

PROGRAMA CURSO: IF4101 Lenguajes para aplicaciones comerciales
I Semestre, 2017

Datos Generales

Sigla del curso: IF4101

Nombre del curso: Lenguajes para aplicaciones comerciales

Tipo de curso: Teórico-Práctico

Créditos: 4

Horas lectivas semanales: 8

Requisitos:

IF3100 Introducción a los Sistemas de Información

IF-4100 Fundamentos de Bases de Datos

Correquisitos: No tiene.

Ubicación en el plan de estudio: V Ciclo

Horario del curso: K 17:00 a 20:50, V 13:00 a 16:50

Tutoría: No

Suficiencia: No

Modalidad: (Presencial/Bajo virtual)

Datos del Profesor

Nombre: Alejandro Ulate Campos, Mci.

Correo electrónico: alejandro.ulate@ucrso.info

Horas consulta presencial: K 13:00 a 17:00

Horas consulta remota: K 13:00 a 17:00

Medio oficial para consulta remota: Hangouts

1. Descripción del curso

Este curso permite al estudiante 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 al y el estudiante incursionar en varios frameworks de aplicaciones distintos.

Objetivos

2. Objetivo General

Desarrollar en el estudiante los conocimientos fundamentales para la creación de aplicaciones de software estudiando el paradigma de aplicaciones en varias capas y profundizando en el uso de la programación orientada a objetos, por eventos, programación visual y basada en componentes.

3. Objetivos Específicos

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

- Desarrollar aplicaciones basadas en Web y Ventanas, que permita:
 - o Minimizar el tiempo de desarrollo de los sistemas.
 - o Utilizar motores de bases de datos para manipular la información desde la lógica de una aplicación.
 - o 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 varias capas.
 - Crear y utilizar componentes software.
 - Implementar un módulo de seguridad en las aplicaciones de software.
 - Implantar las mejores prácticas de desarrollo de la ingeniería de software en las aplicaciones de software.
 - Profundizar en el estudio de las herramientas de desarrollo que se le presentan.
-

4. Contenidos

1. Las aplicaciones de software.

Aplicaciones basadas en Ventanas y Web

Marcos de construcción de aplicaciones (application frameworks)

Los marcos de construcción (Frameworks)

Los marcos de construcción de aplicaciones (application frameworks)

Beneficios

Arquitectura Física: N-Tier Architecture

Concepto

Tipos (Cliente-Servidor, 3-Tier)

Beneficios: Escalabilidad, seguridad y tolerancia a fallas

Arquitectura Lógica: N-Layer Architecture

Concepto

Tipos (3-layers, n-layers)

Beneficios: mantenibilidad, reutilización, distribución del trabajo, flexibilidad, robustez, entre otros

Estudio de las capas lógicas de implementación (presentación, aplicación, negocios y servicio de datos)

Introducción a las Aplicaciones Web

Las aplicaciones Web

Forma de trabajo de las aplicaciones Web

Estructura de una aplicación Web

Introducción al HTML

Formularios HTML como recolectores de información

Scriptlets y expresiones

Servlets

Javasever Pages

XML y API para la manipulación (DOM)

2. Características de los ambientes de desarrollo usados para las aplicaciones comerciales.

Instalación y configuración de ambientes de desarrollo

Tiempo de desarrollo

Facilidades de desarrollo de interfaces

Facilidades para el almacenamiento, acceso y uso de los datos

Especificaciones técnicas y requerimientos

El editor

Manejo de menús

Depuración

3. Aplicaciones basadas en Ventanas

Las aplicaciones de escritorio

El modelo Cliente-Servidor

Componentes de Interfaces Gráficas

Interfaces de Documentos Múltiples (MDI), Menús, controles, formularios, formularios predesarrollados, controles personalizados

Propiedades

Eventos

Librerías y reutilización

Implementación de capas en una aplicación basada en ventanas

Métodos abreviados: mnemonics y atajos de teclado

4. Aplicaciones Web

Las aplicaciones Web

El patrón arquitectónico Model-View-Controller

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

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

Configuración de una aplicación web:

