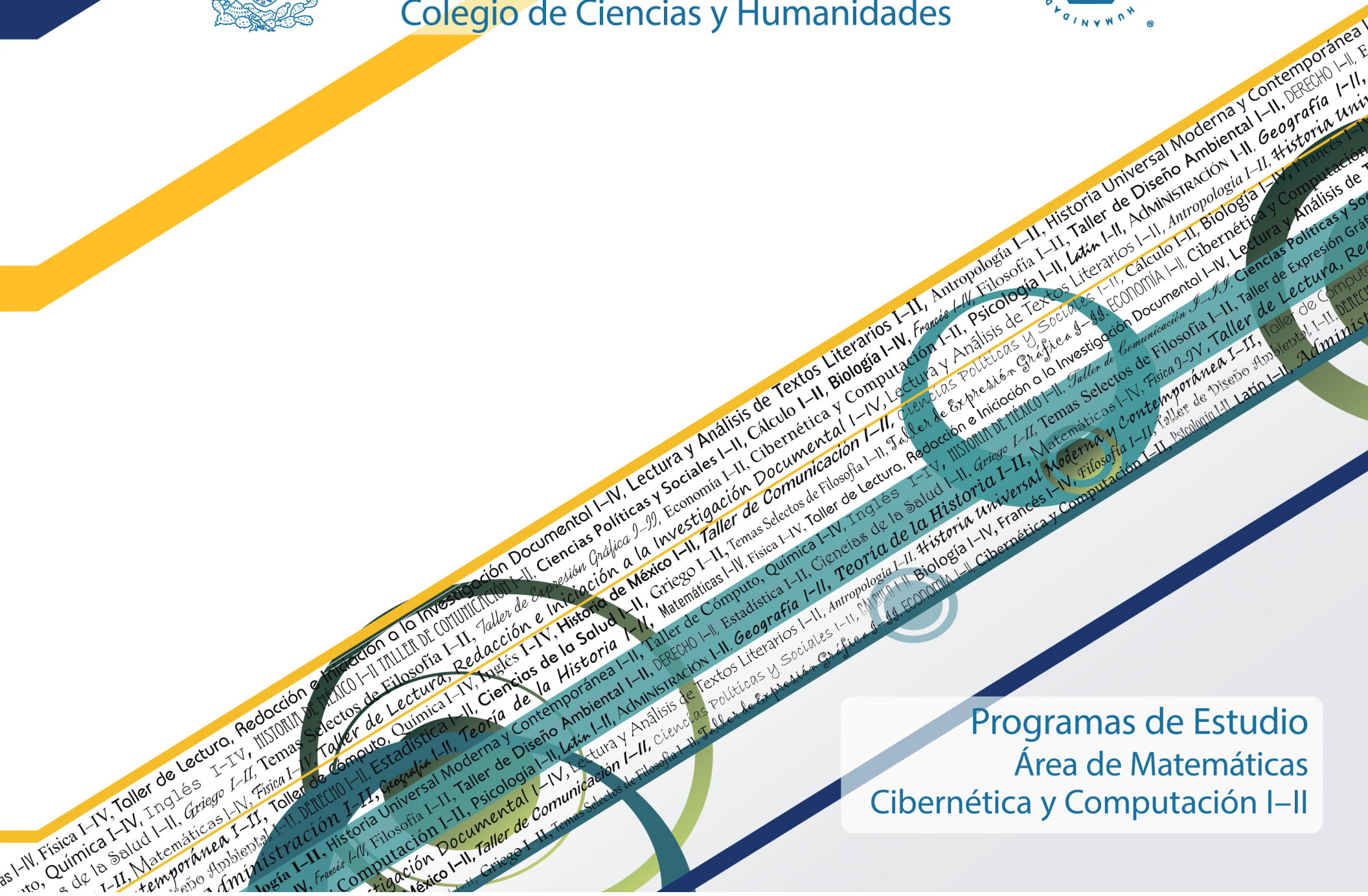
Universidad Nacional Autónoma de México
Escuela Nacional
Colegio de Ciencias y Humanidades



Programas de Estudio
Área de Matemáticas
Cibernética y Computación I–II

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. Circuitos lógicos 20
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 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. 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 estudiante para ingresar a la licenciatura con los conocimientos necesarios para su vida profesional) y está orientado a la formación intelectual ética y social de sus alumnos, considerados sujetos de la cultura y de su propia educación. Esto significa que la enseñanza dirigida al estudiante en la institución le fomentará actitudes y habilidades necesarias para que, por sí mismo, se apropie de conocimientos racionalmente fundados y asuma valores y opciones personales.[1]"

"Desde su origen el CCH adoptó los principios de una educación moderna donde consideró al estudiante como individuo capaz de captar por sí mismo el conocimiento y sus aplicaciones. En este sentido, el trabajo del docente del Colegio consiste en dotar al alumno de los instrumentos metodológicos necesarios para poseer los principios de una cultura científica–humanística.

El concepto de aprendizaje cobra mayor importancia que el de enseñanza en el proceso de la educación, por ello, la metodología aplicada persigue que aprenda a aprender, que la actividad receptiva y creadora no se malgaste y que adquiera capacidad auto informativa.

Para lograr el conocimiento auténtico y la formación de actitudes, el CCH trabaja con una metodología en la que participa el escolar activamente en el proceso educativo bajo la guía del profesor, quien intercambia experiencias con sus colegas en diferentes espacios académicos en su beneficio. De esta manera, el profesor no sólo es el transmisor de conocimientos, sino un compañero responsable del alumno al que propone experiencias de aprendizaje para permitir adquirir nuevos conocimientos y tomar conciencia creciente de cómo proceder para que por su propia cuenta y mediante la información, reflexión rigurosa y sistemática lo logre.

Lo anterior no le quita al docente su autoridad académica respaldada por sus experiencias, habilidades intelectuales y conocimientos.

Al ser un aprendizaje dinámico el promovido por el CCH, el escolar desarrollará una participación activa tanto en el salón de clases como en la realización de trabajos de investigación y prácticas de laboratorios.[2]"

El Colegio de Ciencias y Humanidades, su modelo de educación vanguardista y el desarrollo vertiginoso de

[1] http://www.cch.unam.mx/plandeestudios

[2] http://www.cch.unam.mx/misionyfilosofia

índice

la tecnología, resaltan el compromiso de proporcionar a sus alumnos la educación, el conocimiento y las habilidades que los orienten en la elección de su profesión, así como, a tener los conocimientos antecedentes necesarios para comprender las materias del nivel superior que elijan, a la vez que se enriquezca su cultura básica.

Es importante que el alumno que curse la materia de Cibernética y Computación adquiera los conocimientos fundamentales que le permitan comprender a la Cibernética como una ciencia interdisciplinaria, la cual incluye entre otras a la Computación, la cual le permitirá usar de forma eficiente, ética y segura las tecnologías de la información y la comunicación, asimismo, le permitirá modificar y utilizar la información en todos los campos de la actividad humana.

El alumno, 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 con la cibernética al estudiar los sistemas naturales y artificiales, el manejo de los circuitos lógicos y el álgebra de Boole. Asimismo, adquirirá una metodología para la 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 pertenecen al área de Matemáticas, deben cursarse en dos semestres (quinto y sexto), son opcionales u obligatorias de acuerdo con el área de elección. Los conocimientos adquiridos con antelación, más los aportados por esta materia, fomentarán en los alumnos, la reflexión de los procesos de aprendizaje y la construcción de conocimientos, haciendo énfasis en la investigación y en el desarrollo de habilidades.

El programa contempla tres unidades para la asignatura de Cibernética y Computación I y cuatro para la asignatura de Cibernética y Computación II, con un total de 64 horas por semestre, cursando 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.

Cibernética y Computación I:

En la unidad 1. “La cibernética”, se aborda primeramente desde el punto de vista histórico y se finaliza con el planteamiento de parte del alumno de un sistema.

En la unidad 2. “Circuitos lógicos”, se aborda el álgebra de Boole, las compuertas y circuitos lógicos y se finaliza con el planteamiento y desarrollo de un proyecto, por parte del alumno, que se resuelva utilizando circuitos lógicos.

En la unidad 3. “Metodología de solución de problemas e Introducción a la programación”, se adquiere una metodología que le permite al alumno resolver problemas utilizando el planteamiento de un algoritmo, representando su solución por medio de diagramas de flujo y su codificación en un lenguaje de programación.

Cibernética y Computación II:

En la unidad 1. “Lenguaje 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 programas.

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

En la unidad 3. “Polimorfismo, constructores, colaboración y herencia de clases”, se escriben programas en Java utilizando polimorfismo, constructores, colaboración y herencia de clases.

En la unidad 4. “Interfaz gráfica de usuario”, se desarrollan programas en Java utilizando interfaces gráficas de usuario conjuntamente con la temática aprendida en las unidades anteriores.

Se recomienda que los programas en Java que se sugieran resolver estén al alcance intelectual de los alumnos, el curso no pretende formar programadores, 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.

índice

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 alumno se auto evalúe, y a la vez, le dan información al profesor sobre el logro de los aprendizajes por parte del alumno. Dicha información permitirá dar a conocer al alumno sus avances y al profesor corregir la planeación de sus propuestas de enseñanza–aprendizaje.

Relaciones con el Área y otras asignaturas

Históricamente los pioneros de 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 la asignatura tiene estrecha relación con el área de Matemáticas. Se requiere del conocimiento de los sistemas de numeración, el álgebra de Boole, de los circuitos lógicos, de 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 al alumnos a acceder a conocimientos más especializados contenidos, entre otras asignaturas, a Cibernética y Computación; las asignaturas de Inglés de I a IV son requeridas principalmente en el estudio del lenguaje de programación ya que éste está escrito en Inglés; en Física I y II los alumnos estudian los temas de sistemas, circuitos, planteamiento y resolución de problemas. La asignatura tiene estrecha relación con Taller de Cómputo, los alumnos podrán usar un procesador de texto, una hoja de cálculo, harán investigaciones en Internet, en general harán uso de lo aprendido en el Taller. Con respecto a la asignatura de Taller de Lectura, Redacción e Iniciación a la Investigación Documental I–IV, los alumnos trabajan en el primer semestre con artículos de divulgación científica; en el tercer semestre realizan ensayos académicos; en el cuarto semestre realizan proyectos de investigación. Todo lo anterior conforma las bases para el estudio de la asignatura de 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. Esta asignatura le permite al alumno reflexionar sobre lo que está aprendiendo en otros campos del saber y tomarlo como tema de aplicación en la asignatura de Cibernética y Computación, los alumnos pueden proponer el diseño de sistemas relacionados con cualquier asignatura.

