



Universidad Guadalajara

Centro Universitario del Sur

1. IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

División

DIVISIÓN DE CIENCIAS EXACTAS, NATURALES Y TECNOLÓGICAS

Departamento

Ciencias Computacionales e Innovación Tecnológica

Academia

Programación

Programa(s) educativo(s)

Telemática

Denominación de la unidad de aprendizaje:

Programación II

Clave de la materia:	Horas de teoría:	Horas de práctica:	Carga horaria global:	Valor en créditos:
IG189	48	32	80	8

Tipo de curso:		Nivel en que se ubica:	Prerrequisitos:
C = curso		Licenciatura	Ninguno
CL = curso laboratorio			
L = laboratorio			
P = práctica			
T = taller			
CT = curso - taller	X		
N = clínica			
M = módulo			
S = seminario			

Área de formación:

Básica Particular Obligatoria

Perfil docente:

Lic., en Informática, Lic., en computación, Licenciatura, maestría o doctorado en área afín al área de programación. Experiencia docente mínimo un año.

Elaborado por:

Actualizado por:

L.I. Eduviges de la Torre García, María Griselda Palacios Ruiz	Jorge Lozoya Arandia
---	----------------------

Fecha de elaboración:
aprobación por

Fecha de última

Fecha de última

Fecha de

actualización:

evaluación:

Colegio

Departamental:

Febrero de 2010	Enero 2023	Enero 2023	Enero 2023
-----------------	------------	------------	------------

1. PRESENTACIÓN DE LA UNIDAD DE APRENDIZAJE

Programación II

En esta materia, el objeto de estudio es el paradigma de la programación orientado a objetos, partiendo del análisis, diseño e implementación de situación de baja complejidad.

Aunque el plan curricular de la carrera de la licenciatura de la ingeniería en telemática no indica que esta materia tenga prerequisites, es recomendable que sea cursada después de las materias de Fundamentos de Programación y Programación I. Tiene relación con Estructura de datos e Ingeniería de Software

2. OBJETIVO GENERAL/COMPETENCIA

Analizar y diseñar soluciones a problemas computables con enfoque orientado a objetos e implementación de las mismas en un lenguaje de programación para facilitar, en un futuro, tareas computables a los usuarios que hacen uso de la computadora como herramienta de trabajo.

3. CAMPO DE APLICACIÓN PROFESIONAL DE LOS CONOCIMIENTOS

4. SABERES:

Prácticos	• Identificar las características y comportamientos de las clases de un sistema • Identificar e implementar en el diseño los elementos clave (clase, herencia, polimorfismo, sobrecarga de operadores) de la programación orientada a objetos • Diseñar los elementos estáticos de un sistema • Diseñar los elementos dinámicos de un sistema • Implementar los diseños anteriores en un lenguaje de programación
------------------	---

Teóricos	• Definir e identificar conceptos fundamentales de la programación orientada a objetos • Identificar el modelado de objetos para el diseño e implementación de la solución de un problema computable • Identificar elementos para el diseño UML de elementos estáticos y dinámicos de un programa orientado a objetos • Identificar palabras reservadas de un lenguaje de programación para implementar los diseños orientados a objetos
Formativos	• Abstracción y análisis orientada a objetos • Fomento del trabajo colaborativo • Fomento de autoaprendizaje Descubrir la necesidad de un diseño de la solución antes de la implementación en un lenguaje de programación.

5. CONTENIDO TEMÁTICO (TEÓRICO-PRÁCTICO)

Unidad 1. Paradigmas

- 1.1. Paradigmas
- 1.2. Programación orientada a objetos
- 1.3. Las clases
- 1.4. Objetos
 - 1.4.1. Miembros de una clase
 - 1.4.2. Crear o definir una clase
 - 1.4.3. Programación Crear un objeto a partir de una clase
 - 1.4.4. Acceder a los miembros de una clase
 - 1.4.5. Arreglos de objetos
- 1.5. Encapsulamiento
- 1.6. Polimorfismo
 - 1.6.1. Sobrecarga y sobreescritura de métodos
- 1.7. Herencia
 - 1.7.1. Simple
 - 1.7.2. Múltiple

Unidad 2. Iniciando UML Diagramas de casos de uso.

- 2.2. Diagramas de clases.
- 2.3. Clases.
 - 2.3.1. Control de acceso y visibilidad.
 - 2.3.2. Relaciones entre clases.
 - 2.3.2.1. Dependencia.
 - 2.3.2.2. Asociaciones.
 - 2.3.2.3. Agregación.
 - 2.3.2.4. Composición.
- 2.4. Generalizaciones.
- 2.5. Metaclase.

- 2.6. Clase Terminales.
- 2.7. Clase Abstracta.

Unidad 3. Formularios FRONTEND (interfaz gráfica), BACKEND (Funcionalidad) 3.1.

Introducción.

- 3.2. Tipos.
- 3.3. Propiedades.
- 3.4. Métodos.
- 3.5. Eventos.

Unidad 4. Gráficos GUI (Interfaz gráfica de usuario)

- 4.1. Gráficos GUI (subtemas).
- 4.2. Computación grafica (línea, círculos, colores, rellenos).

