



FACULTAD
INGENIERÍA
& ARQUITECTURA

SYLLABUS

Metodología y Programación Orientada a Objetos

I. INFORMACIÓN GENERAL

Universidad:	Universidad Americana (UAM)	
Facultad:	Ingeniería y Arquitectura (FIA)	
Sede:	Managua	
Carrera:	Ingeniería en Sistemas de Información	
Asignatura:	Metodología y Programación Orientada a Objetos	
Período:	2S 2024	
Total de horas:	64 horas de atención docente, 180 horas totales	
Nombre del profesor:	Norman J. Cash Arcia	José A. Durán García
Horario de consultas:	Jueves, 3:30 – 4:30 p.m.	Horario acordado entre docente y estudiantes.
Correo electrónico:	norman.cash@uamv.edu.ni	jose.duran@uamv.edu.ni
Fecha de entrega:	19 de agosto de 2024	

II. OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Objetivos Genéricos:

- Utiliza tecnologías de la información y comunicación para aprender permanentemente, abstraer, analiza, sintetiza, identifica, plantea, resuelve problemas, investiga y aplica los conocimientos en la práctica.
- Se comunica óptimamente en español y en un segundo idioma, trabaja en equipo, motiva y conduce a metas comunes y desarrolla habilidades para trabajar en contextos internacionales.
- Se compromete con la calidad, actúa en nuevas situaciones, toma decisiones, innova y trabaja autónomamente.
- Es responsable, ético, demuestra responsabilidad social y compromiso ciudadano.

Objetivo general:

Durante el semestre el/la estudiante.

Desarrolla aplicaciones orientadas a objetos, basadas en el paradigma de la programación orientada a objetos como técnica para el desarrollo de software, a través del diseño y la programación de alta calidad, valorando la adquisición de destrezas, técnicas y hábitos de programación correctos en la construcción de programas basados en esta tecnología, considerando los principios éticos y la responsabilidad que implica la personalización de una solución en función del interés del usuario final.

Objetivos específicos:

1. Utiliza correctamente las reglas de sintaxis del lenguaje orientado a objetos Java para poder ejecutar programas con esta tecnología
2. Diseña clases en Java reusando componentes de software incluidos en las bibliotecas del lenguaje de programación, tales como clases y estructuras de datos.
3. Aplica el paradigma orientado a objetos para diseñar estos objetos programables, de tal modo que facilite el mantenimiento y extensión de aplicaciones de software.
4. Programa software integrando elementos reutilizables de las bibliotecas de Java, y elementos propios y crea aplicaciones utilizando el modelo MVC o Model Driven.



III. CONTRATO DIDÁCTICO

Estimado y estimada estudiante:

Con base en el **Reglamento de Régimen Académico y Disciplinario de Grado (2024)** y otras disposiciones estipuladas por la universidad, a continuación, se detallan los lineamientos a seguir durante el semestre para la asignatura Metodología y Programación Orientada a Objetos.

a. UAM-Virtual

UAM-Virtual es el Entorno Virtual de Aprendizaje, es la plataforma en la que los estudiantes y los docentes se apoyarán exclusivamente para el desarrollo de las actividades académicas. Es fundamental que cada estudiante haga uso adecuado de la plataforma UAM-Virtual para realizar las actividades académicas que corresponden al curso.

- El registro y control de asistencia será realizado mediante una actividad en el aula virtual.
- El registro de acumulados será realizado mediante la hoja de calificaciones en el aula virtual (oficial es UAM Class)
- Las consultas al docente se realizarán mediante foros o videollamadas en el aula virtual
- La entrega de tareas y realización de otras actividades se realizarán únicamente por medio del aula virtual, a menos que el docente lo indique de otra manera
- El material y herramientas a utilizarse en el curso estarán disponibles para descarga en el aula virtual

b. Honestidad académica

- Todos los trabajos académicos realizados por los estudiantes deben ser originales, evitando incurrir en plagio académico; para ello se deberá hacer uso de la norma APA (7ma Ed.) para referenciar y citar las fuentes bibliográficas utilizadas.
- Los estudiantes deberán evitar el fraude académico en todo sentido, el cual será penalizado con la pérdida de la calificación asignada a trabajos académicos, tanto individuales como grupales.
- Las herramientas de inteligencia artificial no son consideradas fuentes académicas válidas y en consecuencia los resultados producidos por estas no reemplazan la producción intelectual de los estudiantes, por tanto, quedará eliminado cualquier trabajo académico que sea literalmente el resultado de una producción de IA.



