

<b>PROGRAMA</b>		
<b>MATERIA</b>	<b>Algoritmos 3</b>	
<b>CARRERAS</b>	<b>Tecnicatura Universitaria en Programación Informática</b>	
<b>SEDE</b>	<b>Miguelete</b>	
<b>CARÁCTER DE LA MATERIA</b>	<b>OBLIGATORIA</b>	
<b>PERÍODO DE VIGENCIA</b>	<b>Primer cuatrimestre 2010</b>	<b>DURACION: cuatrimestral</b>
<b>DOCENTES</b>	<b>Nicolás Passerini – Fernando Dodino</b>	
<b>MATERIAS CORRELATIVAS</b>	<b>Cursada de Algoritmos II, Finales de Algoritmos I, Matemática III, Conceptos de Arquitecturas y Sistemas Operativos</b>	
<b>CARGA HORARIA</b>	<b>Clases Teórico-prácticas: 3hs Laboratorios - 7hs</b>	<b>Total de horas semanales: 10 Total de horas de la materia: 160</b>

### **OBJETIVOS:**

Que el alumno:

- Conozca la problemática específica de la construcción de interfaces de usuario, comprendiendo sus requerimientos y restricciones funcionales, tecnológicas, arquitecturales y de usabilidad.
- Comprenda los distintos componentes que se pueden utilizar para construir una interfaz de usuario, las formas de organizar esos componentes, los patrones de diseño más comunes asociados a esta problemática.
- Comprenda la vinculación entre una interfaz de usuario y el modelo de dominio subyacente, y posea las herramientas necesarias para estructurar una aplicación separando las diferentes incumbencias; en particular manejar abstracciones que permitan:
  - lidiar con la problemática tecnológica de las interfaces de usuario sin afectar a los componentes del sistema que no estén directamente vinculados con la tecnología.
  - permitir la evolución independiente de un modelo de dominio y su interfaz con el usuario correspondiente, definiendo formas adecuadas de interacción entre ambos.
- Conozca, experimente y pueda comparar distintas alternativas para la construcción de interfaces de usuario y su vinculación con un modelo de dominio.
- Comprenda la problemática básica de las interfaces remotas.
- Sea capaz de utilizar todas las herramientas de programación, diseño y metodológicas que conoce previamente en el dominio específico de la presentación, en particular diseño orientado a objetos, unit testing, manejo de errores, programación distribuida, seguridad.

- Sea capaz de analizar una tecnología de presentación desde la perspectiva del conjunto de conceptos teóricos incorporados; a partir de ese análisis entender las ventajas y desventajas de diferentes opciones tecnológicas y elegir la más adecuada para un contexto dado, proponiendo a su vez los complementos necesarios para construir una solución apropiada.
- Conozca diferentes opciones tecnológicas utilizadas en la industria y sea capaz de construir aplicaciones sencillas utilizando dichas tecnologías, focalizando en la construcción de aplicaciones administrativas.
- Reconozca y adquiera algunas buenas prácticas en el desarrollo de la interfaz de usuario de una aplicación.

## **CONTENIDOS MÍNIMOS**

- Diferentes tipos de interfaces de usuario, interfaces de línea de comando, gráficas, web, RIA, etc. Diferentes tipos de aplicaciones cliente: clientes pesados, clientes web, application clients. Características y aplicabilidad de cada uno.
- Interfaces orientadas a eventos, pedido-respuesta, basadas en continuations. Aplicaciones client-initiative y application-initiative.
- Concepto de servidor. Arquitecturas cliente-servidor, centralizada, distribuida. Arquitecturas Web, servidores y protocolos propios de la arquitectura web.
- Componentes gráficos usuales de las interfaces de usuario. Diferentes estrategias para describir una vista, sus componentes y la distribución espacial de los mismos: HTML Estático, CSS, generación programática de HTML, server pages, templates, transformación de xml, descripción basada en componentes, descripciones declarativas, generación automática. Problemas característicos de cada estrategia; herramientas que las soportan.
- Vinculación entre la interfaz de usuario y el modelo de dominio subyacente. Conceptos de modelo de la vista, binding. Problemática asociada a transformaciones, validaciones, manejo de errores, excepciones, transacciones e identidad.
- Adaptaciones de los objetos de dominio a la interfaz de usuario. Objetos de nivel de aplicación, casos de uso, servicios, daos. Patrones de interacción, MVC.
- Navegación y manejo del estado conversacional. REST, estado en sesión.
- Aplicaciones distribuidas. Comunicación sincrónica y asincrónica.
- Seguridad.
- Testeo unitario.
- Nociones de usabilidad: concepto, pertinencia, conveniencia de definir y mantener standards.
- Análisis de performance de una interfaz de usuario, pruebas de stress y carga.

## **FUNDAMENTACIÓN**

Este espacio procura poner al alumno en contacto con las múltiples variantes tecnológicas para la construcción de interfaces de usuario que se encuentran presentes actualmente en la industria del software.

La materia busca el doble objetivo de proveer un marco teórico bajo el cual analizar y guiar la construcción de interfaces de usuario, a la vez que aprender herramientas tecnológicas que resulten de aplicación directa en la industria.

Junto con los contenidos teóricos, se presentará múltiples variantes tecnológicas que cubran los diferentes estilos de construcción de interfaces de usuario utilizados en la actualidad. Esta variedad tecnológica permite la visualización de un mismo concepto teórico en distintos ámbitos, con el objetivo de evitar la asociación de un concepto con una herramienta puntual.

