las estructuras condicionales <p>Desarrollo</p> <p>El profesor plantea problemas que empleen estructuras condicionales simples, compuestas, anidadas y múltiples en métodos de una Clase, haciendo énfasis en la importancia de los distintos tipos de operadores relacionales y en la identificación de cuando debe utilizarse que tipo de condición.</p> <p>Cierre</p> <p>El alumno propone un problema similar a los vistos anteriormente y elabora el programa de solución.</p> <p>Extraclase</p> <p>El profesorado plantea nuevos problemas en cuya solución se empleen estructuras condicionales simples, compuestas, anidadas y múltiples, e invita al alumnado a subirlas a la plataforma.</p>

<p>Identifica y codifica excepciones a través de las sentencias try y catch.</p>	<p>Concepto de excepción try y catch</p>	<p>Apertura En equipo, el alumnado investiga los conceptos relativos al manejo de excepción try y catch.</p> <p>Desarrollo El profesorado tomando en cuenta la investigación previa realizada por el alumnado, explica el concepto de excepción relativa a los casos:</p> <ul style="list-style-type: none"> • Convertir a número entero una cadena (String) que no contiene valores numéricos. • Operación de división, en el caso de que el divisor tome el valor de cero. • Intentar acceder o modificar atributos con valores fuera de rango. <p>Explica el uso del try y catch en los métodos de una clase, así como plantea problemas, el alumnado bajo la estricta supervisión del profesorado los desarrolla.</p> <p>Cierre En reunión plenaria el alumnado expone la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p>
<p>Desarrolla programas que involucren la estructura condicional múltiple en los métodos de una Clase.</p>	<p>Estructura condicional múltiple:</p> <ul style="list-style-type: none"> • <i>Switch</i>. <p>Implementación.</p>	<p>Apertura</p> <p>El profesor explica el uso de la estructura condicional múltiple en los métodos de una Clase y realiza ejemplos de su aplicación.</p> <p>Desarrollo</p> <p>El profesor plantea problemas que empleen la estructura condicional múltiple en los métodos de una Clase, los alumnos elaboran los programas correspondientes.</p>
		<p>Cierre</p> <p>En plenaria los alumnos exponen la solución dada a los problemas resueltos y el resto de los alumnos opinan y comparan su solución.</p> <p>Extraclase</p> <p>En equipo, los alumnos investigan los conceptos relativos a la estructura repetitiva <i>for</i>.</p>

<p>Desarrolla programas para resolver problemas que involucren la estructura repetitiva <i>for</i> en los métodos de una Clase.</p> <p>Codifica programas para resolver problemas que involucren estructuras repetitivas <i>for</i>, <i>while</i> y <i>do while</i>.</p>	<p>Estructura repetitiva <i>for</i>.</p> <p>Implementación.</p> <p>Estructura repetitiva</p> <ul style="list-style-type: none"> • <i>for</i> • <i>while</i> • <i>do-while</i> 	<p>Apertura En equipo, el alumnado investiga los conceptos relativos a las estructuras repetitiva <i>for</i>, <i>while</i> y <i>do-while</i>.</p> <p>Desarrollo El profesorado explica los conceptos relativos a las estructuras repetitivas <i>for</i>, <i>while</i> y <i>do-while</i> y plantea problemas que empleen las estructuras <i>for</i>, <i>while</i> y <i>do-while</i> en métodos de una Clase, bajo la estricta supervisión del profesorado el alumnado los desarrolla.</p> <p>Cierre En reunión plenaria el alumnado expone la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>Extraclase El profesorado plantea problemas adicionales en cuya solución se empleen las estructuras repetitivas, e invita al alumnado a resolverlas empleando algunas de las estructuras vistas en clase <i>for</i>, <i>while</i> y <i>do-while</i>, solicita al alumnado a que justifique por qué selecciono la estructura de ciclo, el trabajo desarrollado lo deberán subir a la plataforma.</p> <p>NOTA. Se recomienda plantear problemas relativos a:</p> <ul style="list-style-type: none"> • Asignación del monto de beca con base al promedio obtenido durante el semestre. • Determinar el importe por la compra de diversos productos, considerando que se otorga un porcentaje de descuento con base en la compra realizada. • Convertir la calificación de número a letra. • Solución de: <ul style="list-style-type: none"> ▪ Ecuación de primer grado. ▪ Sistema de ecuaciones simultáneas. ▪ Ecuación de segundo grado. • En un verificentro determinar el promedio de puntos contaminantes de los vehículos atendidos durante un día, así como el vehículo que menos y más contamina. • Operación de un cajero automático. • Con base al sueldo diario, días y horas extras trabajadas, determinar el sueldo bruto de un empleado. • Índice de masa corporal. • Promedios de índice de masa corporal para un grupo de personas.
--	---	---

<p>Desarrolla programas que involucren la estructura repetitiva <i>while</i> en los métodos de una Clase.</p>	<p>Estructura repetitiva: <i>while</i>.</p> <p>Implementación.</p>	<p>Apertura El profesor explica el uso de la estructura repetitiva <i>while</i> en los métodos de una Clase, tomando como base la investigación previa realizada por los alumnos, y desarrolla ejemplos de su aplicación.</p> <p>Desarrollo El profesor plantea problemas que empleen la estructura repetitiva <i>while</i> en los métodos de una Clase, los alumnos elaboran los programas correspondientes.</p> <p>Cierre En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución.</p> <p>Extraclase En equipo los alumnos investigan los conceptos relativos a la estructura repetitiva <i>do-while</i>.</p>
<p>Desarrolla programas que involucren la estructura repetitiva <i>do-while</i> en los métodos de una Clase.</p>	<p>Estructura repetitiva: <i>do-while</i>.</p> <p>Implementación.</p>	<p>Apertura El profesor ejemplifica el uso de la estructura repetitiva <i>do-while</i> en los métodos de una Clase, tomando en cuenta la investigación previa realizada por los alumnos.</p> <p>Desarrollo El profesor plantea problemas que empleen la estructura repetitiva <i>do-while</i> en los métodos de una Clase, los alumnos elaboran los programas correspondientes.</p> <p>Cierre En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución.</p> <p>Extraclase En equipo, los alumnos investigan los conceptos relativos a arreglos unidimensionales.</p>