Unidad 5. Programación concurrente multihilo

- 5.1. Concepto de hilos.
- 5.2. Comparación de un programa sin hilos vs un programa con hilos.
- 5.3. Creación y control de hilos.
- 5.4. Sincronización de hilos

4

6. ESTRATEGIAS DE ENSEÑANZA-APRENDIZAJE

Investigación y lecturas del contenido teórico practico
Examen parcial
Exposición en clase
Realización de prácticas de laboratorio
Actividades de formación integral
Recopilación de prácticas

7. EVALUACIÓN DEL APRENDIZAJE

8.1. Evidencias de aprendizaje	8.2. Criterios de desempeño
Investigación y lecturas del contenido teórico practico	Entregar el tema(s) de investigación en forma electrónica o impresa según indique el profesor. La investigación debe contener lo siguiente: Título, índice, introducción objetivos, desarrollo del tema, conclusiones y bibliografía y en caso de haber consultado en internet web grafía.
Examen parcial	Los exámenes parciales se realizarán escritos (teórico practico) donde deberán contestar las preguntas y realizar los ejercicios de forma clara de acuerdo a las especificaciones en el examen

Exposición de clase	El alumno desarrollara un tema en específico donde presentara el título del tema, desarrollo del tema (conceptos, ejercicios prácticos), síntesis y conclusiones ya sea de forma individual o en equipo. Realizar la codificación de algoritmos y el corrimiento de los mismos Dar solución a problemas computables Entregar la práctica en electrónico y el corrimiento por escrito ambos sin excepción alguna.
Realización de prácticas	
Recopilación de prácticas y programas	Dar solución a problemas especificados por el profesor de acuerdo a las especificaciones. A lo largo del semestre el alumno reunirá en forma electrónica todas las prácticas y programas elaborados incluyendo archivos de reportes, archivos de programas y proyectos de cierre de unidad final. Reunirá los archivos en forma ordenada utilizando carpetas con el nombre del número de práctica o número de programa

8. CALIFICACIÓN

Examen teórico prácticos 25 puntos Prácticas de laboratorio 20 puntos Actividades finales por unidad 25 puntos Actividades de formación integral 5 puntos Trabajo final 25 puntos Total 100 puntos Para acreditar la formación integral deberá comprobar al menos una de las siguientes condiciones: – Tres constancias de actividades extra-escolares. - Un taller deportivo o artístico. El 5% de la formación integral sólo se contabilizará en caso de que la suma de la calificación de las otras actividades sea de 60 o más.	
--	--

9. ACREDITACIÓN

Periodo ordinario. De conformidad con el artículo 20 del Reglamento General de Evaluación y Promoción de Alumnos de la Universidad de Guadalajara, para que el alumno tenga derecho al registro del resultado final de la evaluación en el periodo ordinario,	Periodo extraordinario. De conformidad con el artículo 27 del Reglamento General de Evaluación y Promoción de Alumnos de la Universidad de Guadalajara, para que el alumno tenga derecho al registro de la calificación en el periodo extraordinario,
--	--

establecido en el calendario escolar aprobado por el Consejo General Universitario, se requiere:	se requiere:
I. Estar inscrito en el plan de estudios y curso correspondiente, y	I. Estar inscrito en el plan de estudios y curso correspondiente.
II. Tener un mínimo de asistencia del 80% a clases y actividades registradas durante el curso.	II. Haber pagado el arancel y presentar el comprobante correspondiente.
	III. Tener un mínimo de asistencia del 65% a clases y actividades registradas durante el curso.
	Se exceptúan de este caso las materias de orden práctico que requerirán la repetición del curso (Art. 23 RGEYPA).

10. BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

Ceballos Sierra, Fco Javier (2018)
Programación orientada a objetos con C++
 Madrid RA-MA Editorial
 ISBN: 978-84-9964-754-8
 Clasificación: **005.133 CEB 2018**
5 ejemplares físicos en biblioteca

López Román, Leobardo (2013).



Metodología de la programación orientada a objetos

México D.F. Alfaomega Grupo Editor, S.A. de C.V.
 ISBN: 978-607-707-589-9
 Clasificación: **005.117 LOP 2013**
5 ejemplares físicos en biblioteca

BIBLIOGRAFÍA COMPLEMENTARIA

Vázquez Cuesta, Borja (2017)
Java y C++ paso a paso
 Madrid RA-MA Editorial
 ISBN: 978-84-9964-723-4
 Clasificación: **005.133 VAZ 2017**
5 ejemplares físicos en biblioteca

Fontela, Carlos (2011)

UML modelado de software para profesionales

Alfaomega Grupo Editor Argentino

ISBN: 978-987-1609-22-2

Clasificación: **005.117 FON 2011**

2 ejemplares físicos en biblioteca

Joyanes Aguilar, Luis (2011)

Programación en Java 6. *Algoritmos, programación orientada a objetos e interfaz gráfica de usuario.*

México McGraw-Hill/Interamericana Editores

ISBN: 978-607-15-0618-4

Clasificación: **005.133 JOY 2011**

6 ejemplares físicos en biblioteca

11. RECURSOS COMPLEMENTARIOS (páginas web, mooc's, plataformas, objetos de aprendizaje)

--

Firma:

Presidente de Academia
Mtro. Gerardo Jiménez

Vo.Bo.
Jefe de Departamento
Dr.Jorge Lozoya Arandia