índice

Enfoque disciplinario y didáctico

La asignatura debe propiciar en los alumnos una visión general sobre la Cibernética y la Computación, sus avances, perspectivas y el aprovechamiento de las herramientas computacionales en la solución de problemas utilizando una metodología.

Enfoque disciplinario

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, así como el estudio de los sistemas naturales y artificiales, las partes que los constituyen, enfatizando que son elementos que podrá asociar a cualquier rama del conocimiento, tomando en cuenta que los sistemas se encuentran en cualquier actividad del ser humano.

Circuitos lógicos

Es importante conocer los sistemas de numeración: binario, octal y hexadecimal, ya que son la base del manejo de información en la computadora, así como el álgebra de Boole y su vínculo con el diseño y armado de circuitos lógicos.

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 alumno 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.

Cibernética y Computación II:

Lenguaje de programación orientada a objetos con Java

Reconocimiento de las características de la programación orientada a objetos y las del lenguaje de programación que las utiliza como “Java”, conociendo e implementando clases, con sus atributos, métodos e interfaces para poder crear objetos y así obtener programas sencillos con los principios básicos.

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, múltiples, así como la utilización de estructuras de control iterativas.

índice

Polimorfismo, constructores, colaboración y herencias de clases

Se profundiza en los conceptos de programación orientada a objetos como polimorfismo, constructores, interacción y comunicación entre clases, herencia como temas que refuerzan las características de la programación orientada a objetos con Java.

Interfaz gráfica de usuario

Desarrollo de interfaces visuales que permiten mostrar la solución de problemas de forma agradable a la vista y que engloban los aprendizajes adquiridos.

Enfoque didáctico

Las asignaturas de Cibernética y Computación I y II están enfocadas didácticamente para que los alumnos 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 intensión 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.

índice

Contribución al perfil del egresado

Las asignaturas colaboran sustancialmente con el perfil del egresado, mediante el desarrollo de conocimientos, habilidades y destrezas intelectuales, la evolución de su forma de pensamiento y la adquisición de valores, actitudes y normas, las asignaturas de Cibernética y Computación I y II propician entre otros:

- La valoración de la dimensión tecnológica de los conocimientos que adquiere y aplicación de los mismos en la solución de problemas.
- La valoración del conocimiento científico.
- La habilidad para el manejo de estrategias de solución de problemas usando la computadora.
- 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.
- 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 conocimiento para desarrollar software de aplicación a sus distintas materias.
- El compartir sus conocimientos con sus compañeros.
- El estar preparado para abordar nuevos conocimientos.

índice

Evaluación

La evaluación es un proceso cuya finalidad es determinar en qué medida se lograron los propósitos y aprendizajes marcados en el programa de estudios.

En ella se valora el desarrollo de cada una de las actividades que realiza alumno, el cual debe ser visto como el constructor de su propio aprendizaje. Todas las actividades serán analizadas y tomadas en cuenta por el profesor.

Se plantean tres tipos de evaluaciones: diagnóstica, formativa y sumativa:

La evaluación diagnóstica es la que se realiza al comienzo del curso. Consiste en la recolección de datos tanto de carácter personal como académico en la situación de partida, su finalidad es que el profesor inicie el proceso educativo con un conocimiento real de las características de cada uno de sus alumnos, lo que debe permitirle diseñar estrategias didácticas y acomodar su práctica docente a la realidad del grupo y de sus singularidades individuales.

Se recomienda llevar a cabo una evaluación diagnóstica al inicio del curso y de cada unidad. Puede consistir en un examen escrito o simplemente en un interrogatorio al grupo, consistente en plantear un tema de discusión y dejar que los alumnos opinen. Las actividades extra–clase también permiten diagnosticar si en general los alumnos están adquiriendo los aprendizajes.

La evaluación formativa permite valorar, a través de la recolección continua y sistemática de datos, el funcionamiento del 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 alumno 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 adquiera el aprendizaje. Es claro que el alumno puede llevar a cabo las investigaciones con éxito siempre y cuando el profesor le dé lineamientos detallados y que en cada momento lo esté asesorando, para ello existen varias herramientas, como: rúbricas, lista de verificación y la misma observación de las actividades.

La evaluación sumativa consiste en la recolección y valoración de datos al finalizar el curso escolar. Se alimenta de las evaluaciones diagnóstica y formativa, sirve para contrastar los resultados obtenidos con los propósitos y aprendizajes esperados. Permite evaluar el trabajo del profesor, del grupo y de cada alumno, puede ser el punto de arranque de la evaluación diagnóstica del siguiente periodo escolar.

índice

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 por cuenta propia.

Aprender a hacer

Durante el desarrollo de ambas asignaturas se elaboran prácticas de muy diversa índole que permiten desarrollar habilidades procedimentales. La solución de problemas permite que el alumno aborde diferentes situaciones, las analice y observe las relaciones entre los elementos del problema para obtener su solución, que descubra por él mismo o en equipo el funcionamiento de las herramientas o software 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.

índice

Propósitos generales de la materia

Al finalizar el curso el alumno:

- Aplicará los conceptos de la cibernética a lo largo del curso utilizando el concepto de multidisciplina para entender la relación del hombre y la máquina.
- 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.
- 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 el polimorfismo, los constructores, comunicación entre clases y herencia para construir programas.
- Utilizará los objetos para instrumentar propiedades de la programación orientada a objetos para realizar interfaces gráficas.

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.

índice

Contenidos Temáticos
Cibernética y computación I

Unidad	Nombre de la unidad	Horas
1	La Cibernética	14
2	Circuitos lógicos	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 de programación orientada a objetos con Java	10
2	Estructuras de control de secuencia en Java	22
3	Polimorfismo, constructores, colaboración y herencia de clases	10
4	Interfaz gráfica de usuario	22
	Total	64

índice

CIBERNÉTICA Y COMPUTACIÓN I
Unidad 1. La cibernética

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Modelará un sistema relacionado con un tema de alguna disciplina de su interés, analizando el concepto de cibernética para interrelacionarlo con otras ciencias y los elementos que conforman un sistema.	14 horas

Aprendizajes	Temática	Estrategias sugeridas
El alumno: Comprende la influencia de la cibernética en el desarrollo de la ciencia.	• Definición del concepto de cibernética. • Antecedentes de la cibernética. • Relación de la cibernética con otras ciencias. • Aplicaciones de la cibernética en la actualidad.	**Apertura** Los alumnos, en equipo, investigan los temas relativos a la cibernética, concepto, antecedentes históricos, la relación y aplicación con otras ciencias. **Desarrollo** • El profesor 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 profesor propicia un intercambio de ideas en donde los alumnos exponen su punto de vista relativo al material didáctico observado y los conceptos investigados. **Cierre** El alumno contesta un cuestionario de los conceptos relativos a la cibernética y sus aplicaciones. **Extraclase** Los alumnos buscan un video sobre la cibernética y sus aplicaciones y escriben una reseña.
Describe el trabajo científico sobre la cibernética de Norbert Wiener, Arturo Rosenblueth, Claude Shannon, entre otros.	**Obra de distintos autores en trabajos científicos sobre la cibernética:** • Norbert Wiener. • Arturo Rosenblueth. • Claude Shannon. • Otros.	**Apertura** Los alumnos, en equipo, investigan las aportaciones para el desarrollo de la cibernética de Norbert Wiener, Arturo Rosenblueth y Claude Shannon, entre otros.

índice

Aprendizajes	Temática	Estrategias sugeridas
		Desarrollo • El profesor propicia un intercambio de ideas en donde los alumnos exponen su punto de vista relativo a las aportaciones a la cibernética de Wiener, Rosenblueth, Shannon, entre otros. • Los alumnos obtienen conclusiones. **Cierre** Los alumnos elaboran una línea de tiempo con fechas, autores y aportaciones. **Extraclase** Los alumnos investigan el concepto, clasificación y ejemplos de sistema.
Comprende los componentes de un sistema.	**Sistemas:** • Concepto. • Elementos. • Ambiente. • Clasificación.	**Apertura** En equipo, los alumnos, exponen el tema relativo a los sistemas: concepto, elementos, ambiente y clasificación. **Desarrollo** Los alumnos unifican conceptos y obtienen conclusiones. **Cierre** El alumno elabora un mapa conceptual sobre la clasificación de los sistemas. **Extraclase** El alumno investiga y hace un reporte sobre qué es un sistema de control, lazo abierto, lazo cerrado y retroalimentación.
Comprende los componentes esenciales de un sistema de control	**Sistemas de control:** • Lazo abierto. • Lazo cerrado. • Retroalimentación.	**Apertura** Los alumnos en equipo exponen su reporte. **Desarrollo** Los alumnos, en plenaria, debaten y unifican criterios sobre los conceptos y obtienen conclusiones. **Cierre** El alumno contesta un cuestionario de los conceptos relativos a los sistemas**.** **Extraclase** El alumno investiga qué es un modelo.