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

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

Internacionalización

Implementación de capas en una aplicación web

Validación de entradas

Administración del Estado

Estado de Aplicación y de Sesión

Cookies

Contexto de variables: page, request, session, application

5. Manejo de Excepciones

6. Manejo transaccional desde las aplicaciones

Inicio y finalización de transacciones

COMMIT y ROLLBACK

7. Implementación de IMEC (CRUD) y un maestros detalles en las aplicaciones

8. Uso de procedimientos almacenados desde las aplicaciones

9. Diseño y utilización de reportes en las aplicaciones

10. Pool de conexiones en las aplicaciones

11. Seguridad en las aplicaciones

Autenticación y autorización

Implementación de un módulo de seguridad basado en roles

12. Creación y consumo de Servicios web

13. Generación de instaladores e implantación de la aplicación en producción

14. Aplicaciones para dispositivos móviles

5. Metodología

- El curso presenta un eje de desarrollo práctico y por ende se desarrollan en el laboratorio de cómputo del recinto.
- El profesor 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, el profesor presentará los contenidos prácticos del curso mediante código en vivo. Los y las estudiantes seguirán al profesor durante 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 (J2EE 5, Struts y ASP.NET) que les servirán para efectos de las evaluaciones del curso (exámenes, laboratorios y proyectos) y para sus vidas profesionales.
- Los materiales didácticos estarán disponibles en un grupo virtual creado para el curso.
- Los y las estudiantes desarrollan un laboratorio semanalmente en donde ponen en práctica los temas cubiertos en el curso. Durante la práctica, los y las estudiantes tendrán la oportunidad de evacuar dudas, discutir sus propuestas con el profesor, compañeros y

compañeras. Los laboratorios son a cuaderno abierto; sin embargo, es necesario que los temas abarcados sean retomados para poder realizar los ejercicios en acorde con los escenarios que se propongan.

- Los y las estudiantes desarrollan un proyecto programado donde ponen en práctica y amplían los conocimientos adquiridos en el curso. Dicho proyecto podrá contemplar aplicaciones de escritorio, Aplicaciones Web, aplicaciones para dispositivos móviles, o una combinación de estas. El profesor entregará los requerimientos y las políticas de evaluación de los proyectos con suficiente antelación a la fecha de entrega. Los proyectos podrán tener fechas de avance y las entregas deberán estar acompañadas de la documentación que sea solicitada. (Todo esto se especifica en el enunciado del proyecto).
- Los estudiantes realizan lecturas semanales y se les podría solicitar resúmenes de las mismas. De esta forma se pretende que durante la clase los y las estudiantes tengan conocimiento del tema para poder participar. Los resúmenes serán de carácter individual, únicamente.
- Los exámenes, por ser de carácter práctico en su mayoría, deberán ser defendidos por el o la estudiante si el profesor lo considera necesario. La defensa se llevaría a cabo en la clase subsiguiente a la realización del examen. El o la estudiante será responsable de configurar el ambiente para que el proyecto del examen se ejecute correctamente, hecho que será parte de la evaluación. El y la estudiante, el día del examen, deberá entregar el código fuente del proyecto del examen para realizar su defensa posterior.
- Además, los 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.

6. Evaluación

<i>Descripción</i>	<i>Porcentaje</i>
Examen parcial I	25%
Examen Parcial II	25%
Investigaciones	10%*
Proyecto	20%*
Tareas, Labs, Quices, Giras	20%
Total:	100%

* Para los proyectos grupales, se reserva al menos el 5% del porcentaje asignado, para la realización de la comprobación individual de cada integrante.

