



**PROGRAMA DEL CURSO
IF4101 - LENGUAJES PARA APLICACIONES COMERCIALES
I CICLO DEL 2019**

1 Datos Generales

- Sigla: IF4101
- Nombre: Lenguajes para aplicaciones comerciales
- Tipo de curso: Teórico - práctico
- Créditos: 4
- Horas lectivas: 8
- Requisitos: IF3100 Introducción a los sistemas de información, IF4100 Fundamentos de bases de datos
- Correquisitos: Ninguno
- Ubicación en el plan de estudio: V Ciclo
- Suficiencia: No
- Tutoría: Sí
- Modalidad: Regular - bajo Virtual

2 Descripción

Este curso permite a sus participantes obtener los conocimientos necesarios para desarrollar aplicaciones comerciales de software, haciendo uso de diferentes lenguajes de programación, herramientas de desarrollo y frameworks de aplicaciones. A través del curso se explica el funcionamiento de lenguajes que posibilitan la construcción de aplicaciones que permiten el mantenimiento de los datos y otros archivos de organización de la información, así como el conjunto de herramientas para el diseño de interfaces gráficas de usuario. El curso le permite a la y el estudiante incursionar en al menos dos frameworks de aplicaciones distintos.



3 Objetivo General

Desarrollar en el estudiantado participante los conocimientos fundamentales para la creación de aplicaciones de software, empleando un enfoque de codificación basado en capas y profundizando en el uso de la programación orientada a objetos, por eventos, programación visual y basada en componentes.

4 Objetivos Específicos

Al finalizar el curso el o la estudiante estará en capacidad de:

- Desarrollar aplicaciones basadas en web y ventanas, que permita minimizar el tiempo de desarrollo de los sistemas, utilizar motores de bases de datos para manipular la información desde la lógica de la aplicación y diseñar interfaces agradables para el usuario.
- Aplicar los conceptos de programación visual, por eventos y orientada a objetos en ambientes de desarrollo vigentes.
- Desarrollar aplicaciones de software cuya lógica de implementación esté separada en tres capas.
- Crear y utilizar componentes de software.
- Implementar un módulo de seguridad en las aplicaciones de software.
- Emplear buenas prácticas de desarrollo en las aplicaciones de software.
- Profundizar en el estudio de las herramientas de desarrollo que se le presentan.

5 Contenidos

5.1. Las aplicaciones de software.

5.1.1. Aplicaciones basadas en ventanas y web

5.1.2. Marcos de construcción de aplicaciones (application frameworks)

- i. Los marcos de construcción (frameworks)
- ii. Los marcos de construcción de aplicaciones (application frameworks)
- iii. Beneficios



5.1.3. Arquitectura Física: N-Tier Architecture

- i. Concepto
- ii. Tipos (Cliente-Servidor, 3-Tier)
- iii. Beneficios: Escalabilidad, seguridad y tolerancia a fallas

5.1.4. Arquitectura Lógica: N-Layer Architecture

- i. Concepto
- ii. Tipos (3-layers, n-layers)
- iii. Beneficios: mantenibilidad, reutilización, distribución del trabajo, flexibilidad, robustez, entre otros
- iv. Estudio de las capas lógicas de implementación (presentación, aplicación, negocios y servicio de datos)

5.1.5. Introducción a las aplicaciones web.

- 5.1.5.1 Las aplicaciones web
- 5.1.5.2 Forma de trabajo de las aplicaciones web
- 5.1.5.3 Estructura de una aplicación web
- 5.1.5.4 Introducción al HTML
- 5.1.5.5 Formularios HTML como recolectores de información
- 5.1.5.6 Scriptlets y expresiones
- 5.1.5.7 Servlets
- 5.1.5.8 Javaserer Pages
- 5.1.5.9 XML y API para la manipulación (DOM)

5.2. **Características de los ambientes de desarrollo usados para las aplicaciones comerciales.**

- 5.2.1 Instalación y configuración de ambientes de desarrollo
- 5.2.2 Tiempo de desarrollo
- 5.2.3 Facilidades de desarrollo de interfaces
- 5.2.4 Facilidades para el almacenamiento, acceso y uso de los datos
- 5.2.5 Especificaciones técnicas y requerimientos
- 5.2.6 El editor
- 5.2.7 Manejo de menús
- 5.2.8 Depuración