- Los entregables del curso deberán ser presentados en formato digital, por tanto, es responsabilidad de los estudiantes garantizar el acceso, disponibilidad, versión y buen estado de los archivos alojados en las diferentes plataformas: UAM-Virtual, Drive, y otros, o podrían ser objeto de anulación del puntaje asignado.

c. Normas de comportamiento para las clases

- Asistir mínimamente al 80% de las sesiones de clases, de lo contrario el estudiante reprobará la asignatura, perdiendo el derecho al registro de calificaciones y se reflejará en el acta oficial la nota cero (0).
- La justificación de inasistencias tendrá efectos para reprogramar actividades académicas y tareas, pero no para eliminar dicha inasistencia.
- Asistir puntualmente al aula de clase, y contar con todos los materiales de apoyo necesarios para su aprendizaje, así como participar activamente en las actividades de cada sesión de clases.
- Mantener una actitud respetuosa con sus compañeros, compañeras, docentes y con la comunidad universitaria en general, así como el cuidado de las instalaciones y equipamiento propiedad de la Universidad Americana.
- No se admite el ingreso de alimentos, bebidas, uso de dispositivos móviles, medios de entretenimiento o computadoras personales para fines ajenos a la clase, en el aula y laboratorio.
- Los únicos medios de comunicación válidos entre docente y estudiantes serán UAM-Virtual y el correo electrónico docente y estudiantil respectivamente: @uamv.edu.ni

d. Revisión de calificaciones

Los estudiantes deben dar seguimiento a sus acumulados, y pueden solicitar al docente la revisión de la calificación de una evaluación en un periodo no mayor a cinco días después de entregado el resultado, y de la calificación final de un corte evaluativo en un periodo no mayor a dos días después de publicada la calificación en UAM Class. Es responsabilidad de los estudiantes verificar que las calificaciones grabadas en UAM Class sean conforme a los resultados proporcionados por el docente en UAM Virtual.

IV. PLAN PROGRAMÁTICO DE ACTIVIDADES

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
I UNIDAD: Introducción al lenguaje Java y la orientación a objetos				
Sesión 1: 19 de agosto	Conocer los conceptos básicos de la programación orientada a objetos.	<p>Presentación de la asignatura y objetivos de aprendizaje.</p> <p>El paradigma de la programación orientada a objetos.</p> <p>Filosofía de objetos</p> <ul style="list-style-type: none"> ✓ Problema = Solución ✓ Pensando en objetos: abstracción de objetos del mundo real 	<p>Conferencia Presentación de la asignatura y expectativas</p> <p>Evaluación diagnóstica Los estudiantes contestan una serie de preguntas donde el docente explora los conocimientos previos</p> <p>Clase demostrativa. Mediante la creación de ejemplos sencillos se muestra el uso de entrada y salidas por pantalla, así como el uso de los principales operadores.</p> <p>Lectura individual. Los estudiantes realizan una lectura del material proporcionado por el docente al igual replican los ejercicios propuestos en la lectura y lo guardan en su portafolio digital.</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°1</p> <p>Material de Lectura: Deitel, P & Deitel, H. (2016) Cómo Programar en Java (10. a ed.). PEARSON EDUCACIÓN México.</p> <p>Herramientas de software: UAM-Virtual</p>
Sesión 2		<p>Introducción al lenguaje Java.</p> <p>Elementos del lenguaje.</p> <p>Programación de un programa sencillo.</p> <p>Diagrama de Clases</p> <p>Recorrido con IntelliJIDEA.</p> <p>Estructura de un programa (paquetes, clases, archivos), reglas básicas.</p>	<p>Control de lectura. mediante una serie de preguntas al azar.</p> <p>Presentación de diapositivas.</p> <p>Clase demostrativa. Mediante ejemplos sencillos se explica la estructura básica de un programa en java, reglas básicas y formas de compilar y ejecutar.</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°1</p> <p>Material de Lectura: Deitel, P & Deitel, H. (2016) Cómo Programar en Java (10. a ed.). PEARSON EDUCACIÓN México.</p> <p>Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub</p>