índice

Aprendizajes	Temática	Estrategias sugeridas
Comprende el concepto y la importancia del modelo.	**Modelos:** • Concepto. • Tipos. - Naturales y artificiales. - Analógicos y digitales. - Matemáticos. - Conceptuales. • Relación.	**Apertura** Los alumnos, en equipo, investigan el tema relativo a modelos: concepto, tipos y relación. **Desarrollo** Los alumnos debaten y unifican criterios sobre el concepto, tipos y relación de los modelos. **Cierre** En equipo, los alumnos modelan, por medio de una representación gráfica, una maqueta, etcétera, un sistema sencillo, propuesto por el profesor. **Extraclase** El alumno propone un modelo de un sistema de alguna disciplina de su interés.
Desarrolla el modelo de un sistema.	**Elementos para modelar un sistema:** • Entrada y salida. • Proceso.	**Apertura** En equipos, los alumnos discuten si es viable la modelación a través de una representación gráfica, una maqueta, entre otros del sistema propuesto. Los alumnos analizan los elementos necesarios para representar su modelo. **Desarrollo** • El profesor resuelve dudas y realiza sugerencias. • Teniendo como base el modelo desarrollado, los equipos de trabajo lo construyen. **Cierre** Los alumnos analizan el funcionamiento del modelo del sistema desarrollado. **Extraclase** Los alumnos preparan una exposición del modelo del sistema desarrollado para presentarlo en clase.

Aprendizajes	Temática	Estrategias sugeridas
Explica cómo construyó el modelo del sistema, las partes que lo conforman y su funcionamiento.	**Desarrollo del proyecto.**	**Apertura** Organización de la exposición. **Desarrollo** Los alumnos, en equipo, exponen el funcionamiento de su proyecto, el por qué lo seleccionaron y en dónde lo van a utilizar. **Cierre** Comentan y obtienen conclusiones. **Extraclase** Los alumnos resuelven un cuestionario de toda la unidad como autoevaluación.

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.

índice

Referencias

Para el alumno

Básica

Ashby, W. (1977). *Introducción a la cibernética*. Buenos Aires: Nueva Visión.

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

Sluckin, W. (1956). *Cerebros y máquinas*. Buenos Aires, Argentina: Ediciones Galatea Nueva Visión.

Sluckin, W. (1971). *La cibernética*. Argentina: Ediciones Nueva Visión.

Sistema de Control [en línea]. Recuperado el 20 de enero de 2016 en <http://es.wikipedia.org/wiki/Sistema_de_control>.

Complementaria

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

Quintanilla, *S. Arturo Rosenblueth y Norbert Wiener: dos científicos en la historiografía de la educación contemporánea* [en línea]. Recuperado el 20 de enero de 2016 en <http://www.infoamerica.org/documentos_pdf/wiener1.pdf>.

Shannon, C. *Infoamérica* [en línea]. Recuperado el 31 de octubre de 2016 en <http://www.infoamerica.org/teoria/shannon1.htm>.*Teoría de Sistemas* [en línea]. Recuperado el 22 de febrero de 2016 en <http://www.aprendizaje.com.mx/TeoriaSistemas/Cibernetica/cibernetica.html>.

Wiener, N. *Infoamérica* [en línea]. Recuperado el 31 de octubre de 2016 en <http://www.infoamerica.org/teoria/wiener1.htm>.

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*. Barcelona, 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: Fondeo de Cultura Económica.

Sluckin, W. (1956). *Cerebros y máquinas*. Buenos Aires, Argentina: Ediciones Galatea Nueva Visión.

Wiener, N. (1998). *Cibernética o el control y comunicación en animales y máquinas*. Barcelona: Tusquets Editores.

Norbert Wiener y el origen de la cibernética [en línea]. Recuperado el 1 de febrero de 2016 en <http://www.infoamerica.org/documentos_pdf/wiener2.pdf>.

Unidad 2. Circuitos lógicos

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Utilizará el álgebra de Boole y el sistema de numeración binario para diseñar, construir o simular circuitos lógicos utilizando un protoboard o un simulador.	18 horas

Aprendizajes	Temática	Estrategias sugeridas
El alumno: Convierte números entre los sistemas de numeración binario, octal, decimal y hexadecimal.	**Sistemas de numeración.** • Binario, octal, decimal y hexadecimal. • Conversiones numéricas entre los sistemas: binario, octal, decimal y hexadecimal.	**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. **Desarrollo** Los alumnos, supervisados por el profesor, realizan conversiones numéricas entre los distintos sistemas de numeración investigados. **Cierre** Los alumnos resuelven ejercicios de conversión entre los distintos sistemas de numeración y comprueban sus resultados utilizando la calculadora. **Extraclase** Los alumnos investigan cómo se realizan las operaciones aritméticas básicas en el sistema de numeración binario.
Realiza operaciones aritméticas con el sistema de numeración binario.	**Aritmética del sistema de numeración binario.** • Operaciones de adición, sustracción, multiplicación y división.	**Apertura** En plenaria los alumnos obtienen conclusiones de la investigación extraclase, posteriormente el profesor explica el desarrollo de las operaciones aritméticas con números binarios. **Desarrollo** El alumno, orientado por el profesor, realiza operaciones aritméticas básicas en el sistema de numeración binario.

índice

Aprendizajes	Temática	Estrategias sugeridas
		Cierre 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. **Extraclase** Los alumnos investigan las operaciones booleanas básicas de conjunción, disyunción y negación y entregan un reporte.
Construye tablas de verdad de funciones booleanas.	**Elementos del álgebra de Boole:** • Variable booleana. • Operaciones básicas: conjunción, disyunción y negación. **Función booleana:** • Concepto. • Expresiones booleanas. **Tablas de verdad.**	**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. **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. **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. **Extraclase** El profesor proporciona una serie de ejercicios relativos a funciones booleanas para que los alumnos obtengan las tablas de verdad.
Simplifica funciones booleanas utilizando postulados y teoremas básicos.	**Simplificación de funciones booleanas.** • Postulados. • Teoremas básicos.	**Apertura** Los alumnos investigan y escriben una lista de postulados y teoremas básicos del álgebra de Boole. El profesor, en plenaria, explica el significado y el uso de los postulados y teoremas básicos

índice

Aprendizajes	Temática	Estrategias sugeridas
		Desarrollo En equipo, utilizando los postulados y teoremas básicos, los alumnos simplifican algunas funciones booleanas. Construyen y comparan las tablas de verdad de la función booleana original con la simplificada. **Cierre** En plenaria, cada equipo, explica cómo simplificaron las funciones booleanas. **Extraclase** Los alumnos, en equipo, investiga los conceptos de interruptor, circuito eléctrico, compuertas lógicas y circuito lógico
• Describe los conceptos de interruptor, circuito eléctrico, compuerta lógica y circuito lógico. • Aprende a utilizar el protoboard o un simulador.	**Interruptor y circuito eléctrico.** **Compuertas y circuitos lógicos:** • Compuerta: *And, Or y Not.* **Uso del protoboard o un simulador.**	**Apertura** El profesor solicita a los equipos que expliquen los conceptos de interruptor, circuito eléctrico, compuertas lógicas y circuito lógico para, después, obtener conclusiones. Los alumnos, en equipo, elaboran el diagrama y construyen o simulan un interruptor y un circuito eléctrico que sirva, por ejemplo, para encender y apagar las lámparas de su casa. El profesor explica el funcionamiento de los circuitos integrados y el uso del protoboard o de un simulador. **Desarrollo** En equipo, los alumnos relacionan la tabla de verdad AND, OR y NOT, con los diagramas de las compuertas lógicas y construyen o simulan los circuitos lógicos correspondientes. **Cierre** En equipo, los alumnos representan por medio de un diagrama lógico una función booleana y viceversa. **Extraclase** El alumno construye o simula funciones booleanas del tipo: AB', ABC', A'+B.

índice

Aprendizajes	Temática	Estrategias sugeridas
Construye la función booleana a partir de la tabla de verdad, empleando suma de productos.	**Obtención de la función booleana.** • Suma de productos.	**Apertura** Los alumnos investigan el concepto de suma de productos (mintérminos) 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.
Construye un semisumador.	**Semisumador.** • Planteamiento del problema (caja negra). • Obtención de la tabla de verdad. • Obtención de la función booleana. • Simplificación. • Construcción o simulación del circuito lógico.	**Apertura** Los alumnos, con ayuda del profesor, 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. **Desarrollo** En equipo los alumnos implementan el circuito lógico del semisumador, empleando un protoboard o un simulador y comprueban su funcionamiento. **Cierre** Los alumnos exponen el circuito lógico construido. **Extraclase** Los alumnos investigan y plantean el problema del sumador completo.

índice

Aprendizajes	Temática	Estrategias sugeridas
Construye un sumador completo.	**Sumador completo:** • 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.	**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. **Desarrollo** En equipo, los alumnos implementan el circuito lógico del sumador completo, empleando un protoboard o un simulador comprueban su funcionamiento. **Cierre** Los alumnos exponen el circuito lógico construido. **Extraclase** Los alumnos plantean y diseñan un problema de su interés que se pueda resolver empleando circuitos lógicos.
Diseña circuitos lógicos a partir de un problema cotidiano usando la metodología aprendida.	**Construcción del circuito lógico.**	**Apertura** En equipo los alumnos retoman el problema diseñado en extraclase y lo implementan empleando un protoboard o un simulador y comprueban su funcionamiento. **Desarrollo** Los alumnos, en equipo, presentan sus trabajos explicando los detalles. **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. **Extraclase** Los alumnos resuelven un cuestionario sobre los temas correspondientes a la unidad.

índice

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**: planteamiento del problema y elaboración de tablas de verdad.
- **Pruebas de ejecución**: construcción o simulación de circuitos.
- **Observación** de las actividades realizadas por los alumnos en el salón de clases.
- **Rúbrica**: los elementos que debe cubrir cada una de las exposiciones (presentación: planteamiento, diseño, funcionamiento del circuito presentado).
- **Exámenes**

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 alumno

básica

Kasatkin, V. (1976). *El ABC de la cibernética*. España: Paraninfo.

Circuitos eléctricos [en línea]. Recuperado el 27 de Enero de 2016 en <http://www.profesorenlinea.com.mx/fisica/Tecnologia/CIRCUITOS_ELECTRICOS.htm>.

