



Universidad Guadalajara

Centro Universitario del Sur

**UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DEL SUR**

DIVISIÓN DE CIENCIAS, ARTES Y HUMANIDADES
**DEPARTAMENTO DE CIENCIAS EXACTAS, TECNOLOGIAS Y
METODOLOGIAS**
CARRERA LICENCIATURA EN INGENIERIA EN TELEMÁTICA



PROGRAMA DE ESTUDIO POR COMPETENCIAS

PROGRAMACIÓN II


Mtra. Guillermina López Jiménez
Presidente de la Academia de Ciencias
computacionales


Mtro. Miguel Ángel Rangel Romero
Jefe del Departamento de Ciencias Exactas,
Tecnologías y Metodologías


MTA. Eduviges de la Torre García
Profesor de la Unidad de Aprendizaje

PROGRAMA DE ESTUDIO POR COMPETENCIAS
Formato Base

1. IDENTIFICACIÓN DE LA UNIDAD DE APRENDIZAJE

Centro Universitario del Sur

1.1 DEPARTAMENTO:

Ciencias Exactas, Tecnologías y Metodologías

1.2 ACADEMIA:

Ciencias Computacionales

1.3 NOMBRE DE LA UNIDAD DE APRENDIZAJE:

Programación II

Nota: Estos datos se encuentran en el Plan de Estudios derivados del dictamen.

Clave de la Unidad de Aprendizaje	Horas de teoría	Horas de práctica	Total de horas	Valor de créditos
IF118	48	32	80	8

Tipo de curso:	Nivel en que se ubica	Prerrequisitos	Correquisitos
C= curso	Técnico		
CL= clínica	Técnico superior		
N= práctica	Licenciatura		
T= taller	Especialidad		
CT= curso-taller	Maestría		
	Doctorado		

1.4 ELABORADO POR:

M. C. Octavio Reynaga Fernández
Mtro. Héctor Manuel Rodríguez Gómez

1.5 FECHA DE ELABORACIÓN:

Febrero de 2010

1.6 PARTICIPANTES:

MTA. Eduviges de la Torre García

1.7 FECHA DE APROBACIÓN POR LA INSTANCIA RESPECTIVA:

Academia de Ciencias Computacionales. Julio de 2014

2. UNIDAD DE COMPETENCIA

Unidad de competencia

Análisis y diseño de 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.

El desarrollo de software ha evolucionado de manera tal que se hace necesario, el uso de nuevas herramientas metodológicas para lograr productos competitivos y confiables. Las técnicas de programación han ido cambiando, la razón de dichos cambios ha sido adaptarse a la creciente complejidad de los programas, fue así como surgieron los lenguajes de alto nivel para dar al programador herramientas con las cuales pudiese hacer frente a una complejidad cada vez mayor. La programación orientada a objetos ha tomado las mejores ideas de la programación estructurada y las ha combinado con varios conceptos nuevos y potentes que permiten realizar las tareas de programación desde un nuevo punto de vista.

Aunque el plan curricular de la carrera de la licenciatura de la ingeniería en telemática no indica que esta materia tenga prerrequisitos, es recomendable que sea impartida después de las materias Algoritmos y Programación I.

Esta Unidad de Aprendizaje abona al Perfil de egreso:

- Capacidad para usar técnicas analíticas y heurísticas para la solución de problemas computables.
- Experiencia educacional integrada que le permita la habilidad de aplicar conocimientos pertinentes para la identificación y solución sistemática de problemas prácticos.
- Diseñar y desarrollar software de aplicación utilizando la metodología orientada a objetos.
- Capacidades y destrezas tanto a nivel básico y avanzado para planear, diseñar, y proponer soluciones en los servicios telemáticos. Se formará con valores de responsabilidad, profesionalismo, búsqueda de calidad y excelencia en los servicios y el valor de superación continua para mantenerse actualizado en el área de telemática

3. ATRIBUTOS O SABERES

Saberes	Descripción
<i>Teóricos</i>	<ul style="list-style-type: none">• 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
<i>Metodológicos</i>	<ul style="list-style-type: none">• Análisis orientado a objetos de un problema computable