5.3. Aplicaciones basadas en ventanas

5.3.1 Las aplicaciones de escritorio

5.3.2 El modelo Cliente-Servidor

5.3.3 Componentes de interfaces gráficas

- i. Interfaces de documentos múltiples (MDI), menús, controles, formularios, formularios predesarrollados, controles personalizados
- ii. Propiedades
- iii. Eventos

5.3.4 Librerías y reutilización

5.3.5 Implementación de capas en una aplicación basada en ventanas

5.3.6 Métodos abreviados: mnemonics y atajos de teclado

5.4. Aplicaciones web

5.4.1 Las aplicaciones web

5.4.2 El patrón arquitectónico Model-View-Controller

5.4.3 Librerías de etiquetas para HTML (ejemplos: JSTL, Struts TagLibs)

5.4.4 Servidores web: instalación, configuración, distribución de aplicaciones y ejecución de aplicaciones en el servidor

5.4.5 Configuración de una aplicación web: Archivos descriptores XML de la aplicación web (ejemplos: web.config, app.config, web.xml, struts-config.xml)

5.4.6 Despliegue dinámico en las aplicaciones web: uso de las librerías de etiquetas para HTML

5.4.7 Internacionalización

5.4.8 Implementación de capas en una aplicación web

5.4.9 Validación de entradas

5.4.10 Administración del Estado

- i. Estado de aplicación y de sesión
- ii. Cookies
- iii. Contexto de variables: page, request, session, application

5.5. Manejo de excepciones



- 5.6. Manejo transaccional desde las aplicaciones
- 5.7. Implementación de IMEC (CRUD) y un maestros detalles en las aplicaciones
- 5.8. Uso de procedimientos almacenados desde las aplicaciones
- 5.9. Diseño y utilización de reportes en las aplicaciones
- 5.10. Pool de conexiones en las aplicaciones
- 5.11. Seguridad en las aplicaciones
 - 5.11.1. Autenticación y autorización
 - 5.11.2. Implementación de un módulo de seguridad basado en roles
- 5.12. Creación y consumo de servicios web
- 5.13. Generación de instaladores e implantación de la aplicación en producción

6 Metodología

- El curso presenta un eje de desarrollo práctico, razón por la cual las clases se llevan a cabo en el laboratorio de cómputo del recinto.
- La persona docente desarrolla las clases soportado en diapositivas, modelos UML, código fuente e instancias de bases de datos con contenidos reales. Por ser un curso de desarrollo de software, se presentará los contenidos prácticos del curso mediante codificación en tiempo real. Los y las estudiantes seguirán el desarrollo de la clase de forma que vivencien la secuencia de pasos y de código necesarios para implementar diversos aspectos en las aplicaciones de software. De esta forma, los y las estudiantes desarrollarán varios proyectos estructurados en los distintos marcos de construcción de aplicaciones, que les servirán para efectos de las evaluaciones del curso (exámenes, laboratorios y proyectos) y para sus vidas profesionales.
- Los materiales didácticos y el programa del curso estarán disponibles en el entorno Mediación Virtual. Para registrarse en el curso deben visitar la dirección electrónica <https://mediacionvirtual.ucr.ac.cr>. La persona docente del curso les proveerá la clave de acceso.



- Los y las estudiantes desarrollan un laboratorio semanalmente en donde ponen en práctica los temas cubiertos en el curso. Durante la práctica tendrán la oportunidad de evacuar dudas, discutir sus propuestas con el resto de la clase. Los laboratorios son a cuaderno abierto; sin embargo, es necesario que los temas abarcados sean retomados para poder realizar los ejercicios acorde con los escenarios que se propongan.
- Los y las estudiantes desarrollan proyectos programados donde ponen en práctica y amplían los conocimientos adquiridos en el curso. Los requerimientos y las políticas de evaluación serán entregados con suficiente antelación a la fecha de entrega. Los proyectos tendrán fechas de avance y las entregas deberán estar acompañadas de la documentación que sea solicitada.
- Los y las estudiantes realizan lecturas semanales y presentan resúmenes. De esta forma se pretende que durante la clase tengan conocimiento del tema para poder participar. Los resúmenes serán de carácter individual, únicamente.
- Para las distintas evaluaciones, el medio de entrega y demás detalles se especificarán junto con el enunciado correspondiente.
- Además, los y las estudiantes realizarán exposiciones acerca de temáticas de interés que complementen el desarrollo del curso. Los temas de exposición serán entregados en la segunda semana de clase, así como las fechas y los aspectos a ser evaluados.