Morris, M. *Diseño digital* [en línea]. Recuperado el 18 de mayo de 2016 en <https://urielectronics.files.wordpress.com/2010/11/diseno-digital-morris-mano-en-espanol.pdf>

UNAM, *Álgebra de Boole* [en línea]. Recuperado el 27 de enero de 2016 en <http://ecampus.fca.unam.mx/ebook/imprimibles/informatica/arquitectura_computadoras/Unidad_4.pdf>.

YouTube, *Compuertas lógicas* [en línea]. Recuperado el 27 de enero de 2016 en <https://www.youtube.com/watch?v=jp7qXsGPE&list=PLDE67521563B57B11>.

Complementaria

Heim, K. (1973). *Álgebra de los circuitos lógicos*. España: DOSSAT.

Boole y compuertas [en línea]. Recuperado el 27 de Febrero de 2016 en <http://docentes.uaa.mx/guido/wpcontent/uploads/sites/2/2014/10/Boole-y-compuertas.pdf>.

Circuitos eléctricos [en línea]. Recuperado el 27 de febrero de 2016 en <http://www.profesorenlinea.com.mx/fisica/Tecnologia/CIRCUITOS_ELECTRICOS.htm>.

Puertas lógicas [en línea]. Recuperado el 27 de febrero de 2016 en <http://electronikatualcance.blogspot.mx/2011/10/puertas-logicas-and-or-not.html>.

UNAM, *Álgebra de Boole* [en línea]. Recuperado el 27 de enero de 2016 en <http://ecampus.fca.unam.mx/ebook/imprimibles/informatica/arquitectura_computadoras/Unidad_4.pdf>.

índice

Para el profesor

Básica

Kasatkin, V. (1976). *El ABC de la cibernética*. España: Paraninfo.

Morris, M. (1988). *Lógica digital y diseño de computadores*. México: Prentice–Hall Hispanoamericana.

Tocci, J. (1987). *Sistemas digitales, principios y aplicaciones*. México: Prentice–Hall Hispanoamericana.

Amaya, L. (1989). *Manual electronics WorkBench* [en línea]. Recuperado el 27 de febrero de 2016, en <http://www.fisicacollazos.260mb.com/archivos/Manual%20Electronics%20WorkbenchECI.pdf?ckattempt=1>.

Autómatas y gramáticas [en línea]. Recuperado el 27 de febrero de 2016 en <http://www.monografias.com/trabajos16/automatas-y-gramaticas/automatas-y-gramaticas.shtml#ixzz3sIsIkK66>.

Giovanni, L. (2010). *Uso del protoboard en proyectos electrónicos* [en línea]. Recuperado 27 de febrero de 2016 en <https://zonaemec.files.wordpress.com/2013/06/tutorial-uso-del-protoboard.pdf>.

Puertas lógicas [en línea]. Recuperado el 27 de febrero de 2016 en <http://electronikatualcance.blogspot.mx/2011/10/puertas-logicas-and-or-not.html>.

Referencia complementaria para el profesor

Floyd, L. (1998). *Fundamentos de sistemas digitales*. España: Prentice–Hall Internacional.

Garza, J. *et al.* (2006). *Sistemas digitales y electrónica digital*. México: Pearson.

Heim, K. (1973). *Álgebra de los circuitos lógicos*. España: DOSSAT.

Morris, M. (1988). *Ingeniería computacional*. México: Prentice–Hall Hispanoamericana.

Puertas lógicas [en línea]. Recuperado el 27 de febrero de 2016 en <http://electronikatualcance.blogspot.mx/2011/10/puertas-logicas-and-or-not.html>

índice

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

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Aplicará la metodología de solución de problemas mediante la construcción de algoritmos y la codificación en el lenguaje de programación Java para tener una visión integral del proceso de solución.	32 horas

Aprendizaje	Temática	Estrategias sugeridas
El alumno: • Define el concepto de problema. • Identifica los elementos de un problema.	**Definiciones y conceptos generales de un problema.** **Elementos y relaciones del problema:** • Entrada. • Proceso. • Salida.	**Apertura** El alumno orientado por el profesor propone problemas de la vida cotidiana y sus posibles soluciones. **Desarrollo** El alumno, en equipo, investiga la definición y algunos elementos de un problema, así como sus relaciones. **Cierre** Los alumnos contestan un cuestionario sobre la temática, resaltando los elementos que tienen en común los problemas. **Extraclase** Los alumnos plantean un problema resaltando la entrada, proceso y salida.
Describe la diferencia entre problemas determinísticos, probabilísticos, secuenciales y condicionales.	**Tipos de problema:** • Determinísticos. • Probabilísticos. • Secuenciales. • Condicionales.	**Apertura** En equipo los alumnos investigan la temática y hacen un resumen. **Desarrollo** El profesor les presenta ejemplos de los cuatro tipos de problemas y los alumnos determinan el tipo de problema, mencionando su entrada y salida. **Cierre** En equipo, los alumnos plantean un problema de cada tipo. **Extraclase** Loa alumnos contestan un cuestionario que destaque las diferencias entre los cuatro tipos de problemas.

índice

Aprendizaje	Temática	Estrategias sugeridas
Conoce las etapas de la metodología de solución de problemas.	**Etapas de la metodología de solución de problemas:** • Planteamiento del problema. • Análisis del problema. • Diseño de la solución del problema: - Elaboración de algoritmos. - Representación del algoritmo a través de pseudocódigo y diagrama de flujo. - Prueba de escritorio.	**Apertura** En equipo, los alumnos, investigan el concepto y las etapas de la metodología de solución de problemas. **Desarrollo** En plenaria, los equipos junto con el profesor, formulan un problema y lo resuelven a través de todas las etapas de la metodología de solución de problemas. **Cierre** El profesor realiza un resumen de las etapas de la metodología de solución de problemas. **Extraclase** Los alumnos proponen un problema, desarrollan una posible solución y entregan una reseña.
Analiza el resultado de expresiones aritméticas utilizando la jerarquía de las operaciones.	**Expresiones y operadores aritméticos:** • Asignación. • Operadores aritméticos. • Jerarquía de operadores aritméticos. • Evaluación de expresiones aritméticas.	**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.
Construye expresiones lógicas utilizando operadores relacionales y lógicos.	**Expresiones y operadores relacionales y lógicos:** • Operadores relacionales. • Operadores lógicos. • Jerarquía de operadores lógicos. • Evaluación de expresiones lógicas.	**Apertura** En equipo los alumnos investigan la temática. **Desarrollo** El profesor organiza a los alumnos en equipos para que propongan ejemplos y expongan los temas.

índice

Aprendizaje	Temática	Estrategias sugeridas
		Cierre En equipo, los alumnos, resuelven ejercicios sobre expresiones lógicas. **Extraclase** Los alumnos proponen expresiones lógicas, para evaluarlas y analizar porque unas son verdaderas y otras falsas.
• Conoce el concepto de algoritmo, diagrama de flujo y pseudocódigo. • Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas secuenciales.	**Concepto de algoritmo, diagrama de flujo y pseudocódigo.** **Elaboración de algoritmos secuenciales.** **Representación de algoritmos secuenciales a través de diagramas de flujo y pseudocódigo.**	**Apertura** Los alumnos, investigan el concepto de algoritmo, diagrama de flujo y pseudocódigo. **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. **Cierre** En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema secuencial propuesto por el profesor. **Extraclase** Los alumnos proponen y construyen la solución de un problema secuencial. Entregan un reporte que incluya las etapas de la metodología de solución de problemas.
• 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.	**Lenguaje de programación Java:** • Historia del lenguaje. • Características: - Tipos de aplicaciones. - Arquitectura neutral. - Lenguaje orientado a objetos. - Disponibilidad de un amplio conjunto de bibliotecas. - Interpretado. - Robusto. - Distribuido.	**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. **Desarrollo** En plenaria los equipos exponen la investigación realizada y obtienen conclusiones. **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

índice

Aprendizaje	Temática	Estrategias sugeridas
	• Entorno de desarrollo del lenguaje de programación: - JDK (kit de desarrollo de Java). - IDE (interfaz de entorno de desarrollo).	de desarrollo del lenguaje de programación Java (Eclipse, Netbeans, JCreator, JBuilder, entre otros). Los alumnos realizan un programa de prueba en Java propuesto por el profesor. **Extraclase.** Contestan un cuestionario sobre la temática vista.
• Realiza programas empleando el método de salida de datos. • Realiza programas empleando la Clase *Scanner* para la entrada de datos.	**Pasos para implementar un programa con el lenguaje de programación Java y el entorno de desarrollo.** • Creación de un proyecto. • Declaración de la Clase. • Método *main*. • Empleo de los métodos *System.out. print y System.out.println*. • Errores sintácticos y lógicos. • Ejecución del programa. **Introducción de datos desde el teclado.** • La Clase *Scanner*. • Definición del objeto de la Clase *Scanner*. • Método *System.in*. • Tipos de datos primitivos.	**Apertura** • 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 *main*. • Los alumnos ejecutan el programa observando, el resultado. **Desarrollo** • En equipo, los alumnos investigan los conceptos relativos a la Clase *Scanner*, definición del objeto de la Clase *Scanner* y el método *System.in*. • El profesor da un ejemplo sobre el uso de la Clase *Scanner*. **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. **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.
Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales.	**Estructuras condicionales.** • Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas condicionales.	**Apertura** En equipo, los alumnos, investigan el concepto de estructura condicional. **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. **Cierre** En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema condicional propuesto por el profesor. **Extraclase** 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.

índice

Aprendizaje	Temática	Estrategias sugeridas
Construye programas de computadora que resuelvan problemas condicionales.	**Tipo de dato primitivo: Lógico.** **Sentencias condicionales:** • Simple *if.* • Doble *if – else.*	**Apertura** Los alumnos, investigan el concepto de Sentencia Condicional **Desarrollo** • 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. **Cierre** En equipo, los alumnos, escriben el código de un programa relativo a un problema condicional propuesto por el profesor y comprueban su funcionamiento. **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.
Elabora el algoritmo, diagrama de flujo y pseudocódigo para problemas condicionales múltiples.	**Elaboración de algoritmos, diagramas de flujo y pseudocódigo de problemas:** • Sentencias condicionales anidadas. • Condicionales múltiples.	**Apertura** En equipo, los alumnos investigan el concepto de estructura condicional anidada y múltiple. **Desarrollo** En plenaria, los alumnos obtienen conclusiones. A partir de una lista de problemas condicionales anidadas y múltiples, propuesta por el profesor, los alumnos auxiliados por él, encuentran su algoritmo, diagrama de flujo y pseudocódigo. **Cierre** En equipo, los alumnos construyen el algoritmo, diagrama de flujo y pseudocódigo de problemas condicionales anidados y múltiples propuestos por el profesor. **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.