<p>Resuelve problemas que involucren el uso de los arreglos unidimensionales en los métodos de una Clase.</p> <p>Conoce el concepto de estructuras de datos en Java.</p> <p>Conoce el concepto de estructura de datos estática.</p> <p>Describe el concepto de los arreglos unidimensionales en los métodos de una clase.</p>	<p>Arreglos unidimensionales:</p> <ul style="list-style-type: none"> • Concepto. • Declaración. • Tipos. • Uso. • Recorrido • Estructura repetitiva for-each 	<p>Apertura El profesor, tomando en cuenta la investigación previa realizada por los alumnos, explica el uso de arreglos unidimensionales en los métodos de una Clase.</p> <p>Desarrollo El profesor plantea problemas en los que se utilicen arreglos unidimensionales en los métodos de una Clase, explica las estructuras de datos estática, el uso de arreglos unidimensionales en los métodos de una Clase y el uso de la estructura repetitiva for-each, el alumnado elabora los programas bajo la guía del profesor.</p> <p>Cierre En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución en forma ordenada y respetuosa.</p> <p>Extraclase Los alumnos resuelven problemas, planteados por el profesor, que empleen arreglos unidimensionales en los métodos de una Clase.</p>
<p>Desarrolla programas que involucren el uso de los arreglos unidimensionales en los métodos de una Clase.</p> <p>Codifica programas que involucren el uso de los arreglos unidimensionales en los métodos de una clase que involucren la estructura repetitiva for each.</p>	<p>Aplicación de arreglos unidimensionales.</p> <ul style="list-style-type: none"> • Acceso • Recorrido <p>Implementación.</p>	<p>Apertura En equipo los alumnos comparan el código de los problemas planteados y resueltos, el profesor revisa, opina y, de ser necesario, modifica el código de los programas.</p> <p>Desarrollo En plenaria los equipos exponen la solución a los problemas resueltos, opinan y comparan sus programas.</p>
		<p>Cierre Los alumnos resuelven un cuestionario relativo a arreglos unidimensionales en los métodos de una Clase.</p> <p>Extraclase Los alumnos investigan los conceptos relativos a arreglos bidimensionales.</p> <p>Apertura El profesorado propone al menos 3 problemas que se resuelvan empleando arreglos unidimensionales.</p>

		<p>Desarrollo En equipos de trabajo, el alumnado elige alguno de los problemas planteados relativos a arreglos unidimensionales en los métodos de una clase, bajo la estricta supervisión del profesor, el alumnado los desarrolla.</p> <p>Cierre En reunión plenaria los equipos de trabajo exponen la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>NOTA. Se recomienda plantear problemas relativos a:</p> <ul style="list-style-type: none"> • Calcular el promedio de un grupo de 10 estudiantes. • Con base a las calificaciones obtenidas de los 10 estudiantes: • Acomodarlas. • Menor a mayor. • Mayor a menor • Con base al primer apellido de los 10 estudiantes, acomodarlos en orden alfabético.
<p>Realiza programas que involucren el uso de los arreglos bidimensionales.</p> <p>Describe los conceptos de los arreglos bidimensionales en los métodos de una clase.</p>	<p>Arreglos bidimensionales.</p> <ul style="list-style-type: none"> • Concepto. • Declaración. • Uso. • Acceso • Recorrido 	<p>Apertura El profesor, tomando en cuenta la investigación realizada por los alumnos, explica el uso de arreglos bidimensionales y propone ejemplos.</p> <p>Desarrollo El profesor plantea problemas que empleen arreglos bidimensionales, los alumnos elaboran los programas correspondientes.</p> <p>Cierre En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución.</p> <p>Extraclase Los alumnos resuelven problemas, planteados por el profesor, que empleen arreglos bidimensionales y estructuras repetitivas.</p> <p>Apertura En equipo, el alumnado investiga los conceptos relativos a los arreglos bidimensionales.</p> <p>Desarrollo El profesorado tomando en cuenta la investigación previa realizada por el alumnado, explica el uso de arreglos bidimensionales en los métodos de una Clase y plantea problemas relativos a arreglos bidimensionales en los métodos de</p>

		<p>una clase, bajo la estricta supervisión del profesorado el alumnado los desarrolla.</p> <p>Cierre En reunión plenaria el alumnado expone la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p>
<p>Desarrolla programas que involucren el uso de los arreglos bidimensionales en los métodos de una Clase.</p> <p>Codifica programas que involucren el uso de los arreglos bidimensionales en los métodos de una clase.</p> <p>Elabora un proyecto que utilice las sentencias vistas hasta el momento, incluyendo arreglos unidimensionales y / o bidimensionales.</p>	<p>Aplicación de los arreglos bidimensionales:</p> <ul style="list-style-type: none"> • Con estructuras repetitivas. • Acceso. • Recorrido. <p>Implementación.</p>	<p>Apertura En equipo, los alumnos comparan el código de los problemas planteados y resueltos, el profesor revisa, opina y de ser necesario modifica el código de los programas.</p> <p>Desarrollo En plenaria los equipos exponen la solución a los problemas resueltos, opinan y comparan sus programas.</p> <p>Cierre Los alumnos resuelven un cuestionario relativo a arreglos bidimensionales en los métodos de una Clase.</p> <p>Extraclase Los alumnos investigan los conceptos relativos a la programación orientada a objetos.</p> <p>Apertura El profesorado propone al menos 3 problemas que se resuelvan empleando arreglos bidimensionales.</p> <p>Desarrollo En equipos de trabajo, el alumnado elige alguno de los problemas planteados relativos a arreglos bidimensionales en los métodos de una clase, bajo guía del profesor el alumnado los desarrolla.</p> <p>Cierre En reunión plenaria los equipos de trabajo exponen la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>NOTA. Se recomienda plantear problemas relativos a:</p> <ul style="list-style-type: none"> • Suma y resta de matrices. • Sumatoria de la diagonal principal de una matriz.