Consideraciones sobre la evaluación

- ❑ Todos los trabajos o asignaciones deberán ser entregados en la fecha y hora que se indique o antes. Para cada trabajo a entregar el profesor habilitará un link en la página del curso para que suban los archivos correspondientes, no se aceptarán trabajos enviados fuera de la fecha y hora límite indicada o entregados por otros medios. Es responsabilidad de cada estudiante o grupo de trabajo enviar el archivo con suficiente tiempo antes de la hora límite.
- ❑ Todo trabajo escrito deberá entregarse en formato de documento abierto (odt) o en formato de documento portátil (pdf).
- ❑ Los trabajos en los que deban crear un programa de software deberán ser entregados con el código fuente completo y en el formato adecuado para poder ser revisado, compilado y ejecutado, además de un script o backup de la base de datos en caso de que aplique. Si un programa tiene errores de compilación o algún otro tipo de error que impida su ejecución será calificado con nota cero.
- ❑ Sólo se reponen evaluaciones según lo estipulado en el artículo 24 del Reglamento de Régimen Académico de la UCR y utilizando el procedimiento ahí descrito.
- ❑ Para fines del curso se ha creado un espacio en Internet. El estudiante deberá registrarse en el mismo, para recibir y/o entregar información y trabajos del curso. En la primera clase el profesor indicará a los alumnos el medio de comunicación oficial del curso.
- ❑ Se creará una lista de correos. El estudiante debe estar pendiente de este medio.
- ❑ Todos los trabajos se deben presentar en formato electrónico (por escrito, se les avisará con antelación) en horario lectivo al profesor.
- ❑ La redacción y ortografía se tomarán en cuenta como rubros de evaluación.
- ❑ **No se acepta** la entrega de trabajos vía correo electrónico.
- ❑ Todos los trabajos serán entregados sin excepción al profesor en las fechas programadas originalmente por el mismo.
- ❑ Los quices **no se repondrán** por ningún motivo y la ausencia a uno de ellos equivale a un cero, se aplicarán de manera semanal/aleatoria y en cualquier momento durante la lección, abarcando la temática vista en clase, o material previamente asignado.
- ❑ **No se repondrá** ningún examen al estudiante que no presente un certificado médico válido (CCSS), dentro de los ocho días hábiles siguientes a la realización de la prueba.
- ❑ **La no presentación** de uno o más proyectos y/o investigaciones, elimina la opción de realizar el examen final del curso.
- ❑ El proyecto programado se realizará en grupos de 3-4 personas como máximo.
- ❑ Si algún estudiante es expulsado(a) de su sub-grupo de trabajo debido a incumplimiento de labores justificadas adecuadamente, perderá los puntos del trabajo respectivo y no se le asignará a ningún otro grupo ni podrá hacerlo individualmente. Las pruebas respectivas para expulsar a un compañero(a) del grupo deben venir por escrito y firmadas por los miembros del grupo donde se exponga la situación presentada.
- ❑ El proyecto debe ser terminado completamente para poder ser evaluado integralmente. El profesor puede evaluar que un proyecto cumplió o no con los objetivos indicados y determinar si el proyecto fue exitoso o no.
- ❑ Las normativas de los proyectos serán entregadas por parte del profesor, con suficiente antelación para ser resuelto.

- El desglose de la evaluación del proyecto será entregada por el profesor junto con el enunciado del proyecto. El profesor se reserva un porcentaje de la calificación de los proyectos para evaluación individual y demostración de lo aprendido.
- **La no presentación** de código fuente en tareas o proyectos programados, se calificará con nota cero. Aquellos programas que no cumplan con los requerimientos mínimos establecidos en el enunciado se calificará con nota cero.
- La comprobación de que alguna tarea individual, proyecto o examen es una copia, se aplicará las sanciones que contemple el reglamento de Régimen Académico Estudiantil. Y puede ser consultado en: http://cu.ucr.ac.cr/normativ/regimen_academico_estudiantil.pdf

7. Cronograma

SEMANA 1	ACTIVIDADES
Tema 1-2	Inicio de Lecciones Entrega programa de curso Introducción Asignación de Lecturas
SEMANA 2	ACTIVIDADES
Tema 3	Clases Magistrales Asignación de Lecturas Taller/Prácticas/Aplicaciones
SEMANA 3	ACTIVIDADES
Tema 4	Clases Magistrales Asignación de Lecturas Taller/Prácticas/Aplicaciones
SEMANA 4	ACTIVIDADES
Tema 5	Clases Magistrales Asignación de Lecturas Taller/Prácticas/Aplicaciones Proyecto Programado /Investigación
SEMANA 5	ACTIVIDADES
Semana Santa	Para la casa Asignación de Lecturas Taller/Prácticas/Aplicaciones
SEMANA 6	ACTIVIDADES
Tema 6	Clases Magistrales Asignación de Lecturas
Semana 7	ACTIVIDADES
Primer Examen	Examen Parcial I
SEMANA 8	ACTIVIDADES
Tema 7	Asignación de Lecturas Taller/Prácticas/Aplicaciones

SEMANA 9	ACTIVIDADES
Tema 8	Asignación de Lecturas Clases magistrales Taller/Prácticas/Aplicaciones
SEMANA 10 y 11	ACTIVIDADES
Tema 9-10	Clases Magistrales Asignación de Lecturas Taller/Prácticas/Aplicaciones
SEMANA 12	ACTIVIDADES
Tema 11-12	Proyecto Programado /investigación
SEMANA 13 y 14	ACTIVIDADES
Tema 13-14	Clases Magistrales Asignación de Lecturas
SEMANA 15 y 16	ACTIVIDADES
Examen final Proyectos	Examen Parcial II Entrega de Proyectos e investigaciones
SEMANA 17 y 18	ACTIVIDADES
Entrega de Notas Ampliación.	