7 Evaluación

DESCRIPCIÓN	PORCENTAJE
Examen Parcial I	20%
Examen Parcial II	20%
Investigación Aplicada	5%
Proyectos programados	40%
Tareas y Laboratorios	15%

7.1 Consideraciones sobre la evaluación

- Según lo establecido en las resoluciones VD-R-8458-2009 y VD-R-9374-2016, se utilizará un entorno virtual de aprendizaje específico para el curso, en la



plataforma oficial institucional Mediación Virtual (<https://mediacionvirtual.ucr.ac.cr>). El entorno será utilizado para la entrega del programa del curso, material, enunciados de evaluaciones, entre otros, por parte del profesor, así como para entrega y/o realización de evaluaciones y otros entregables por parte del estudiante cuando corresponda.

- Según lo establecido en la resolución R-2664-2012, que establece el correo institucional con el dominio @ucr.ac.cr como la herramienta oficial para las comunicaciones de toda la comunidad universitaria, se utilizará el correo institucional como medio oficial de comunicación entre profesores y estudiantes, por lo cual el estudiante debe tenerlo activo y revisarlo continuamente.
- Durante las evaluaciones el uso de teléfonos celulares, tabletas o cualquier otro dispositivo de comunicación está totalmente prohibido - a excepción que el profesor indique lo contrario - dentro y fuera del aula mientras el estudiante no haya hecho entrega de su evaluación. Dichos dispositivos deberán permanecer apagados y guardados en su bolso o bulto.
- Los criterios de calificación de cada evaluación serán especificados en el enunciado de la misma.
- Toda evaluación será comunicada al estudiante al menos 5 días hábiles antes de realizarse, a excepción de las pruebas cortas o “quizes”, de acuerdo con lo especificado en los artículos 15 y 18 del Reglamento de Régimen Académico Estudiantil.
- En caso de ausencia a alguna evaluación, se procederá según lo establecido en el Artículo 24 del Reglamento de Régimen Académico Estudiantil.
- Ante la detección de una posible copia o plagio, total o parcial, en cualquier evaluación, se procederá de acuerdo con lo establecido en el Reglamento de Orden y Disciplina Estudiantil.
- Las fechas del cronograma están sujetas a cambio dependiendo del avance en los contenidos.



8 Docentes del curso

GRUPO	AULA	DOCENTE	HORARIO	CONSULTA
RECINTO DE GOLFITO				
01	17	MSc Gabriela Loaiza Mora ana.loaiza@ucr.ac.cr	M 17 a 21:00 V 17 a 21:00	M 14 a 16:00 V 14 a 16:00
SEDE DEL ATLÁNTICO, RECINTO DE GUÁPILES				
31	Lab 2	Mag Carlos Felipe Escalante Solano carlosfelipe.escalante@ucr.ac.cr	K 13 a 16:50 M 08 a 11:50	V 08 a 11:50
SEDE DEL ATLÁNTICO, RECINTO DE PARAÍSO				
21	Lab 14	MGP Alvaro Mena Monge alvaro.mena@ucr.ac.cr	K 08 a 11:50 J 13 a 16:50	K 13 a 17:00
22	Lab 14	MP Cristian Guillén Méndez cris.guillen0@gmail.com	M 17 a 19:50 J 17 a 19:50	S 08 a 11:50
SEDE DEL ATLÁNTICO, RECINTO DE TURRIALBA				
01	Lab B	Lic Nelson Méndez Montero n.mendezmontero@gmail.com	L 08 a 11:50	V 10 a 11:50
02	Lab B	Bach Jorge Román Solano jorge.roman@hotmail.com	J 13 a 16:50	K 10 a 11:50
SEDE DEL CARIBE, RECINTO DE LIMÓN				
01	Lab 02	Lic Luis West Grant luiswest@gmail.com	K 17 a 20:50 J 17 a 20:50	V 16 a 19:50
SEDE DEL CARIBE, RECINTO DE SIQUIRRES				
01	18	Lic Olivier Blanco Sandí olivier.blanco@ucr.ac.cr	L 08 a 11:50 M 13 a 16:50	M 08 a 11:50
SEDE DE GUANACASTE, RECINTO DE LIBERIA				
01	09	MP Luis Fernando Charpen- tier Gonzalez	S 07 a 15:00	J 15 a 19:00