<i>Saberes</i>	Descripción
	<ul style="list-style-type: none"> • Diseño orientado a objetos de un problema computable • Construir software con técnicas orientadas a objetos
Técnicos	<ul style="list-style-type: none"> • Identificar y definir la funcionalidad de la aplicación a construir para construir los CDU • Identificación de clases abstractas y concretas, para definir las responsabilidades sus comportamientos y atributos • Identificación y aplicar en el diseño e implementación del mismo los tipos de relaciones entre las clases • Identificar y aplicar en el diseño e implementación sobrecarga y polimorfismo. • Implementación de diseño orientado a objetos
<i>Formativos</i>	<ul style="list-style-type: none"> • 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.

4. CONTENIDO TEÓRICO PRÁCTICO

Contenido Teórico Práctico
<p>Unidad I Generalidades</p> <ul style="list-style-type: none"> 1.1. Repaso programación estructurada <ul style="list-style-type: none"> 1.1.1. Identificadores 1.1.2. Tipos de datos 1.1.3. Constantes y variables 1.1.4. Expresiones aritméticas y lógicas 1.1.5. Funciones de cadena y matemáticas 1.1.6. Estructura de control secuencial 1.1.7. Estructura de control selectiva (simple, doble, múltiple) 1.1.8. Estructuras de control repetitiva (condicionada, no condicionada) 1.1.9. Modularidad (funciones) 1.2. Conceptos básicos de programación orientada a objetos <ul style="list-style-type: none"> 1.2.1. UML y Tipos de diagramas UML 1.2.2. Clase <ul style="list-style-type: none"> Objetos Herencia Polimorfismo <p>Unidad II Iniciando UML</p> <ul style="list-style-type: none"> 2.1. Diagramas de casos de uso <ul style="list-style-type: none"> Incluidos

Contenido Teórico Práctico

- Extendidos
- 2.2. Diagramas de clases
 - Clases
 - Control de acceso y visibilidad
- 2.3. Diagramas de objetos
- 2.4. Relaciones entre clases
 - 2.4.1. Dependencia
 - 2.4.2. Asociaciones
 - Agregación
 - Composición
 - 2.4.3. Generalizaciones
 - Metaclase
 - Clase Terminales
 - Clase Abstracta

Unidad III Herencia

- 3.1. Herencia
 - Simple
 - Múltiple
- 3.2. Sobrecarga y sobrescritura de métodos

Unidad IV Introducción a Interfaz de usuario y diagramas

- 4.1. Interfaz de usuario para aplicaciones
- 4.2. Diagramas de estados y transiciones
- 4.3. Diagramas de secuencia

5. TAREAS O ACCIONES

Tareas o acciones

- Ejercicios/trabajos en equipo
- Ejercicios/trabajos individual
- Exámenes parciales
- Repentinos
- Actividades extracurriculares

*Los denominados ejercicios (ya sea en equipo o individual) pueden ser diagramas UML de diseño y/o codificación del diseño; también puede ser un producto solicitado como resultado de una investigación.

*Las rubricas correspondientes se indican en cada actividad. Para todas las actividades, la calificación puede ser de 0 a 100, la puntuación máxima se indicará en cada ejercicio.

*Las actividades fuera de tiempo solo se revisarán mas no podrán obtener puntuación alguna que abone a la calificación final del curso.

Tareas o acciones

* El trabajo final será un ejercicio de análisis, diseño e implementación.

NOTA: La puntuación por concepto de Actividades extracurriculares se otorgará cuando se entregue el comprobante de la misma y únicamente si la puntuación total que se obtenga es mayor o igual a 60% de puntos por concepto de ejercicios en equipo, ejercicios individuales, exámenes parciales y producto final.