		<ul style="list-style-type: none"> • Con base en la producción trimestral de maíz, frijol, o cualquier otro producto de 10 diferentes estados de la República Mexicana, determinar: <ul style="list-style-type: none"> ▪ Producción anual de la República Mexicana ▪ Estado de la República Mexicana con menor y mayor producción anual ▪ Trimestre con menor y mayor producción. • Con base en el sueldo bruto de un empleado, calcular el impuesto sobre la renta que a retener.
<p>Desarrolla un proyecto que utilice las sentencias vistas hasta el momento, incluyendo los arreglos.</p> <p>Conoce el concepto de estructura de datos dinámica.</p> <p>Describe el concepto de ArrayList en los métodos de una clase.</p> <p>Codifica programas que involucren el uso de los ArrayList en los métodos de una clase.</p>	<p>Diseño de una aplicación.</p> <p>Planteamiento del problema:</p> <ul style="list-style-type: none"> • Diseño. • Desarrollo. <p>ArrayList</p> <ul style="list-style-type: none"> • Concepto. • Declaración. • Añadir elementos. • Eliminar elementos • Acceso a los elementos • Recorrido • Implementación. • 	<p>Apertura</p> <p>El profesor organiza a los equipos para la exposición de su aplicación.</p> <p>Desarrollo</p> <p>Los alumnos, en equipo, exponen el proceso que siguieron para desarrollar su aplicación y su ejecución.</p> <p>Cierre</p> <p>Todo el grupo observa los proyectos entregados y hacen comentarios para enriquecer los trabajos finales.</p> <p>Extraclase</p> <p>Los alumnos resuelven un cuestionario de los temas correspondientes a la unidad como autoevaluación.</p> <p>Apertura En equipo, el alumnado investiga los conceptos relativos a las estructuras de datos dinámicas en Java, ArrayList.</p> <p>Desarrollo El profesorado tomando en cuenta la investigación previa realizada por el alumnado, explica el uso de los ArrayList en los métodos de una clase y plantea problemas relativos a ArrayList bajo la estricta supervisión del profesorado el alumnado los desarrolla.</p> <p>Cierre En reunión plenaria el alumnado expone la solución de los problemas planteados, en forma ordenada y respetuosa, sus compañeros opinan y comparan su solución.</p> <p>NOTA. Se recomienda plantear problemas relativos a:</p> <ul style="list-style-type: none"> • Listado de personas con sus respectivos índices de masa corporal, para ello es necesario crear dos clases (“Persona” y “ListadoIMC”) en esta última clase se implementará un ArrayList de objetos persona.

		<ul style="list-style-type: none">• Programa de registro biológico: para ello se debe plantear la clase “Especie” con sus atributos nombre, tipo y población, además de crear la clase “RegistroBiologico” que haga uso de la clase ArrayList , con las funcionalidades de agregar y visualiza• Tienda virtual de libros: modelar las clases “Libro”, “Inventario” con sus respectivos atributos y métodos, así como la clase “TiendaVirtual”
--	--	--

Alumnado

- Blasco, F. (2019). *Programación orientada a objetos en Java*. Ediciones de la U.
<http://www.ebooks7-24.com.pbidi.unam.mx:8080/?il=10026>
- Joyanes, A. L., & Zahonero, M. I. (2014). *Programación en c, c++, java y UML. (2a. ed.)*. McGraw-Hill Interamericana.
<https://ebookcentral.proquest.com/lib/bibliodgbmhe/detail.action?docID=3225314>.
- Malik, D. S. (2013). *Programación java del análisis de problemas al diseño de programas (5a. ed.)*. CENGAGE Learning.
<https://ebookcentral.proquest.com/lib/bibliodgbsp/detail.action?docID=3430513>
- Prieto, N., Marqués, F., & Llorens, M. (2013). *Empezar a programar usando java (2a. ed.)*. Editorial de la Universidad Politécnica de Valencia.
<https://ebookcentral.proquest.com/lib/bibliodgbsp/detail.action?docID=3217647>
- Sznaidleder, P. (2020). *Java a Fondo. (4a. ed.)*. Alfaomega
<https://fadunam-bibliotecasdigitales-com.pbidi.unam.mx:2443/read/9786075386102/index?page=163>
- Olsson, M. (2013). *Java quick syntax reference*. Berkeley, California: Apress.
- Wu, C. (2008). *Programación en Java*. México: McGraw Hill.
- Ávila, J. y Baltazar, J. M. (2023). *Estructuras condicionales*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/estructuras-condicionales>
- Juventino, (2022). *Sintaxis y sentencias. En Lenguaje de programación orientado a objetos con Java*. Portal Académico del CCH, UNAM.
<https://portalacademico.cch.unam.mx/cibernetica2/programacion-java/sintaxis-sentencias>

Profesorado

- Blasco, F. (2019). *Programación orientada a objetos en Java*. Ediciones de la U.
<http://www.ebooks7-24.com.pbidi.unam.mx:8080/?il=10026>
 - Chávez, C. P. (2017). *Informática 2*. Pearson HispanoAmerica
<https://bookshelf-ref.vitalsource.com/books/9786073239073>
 - Joyanes, A. L., & Zahonero, M. I. (2014). *Programación en c, c++, java y UML. (2a. ed.)*. McGraw-Hill Interamericana.
<https://ebookcentral.proquest.com/lib/bibliodgbmhe/detail.action?docID=3225314>.
 - Malik, D. S. (2013). *Programación java del análisis de problemas al diseño de programas (5a. ed.)*. CENGAGE Learning.
<https://ebookcentral.proquest.com/lib/bibliodgbsp/detail.action?docID=3430513>
 - Prieto, N., Marqués, F., & Llorens, M. (2013). *Empezar a programar usando java (2a. ed.)*. Editorial de la Universidad Politécnica de Valencia.
<https://ebookcentral.proquest.com/lib/bibliodgbsp/detail.action?docID=3217647>
 - Sznaidleder, P. (2020). *Java a Fondo. (4a. ed.)*. Alfaomega
<https://fadunam-bibliotecasdigitales-com.pbidi.unam.mx:2443/read/9786075386102/index?page=163>
 - García, L., et al. (2003). *Construcción lógica de programas. Teoría y problemas resueltos*. México: Alfaomega.
 - Eckel, B. (2007). *Piensa en Java*. Madrid: Pearson–Prentice Hall.
 - Joyanes, L. (1990). *Problemas de metodología de la programación, 468 problemas resueltos*. México: Mc Graw–Hill.
- Software optional para el Desarrollo de la POO**
- BlueJ.
 - Replit (replit.com).
 - Eclipse.
 - NetBeans.

EVALUACIÓN.

En las estrategias sugeridas para cada sesión se proponen actividades que brindan evidencias del nivel del logro de los aprendizajes planteados. La evaluación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos obtenidos en cada sesión se sugieren los siguientes tipos e instrumentos de evaluación:

Diagnóstica

Conocer las características del grupo para trazar la ruta de implementación.

- preguntas formuladas al inicio de la unidad y de cada clase
- análisis de las respuestas dadas en los trabajos extraclase.
- Cuestionarios o actividades interactivas del portal académico.

Formativa.

Evaluación para la reorientación para el logro de los aprendizajes.

- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento de los alumnos.

Sumativa

Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios.

- Pruebas de ejecución: implementación de programas.
- Listas de verificación: para evaluar los trabajos extraclase.
- Observación: de las actividades realizadas en el salón de clase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Instrumentos de evaluación:

- Lista de cotejo
- Rúbrica
- Guía de observación
- Pruebas objetivas
- Pruebas de ejecución
- Autoevaluación
- Coevaluación

- Portafolio

Nota: La ponderación de cada actividad será asignada por el profesor.