Aprendizaje	Temática	Estrategias sugeridas
Construye programas de computadora que resuelvan problemas que involucren la toma de decisiones múltiples.	**Sentencias condicionales anidadas.** **Sentencias condicionales múltiples.** • Switch.	**Apertura** Los alumnos investigan el concepto de sentencia condicional anidada y múltiple. **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. **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. **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.
Elabora el algoritmo, diagrama de flujo y pseudocódigo para resolver problemas de estructura de ciclo.	**Estructuras de ciclo.** • 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.	**Apertura** En equipo, los alumnos investigan el concepto de estructuras de ciclo. **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. **Cierre** En equipo, los alumnos, construyen el algoritmo, diagrama de flujo y pseudocódigo de un problema de estructura de ciclo propuesto por el profesor. **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.

índice

Aprendizaje	Temática	Estrategias sugeridas
Construye programas de computadora que empleen la sentencia *for*.	**Estructuras de control de ciclo.** • Sentencia *for*.	**Apertura** Los alumnos, investigan el concepto de estructuras de control de ciclo *for*. **Desarrollo** En plenaria, los equipos obtienen conclusiones. A partir de la lista de problemas de estructuras de control de ciclo *for*, de la sección anterior, los alumnos escriben, auxiliados por el profesor, el código del programa correspondiente y comprueban su funcionamiento. **Cierre** En equipo, los alumnos escriben el código de un programa relativo a un problema de estructuras de control de ciclo *for* propuesto por el profesor y comprueban su funcionamiento. **Extraclase** Los alumnos proponen y construyen el código de un programa relativo a un problema de estructuras de control de ciclo *for*. Entregan un reporte que incluya el código y evidencias de la ejecución.
Elabora el algoritmo, diagrama de flujo y pseudocódigo de problemas de ciclo (cíclicos) que satisfagan una condición.	**Estructuras de ciclo.** • 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.	**Apertura** En equipo, los alumnos investigan el concepto de ciclo que satisfaga una condición. **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. **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. **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.

índice

Aprendizaje	Temática	Estrategias sugeridas
Construye programas de computadora que involucren la sentencia *while*.	**Estructura de control de ciclo.** • Sentencia *while*.	**Apertura** Los alumnos, investigan el concepto de Estructuras de control de ciclo *while*. **Desarrollo** En plenaria, los equipos obtienen conclusiones. A partir de la lista de problemas de estructuras de control de ciclo *while*, de la sección anterior, los alumnos escriben, auxiliados por el profesor, el código del programa correspondiente y comprueban su funcionamiento. **Cierre** En equipo, los alumnos escriben el código de un programa relativo a un problema de estructuras de control de ciclo *while* propuesto por el profesor y comprueban su funcionamiento. **Extraclase** Los alumnos proponen y construyen el código de un programa relativo a un problema que utilice estructuras de control de ciclo *while*. Entregan un reporte que incluya el código y evidencias de la ejecución.

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 extra clase y de preguntas formuladas al inicio de cada clase.

Formativa

- **Listas de verificación**: para evaluar los trabajos extra clase.
- **Pruebas de ejecución**: codificación del programa.
- **Observación**: de las actividades realizadas en el salón de clases.
- **Rúbrica**: para evaluar las investigaciones o exposiciones realizadas.
- **Exámenes.**

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.

índice

Referencias

Para el alumno

Básica

Cairó, O. (2003). *Metodología de la programación. Algoritmos, diagramas de flujo y programas*. México: Alfaomega.

Ceballos, F. J. (2006). *Java: curso de programación*. México: Alfaomega.

Eckel, B. (2007). *Piensa en Java.* Madrid: Pearson–Prentice Hall.

Joyanes, L. (2003). *Fundamentos de la programación. Algoritmos, estructura de datos y objetos*, México: Mc. Graw–Hill.

Moisset, D. (2016). *Curso de programación Java* [en línea]. Recuperado 27 de enero de 2016 en <www.javaya.com.ar>.

Complementaria

Sznajdleder, P. (2012). *Java a fondo.* México: Alfaomega.

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

Curso Java: *Código facilito* [en línea]. Recuperado el 24 de febrero de 2016 en <http://codigofacilito.com/cursos/JAVA>.

Edu Java. *Primer programa, con Eclipse en español. Tutorial 1 programación Java* [en línea]. Recuperado el 24 de febrero de 2016 en <https://www.youtube.com/watch?v=Y5QI2IAoIjw&index=1&list=PLBBE853AB00AF7765>.

Universo Java. *Aprende Java desde cero* (2015) [en línea]. Recuperado el 24 de febrero de 2016 en <http://universojava.com/tutorial-java-para-principiantes-2-instalar-el-eclipse-y-ejemplo-hola-mundo/>.

Para el profesor

Básica

Bores, R. (1995). *Computación, metodología, lógica computacional y programación*, México: McGraw–Hill.

Cairó, O. (2003). *Metodología de la programación. Algoritmos, diagramas de flujo y programas*. México: Alfaomega.

Eckel, B. (2007). *Piensa en Java*. Madrid: Pearson–Prentice Hall.

Garcia, L. *et al*. (2003). *Construcción lógica de programas. Teoría y problemas resueltos*. México: Alfaomega.

Joyanes, L. (1990). *Problemas de metodología de la programación*, *468 problemas resueltos*. México: Mc Graw–Hill.

Complementaria

Ceballos, F. J. (2006). *Java: curso de programación*. México: Alfaomega.

Sznajdleder, P. (2012). *Java a fondo.* México: Alfaomega.

Wu, C. (2001*). Introducción a la programación orientada a objetos con Java*. Madrid: McGraw Hill/ Interamericana de España.

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

Vogel, L. (2015). *Eclipse IDE–tutorial* [en línea]. Recuperado el 24 de febrero de 2016 en <http://www.vogella.com/tutorials/Eclipse/article.html>.

índice

CIBERNÉTICA Y COMPUTACIÓN II
Unidad 1. Lenguaje de programación orientada a objetos con Java

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Conocerá las características del lenguaje de programación orientado a objetos Java y su entorno de desarrollo, definiendo clases, atributos y métodos para la implementación de objetos en programas.	10 horas

Aprendizajes	Temática	Estrategias sugeridas
El alumno: • Conoce los conceptos básicos de la programación orientada a objetos • Conoce la organización general de un programa en Java como lenguaje orientado a objetos.	**Lenguaje de programación orientado a objetos** **Conceptos básicos de la programación orientada a objetos** • Identidad. • Atributos o características. • Comportamiento. • Abstracción. • Encapsulamiento. **Organización general de un programa en Java** • Comentarios. • Uso de bibliotecas. • Identificadores. • Palabras reservadas. • Sentencias. • Tipos de datos primitivos. • Bloque de código. • Operadores. • Expresiones.	**Apertura** 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. **Desarrollo** En plenaria los equipos exponen la investigación realizada y obtienen conclusiones. **Cierre** El profesor utilizando un programa ejemplo muestra a los alumnos la organización general de un programa Java. **Extraclase** Los alumnos investigan la definición de Clase, atributo y método, así como su implementación en el lenguaje Java.
• Describe los conceptos de Clase y atributo del lenguaje Java. • Implementa programas utilizando Clases y atributos.	**Clases:** • Definición. • Declaración.	**Apertura** En plenaria, el profesor propicia el intercambio de opiniones para definir qué es una Clase y sus atributos.

índice

Aprendizajes	Temática	Estrategias sugeridas
	• Atributos. - Definición. - Declaración. - Niveles de visibilidad. • Implementación.	**Desarrollo** Los alumnos escriben programas, utilizando el IDE, en donde codifiquen Clases y sus atributos. **Cierre** Los alumnos contestan un cuestionario. **Extraclase** Los alumnos investigan qué es un método e instanciación de objetos.
• Describe los conceptos de métodos del lenguaje java. • Conoce cómo instanciar objetos a partir de una Clase. • Implementa programas utilizando métodos.	**Métodos:** • Definición. • Declaración. • Parámetros. **Métodos:** • *Geter.* • *Seter.* **Objetos:** • Definición. • Declaración (instanciación). **Implementación.**	**Apertura** En plenaria, el profesor propicia el intercambio de opiniones para definir qué son los métodos e instanciación de objetos; ejemplifica la declaración y uso de los métodos e instanciación de objetos utilizando un ejercicio de la clase anterior. **Desarrollo** Los alumnos escriben programas en donde implementen los métodos e instancien los objetos de los ejercicios desarrollados previamente. Los alumnos ejecutan los programas observando, los posibles errores y el resultado obtenido. **Cierre** En equipo, los alumnos resuelven un problema propuesto por el profesor donde implementen: Clases, atributos y métodos. **Extraclase** • Los alumnos proponen y resuelven un problema.
Empleará la Clase *Scanner* para la entrada de datos en la creación de un programa.	**La Clase *Scanner.*** • Importar la Clase java.util.Scanner. • Definición del objeto de la Clase *Scanner* (instanciación). • Método system.in • Introducción de datos desde el teclado. **Errores sintácticos y lógicos.** **Ejecución del programa.**	**Apertura** El profesor da un ejemplo sobre el uso de la Clase *Scanner* y de cómo se instancia un objeto de ésta Clase para poder introducir datos desde el teclado. **Desarrollo** El profesor propone un ejemplo de entrada y salida de datos, haciendo énfasis en cómo se instancia un objeto de la Clase *Scanner.*

índice

Aprendizajes	Temática	Estrategias sugeridas
		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 *Scanner.* **Extraclase** Los alumnos proponen un problema y elaboran el programa que considere todos los conceptos vistos en la unidad.
Reafirma los conceptos adquiridos en la unidad.	**Implementa una aplicación.**	**Apertura** El profesor organiza a los alumnos para que presenten sus trabajos. **Desarrollo** Los alumnos exponen sus trabajos. **Cierre** Los alumnos obtienen conclusiones. **Extraclase** Los alumnos resuelven un cuestionario acerca de los conceptos vistos en la unidad.