6. EVALUACIÓN DE DESEMPEÑO

Evidencias de desempeño	Criterios de desempeño profesional	Campo de aplicación
Diseño	<p>Cumplir con las especificaciones, diagramas y documentación que se solicite; pudiendo ser diagramas UML y/o interfaz gráfica así como datos tabulares, según se solicite.</p> <p>Estos elementos deben ser únicos y creativos, ya sea de tarea o en las sesiones de clase; individuales o en equipo; según se indique.</p>	<ul style="list-style-type: none"> • Biblioteca • Actividad extra de clase • Laboratorio
Codificación	<p>Cumplir con las especificaciones requeridas y expresadas en el diseño realizado así como en la documentación que sea requerida.</p> <p>Utilizar los conceptos de orientación a objetos, indentación, estructuras de control adecuadas, documentación interna (comentarios).</p>	
	Actividades a realizarse de	Biblioteca

Evidencias de desempeño	Criterios de desempeño profesional	Campo de aplicación
Ejercicios individuales	<p>forma presenciales o extraclase; pueden ser</p> <ul style="list-style-type: none"> •Diagramas UML y/o codificación con la documentación solicitada. •Productos los cuales deben de expresar ideas propias, claras, concisas, ortográficamente correcto, relacionadas con el tema que se esté tratando incluyendo las fuentes bibliográficas; también deben de ser únicos, creativos. <p>Pueden ser resultado de una investigación, de mesas redondas o grupos de trabajo entre otras formas de trabajo; que se realicen en sesiones presenciales.</p>	Actividad extra de clase Laboratorio de software
Ejercicios en equipo	<p>Actividades a realizarse en grupos de 3 o 5 personas, según se indique</p> <p>De manera presencial o extraclase en los cuales, cada miembro del mismo debe de tener una participación activa y colaborativa en el equipo.</p> <ul style="list-style-type: none"> •Diagramas UML y/o codificación con la documentación solicitada. •Algún productos los cuales deben de expresar ideas propias, claras, concisas, ortográficamente correcto, relacionadas con el tema que se esté tratando incluyendo las fuentes bibliográficas; también 	Biblioteca Actividad extra de clase Laboratorio de software

Evidencias de desempeño	Criterios de desempeño profesional	Campo de aplicación
Exámenes parciales y/o repentinos (teórico/práctico)	<p>deben de ser únicos, creativos. (se realizarán mesas redondas o grupos de trabajo entre otras formas de trabajo)</p> <p>Incluye ejercicios teórico-prácticos (programas y/o información teórica) a realizar por parte del participante del curso de forma individual o en equipo –según se indique- y de manera presencial. Los exámenes repentinos pueden ser en cualquier día</p>	Laboratorio de software
<p>Notas generales:</p> <ul style="list-style-type: none"> • Los criterios requeridos se indicarán en cada ejercicio. • No se permitirá la entrega de evidencias fuera de la fecha requerida. • A quien se sorprenda con “acordeón” o material auxiliar en cualquier formato que no se haya indicado como requerido, ya sea en los ejercicios individuales o exámenes se suspenderá dicha actividad y se anulará dicha participación, sin tener derecho a recuperarla. • Para trabajos iguales, la calificación será cero para todos estos trabajos. • Los productos deberán de ser creativos y con ideas originales. 		

7. CALIFICACIÓN

Unidad de competencia	
Ejercicios/trabajos en equipo.....	25%
Ejercicios/trabajos individual.....	25%
Exámenes parciales.....	25%
Exámenes repentinos.....	25%
Actividades extracurriculares.....	5%

*Los denominados ejercicios (ya sea en equipo o individual) pueden ser diagramas UML de diseño y/o codificación (programas) del diseño; también puede un producto solicitado como resultado de una investigación.

*Las rubricas correspondientes se indican en cada actividad. Para todas las actividades, la calificación podrá ser de 0 a 100, para cada ejercicio, se indicará la puntuación máxima.

*Las actividades fuera de tiempo solo se revisarán mas no podrán obtener puntuación alguna que abone a la calificación final del curso.

* El trabajo final será un ejercicio de análisis, diseño e implementación.

* La puntuación por concepto de Actividades extracurriculares se otorgará cuando se entregue el comprobante de la misma y únicamente si la puntuación total que se obtenga es mayor o igual a 60% de puntos por concepto de ejercicios en equipo, ejercicios individuales, exámenes parciales y producto final.