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

El tema de estos dos pilares importantes dentro de la programación orientada a objetos se pospuso hasta esta unidad, esperando que el alumnado fuera asimilando los conceptos básicos, y finalmente para lograr reutilizar los objetos creados y darle un mayor dinamismo y simplificación al código se implementan la herencia y el polimorfismo.

<p>Propósito:</p> <p>Al finalizar la unidad el alumnado:</p> <p>Implementará programas en Java utilizando herencia, polimorfismo, y colaboración de Clases para aprovechar las bondades de la programación orientada a objetos como mecanismos de reutilización de código y especialización de clases.</p>	<p>Tiempo: 10 14 horas</p>
---	---------------------------------------

Aprendizajes	Temática	Estrategia sugeridas
<p>El alumno:</p> <p>Conoce el concepto de polimorfismo y constructor.</p> <p>Desarrolla programas que involucren polimorfismo y constructores.</p> <p>Conoce el concepto de herencia como mecanismo de especialización y reutilización de código</p>	<p>Concepto de polimorfismo.</p> <p>Concepto de constructor.</p> <p>Implementación de constructores con polimorfismo.</p> <p>Herencia y jerarquía de objetos: clase, superclase y subclase (diferencias de super y this).</p> <p>Agregación de atributos y métodos en clases descendientes</p>	<p>Apertura</p> <p>El profesorado explica el concepto de polimorfismo, constructores y su implementación.</p> <p>Desarrollo</p> <p>En equipo el alumnado resuelven un problema planteado por el profesor, tomando en cuenta las características del polimorfismo y constructores.</p> <p>Cierre</p> <p>Profesorado y alumnado en plenaria, los equipos presentan y explican la forma como resolvieron el problema.</p> <p>Extraclase</p> <p>El alumnado plantean y resuelven un problema que requiera la inicialización de atributos (constructor).</p> <p>Apertura</p> <p>Profesorado y alumnado: Revisar la clase Object y los wrapper de Java: Boolean, Float, Double, Integer en la documentación en línea de Java y señala las diferencias con los tipos primitivos. (Las clases wrapper y su diferencia con los tipos primitivos)</p> <p>Desarrollo:</p> <p>Profesorado y alumnado en plenaria se revisa conceptos básicos para identificar elementos de clases y de instancia, atributos y métodos, revise la congruencia en la nomenclatura (mayúsculas, minúsculas, capitalización); arme un árbol de clases de acuerdo con la jerarquía considerando los elementos nuevos y comunes. Introducir los símbolos UML para herencia.</p>

		<p><i>Docente y alumnos</i> (en equipo o individual): realizar la escritura de varios objetos realizados en clase y de tipos de datos primitivos. Por ejemplo <code>System.out.println(new Integer(5));</code>, aplicar algo parecido para las clases realizadas a lo largo del semestre.</p> <p>Profesorado: (opcionalmente) agregar un arreglo polimórfico de Object, realizar el recorrido de los objetos escribiéndolos a la pantalla.</p> <pre>for(Object o : miArreglo) System.out.println(o.getClass().getName()+ ' '+o);</pre> <p>Cierre:</p> <p>Profesorado: introducir los términos: clase, objeto (instancia), superclase, subclase, clase base, <code>super</code> y <code>this</code>.</p> <p>Profesorado y alumnado: Concluir la ubicación de las clases creadas previamente a lo largo del curso. ¿Por qué es posible utilizar el método <code>getClass()</code> en todos los objetos que nos son Object?, ¿Por qué los objetos aparecen en un mismo formato que parece no proporcionar información del contenido del objeto?</p> <p>Profesorado: Proponer una jerarquía de utilizando las clases previas en el curso</p> <p>Extraclase:</p> <p><i>Estudiantes:</i> Agregar métodos o atributos a subclases a partir de las clases planteadas en el curso. Por ejemplo: Agregar <code>desviación_Estándar</code> y otra información estadística a clase <code>Estadística</code> para aprovechar los datos almacenados previamente y el valor del promedio; <code>Círculo</code> y <code>Triángulo</code>, agregar <code>área</code> o <code>perímetro</code>, lo que no tengan; A partir de <code>Figuras2D</code>, crear <code>Figuras3D</code> agregando nuevos atributos y cálculos como el del volumen</p>
--	--	---

	<p>Colaboración de clases.</p>	<p>Apertura: Docentes y estudiantes: Revisar la actividad extraclase y aclarar dudas.</p> <p>Desarrollo: Docente y estudiantes: Crear clases de prueba, utilice arreglos polimórficos a partir de Object:</p> <pre>Object[] arregloDeObjetos = new Object[10]; Object[0] = new Float(6.0); Object[1] = new TrianguloEquilatero(1); //...</pre> <p>O simplemente:</p> <pre>Object unObjeto = new TriánguloEquilatero(1);</pre> <p>Revise cómo cambian los comportamientos cuando se trata de una super clase y su(s) subclase(s). Introducir los símbolos UML para colaboración.</p> <p>Cierre: Concluir que los objetos pueden estar compuestos por atributos de datos primitivos o bien contener objetos agregados al mismo.</p> <p>Extraclase: Estudiantes: colocar las variables declaradas dentro del método como atributos (privados) y observar el comportamiento, dividir las acciones de creación, asignación de valores para pruebas y escritura en métodos por separado, observar que el alcance en toda la clase. Realizar la escritura de las conclusiones.</p>
<p>Utiliza el concepto de polimorfismo como mecanismo de especialización y automatización</p>	<p>Sobrecarga (compilación estática, métodos con firmas distintas)</p>	<p>Apertura: Estudiantes y docente: en plenaria, discuten el concepto de sobrecarga estática; se explica el término de firma de un método y su importancia en la sobrecarga (polimorfismo estático)</p> <p>Desarrollo: El docente selecciona un problema que involucre la creación de una clase que requiera múltiples constructores, en una clase principal se crearán objetos utilizando los diferentes constructores de la clase, en conjunto responderán las siguientes preguntas:</p> <ul style="list-style-type: none"> ¿Porque debe(n) implementarse constructores en las subclases? ¿Qué utilidad tiene sobrecargar constructores? ¿Cuál es el pilar de la POO que permite que una clase pueda contener múltiples constructores?