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**: codificación del programa.
- **Observación** de las actividades realizadas en el salón de clases.
- **Rúbrica**: para evaluar las investigaciones o exposiciones realizadas.
- **Exámenes.**

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.

índice

Referencias

Para el alumno

Básica

Ceballos, F. (2015). *Java: interfaces gráficas y aplicaciones para Internet*. Paracuellos de Jarama, Ra–Ma.

Eckel, B. (2007). *Piensa en Java*. Madrid: Pearson–Prentice Hall.

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

Moisset, D. (2016). *Curso de programación Java* [en línea]. Recuperado 27 de enero de 2016 en <www.javaya.com.ar>.

Vogel, L. (2015). *Eclipse IDE–tutorial* [en línea]. Recuperado el 24 de febrero de 2016 en <http://www.vogella.com/tutorials/Eclipse/article.html>.

Complementaria

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

Olsson, M. (2013). *Java quick syntax reference*. Berkeley, California: Apress.

Streib, J. T. & Soma, T. (2014). *Guide to Java: a concise introduction to programming*. London: Springer.

Sznajdleder, P. (2012). *Java a fondo*. México: Alfaomega.

Weiss, M. A. & Weiss, M. A. (2013). *Estructura de datos en Java*. Madrid: Pearson Educación.

Para el profesor

Básica

Arnow, D. & Weiss, G. (2000). *Introducción a la programación con Java*. Madrid: Addison–Wesley

Villalobos J. A. (2008*). Introducción a las estructuras de datos: aprendizaje activo basado en casos: un enfoque moderno usando Java, UML, Objetos y Eclipse*. México: Pearson Educación.

Malik, D. S. (2013). *Programación Java: del análisis de problemas al diseño de programas*. Recuperado 29 de febrero de 2016 en <http://site.ebrary.com.pbidi.unam.mx:8080/lib/bibliodgbsp/detail.action?docID=11087780>.

Moisset, D. (2016). *Curso de programación Java* [en línea]. Recuperado 27 de enero de 2016 en <www.javaya.com.ar>.

Rodríguez, A. *Curso aprender programación Java desde cero* [en línea]. Recuperado 29 de febrero de 2016 en <http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188>.

Complementaria

Ceballos, F. (2006). *Java 2: curso de programación*. México: Ra–Ma.

Eckel, B. (2007). Piensa en Java. Madrid: Pearson–Prentice Hall

Sznajdleder, P. (2012). *Java a fondo.* México: Alfaomega.

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

Vogel, L. *Eclipse IDE – Tutorial* (2015) [en línea]. Recuperado el 24 de febrero de 2016 en <http://www.vogella.com/tutorials/Eclipse/article.html>.

índice

Unidad 2. Estructuras de control de secuencia en Java

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Utilizará las estructuras de control de secuencia para la resolución de problemas a través del lenguaje de programación orientado a objetos con Java.	22 horas

Aprendizajes	Temática	Estrategias sugeridas
El alumno: Desarrolla programas que involucren las estructuras condicionales simples, compuestas y anidadas en los métodos de una Clase.	**Estructuras condicionales:** • Simples: *if*. • Compuestas: *if–else* - Operadores: ○ Matemáticos. ○ Relacionales. ○ Lógicos. - Anidadas. **Implementación.**	**Apertura** • En equipo, los alumnos investigan los conceptos relativos a las estructuras condicionales simples, compuestas y anidadas. • El profesor proporciona ejemplos. **Desarrollo** El profesor plantea problemas que empleen estructuras condicionales simples, compuestas y anidadas en métodos de una Clase, haciendo énfasis en el uso de los distintos tipos de operadores. Los alumnos elaboran los programas correspondientes. **Cierre** El alumno propone un problema similar a los vistos anteriormente y elabora el programa para resolverlo. **Extraclase** En equipo, los alumnos investigan los conceptos relativos a la estructura condicional múltiple.
Desarrolla programas que involucren la estructura condicional múltiple en los métodos de una Clase.	**Estructura condicional múltiple:** • *Switch*. **Implementación.**	**Apertura** El profesor explica el uso de la estructura condicional múltiple en los métodos de una Clase y realiza ejemplos de su aplicación. **Desarrollo** El profesor plantea problemas que empleen la estructura condicional múltiple en los métodos de una Clase, los alumnos elaboran los programas correspondientes.

índice

Aprendizajes	Temática	Estrategias sugeridas
		Cierre En plenaria los alumnos exponen la solución dada a los problemas resueltos y el resto de los alumnos opinan y comparan su solución. **Extraclase** En equipo, los alumnos investigan los conceptos relativos a la estructura repetitiva *for*.
Desarrolla programas para resolver problemas que involucren la estructura repetitiva *for* en los métodos de una Clase.	**Estructura repetitiva *for*.** **Implementación.**	**Apertura** El profesor explica el uso de la estructura repetitiva *for* en los métodos de una Clase, tomando como base la investigación previa realizada por los alumnos, y desarrolla ejemplos de su aplicación. **Desarrollo** El profesor plantea problemas que empleen la estructura repetitiva *for* en los métodos de una Clase, los alumnos elaboran los programas correspondientes. **Cierre** En plenaria los alumnos exponen la solución dada a los problemas resueltos y el resto de los alumnos opinan y comparan su solución. **Extraclase** En equipo, los alumnos investigan los conceptos relativos a la estructura repetitiva *while*.
Desarrolla programas que involucren la estructura repetitiva *while* en los métodos de una Clase.	**Estructura repetitiva: *while*.** **Implementación.**	**Apertura** El profesor explica el uso de la estructura repetitiva *while* en los métodos de una Clase, tomando como base la investigación previa realizada por los alumnos, y desarrolla ejemplos de su aplicación. **Desarrollo** El profesor plantea problemas que empleen la estructura repetitiva *while* en los métodos de una Clase, los alumnos elaboran los programas correspondientes. **Cierre** En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución. **Extraclase** **En equipo los alumnos investigan los conceptos relativos a la estructura repetitiva** *do–while*.

índice

Aprendizajes	Temática	Estrategias sugeridas
Desarrolla programas que involucren la estructura repetitiva *do–while* en los métodos de una Clase.	**Estructura repetitiva: *do–while.*** **Implementación.**	**Apertura** El profesor ejemplifica el uso de la estructura repetitiva *do–while* en los métodos de una Clase, tomando en cuenta la investigación previa realizada por los alumnos. **Desarrollo** El profesor plantea problemas que empleen la estructura repetitiva *do–while* en los métodos de una Clase, los alumnos elaboran los programas correspondientes. **Cierre** En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución. **Extraclase** En equipo, los alumnos investigan los conceptos relativos a arreglos unidimensionales.
Resuelve problemas que involucren el uso de los arreglos unidimensionales en los métodos de una Clase.	**Arreglos unidimensionales:** • Concepto. • Declaración. • Tipos. • Uso.	**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. **Desarrollo** El profesor plantea problemas en los que se utilicen arreglos unidimensionales en los métodos de una Clase, los alumnos elaboran los programas correspondientes. **Cierre** En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución. **Extraclase** Los alumnos resuelven problemas, planteados por el profesor, que empleen arreglos unidimensionales en los métodos de una Clase.
Desarrolla programas que involucren el uso de los arreglos unidimensionales en los métodos de una Clase.	**Aplicación de arreglos unidimensionales.** • Acceso. • Recorrido. **Implementación.**	**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. **Desarrollo** En plenaria los equipos exponen la solución a los problemas resueltos, opinan y comparan sus programas.

índice

Aprendizajes	Temática	Estrategias sugeridas
		Cierre Los alumnos resuelven un cuestionario relativo a arreglos unidimensionales en los métodos de una Clase. **Extraclase** Los alumnos investigan los conceptos relativos a arreglos bidimensionales.
Realiza programas que involucren el uso de los arreglos bidimensionales.	**Arreglos bidimensionales.** • Concepto. • Declaración. • Uso.	**Apertura** El profesor, tomando en cuenta la investigación realizada por los alumnos, explica el uso de arreglos bidimensionales y propone ejemplos. **Desarrollo** El profesor plantea problemas que empleen arreglos bidimensionales, los alumnos elaboran los programas correspondientes. **Cierre** En plenaria los alumnos exponen la solución a los problemas resueltos, opinan y comparan su solución. **Extraclase** Los alumnos resuelven problemas, planteados por el profesor, que empleen arreglos bidimensionales y estructuras repetitivas.
Desarrolla programas que involucren el uso de los arreglos bidimensionales en los métodos de una Clase.	**Aplicación de los arreglos bidimensionales:** • Con estructuras repetitivas. • Acceso. • Recorrido. **Implementación.**	**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. **Desarrollo** En plenaria los equipos exponen la solución a los problemas resueltos, opinan y comparan sus programas. **Cierre** Los alumnos resuelven un cuestionario relativo a arreglos bidimensionales en los métodos de una Clase. **Extraclase** Los alumnos investigan los conceptos relativos a la programación orientada a objetos.

índice

Aprendizajes	Temática	Estrategias sugeridas
Desarrolla un proyecto que utilice las sentencias vistas hasta el momento, incluyendo los arreglos.	**Diseño de una aplicación.** **Planteamiento del problema:** • Diseño. • Desarrollo.	**Apertura** El profesor organiza a los equipos para la exposición de su aplicación. **Desarrollo** Los alumnos, en equipo, exponen el proceso que siguieron para desarrollar su aplicación y su ejecución. **Cierre** Todo el grupo observa los proyectos entregados y hacen comentarios para enriquecer los trabajos finales. **Extraclase** Los alumnos resuelven un cuestionario de los temas correspondientes a la unidad como autoevaluación.