8. ACREDITACIÓN

De conformidad a lo que establece el **Art. 20** del "Reglamento General de Evaluación y Promoción de Alumnos de la U. de G.":

Para que el alumno tenga derecho al registro del resultado final de la evaluación en el **periodo ordinario** establecido en el calendario escolar aprobado por el H. Consejo General Universitario, se requiere:

- I. Estar inscrito en el plan de estudios y curso correspondiente, y
- II. Tener un mínimo de asistencia del 80% a clases y actividades registradas durante el curso.

De conformidad a lo que establece el **Art. 27** del "Reglamento General de Evaluación y Promoción de Alumnos de la U. de G.":

Para que el alumno tenga derecho al registro de la calificación en el **periodo extraordinario**, se requiere:

- I. Estar inscrito en el plan de estudios y curso correspondiente.

- 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.

De conformidad a lo que establece el **Art. 25** del "Reglamento General de Evaluación y Promoción de Alumnos de la U. de G.":

La evaluación en periodo extraordinario se calificará atendiendo a los siguientes criterios:

- I. La calificación obtenida en periodo extraordinario, tendrá una ponderación del 80% para la calificación final;
- II. La calificación obtenida por el alumno durante el periodo ordinario, tendrá una ponderación del 40% para la calificación en periodo extraordinario, y
- III. La calificación final para la evaluación en periodo extraordinario será la que resulte de la suma de los puntos obtenidos en las fracciones anteriores.

9. BIBLIOGRAFÍA

9.1 BIBLIOGRAFÍA BÁSICA

- Anderson, J., Franceschi H. (2008). (Clásico) Java 6 Illuminated an active learning approach. Sudbury Jones and Bartlett Publishers.
2 Ejemplares Disponibles En Biblioteca.
- Arlow, J., Neustadt, I. (2006). (Clásico) UML 2. Madrid: Anaya Multimedia.
2 Ejemplares Disponibles En Biblioteca.
- Arnow, D. (2001). (Clásico) Introducción a la programación con JAVA. Un enfoque orientado a objetos. España: Pearson education.
7 Ejemplares Disponibles En Biblioteca.
- Cadenhead, R. (2012). Java 7. Madrid: Anaya Multimedia.
2 Ejemplares Disponibles En Biblioteca.
- Durán, F., Gutierrez, F., Pimentel, E. (2007). (Clásico) Programación orientada a objetos con java. Madrid: Thomson.
1 Ejemplares Disponibles En Biblioteca.
- Fain, Y. (2011). Programación Java. Madrid: Anaya Multimedia.
7 Ejemplares Disponibles En Biblioteca.
- García, J., Ortín, F. (2010). Programación orientada a objetos avanzada. Santiago de Compostela: Andavira.
2 Ejemplares Disponibles En Biblioteca.

- Joyanes, L. Zahonero, I. (2010). Programación en C, C++, Java y UML. México: McGraw Hill/Interamericana.
5 Ejemplares Disponibles En Biblioteca.
- Joyanes, J. (1998). (Clásico) Programación orientada a objetos. Madrid: McGraw-Hill.
9 Ejemplares Disponibles En Biblioteca.
- Joyanes, J. (2008). (Clásico) Fundamentos de programación algoritmos, estructura de datos y objetos. Madrid: McGraw-Hill/Interamericana de España.
6 Ejemplares Disponibles En Biblioteca.
- Larman, C. (1999). (Clásico) UML y patrones introducción al análisis y diseño orientado a objetos. México: Alfaomega.
5 Ejemplares Disponibles En Biblioteca.
- López, L. (2011). Programación estructurada y orientada a objetos un enfoque algorítmico. México: Alfaomega.
3 Ejemplares Disponibles En Biblioteca.
- López, L. (2008). (Clásico) Metodología de la programación orientada a objetos. México: Alfaomega.
10 Ejemplares Disponibles En Biblioteca.
- Rumbaugh, J. (1997). (Clásico) Modelado y diseño orientados a objetos. España: Prentice Hall Internacional.
1 Ejemplar Disponible En Biblioteca.
- Moldes, F. (2011). Java 7. Madrid: Anaya Multimedia.
2 Ejemplares Disponibles En Biblioteca.
- Meyer, B. (1999). (Clásico) Construcción de software orientado a objetos. Madrid: Prentice Hall Iberia. **1 Ejemplar disponible en biblioteca**
- Piza, H. (2008). (Clásico) Aprendiendo java un enfoque orientado a objetos. México: Universidad de Guadalajara Centro Universitario de los Valles.
6 Ejemplares Disponibles En Biblioteca.
- Pérez, J. (2006). (Clásico) Problemas resueltos de programación en lenguaje Java. Madrid: International Thomson Editores Spain.
3 Ejemplares Disponibles En Biblioteca.
- Sines, A. (2002). (Clásico) Aprendiendo programación orientada a objetos en 21 lecciones avanzadas. México Pearson.
3 Ejemplares Disponibles En Biblioteca.