		<p>Para crear múltiples constructores ¿En qué se deben diferenciar?</p> <p>¿Existen diferencias entre los objetos creados con los diversos constructores?</p> <p>¿Cuáles?</p> <p>El Docente selecciona una clase en la cual se deba sobrecargar uno de sus métodos para que funcione de manera distinta de acuerdo a su firma del método</p> <p>Cierre</p> <p>El docente resalta la importancia de la sobrecarga el polimorfismo dentro de la POO.</p>
	<p>Polimorfismo o sobreescritura, métodos con firmas iguales en distintas clases de una jerarquía y uso de super y this en el contexto de la sobre carga dinámica.</p>	<p>Apertura:</p> <p>En plenaria: Revisar la actividad extraclase.</p> <p>Desarrollo:</p> <p>Se revisa el ejemplo de escritura de objetos:</p> <pre>for(Object o : miArreglo) System.out.println(o.getClass().getName()+ ' '+o);</pre> <p>Revisar que en la práctica escribir un objeto puede hacerse también con o.toString()</p> <p>Revisar que clases implementan toString() (Float, Double, Integer, ...)</p> <p>Implemente en un par de clases desarrolladas en el semestre el método toString() y vea el comportamiento de la escritura, ¿se sobreescribe? A este mecanismo se conoce como polimorfismo (dinámico).</p> <p>Cierre:</p> <p>Implemente en el resto de las clases su método Implementación del método toString()</p> <p>Extraclase:</p> <p>Implemente otros métodos como redefinición de otros métodos: getArea() en Óvalo, getInterpretaciónIMC() en Persona y subclases,</p>
	<p>Encapsulamiento y modificador protected.</p>	<p>Apertura:</p> <p><i>Docente y estudiantes.</i> Revisan los ejercicios realizados en extraclase, ¿Qué debe implementarse el encapsulamiento (todos los atributos en private)? Plantee el uso de la visibilidad con protected. ¿En qué escenarios es correcto utilizar protected (no usar en atributos)?</p> <p>Desarrollo:</p> <p>Estudiantes: en equipo, realizar la correcta definición de los métodos getArea() utilizando getters y setter (modificar los que corresponda en getArea de la super y la sub clase.</p>

		<p>Desarrollo de otros métodos implementando la encapsulación correspondiente.</p> <p>Cierre: Docente y estudiantes: concluyen en que casos se utiliza this o super para el correcto llamado de los métodos.</p> <p>Extraclase: Estudiantes: Realizan un cuestionario de repaso con los temas vistos, realizan la implementación de dos subclases más, indicando porque eligieron cada visibilidad, en qué casos ocupan this o super y qué impacto tiene ocupar una u otra visibilidad o alcance</p>
	<p>Clases de interfaz y de implementación (métodos abstractos).</p>	<p>Apertura: <i>Docente y estudiantes.</i> Revisar los ejercicios realizados en extraclase.</p> <p>Desarrollo Profesorado y alumnado: Crear o recuperar, de los ejercicios previos, una Función $f(x)$ para tabular o para calcular una integral de Riemann o la derivada por aproximaciones, Establecer que debe modificarse $f(x)$ para poder realizarse las mis mismas acciones con otra función. Determinar que si se redefine $f(x)$ el algoritmo empleado es el mismo, implementar $f(x)$ como un método abstracto para forzar a las subclases a implementarla. Codificar la implementación como una clase aparte, como una clase contenida dentro de la clase principal y como una clase anónima embebida dentro del código.</p> <p>Cierre Profesorado y alumnado: agregar otro método a la clase base y que haga uso de $f(x)$, observar como las clases derivadas utilizan las relaciones de polimorfismo.</p> <p>Extraclase Estudiantes: implemente subclases para redefinir la descripción del índice de masa corporal de adultos a estudiantes de quince años, hombres y mujeres para implementarlo en otras poblaciones.</p>
	<p>Interpretación y modelado de una jerarquía de objetos.</p>	<p>Apertura: Profesorado y alumnado: Revisar los ejercicios realizados en extraclase, aclarar dudas y generar conclusiones.</p> <p>Desarrollo Profesorado y alumnado en equipos: Realice la jerarquía de clases (desde Object) de todas las clases desarrolladas en el curso.</p> <p>Cierre Profesorado y alumnado: realice conclusiones de qué principios de la programación orientada a objetos se implementan, indique en qué casos hay sobrecarga y sobreescritura (polimorfismo), qué diferencias hay entre usar this o super y a qué método se llama en cada clase dependiendo de la jerarquía.</p>

<p>Comprende la colaboración de Clases para la resolución de problemas.</p>	<p>Interacción y comunicación entre Clases.</p>	<p>Apertura Los alumnos investigan qué se requiere para interactuar y comunicarse entre Clases, en plenaria los alumnos concluyen cómo interactúan y se comunican las Clases en un programa.</p> <p>Desarrollo En equipo los alumnos resuelven un problema propuesto por el profesor, en el que se empleen al menos dos Clases.</p> <p>Cierre En plenaria, bajo la supervisión del profesor, los equipos presentan y explican la forma como resolvieron el problema.</p> <p>Extraclase Los alumnos resuelven un problema, planteado por el profesor, que requiera el uso de dos o más Clases, un constructor y alguna estructura repetitiva.</p>
<p>Desarrolla programas que involucren la colaboración de Clases.</p>	<p>Interacción y comunicación entre Clases.</p>	<p>Apertura En equipo, los alumnos comparan el código del trabajo extraclase, el profesor revisa, opina y, de ser necesario, modifica el código del programa.</p> <p>Desarrollo En plenaria los equipos exponen la solución al problema resuelto, opinan y comparan sus programas.</p> <p>Cierre En equipo, los alumnos proponen y resuelven un problema que requiera el uso de dos o más Clases, un constructor y alguna estructura repetitiva.</p> <p>Extraclase Los alumnos investigan los conceptos relativos a la herencia entre Clases.</p>

<p>Comprende el concepto de herencia en la resolución de un problema.</p>	<p>Herencia:</p> <ul style="list-style-type: none"> • Superclase. • Subclase. • Ventajas. 	<p>Apertura Los alumnos investigan, en equipo, el concepto de herencia. En plenaria el profesor solicita la opinión de los alumnos y destaca que una característica esencial de la herencia es que permite evitar la duplicidad de código, que aquello que es común a varias Clases se escribe sólo una vez (reutilizabilidad).</p> <p>Desarrollo El profesor plantea un problema mencionando las características de la superclase y de las subclases, los alumnos en equipo, lo resuelven.</p> <p>Cierre En plenaria, los equipos exponen y explican su solución al problema.</p> <p>Extraclase Los alumnos resuelven un problema, planteado por el profesor, en el que para su solución se requiera emplear la herencia, la colaboración entre Clases y alguna estructura repetitiva.</p>
---	---	---

Software optional para el Desarrollo de la POO

- BlueJ.
- Replit (replit.com).
- Eclipse.
- NetBeans.