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
		Formas de compilar y ejecutar un programa en java.		
Sesión 3		<p>Diseño e Implementación de un Diagrama de Clases.</p> <p>Tipos de relaciones: asociación, agregación, composición, generalización</p> <p>Objetos y colaboraciones</p>	<p>Conferencia dialogada. El docente explica en qué consiste la programación orientada a objetos.</p> <p>Práctica. El docente realiza ejercicios de abstracción de clases, relaciones y objetos</p> <p>Demostración. Mediante ejemplos de programación previamente programados, el docente explica la filosofía de objetos.</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°2</p> <p>Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación</p> <p>Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub</p>
Sesión 4		Operadores, Sentencias condiciones y recursivas.	<p>Control de lectura: el docente realiza preguntas sobre la lectura orientada en la sesión anterior.</p> <p>Presentación con diapositivas.</p> <p>Clase demostrativa. Mediante una serie de ejercicios el docente explica el uso de la estructura de control if, else, if anidados.</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°2</p> <p>Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación</p> <p>Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub</p>
Sesión 5		<p>Clases: encapsulación, constructores, métodos.</p> <p>Enum</p> <p>Herencia</p>	<p>Clase demostrativa. El docente se apoya de un proyecto de ejemplo para explicar la implementación y creación de las clases.</p> <p>Implementación del proyecto pizza, implementando herencia, condicionales, encapsulación y Enumeradores</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°3</p> <p>Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación</p>

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
Sesión 6				Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub
Sesión 7		Sentencia de control switch Sentencia de control: for, while y do ... while	Clase demostrativa. Mediante una serie de ejercicios el docente explica la aplicación de clases, enum, constructores. Revisión de Proyectos de curso en clase. Implementación del Proyecto pizza. Asignación de tarea para el primer corte.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°4 Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub
Sesión 8		Arreglos y Objeto ArrayList ✓ Arreglo unidimensional ✓ Arreglos multidimensional ✓ Cadena de caracteres Colecciones y la clase ArrayList ✓ Creación ✓ Métodos	Clase demostrativa. El docente explica arreglos y cadenas en Java y mediante la técnica learning by doing, los estudiantes aplican estos para crear objetos más complejos. Clase demostrativa. El docente mediante ejemplos de explica el uso de la clase ArrayList y los iteradores	Documentos académicos: Syllabus de la asignatura Guía didáctica N° 4 Guía didáctica N° 5 Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación
Sesión 9		Iteradores Clase String. Interfaz	Presentación con diapositivas	Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
Sesión 10		I Corte	Evaluación I Corte	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N° 5</p> <p>Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación</p> <p>Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub</p>
II UNIDAD: Persistencia de Datos con Java				
Sesión 11	Conocer que es JDBC. Conocer como accede Java a los datos. Conocer que es JPA.	Que es JDBC. Que es JPA. Patron Singleton. Patron DAO.	<p>Clase demostrativa. Mediante el proyecto de clase el docente implementa JPA.</p> <p>Presentación con diapositivas.</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°6</p> <p>Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación</p> <p>Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub</p>
Sesión 12			<p>Clase demostrativa. Mediante el proyecto de clase el docente implementa JPA.</p>	<p>Documentos académicos: Syllabus de la asignatura Guía didáctica N°6</p> <p>Material de Lectura: Deitel, P., Deitel, H. (2016). Java. Cómo programar. 10ma. Edición. México: Pearson Educación</p>