GRUPO	AULA	DOCENTE	HORARIO	CONSULTA
luischar@gmail.com				
SEDE DE OCCIDENTE, RECINTO DE GRECIA				
02	Lab A	MCI David Cruz Alvarado david.cruz@ucr.ac.cr	L 17 a 20:50 M 17 a 20:50	J 17 a 20:50
SEDE DE OCCIDENTE, RECINTO DE SAN RAMÓN				
01	Bib 1	MCI Francisco Blanco Chavarría francisco.blanco_c@ucr.ac.cr	L 08 a 11:50 V 08 a 11:50	L 13 a 16:50
SEDE DEL PACÍFICO				
01	Lab 1	Lic Melvin Araya González melvin.arayagonzalez@ucr.ac.cr	L 17 a 19:50 M 08 a 11:50	M 13 a 17:00

9 Cronograma

SEM	FECHA	TEMA O ACTIVIDAD
01	11 - 17 MAR	Las aplicaciones de software
02	18 - 24 MAR	Introducción al desarrollo web en Framework 1
03	25 - 31 MAR	Introducción al desarrollo web en Framework 1
04	01 - 07 ABR	Implementación de IMEC (CRUD) y un maestros detalles en las aplicaciones
05	08 - 14 ABR	Manejo transaccional desde las aplicaciones
06	15 - 21 ABR	Semana Santa
07	22 - 28 ABR	Semana universitaria - Manejo de excepciones
08	29 - 05 MAY	Uso de procedimientos almacenados desde las aplicaciones
09	06 - 12 MAY	Creación y consumo de servicios web
10	13 - 19 MAY	Introducción al desarrollo web en Framework 2
11	20 - 26 MAY	Examen Parcial I - Introducción al desarrollo web en Framework 2



SEM	FECHA	TEMA O ACTIVIDAD
12	27 - 02 JUN	Implementación de IMEC (CRUD) y un maestros detalles en las aplicaciones
13	03 - 09 JUN	Seguridad en las aplicaciones
14	10 - 16 JUN	Aplicaciones basadas en ventanas
15	17 - 23 JUN	Diseño y utilización de reportes en las aplicaciones
16	24 - 30 JUN	Examen Parcial II - Generación de instaladores e implantación de la aplicación en producción
17	01 - 07 JUL	Entrega de proyecto final

Referencias obligatorias

- Aravinth, A. and Machiraju, S. (2018). *Beginning Functional JavaScript: Uncover the Concepts of Functional Programming with EcmaScript 8*. Apress, Berkeley, CA.
- Freeman, A. (2017). *Pro ASP.NET Core MVC 2*. Apress, Berkeley, USA.
- Freeman, A. (2018a). *Pro Angular 6*. Apress, Berkeley, USA.
- Freeman, A. (2018b). *Pro Entity Framework Core 2 for ASP.NET Core MVC*. Apress, Berkeley, USA.
- Prasad, S. (2017). *Beginning Spring Boot 2: Applications and Microservices with the Spring Framework*. Apress, New York, USA.
- Walls, C. (2018). *Spring in Action*. Manning Publications, Shelter Island, NY.

Referencias secundarias

- Tutorial: Intro to react – react.
- Borggreve, B. (2018). *Server-Side Enterprise Development with Angular*. Packt Publishing, Birmingham, UK.
- Deinum, M. (2018). *Spring Boot 2 Recipes: A Problem-Solution Approach*. Apress, Berkeley, CA.
- Elliott, E. and Elliott, E. (2016). Top javascript frameworks and topics to learn in 2017.
- Martin, R. C. (2012). *Clean Code*. Prentice Hall; 1 edition (August 11, 2008).
- Platzi (2019). Github vs gitlab — platzilive.