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**: implementación de programas.
- **Observación** de las actividades realizadas en el salón de clases.
- **Rúbrica**: para evaluar las investigaciones realizadas.
- **Exámenes.**

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 es asignada por el profesor.

índice

Referencias

Para el alumno

Básica

Jiménez, J. (2014). *Fundamentos de programación, diagramas de flujo, diagramas N–S, pseudocódigo y Java*. México: Alfaomega.

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

Introducción a la programación Java, parte 1: conceptos básicos del lenguaje Java [en línea]. Recuperado 29 de febrero de 2016 en <http://www.ibm.com/developerworks/ssa/java/tutorials/j-introtojava1/>.

Moisset, D. (2016). *Curso de programación Java* [en línea]. Recuperado 27 de enero de 2016 en <www.javaya.com.ar>.

Sánchez J. (2016). *Java 2* [en línea]. Recuperado el 26 de febrero de 2016 en <http://www.jorgesanchez.net/programacion/manuales/Java.pdf>.

Complementaria

Ceballos, F. (2015). *Java: interfaces gráficas y aplicaciones para Internet*. Paracuellos de Jarama: Ra–Ma.

Olsson, M. (2013). *Java quick syntax reference*. Berkeley, California: Apress.

Sierra, M. (2014). *Programador Java certificado: curso práctico*. Madrid: Ra–Ma.

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

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

Para el profesor

Básica

Garcia, 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.

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

Martínez, J. *Fundamentos de programación en Java* [en línea]. Recuperado el 1 de febrero de 2016 en <http://bit.ly/1m9hed4>.

Complementaria

Ceballos, F. J. (2006). *Java: curso de programación*. México: Alfaomega.

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

Wu, C., (2001). *Introducción a la programación orientada a objetos con Java*. Madrid: McGraw Hill/ Interamericana de España.

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

Vogel, L. (2015). *Eclipse IDE–tutorial* [en línea]. Recuperado el 24 de febrero de 2016 en <http://www.vogella.com/tutorials/Eclipse/article.html>.

índice

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

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Implementará programas en Java utilizando polimorfismo, constructores, colaboración y herencia de Clases para aprovechar las bondades de la programación orientada a objetos.	10 horas

Aprendizajes	Temática	Estrategia sugeridas
El alumno: • Conoce el concepto de polimorfismo y constructor. • Desarrolla programas que involucren polimorfismo y constructores.	**Concepto de polimorfismo.** **Concepto de constructor.** **Implementación de constructores con polimorfismo.**	**Apertura** El profesor explica el concepto de polimorfismo, constructores y su implementación. **Desarrollo** En equipo los alumnos resuelven un problema planteado por el profesor, tomando en cuenta las características del polimorfismo y constructores. **Cierre** En plenaria, los equipos presentan y explican la forma como resolvieron el problema. **Extraclase** Los alumnos plantean y resuelven un problema que requiera la inicialización de atributos (constructor).
Comprende la colaboración de Clases para la resolución de problemas.	**Interacción y comunicación entre Clases.**	**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. **Desarrollo** En equipo los alumnos resuelven un problema propuesto por el profesor, en el que se empleen al menos dos Clases. **Cierre** En plenaria, bajo la supervisión del profesor, los equipos presentan y explican la forma como resolvieron el problema. **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.

índice

Aprendizajes	Temática	Estrategia sugeridas
Desarrolla programas que involucren la colaboración de Clases.	**Interacción y comunicación entre Clases.**	**Apertura** En equipo, los alumnos comparan el código del trabajo extraclase, el profesor revisa, opina y, de ser necesario, modifica el código del programa. **Desarrollo** En plenaria los equipos exponen la solución al problema resuelto, opinan y comparan sus programas. **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. **Extraclase** Los alumnos investigan los conceptos relativos a la herencia entre Clases.
Comprende el concepto de herencia en la resolución de un problema.	**Herencia:** • Superclase. • Subclase. • Ventajas.	**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). **Desarrollo** El profesor plantea un problema mencionando las características de la superclase y de las subclases, los alumnos en equipo, lo resuelven. **Cierre** En plenaria, los equipos exponen y explican su solución al problema. **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.

índice

Aprendizajes	Temática	Estrategia sugeridas
Desarrolla programas que involucren la herencia de Clases.	**Implementación de la herencia de Clases.**	**Apertura** En equipo los alumnos comparan el código del problema planteado y resuelto, el profesor revisa, opina y de ser necesario modifica el código del programa. **Desarrollo** En plenaria los equipos exponen la solución al problema resuelto, opinan y comparan sus programas. **Cierre** En equipo los alumnos proponen y resuelven un problema que requiera el uso de la herencia de Clases y lo implementan. **Extraclase** Los alumnos resuelven un cuestionario de los temas correspondientes a la unidad.

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 extra clase y de preguntas formuladas al inicio de cada clase.

Formativa

- **Listas de verificación**: para evaluar los trabajos extra clase.
- **Pruebas de ejecución**: codificación del programa.
- **Observación**: de las actividades realizadas en el salón de clases.
- **Rúbrica**: para evaluar las investigaciones o exposiciones realizadas.
- **Exámenes.**

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.

índice

Referencias

Para el alumno

Básica

Jiménez, J. (2014). *Fundamentos de programación, diagramas de flujo, diagramas N–S, pseudocódigo y Java*. México: Alfaomega.

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

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

Moisset, D. (2016). *Curso de programación Java* [en línea]. Recuperado 27 de enero de 2016 en <www.javaya.com.ar>.

Pérez G. (2008), *Aprendiendo Java y POO.* [en línea]. Recuperado 29 de febrero de 2016 en <http://es.slideshare.net/jpincay/aprendiendo-java-y-poo>.

Complementaria

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>.

Para el profesor

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.

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

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

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>

Complementaria

Ceballos, F. (2006). *Java: 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]. Recuperado 29 de febrero de 2016 en <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>.

Henao, C., *Ejemplo de polimorfismio* [en línea]. Recuperado 29 de febrero de 2016 en <http://codejavu.blogspot.mx/2013/06/ejemplo-de-polimorfismo-en-java.html>.

índice

Unidad 4. Interfaz gráfica de usuario

Propósito:	**Tiempo:**
Al finalizar la unidad el alumno: Desarrollará programas en Java utilizando interfaces gráficas de usuario para aplicar y ampliar sus conocimientos de la programación orientada a objetos.	22 horas

Aprendizaje	Temática	Estrategias sugeridas
El alumno: Conoce las características de la Clase *Swing.*	**Concepto de Interfaz gráfica de usuario (GUI).** **La Clase *Swing.*** **Componentes *javax.swing.*** **La Clase AWT como antecedente de la Clase *Swing.*** **Relación de la Clase *Swing* con la librería AWT: *java.awt.* y java.awt.event.****	**Apertura** El profesor explica el concepto de interface gráfica de usuario (GUI) y su relación con la Clase *Swing.* **Desarrollo** • Loa alumnos hacen una investigación documental sobre los componentes *Swing.* • El profesor presenta un ejemplo del uso de componentes. • En plenaria, los alumnos, obtienen conclusiones. **Cierre** El profesor explica que la Clase *Swing* tuvo como antecedente a la Clase AWT (*Abstract Window Toolkit*). **Extraclase** Los alumnos proponen un proyecto que se pueda solucionar usando etiquetas y botones.
Elabora programas con una interfaz gráfica de usuario, aplicando las Clases: *JFrame, JLabel y JButton*	**Clase *Swing*:** • *JFrame.* • *JLabel.* • *JButton.*	**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** Los alumnos ejecutan los programas para verificar sus resultados.

índice

Aprendizaje	Temática	Estrategias sugeridas
		Extraclase El alumno plantea y elabora un proyecto que integre las Clases estudiadas para la realización de un programa con interfaz gráfica.
Propone un proyecto que utilice las Clases: *JFrame, JLabel y JButton***.**	**Clase *Swing*:** • *JFrame.* • *JLabel.* • *JButton.*	**Apertura** En equipo los alumnos comparan el código del proyecto planteado y resuelto en extraclase, el profesor revisa, opina y de ser necesario modifica el código del programa. **Desarrollo** En plenaria los equipos exponen algunos de los proyectos elaborados, opinan y comparan sus programas. **Cierre** En equipo, los alumnos proponen y resuelven un proyecto que para su solución requiera el uso de las Clases *JFrame, JLabel y JButton* más el uso de alguna temática previamente estudiada. **Extraclase** Los alumnos investigan los conceptos relativos a las Clases *JTextField, JTextArea y ComboBox.*
Elabora programas con interfaz gráfica de usuario aplicando las Clases: *JTextField, JTextArea y JComboBox.*	**Clase *Swing*:** • *JTextField.* • *JTextArea.* • *JComboBox.*	**Apertura** En plenaria, los alumnos intercambian información sobre la investigación realizada extraclase y el profesor completa la información sobre el uso de las Clases: *JTextField, JTextArea*, *JComboBox* para crear un programa con interfaz gráfica. **Desarrollo** Los alumnos organizados en equipo resuelven problemas, sugeridos por el profesor, empleando tanto las Clases expuestas en esta sesión y las vistas anteriormente. **Cierre** Los alumnos ejecutan los programas para verificar sus resultados. **Extraclase** El alumno plantea y elabora un proyecto con interfaz gráfica que integre las Clases vistas en esta sesión más las estudiadas anteriormente.