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
				Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub
Sesión 13		Persistence API. Relaciones OneToMany. ManyToOne. ManyToMany. Collections. Master-Details.	Clase demostrativa. El docente enseña a persistir la información con relaciones.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°7 Material de Lectura: https://www.ibm.com/docs/es/was-liberty/nd?topic=liberty-java-persistence-api-jpa Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub
Sesión 14		Composite Keys. Enum Listener.	Clase demostrativa. El docente utilizando la metodología Learning by doing explica la temática a abordar.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°7 Material de Lectura: https://www.ibm.com/docs/es/was-liberty/nd?topic=liberty-java-persistence-api-jpa Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
Sesión 15		Practica de JPA.	Clase demostrativa. El docente utilizando la metodología Learning by doing explica la temática a abordar.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°8 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate Git - GitHub
III UNIDAD: Creando UI con OPENXAVA.				
Sesión 16	Conocer e Implementar el paradigma Model Driven con Openxava. Crear un proyecto con Openxava.	Que es Model Driven. Que es una aplicación Monolitica. Desarrollo Agil. Conociendo un modelo de Dominio sencillo con Openxava.	Presentación. Ejercicio Guiado. Explicación y Discusión en clase.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°8 Guía didáctica N°9 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub
Sesión 17		Calculando valores por defecto. Refinando la interfaz de usuario. Herencia de Superclases Mapeadas. Herencia de Entidades. Herencia de Vista.		
Sesión 18		Lógica de Negocio. Propiedades Calculadas.		
Sesión 19		Validaciones. @EntityValidator <ul style="list-style-type: none"> ✓ Validaciones al borrar. ✓ Anotaciones. Acciones. <ul style="list-style-type: none"> ✓ Acciones Personalizadas. 		Documentos académicos: Syllabus de la asignatura Guía didáctica N°10 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
		✓ Modificando el comportamiento normal.		OpenXava Git - GitHub
Sesión 20		Comportamientos y Lógica de Negocio. Referencias y Colección.	Presentación. Ejercicio Guiado. Explicación y Discusión en clase. II Corte Evaluación	Documentos académicos: Syllabus de la asignatura Guía didáctica N°10 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub
IV UNIDAD: PRIMEFACES.				
Sesión 21	Desarrollar aplicaciones web dinámicas utilizando este framework basado en JavaServer Faces (JSF). Integrando y personalizando	¿Qué es JSF? ¿Qué es Primefaces? Componentes de Primefaces. Mi primer Primefaces.	Conferencia dialogada. El docente introduce y explica que es JSF, componentes de primefaces. Práctica. Mediante la técnica learning by doing los estudiantes junto con el docente crean su primer Primefaces.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°11 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub
Sesión 22	componentes visuales avanzados de PrimeFaces para crear interfaces de usuario interactivas y responsivas, aplicando los	Creando un formulario Botones.	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°11 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
Sesión 23	principios de la programación orientada a objetos y siguiendo buenas prácticas de desarrollo de software.	DataTable. DataView.	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°12 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git – GitHub
Sesión 24		Gmap.	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°12 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git – GitHub
Sesión 25		Otros componentes. PickList. OrderList. Panel	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°13 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git – GitHub
Sesión 26		ConfirmDialog ConfirmPopup. Dialog. OverlayPanel. Menu.	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°13 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git – GitHub

Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
Sesión 27		Charts	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°14 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub
Sesión 28		Messages. File. Validation.	Sesión demostrativa. El docente, junto con los estudiantes, realiza una práctica desarrollando una interfaz web en PrimeFaces, aplicando conceptos de programación orientada a objetos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°14 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub
Sesión 29		Proyecto de Prueba	Revisión de avances de proyectos	Documentos académicos: Syllabus de la asignatura Guía didáctica N°15 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub
Sesión 30				
Sesión 31		Revisión de Proyectos.	Discusión de grupo	Documentos académicos: Syllabus de la asignatura Guía didáctica N°16 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate



Fecha	Objetivos de aprendizaje	Contenido	Estrategias	Recursos básicos de enseñanza
				OpenXava Git - GitHub
Sesión 32		Revisión de Proyectos.	Disertación de proyectos.	Documentos académicos: Syllabus de la asignatura Guía didáctica N°16 Herramientas de software: UAM-Virtual IntelliJ IDEA Ultimate OpenXava Git - GitHub

V. METODOLOGÍA

La enseñanza se concibe como un proceso que impulsa el desarrollo de los estudiantes, que rescata la coherencia entre conocimiento y vida, que propicia la acción formadora del docente, que se centra en el reconocimiento de la singularidad de cada alumno, y en el rol protagónico que le corresponde desempeñar frente a su aprendizaje. Si el aprendizaje es el resultado de la actividad cognitiva y afectiva del estudiante, el docente no es sino un acompañante, mediador, guía o facilitador de ese proceso. Sus acciones sólo cobran sentido en la medida en que acompañan, apoyan y facilitan efectivamente el aprendizaje del estudiante.