- Soroka, B. (2006) (Clásico) Java 5 objects first. Sudbury, Massachusetts Jones and Bartlett.
3 Ejemplares Disponibles En Biblioteca.
 - Wang, P. (2000). (Clásico) Java con programación orientada a objetos y aplicaciones. México: International Thomson Editores.
3 Ejemplares Disponibles En Biblioteca.
- Wu, Thomas. (2008). (Clásico) Programación en Java introducción a la programación orientada a objetos. México: McGraw-Hill-Interamericana. **8 Ejemplares Disponibles En Biblioteca.**

9.2 BIBLIOGRAFÍA COMPLEMENTARIA

- Cairó, O. (1995). (Clásico) Metodología de la programación algoritmos, diagramas de flujo y programas. México: Alfaomega.
5 Ejemplares Disponibles En Biblioteca.
- Braude, E. (2003). (Clásico) Ingeniería de software una perspectiva orientada a objetos. México: Alfaomega.
8 Ejemplares Disponibles En Biblioteca.
- Burd, B. (2005). (Clásico) Beginning programming with java for dummies. Indianapolis: Wiley Publishing.
3 Ejemplares Disponibles En Biblioteca.
- Charte, O. (2005). (Clásico) Introducción a la programación. Madrid: Anaya Multimedia.
1 Ejemplar Disponible En Biblioteca.
- Cowell, J. (1997). (Clásico) Essential java fast how to write object oriented software for the internet. London: Springer.
1 Ejemplar Disponible En Biblioteca.
- Dean, J., Dean Raymond. (2009). Introducción a la programación con JAVA. México McGraw-Hill/Interamericana Editores.
1 Ejemplar Disponible En Biblioteca.
- Deitel, J., Deitel, M. (2008). (Clásico) Cómo programar en Java. México: Pearson Prentice Hall.
6 Ejemplares Disponibles En Biblioteca.

10. CURRICULUM VITAE DEL PROFESOR



DATOS GENERALES:

Nombre: Ediviges de la Torre García
Correo electrónico: edu@cusur.udg.mx

FORMACIÓN PROFESIONAL BASICA:

Licenciatura en Informática por el Instituto Tecnológico de Ciudad Guzmán

ESTUDIOS DE POSTGRADO:

Maestría en tecnologías para el aprendizaje

CERTIFICACIONES:

Sun certified programmer, for the java plataform, standard edition 6.

NOMBRAMIENTOS ACADEMICOS ACTUAL:

Profesor de asignatura B

CENTRO LABORAL ACTUAL:

Centro Universitario del Sur

CARGOS ADMINISTRATIVOS DESEMPEÑADOS:

- Actualmente es parte del personal de Apoyo a la Unidad de Mutimedia Instruccional del Centro Universitario del Sur
- Asesor técnico y de diseño instruccional en el área de cursos en línea del Centro Universitario del Sur de Enero de 2001 a Junio de 2013.
- Área capacitación del Centro de cómputo del Centro Universitario del Sur, como responsable de la organización de curso de ofimática en los ciclos escolares 1999A al 2004B
- Instructor y asesor de cursos de ofimática a los docentes del CUSur del calendario escolar 1999A al 2004B.