índice

Aprendizaje	Temática	Estrategias sugeridas
Propone un proyecto que utilice las Clases*: JTextField, JTextArea y JComboBox.*	**Clase *Swing*:** • *JTextField.* • *JTextArea.* • *JComboBox.*	**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. **Desarrollo** En plenaria, los equipos exponen algunos de los proyectos elaborados, opinan y comparan sus programas. **Cierre** En equipo los alumnos proponen y resuelven un problema que para su solución requiera el uso de las Clases *JTextField, JTextArea y JComboBox* y algunas de las Clases previamente estudiadas. **Extraclase** Los alumnos afinan sus proyectos personales agregando lo aprendido en el cierre de esta sesión.
Elabora programas con interfaz gráfica de usuario aplicando las Clases: *JMenuBar, JMenu, JMenuItem.*	**Clase *Swing*:** • *JMenuBar.* • *JMenu.* • *JMenuItem.*	**Apertura** En profesor expone el uso de las Clases: *JMenuBar, JMenu, JMenuItem*, para crear programas con interfaz gráfica de usuario. **Desarrollo** Los alumnos organizados en equipo resuelven problemas, sugeridos por el profesor, empleando las Clases expuestas. **Cierre** Los alumnos ejecutan los programas para verificar sus resultados. **Extraclase** El alumno plantea y elabora un proyecto con interfaz gráfica que integre las Clases vistas en esta sesión más las estudiadas anteriormente.
Elabora programas con interface gráfica de usuario aplicando las Clases: *JCheckBox, JRadioButton.*	**Clase *Swing*:** • *JCheckBox.* • *JRadioButton.*	**Apertura** En profesor expone el uso de las Clases: *JCheckBox* y *JradioButton*, para crear programas con interfaz gráfica de usuario. **Desarrollo** Los alumnos, organizados en equipo, resuelven problemas sugeridos por el profesor, empleando las Clases expuestas y las vistas con antelación.

índice

Aprendizaje	Temática	Estrategias sugeridas
		Cierre Los alumnos ejecutan los programas para verificar sus resultados. **Extraclase** El alumno plantea y elabora un proyecto con interfaz gráfica que integre las Clases vistas en esta sesión más las estudiadas anteriormente.
Elabora programas con interfaz gráfica de usuario aplicando las Clases: *setColor, drawLine, drawRect, drawRoundRect, drawOval, drawPolygon.*	**Clase *Graphics*:** • *setColor.* • *drawLine.* • *drawRect.* • *drawRoundRect.* • *drawOval.* • *drawPolygon.*	**Apertura** En profesor expone el uso de las Clases: *setColor, drawLine, drawRect, drawRoundRect, drawOval, drawPolygon*, para crear un programa con interfaz gráfica. **Desarrollo** Los alumnos, organizados en equipo, resuelven problemas sugeridos por el profesor empleando las Clases expuestas y las vistas con antelación. **Cierre** Los alumnos ejecutan el programa para verificar sus resultados. **Extraclase** El alumno plantea y elabora un proyecto que integre las Clases vistas en esta sesión.
Elabora programas con interfaz gráfica de usuario aplicando las Clases: *fillRect, fillRoundRect, fillOval, fillPolygon*	**Clase Graphics:** • *fillRect.* • *fillRoundRect.* • *fillOval.* • *fillPolygon.*	**Apertura** En profesor expone el uso de las Clases: *fillRect, fillRoundRect, fillOval, fillPolygon*, para crear un programa con interfaz gráfica de usuario. **Desarrollo** Los alumnos, organizados en equipo, resuelven problemas sugeridos por el profesor, empleando las Clases expuestas y las vistas con antelación. **Cierre** Los alumnos ejecutan el programa para verificar sus resultados. **Extraclase** El alumno plantea y elabora un proyecto que integre las Clases vistas en esta sesión.

índice

Aprendizaje	Temática	Estrategias sugeridas
Propone un proyecto que utilice las Clases: *setColor, drawLine, drawRect, drawRoundRect, drawOval, drawPolygon, fillRect, fillRoundRect, fillOval, fillPolygon*.	**Preparación del proyecto.**	**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. **Desarrollo** En plenaria los equipos exponen algunos de los proyectos elaborados, opinan y comparan sus programas. **Cierre** Los alumnos afinan o corrigen sus proyectos tomando en cuenta las observaciones hechas por el profesor, así como por sus compañeros. **Extraclase** Los alumnos preparan la presentación de sus proyectos.
Desarrolla un proyecto que integre las Clases estudiadas en esta unidad.	**Desarrollo de un proyecto.**	**Apertura** En plenaria el grupo se pone de acuerdo para organizar la forma de exponer el proyecto. **Desarrollo** 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 manejaron. **Cierre** Los alumnos obtienen conclusiones de las exposiciones y entregan el reporte del proyecto.

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

- **Rúbrica de investigación:** componentes de las clases *Swing* y *Graphics*.
- **Listas de verificación:** criterios para evaluar el trabajo en equipo.
- **Reporte de realización de proyectos:** criterios para evaluar los proyectos individuales.
- **Participación individual**.
- **Exámenes**.

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 alumno

Básica

Eckel, B. (2007*). Piensa en Java*. Madrid: Pearson–Prentice Hall.

Berzal, F. *Componentes Swing* [en línea]. Recuperado el 27 de enero de 2016 en <http://elvex.ugr.es/decsai/java/pdf/D1-swing.pdf>.

Henao, C. *CoDejaVu* [en línea]. Recuperado el 27 de enero de 2016 en <http://codejavu.blogspot.mx/2013/08/que-es-java-swing.html>.

Java GUI la librería Swing [en línea]. Recuperado el 27 de enero de 2016 en <http://es.slideshare.net/lauriz19cour/java-gui-la-librera-swing-7932755>.

Uso de Java Swing [en línea]. Recuperado el 27 de enero de 2016 en <http://www.uv.mx/personal/mansuarez/files/2013/08/Uso-de-Java-Swing.pdf>.

Complementaria

García de Jalón, J. (2000). *Aprenda Java como si estuviera en primero.* España: Tecnum.

Olsson, M. (2013). *Java quick syntax reference*. Berkeley, California: Apress.

Sierra, M. (2014). *Programador Java certificado: curso práctico.* Madrid: Ra–Ma.

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

Uso de layouts [en línea]. Recuperado el 27 de enero de 2016 en <http://chuwiki.chuidiang.org/index.php?title=Uso_de_Layouts>.

índice

Para el profesor

Básica

Ceballos, F. (2015). *Java: interfaces gráficas y aplicaciones para Internet*. España: Ra–Ma

Ceballos, F. (2015). *Java. Interfaces gráficas y aplicaciones para Internet* [en línea]. Recuperado el 24 de febrero de 2016 en <http://es.scribd.com/doc/8773908/Ceballos-Java-2-Interfaces-Graficas-y-Aplicaciones-para-Internet-4Ed#scribd>.

Polinux. *Aplicaciones gráficas en Java librería Swing* [en línea]. Recuperado el 27 de enero de 2016 en <https://juanalbertogt.files.wordpress.com/2012/06/charlaswing.pdf>

Oracle, How to use HTML in swing components [en línea]. Recuperado el 27 de enero de 2016 en <http://docs.oracle.com/javase/tutorial/uiswing/components/html.html>.

Tutorials point, learn Java Swing [en línea]. Recuperado el 27 de enero de 2016 en <http://www.tutorialspoint.com/swing/index.htm>.

Complementaria

Calderón, M. *Swing, la solución actual de Java para crear GUIS* [en línea]. Recuperado el 27 de enero de 2016 en <http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html>.

Eckel, B. *Thinking in Java* (2012) [en línea]. Recuperado el 27 de Febrero de 2016 en <http://www2.uned.es/infor-3-programacion-concurrente/CDROM/Java/Thinking%20in%20Java%20-%203rd.pdf>.

FormDev Software GmbH. JFormDesigner 5.1 documentation [en línea]. Recuperado el 27 de enero de 2016 en <http://download.formdev.com/files/jformdesigner/5.1/JFormDesignerDoc-5.1.pdf>.

Oracle, Package javax.swing [en línea]. Recuperado el 27 de enero de 2016 en <http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>.

Oracle, Trail: Creating a GUI With JFC/Swing [en línea]. Recuperado el 27 de enero de 2016 en <http://docs.oracle.com/javase/tutorial/uiswing/index.html>.

índice

Dr. Enrique Graue Wiechers
Rector
Dr. Leonardo Lomelí Vanegas
Secretario General
Ing. Leopoldo Silva Gutiérrez
Secretario Administrativo
Dr. Alberto Ken Oyama Nakagawa
Secretario de Desarrollo Institucional
Dr. César Iván Astudillo Reyes
Secretario de Atención a la Comunidad Universitaria
Dra. Mónica González Contró
Abogada General
Mtro. Néstor Martínez Cristo
Director General de Comunicación Social

Dr. Jesús Salinas Herrera
Director General
Ing. Miguel Ángel Rodríguez Chávez
Secretario General
Lic. José Ruiz Reynoso
Secretario Académico
Lic. Aurora Araceli Torres Escalera
Secretaria Administrativa
Lic. Delia Aguilar Gámez
Secretaria de Servicios de Apoyo al Aprendizaje
Mtra. Beatriz A. Almanza Huesca
Secretaria de Planeación
Dra. Gloria Ornelas Hall
Secretaria Estudiantil
Dr. José Alberto Monzoy Vásquez
Secretario de Programas Institucionales
Lic. María Isabel Gracida Juárez
Secretaria de Comunicación Institucional
M. en I. Juventino Ávila Ramos
Secretario de Informática

Directores en Planteles:
Azcapotzalco **Lic. Sandra Guadalupe Aguilar Fonseca**
Naucalpan **Dr. Benjamín Barajas Sánchez**
Vallejo **Mtro. José Cupertino Rubio Rubio**
Oriente **Lic. Víctor Efraín Peralta Terrazas**
Sur **Mtro. Luis Aguilar Almazán**

Para la elaboración de este Programa se agradece la participación de: Everardo Beltrán Herrera, María del Carmen Calderón Villa, Jeanett Figueroa Martínez, Gilberto Fuentes Romero, Elías García Santillán, Juan Gutiérrez Quiroz, Armando Hernández Solís, Víctor Manuel Pérez Torres, Francisco Quezada Campos, Rafael Ramírez García, Asunción Reynoso Díaz, Cecilio Rojas Espejo, César Talavera Gómez, Ignacio Rafael Vázquez Torre.

45H
1971 2016
Escuela Nacional Colegio
de Ciencias y Humanidades