Evaluación

La relación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos contenidos en cada sesión se sugieren los siguientes tipos e instrumentos de evaluación:

Diagnóstica

- Conocer las características del grupo para trazar la ruta de implementación.
- Preguntas formuladas al inicio de la unidad y de cada clase
- Análisis de las respuestas dadas en los trabajos extra-clase.
- Cuestionarios o actividades interactivas del portal académico...

Formativa.

- Evaluación para la reorientación para el logro de los aprendizajes.

- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento de los alumnos.
- Trabajo en equipo, respeto y responsabilidad.
- Citación adecuada de trabajo.

Sumativa

- Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios.
- Listas de verificación: para evaluar los trabajos extraclase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Nota: La ponderación de cada actividad será asignada por el profesor.

Unidad 4. Interfaz gráfica de usuario

En virtud de que todo software usado por el alumnado está compuesto de ventanas, botones de herramientas, menús, cuadros de diálogos es importante que cambie de un modo consola a un modo gráfico, por lo que, lo programado anteriormente lo transformará a través de la creación de objetos del paquete Swing para obtener un ambiente gráfico.

Propósito:

Al finalizar la unidad el alumnado

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

Implementará programas con colaboración de clases en Java para aplicar sus conocimientos de programación orientada a objetos utilizando interfaces gráficas de usuario con formularios y su comparación con las interfaces de consola.

Tiempo:

22 16 horas

Aprendizaje	Temática	Estrategias sugeridas
<p>El alumno:</p> <p>Conoce las características de la Clase <i>Swing</i>.</p> <p>Aplicará los mecanismos de colaboración entre clases para implementar interfaces de usuario (presentación) con las clases de la lógica (modelado).</p>	<p>Concepto de Interfaz gráfica de usuario (GUI).</p> <p>La Clase <i>Swing</i>.</p> <p>Componentes <i>javax.swing</i>.</p> <p>La Clase <i>AWT</i> como antecedente de la Clase <i>Swing</i>.</p> <p>Relación de la Clase <i>Swing</i> con la librería <i>AWT</i>: <i>java.awt.*</i> y <i>java.awt.event.*</i></p> <p>Interfaces de usuario: consola e interfaces formularios.</p> <p>Diagramas de secuencia (usuario, programa principal y clases de lógica).</p>	<p>Apertura</p> <p>El profesor explica el concepto de interface gráfica de usuario (GUI) y su relación con la Clase <i>Swing</i>.</p> <p>Desarrollo</p> <ul style="list-style-type: none"> Los alumnos hacen una investigación documental sobre los componentes <i>Swing</i>. El profesor presenta un ejemplo del uso de componentes. En plenaria, los alumnos, obtienen conclusiones. <p>Cierre</p> <p>El profesor explica que la Clase <i>Swing</i> tuvo como antecedente a la Clase <i>AWT</i> (<i>Abstract Window Toolkit</i>).</p> <p>Extraclase</p> <p>Los alumnos proponen un proyecto que se pueda solucionar usando etiquetas y botones.</p> <p>Apertura</p> <p>El profesor explica el concepto de interfaz de usuario y los tipos de interfaces (Consola y Formulario)</p> <p>Desarrollo</p> <p>El profesor plantea el diagrama de flujo de un problema sencillo donde el usuario</p>

		<p>interactúe con el programa principal y este con sus clases.</p> <p>El alumno planteará el esquema del diagrama de secuencia para la interfaz tipo consola</p> <p>El alumno planteará el esquema del diagrama de secuencia para la interfaz tipo Formulario</p> <p>Los alumnos observarán las diferencias y la necesidad de crear un formulario</p> <p>Cierre</p> <p>El profesor explicará los elementos fundamentales de los diagramas de secuencia para ambas interfaces.</p> <p>Extraclase</p> <p>Estudiantes: Revisan (en modo consola) que instrucciones involucran interacción en el diagrama de secuencia y/o de flujo una interacción entre el programa principal y el usuario, y entre el programa principal y las clases de la lógica. Preparan conclusiones para la siguiente sesión.</p>
	<p>Diferencias entre mecanismos de interacción en las interfaces de consola y formularios.</p> <p>Sistema de coordenadas en pantalla</p>	<p>Apertura</p> <p>Estudiantes y docente: Concluyen que las sentencias de lectura y escritura son interacciones del programa principal con el usuario, y los cálculos se quedan entre la lógica y el programa principal. Asocian que por cada instrucción de lectura (en modo consola) se requiere escribir algo en pantalla y leer un dato. Detallan el mecanismo de declaraciones, instanciación y uso de métodos de un objeto de la clase Scanner para tener acceso a los datos proporcionados por el usuario mediante System.in y System.out</p> <p>Desarrollo</p> <p>Docente: muestra algunos formularios de ejemplo.</p> <p>Docente y estudiantes: identifican los tipos de componentes de los formularios e indican qué tipo de información pueden recuperar (cadenas y booleanos). Analizan que elementos comunes y distintivos hay entre cada componente en una interfaz gráfica de usuario.</p> <p>Estudiantes: realizan una investigación para ver los mecanismos de conversión entre cadenas y otros tipos de datos.</p> <p>Docente: realiza un resumen de acciones que pueden realizarse con las cadenas para realizar conversiones de valores primitivos y arreglos.</p> <p>Cierre:</p> <p>Estudiantes y docente: Realizan ejemplos de conversiones de datos y revisan qué tipo de excepciones pueden generarse. Identifican que un get entre el usuario y el programa principal se traduce en un setter con la clase de la lógica y viceversa.</p> <p>Extraclase:</p> <p>Estudiantes: Realizan ejercicios de conversión entre datos primitivos y cadenas; hacen un listado de tipos de componentes que pueden encontrarse en una interfaz de usuario gráfica, características comunes y distintivas de cada componente.</p>

Aplicará los conceptos de herencia para modelar e interpretar la jerarquía de clases de AWT/Swing para la construcción de interfaces gráficas de usuario (GUI)

Elabora programas con una interfaz gráfica de usuario, aplicando las Clases: *JFrame*, *JLabel* y *JButton*

Jerarquía de clases de Swing

Clase *Swing*:

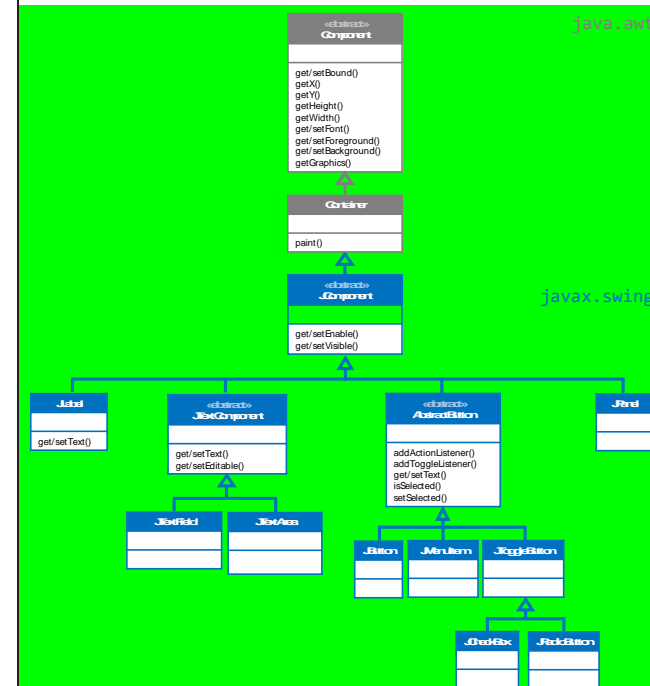
- *JFrame*.
- *JLabel*.
- *JButton*.