Para el desarrollo y alcance de objetivos de la asignatura Análisis y Diseño de Sistemas se utilizará el aprendizaje experiencial, que facilita espacios para construir aprendizajes significativos desde la experimentación, utilizando principalmente la estrategia de aprendizaje basado en proyectos (ABP).

Dado que el aprendizaje es visto como un proceso de descubrimiento, de búsqueda, de solución de problemas, no de mera repetición de conocimientos, el aprender a aprender, la investigación, la práctica, la búsqueda de significados, el establecimiento de vínculos entre lo que ya se sabe y lo nuevo por conocer, el aprender de otros y con otros, se convierten en ejes fundamentales de dicho proceso.

Para el aprendizaje de esta asignatura será necesario un conjunto de estrategias prácticas para el estudio individual y grupal, dentro de las cuales están las siguientes:

- Cuestionarios
- Investigaciones
- Resolución de problemas
- Control de lectura
- Exposiciones
- Casos de estudio (proyecto)

Por tratarse de una asignatura de formación específica que sirve de base para futuras asignaturas de ingeniería, se requiere un seguimiento cercano de parte del docente para identificar y remediar los problemas de aprendizaje individual, así como los relacionados con la enseñanza misma.

VI. EVALUACIÓN DEL APRENDIZAJE

La evaluación se concibe como un proceso dinámico, abierto, sistémico, contextualizado e inherente al desarrollo de las acciones de aprendizaje y enseñanza. Se desarrolla a lo largo del semestre, pues no se trata de acciones puntuales aisladas. Debe cumplir tanto la función social como pedagógica, implica el desarrollo de acciones de regulación y autorregulación, a partir de procesos autorreflexivos (tanto de los docentes, sobre su práctica educativa, como de los estudiantes acerca de qué y cómo aprenden).

A continuación, se detallan las estrategias de evaluación de los aprendizajes en la asignatura Análisis y Diseño de Sistemas:

Estrategias de evaluación	Fecha	Puntaje
Evaluación diagnóstica	19-Ago	-
PRIMER CORTE EVALUATIVO		
Participación en la Jornada Tecnológica: Systech 2024	30-Ago	10
Control de lecturas	Todo el corte	-
Asistencia	Todo el corte	20
Resolución de ejercicios introductorios al lenguaje Java	Del 26 de agosto al 6 de septiembre	-
Diagrama de clases	Del 9 al 13 de septiembre	30
Ejecución de Prototipo. Almacenamiento GITHUB	18 de septiembre	-
Examen de Evaluación de Java	Del 19 al 20 de septiembre	40
SEGUNDO CORTE EVALUATIVO		
Control de lecturas	Todo el corte	-
Asistencia	Todo el corte	20
Validaciones	Del 14 al 18 octubre	-
Examen de Evaluación de JPA, Entities	Del 21 al 23 de octubre	30
Creación de Entities	Del 24 al 29 de octubre	20
Prueba de CRUD	23 de octubre	30
TERCER CORTE EVALUATIVO		
Control de lecturas	Todo el corte	-



Asistencia	Todo el corte	20
Diseño UI del proyecto	Del 11 al 15 de noviembre	10
Implementación (parte 1)	Del 18 al 22 de noviembre	30
Implementación (parte 2)	Del 25 al 29 de noviembre	-
Defensa del Proyecto	Del 2 al 6 de diciembre	20
Presentación con el usuario	Del 2 al 6 de diciembre	20



VII. REFERENCIAS BIBLIOGRÁFICAS

Obligatorias:

DEITEL, P. y. (2016). Java. Cómo Programar. Décima edición. México: PEARSON EDUCACIÓN,

Complementarias:

Joyanes, L., Zahonero, I. (2002). Programación en Java 2. Algoritmos, Estructura de Datos y Programación Orientada a Objetos. Madrid: McGraw-Hill

VIII. HERRAMIENTAS DE SOFTWARE

IntelliJ IDEA.Ultimate [Software de computadora].

Open Xava

PrimeFaces