Al mismo tiempo aprender diferentes herramientas con objetivos similares tiene por finalidad que el alumno se familiarice con el proceso de incorporar una nueva herramienta tecnológica a su base de conocimientos, lo que resulta particularmente necesario en una industria que se encuentra actualmente en permanente evolución y renovación como es la de las interfaces de usuario.

### **METODOLOGÍA DE EVALUACIÓN**

La evaluación se realizará a partir de las actividades prácticas. El alumno deberá demostrar su comprensión de la teoría en:

- trabajos de desarrollo, que le permitan aplicar sus conocimientos teóricos en una aplicación real de mediana complejidad.
- trabajos de investigación, que le permitan aprender nuevas tecnologías en forma autónoma y analizarlas a la luz del marco teórico adquirido.

### **METODOLOGÍA DIDÁCTICA**

Los contenidos del espacio se organizan en clases teóricas y prácticas. Los contenidos teóricos se acompañarán en las clases prácticas por el aprendizaje de tecnologías seleccionadas específicamente para la aplicación de cada uno de ellos.

Las clases prácticas se realizarán en laboratorio y los alumnos (individualmente o en parejas) construirán aplicaciones que permitan poner en práctica los contenidos teóricos, con la guía del profesor.

También se prevé destinar tiempo de las clases prácticas para el seguimiento de los trabajos prácticos, respondiendo consultas de diseño y destrabando problemas que pudieran aparecer con las diferentes tecnologías utilizadas.

### **PROGRAMA ANALÍTICO**

#### **UNIDAD 1: Introducción a las Interfaces de Usuario**

Diferentes tipos de interfaces de usuario, interfaces de línea de comando, gráficas, web, RIA, otras. Diferentes tipos de aplicaciones cliente: clientes

pesados, clientes web, application clients. Características y aplicabilidad de cada uno.

### **UNIDAD 2: Diseño de Interfaces de Usuario**

Programación de interfaces de usuario. Concepto de componente, eventos y binding. Diferentes mecanismos de interacción entre vista y modelo (MVC y variantes). Mecanismos para distribución gráfica de los componentes, concepto de layout. Navegación. Herramientas conceptuales para modelado del comportamiento de una aplicación. Patrones de diseño de interfaces de usuario.

### **UNIDAD 3: Introducción a la Tecnologías Web**

Tecnologías web básicas: HTTP, HTML, CSS, Javascript. Generación de HTML basada en templates. Manejo del estado en aplicaciones web. Aplicaciones REST. Manejo de la navegación, formularios, actualizaciones parciales y asíncronas (AJAX). Patrones de diseño web.

### **UNIDAD 4: Abstracciones para Construcción de Aplicaciones Web**

Desarrollo web basado en componentes. Extensión de los conceptos de binding y eventos a la programación web. Abstracciones para el manejo de navegación. Continuations.

### **UNIDAD 5: Rich Internet Applications**

Comparación de las tecnologías para producir aplicaciones RIA: Javascript, Flash, XUL, otras. Conceptos básicos de programación distribuida, mecanismos de comunicación: Web Services, SOAP, RPC, Ajax. Arquitecturas cliente-servidor. Comunicación sincrónica y asíncrona. Patrones de diseño para aplicaciones distribuidas; introducción a los conceptos de servicio, facade, DTO.

## **TRABAJOS DE LABORATORIO**

Los trabajos de laboratorio se llevarán a cabo en tres etapas:

- **Objetivo 1: *Fundamentos de las Interfaces de Usuario***

En la primera etapa se realizarán actividades prácticas sencillas con el objetivo de familiarizarse con los conceptos de base y con algunos de los aspectos tecnológicos más frecuentemente utilizados.

- **Objetivo 2: *Diseño de Interfaces de Usuario***

Desarrollo de una interfaz de usuario completa sobre un dominio de mediana complejidad, que permita encontrarse con problemas de diseño que resulten interesantes.

En este trabajo práctico se evaluará la posibilidad del alumno de aplicar los contenidos teóricos aprendidos en una aplicación concreta.

- **Objetivo 3: Autonomía y capacidad de análisis.**

Se enfrentará al alumno a nuevas tecnologías para la construcción de interfaces de usuario. El alumno deberá hacer uso de los recursos disponibles para poner en marcha por sus propios medios una interfaz sencilla utilizando una tecnología para él previamente desconocida. Asimismo se lo impulsará a analizar el nuevo marco de trabajo a la luz de los conceptos teóricos adquiridos.

En un ambiente de cambio tecnológico permanente, la capacidad de incorporar nuevas herramientas y de extrapolar los conocimientos teóricos a nuevos ámbitos es de vital importancia para un programador profesional.

## **BIBLIOGRAFÍA**

### ***Diseño de sistemas aplicado a las interfaces de usuario***

- *Designing Object-Oriented User Interfaces* - Dave Collins.
- *Domain-Driven Design: Tackling Complexity in the Heart of Software.* - Eric Evans.
- *Patterns of Enterprise Application Architecture.* - Martin Fowler.

### **Usabilidad**

- *Don't Make Me Think: A Common Sense Approach to Web Usability* - Steve Krug.
- *The Design of Everyday Things* - Donald Norman.

### **Tecnología**

- *Wicket in Action*  
- Martijn Dashorst and Eelco Hillenius
- *Flex 3 in Action*  
- Tariq Ahmed

### ***Nociones básicas de diseño y programación (bibliografía de soporte y referencia)***

- *Design Patterns: Elements of Reusable Object-Oriented Software* - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides.
- *The Pragmatic Programmer: From Journeyman to Master*  
- Hunt, Andrew & Thomas, David