Apertura:

Docente y estudiantes: realizan una revisión de la actividad extraclase para identificar los tipos de componentes y proponer una jerarquía de clases que los describa.

Desarrollo:

Docente: introduce el siguiente diagrama y en plenaria ubican los elementos comunes y distintivos.



Docente y estudiantes realizan una revisión rápida de la documentación para ver otros métodos disponibles en cada clase

Apertura

- El profesor expone la importancia y usos de programas con interfaz gráfica de usuario.
- En profesor expone el uso de las Clases: *JFrame*, *JLabel* y *JButton*, para crear un programa con interfaz gráfica.

Desarrollo

Los alumnos, organizados en equipos, resuelven problemas sugeridos por el profesor, empleando las Clases expuestas.

Cierre

		<p>Los alumnos ejecutan los programas para verificar sus resultados.</p>
		<p>Extraclase El alumno plantea y elabora un proyecto que integre las Clases estudiadas para la realización de un programa con interfaz gráfica.</p>
<p>Propone un proyecto que utilice las Clases: JFrame, JLabel y JButton.</p>	<p>Contenedores: JFrame y JPanel Captura y presentación de texto: JLabel, JTextField, JTextArea Clase Swing:</p> <ul style="list-style-type: none"> • <i>JFrame.</i> • <i>JLabel.</i> • <i>JButton.</i> 	<p>Apertura: Estudiantes: Guiados por el profesor hacen una revisión conceptual de las interacciones GUI usando clases AWT/Swing de JFrame, JPanel, JLabel, JTextField y JTextArea utilizando la documentación oficial de Java que se encuentra en línea.</p> <p>Desarrollo: Explicación en plenaria de los conceptos clave: JFrame como contenedor principal, JPanel como subcontenedor, JLabel como herramienta para mostrar texto o imágenes de forma estática, JTextField como forma de entrada de texto y JTextArea como una entrada de texto multi renglón editable. El docente desarrolla un ejemplo sencillo en donde se crea una ventana principal mediante un JFrame y además crea un contenedor mediante el uso de JPanel, posteriormente agrega elementos JLabel, JTextField y JTextArea. A continuación, en equipo los alumnos desarrollan pequeños programas haciendo énfasis en usar su creatividad para hacer variaciones del ejemplo desarrollado por el profesor.</p> <p>Cierre: Los alumnos exponen por equipo el ejemplo que desarrollaron.</p> <p>Extraclase: Se les pide a los alumnos que ahora de forma individual realicen otro programa con la finalidad de afianzar los conocimientos vistos en esta sesión, en caso de dudas el profesor podrá tener una retroalimentación con ellos mediante la plataforma de comunicación que haya establecido (Teams, Classroom, Telegram, WhatsApp, etcétera)</p>

<p>Manejo de eventos: ActionListener y KeyListener</p>	<p>Apertura: Estudiantes y docente: En plenaria se discuten los eventos y los escuchadores (listener), los tipos y su utilidad en las interfaces de usuario en Java.</p> <p>Desarrollo: El docente propone un ejemplo donde se tenga que accionar un evento que involucre hacer clic en un botón, mediante un ActionListener se debe permitir captar el evento y realizar una acción. Por ejemplo, al presionar un botón de la interfaz, se tiene que cambiar el color del fondo del JPanel.</p> <p>El docente propone un ejemplo donde se tenga que presionar una tecla y mediante un KeyListener se debe permitir captar el evento y realizar una acción. Por ejemplo, al presionar una tecla, se tiene que visualizar en un JLabel todas las letras presionadas por el usuario.</p> <p>El grupo se organiza en equipos y resuelven diferentes ejercicios que involucren el uso de eventos y escuchadores.</p> <p>Cierre: Los alumnos exponen sus resultados al resto del grupo. En plenaria se comenta la importancia de las acciones y escuchadores dentro de las interfaces de usuario en Java. El alumno genera sus conclusiones.</p> <p>Extraclase: El docente propone un conjunto de ejercicios para reforzar los conocimientos que involucren acciones y escuchadores dentro de una Interfaz gráfica de usuario</p>
<p>Ejecución de acciones por medio de botones: JButton, JCheckBox, JRadioButton</p>	<p>Apertura: Estudiantes y docente: En plenaria se discuten los conceptos básicos, funcionalidad y características de elementos interactivos en Java, <i>JCheckBox</i>, <i>JButton</i> y <i>JRadioButton</i></p> <p>Desarrollo: Con práctica guiada, el grupo codifica un ejemplo de interfaz que contenga tres secciones: Selección de alternativas con <i>JCheckBox</i>, selección de una alternativa entre varias con <i>JRadioButton</i> y una última sección de procesamiento o confirmación con <i>JButton</i>. Se utiliza el método <i>addActionListener()</i> y <i>isSelected()</i> de <i>JRadioButton</i> para detectar las acciones del usuario</p> <p>Cierre: Se discute en plenaria los usos, ventajas y desventajas de los tres elementos</p>