8. Normativa del Curso

- Se les facilitará documentación sobre presentaciones orales y sobre formato de documentos de investigación, con base en los cuales serán evaluados sus trabajos.
- El material utilizado en el desarrollo del curso podrá ser tanto en el idioma español como en inglés.
- El estudiante se ve en la obligación de tener suficientes respaldos como lo considere necesario de los trabajos por entregar.
- Cada estudiante tiene la obligación de manejar respaldos de sus trabajos y tareas sin excepción alguna durante todo el semestre, y estos pueden ser solicitados en cualquier momento luego de calificados.

9. Bibliografía

- Barcia, Roland et al. **Persistence in the Enterprise: A Guide to Persistence Technologies**. IBM Press, USA, 2008 *
- Bauer Christian, King Gavin. **Java Persistence with Hibernate**. Manning Publications Co. USA 2007 *
- Brown, Donald; Davis, Chad Michael; Stanlick, Scott. **Struts 2 in Action**. Manning Publications, USA, 2008
- Brown, Erik. **Windows Forms Programming with C#**. Segunda Edición, Manning Publications Co. 2006 *
- Doray, Arnold. **Beginning Apache Struts From Novice to Professional**. Apress. USA. 2006
- Feuerstein Steven, Harrison Guy. **MySQL Stored Procedure Programming**. USA: O'Reilly 2006 *
- Fitzgerald, Michael. **Learning XSLT**. O'Reilly. USA. 2004
- FitzGerald Neil, Edkins James, Jonker Annette, Voloshko Michael. **Crystal Reports® XI: Official Guide**. USA: Sams 2006.
- Geary David, Horstmann Cay. **Core JavaServer™ Faces**. 2da. Edición. Prentice Hall. 2007.
- Jendrock Eric, et al. **The Java EE 5 Tutorial**. Sun Microsystems, 2008 *
Disponible en: <http://java.sun.com/javase/5/docs/tutorial/doc/index.html>
- Heffelfinger, David R. **JasperReports for Java Developers: Create, Design, Format, and Export Reports with the World's Most Popular Java Reporting Library**. Packt Publishing. UK. 2006
- Holmes, James. **Struts-The Complete Reference**, 2da. Edición. USA: McGraw-Hill/Osborne 2007.
- Husted Ted N, Dumoulin Cedric, Franciscus George, Winterfeldt David. **Struts in Action: Building web applications with the leading Java framework**. Manning Publications Co. USA. 2003 **
- Lhotka, Rockford. **Expert C# 2008 Business Objects**. Apress, USA, 2009 *
- McLaughlin Brett D., Edelson Justin. **Java and XML**. 3era. Edición. USA: O'Reilly, 2006
- Olsson, Tommy; O'Brien, Paul. **The Ultimate CSS Reference**. SitePoint, USA, 2008
- Sanders, William B. **ASP.NET 3.5 A Beginner's Guide**. McGraw-Hill, USA, 2009 **
- Sarknas, Paul. **Pro ASP.NET 2.0 E-Commerce in C# 2005**. Apress, USA, 2006 *
- Stefano, Ceri ...[et al.] **Designing Data Intensive Web Applications**. Morgan Kaufmann Publishers, USA, 2003 *
- Apache Software Foundation, **Struts** <http://struts.apache.org/userGuide/index.html>
- **Sun Microsystems** <http://java.sun.com/>
- **ASP.NET Microsoft** <http://www.asp.net>
- Halvorson, Michael. **Microsoft Visual Basic 2008 Step by Step**. Primera edición. Microsoft Press, Washington, 2008.
- Stephens, Rod. **Visual Basic 2008 Programmer's Reference**. Primera edición. Wiley Publishing. Indianapolis, 2008.

- Sharp, John. **Microsoft C# 2008 Step by Step**. Primera edición. Microsoft Press, Washington, 2008.
- Ramírez, José Felipe. **Aprenda practicando Visual Basic 2005 usando Visual Studio 2005**. Primera edición. Pearson Educación, México, 2007.

10. Otras referencias

Además de estas fuentes de información, el profesor pondrá a disposición de los estudiantes material electrónico a su alcance tal como conferencias nacionales o internaciones, blogs nacionales o internaciones y cualquier otra fuente de información.



Universidad de Costa Rica
Sede de Occidente
Bachillerato en Informática Empresarial