<p>Elabora programas con interfaz gráfica de usuario aplicando las Clases: <i>JTextField</i>, <i>JTextArea</i> y <i>JComboBox</i>.</p>	<p>Clase Swing:</p> <ul style="list-style-type: none"> • <i>JTextField</i>. • <i>JTextArea</i>. • <i>JComboBox</i>. 	<p>Apertura En plenaria, los alumnos intercambian información sobre la investigación realizada extraclase y el profesor completa la información sobre el uso de las Clases: <i>JTextField</i>, <i>JTextArea</i>, <i>JComboBox</i> para crear un programa con interfaz gráfica.</p> <p>Desarrollo Los alumnos organizados en equipo resuelven problemas, sugeridos por el profesor, empleando tanto las Clases expuestas en esta sesión y las vistas anteriormente.</p> <p>Cierre Los alumnos ejecutan los programas para verificar sus resultados.</p> <p>Extraclase El alumno plantea y elabora un proyecto con interfaz gráfica que integre las Clases vistas en esta sesión más las estudiadas anteriormente.</p>
<p>Propone un proyecto que utilice las Clases: <i>JTextField</i>, <i>JTextArea</i> y <i>JComboBox</i>.</p>	<p>Clase Swing:</p> <ul style="list-style-type: none"> • <i>JTextField</i>. • <i>JTextArea</i>. • <i>JComboBox</i>. 	<p>Apertura En equipo los alumnos comparan el código del proyecto planteado y resuelto en el trabajo extraclase, el profesor revisa, opina y de ser necesario modifica el código del programa.</p> <p>Desarrollo En plenaria, los equipos exponen algunos de los proyectos elaborados, opinan y comparan sus programas.</p> <p>Cierre En equipo los alumnos proponen y resuelven un problema que para su solución requiera el uso de las Clases <i>JTextField</i>, <i>JTextArea</i> y <i>JComboBox</i> y algunas de las Clases previamente estudiadas.</p> <p>Extraclase Los alumnos afinan sus proyectos personales agregando lo aprendido en el cierre de esta sesión.</p>

<p>Elabora programas con interfaz gráfica de usuario aplicando las Clases: <i>JMenuBar</i>, <i>JMenu</i>, <i>JMenuItem</i>.</p>	<p>Clase Swing:</p> <ul style="list-style-type: none"> • <i>JMenuBar</i>. • <i>JMenu</i>. • <i>JMenuItem</i>. 	<p>Apertura En profesor expone el uso de las Clases: <i>JMenuBar</i>, <i>JMenu</i>, <i>JMenuItem</i>, para crear programas con interfaz gráfica de usuario.</p> <p>Desarrollo Los alumnos organizados en equipo resuelven problemas, sugeridos por el profesor, empleando las Clases expuestas.</p> <p>Cierre Los alumnos ejecutan los programas para verificar sus resultados.</p> <p>Extraclase El alumno plantea y elabora un proyecto con interfaz gráfica que integre las Clases vistas en esta sesión más las estudiadas anteriormente.</p>
<p>Elabora programas con interface gráfica de usuario aplicando las Clases: <i>JCheckBox</i>, <i>JRadioButton</i>.</p>	<p>Clase Swing:</p> <ul style="list-style-type: none"> • <i>JCheckBox</i>. • <i>JRadioButton</i>. 	<p>Apertura En profesor expone el uso de las Clases: <i>JCheckBox</i> y <i>JRadioButton</i>, para crear programas con interfaz gráfica de usuario.</p> <p>Desarrollo Los alumnos, organizados en equipo, resuelven problemas sugeridos por el profesor, empleando las Clases expuestas y las vistas con antelación.</p>
<p>Propone un proyecto que utilice las Clases: <i>setColor</i>, <i>drawLine</i>, <i>drawRect</i>, <i>drawRoundRect</i>, <i>drawOval</i>, <i>drawPolygon</i>, <i>fillRect</i>, <i>fillRoundRect</i>, <i>fillOval</i>, <i>fillPolygon</i>.</p>	<p>Preparación del proyecto.</p>	<p>Apertura En equipo los alumnos comparan el código del proyecto planteado y resuelto, el profesor revisa, opina y de ser necesario modifica el código del programa.</p> <p>Desarrollo En plenaria los equipos exponen algunos de los proyectos elaborados, opinan y comparan sus programas.</p> <p>Cierre Los alumnos afinan o corrigen sus proyectos tomando en cuenta las observaciones hechas por el profesor, así como por sus compañeros.</p> <p>Extraclase Los alumnos preparan la presentación de sus proyectos.</p>

<p>Desarrolla un proyecto que integre las Clases estudiadas en esta unidad.</p> <p>Integrar los conceptos de la POO en el desarrollo de una aplicación con GUI</p>	<p>Desarrollo de un proyecto.</p> <p>Proyecto separando lógica de presentación en un formulario con la jerarquía de clases Swing</p>	<p>Apertura</p> <p>En plenaria el grupo se pone de acuerdo para organizar la forma de exponer el proyecto.</p> <p>Desarrollo</p> <p>Los alumnos exponen ante el grupo el proyecto realizado, el profesor explica y cuestiona a los alumnos acerca de las características de los objetos y las características de la información que manejan.</p> <p>Cierre</p> <p>Los alumnos obtienen conclusiones de las exposiciones y entregan el reporte del proyecto.</p> <p>Usando alguna plataforma de IA generativa el grupo busca ejemplos prácticos que implementen una interfaz gráfica que contenga los elementos que se revisaron en la unidad de usuario.</p>
--	---	---

Alegsa, Leandro. (2023). Definición de GUI. Recuperado de <https://www.alegsa.com.ar/Dic/gui.php>

Hipertexto.info. (2019). La interfaz gráfica. [en línea] Disponible en <http://www.hipertexto.info/documentos/interfaz.htm>

Java México User Group. (2019). Código JAVA: Cambiar el color de fondo de un JFrame y JPanel. [Línea] https://www.javamexico.org/blogs/edbast/codigo_java_cambiar_el_color_de_fondo_de_un_jframe_y_jpanel

Software auxiliar

- BlueJ. <https://www.bluej.org/>
- Replit. <https://replit.com/>
- Eclipse. <https://www.eclipse.org/>
- NetBeans. <https://netbeans.apache.org/>

Evaluación

La relación de cada sesión se integra en la evaluación sumativa. Para evaluar los productos contenidos en cada sesión se sugieren los siguientes tipos e instrumentos de evaluación:

Diagnóstica

- Conocer las características del grupo para trazar la ruta de implementación.
- Preguntas formuladas al inicio de la unidad y de cada clase
- Análisis de las respuestas dadas en los trabajos extra-clase.
- Cuestionarios o actividades interactivas del portal académico...

Formativa.

- Evaluación para la reorientación para el logro de los aprendizajes.
- Pruebas de ejecución: codificación del programa.
- Observación: de las actividades realizadas en el salón de clases.
- Cuestionarios: para verificar el seguimiento o aprovechamiento de los alumnos.
- Trabajo en equipo, respeto y responsabilidad.
- Citación adecuada de trabajo.

Sumativa

- Análisis de los resultados obtenidos para determinar si se alcanzaron los niveles de aprendizajes previstos en el programa de estudios.
- Listas de verificación: para evaluar los trabajos extraclase.
- Rúbrica: para evaluar las investigaciones o exposiciones realizadas.
- Exámenes.

Nota: La ponderación de cada actividad será asignada por el profesor.