

PROGRAMA de Construcción de Interfaces de Usuario

Carreras: Tecnicatura Universitaria en Programación Informática - Licenciatura en Informática

Asignatura: Construcción de Interfaces de Usuario

Núcleo al que pertenece: Básico

Profesores: Nahuel Palumbo y Fernando Dodino

Asignaturas Correlativas: Programación con Objetos II

Objetivos:

Que el estudiantado:

- 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.
- Conozca, experimente y pueda comparar distintas alternativas para la descripción de interfaces de usuario/a y su vinculación con el modelo de dominio subyacente, y comprenda la pertinencia tanto de permitir la evolución independiente del modelo de dominio y de la interfaz con el usuario/a como de acotar el impacto de la problemática tecnológica de las interfaces de usuario en los componentes no vinculados directamente con la tecnología.
- Sea capaz de utilizar distintas herramientas de programación, diseño y metodológicas que conoce previamente en el dominio específico de la presentación, en particular diseño orientado a objetos, unit testing y manejo de errores.
- Comprenda la problemática básica de las interfaces remotas.
- Conozca diferentes opciones tecnológicas utilizadas en la industria y adquiera algunos criterios que lo ayuden a discernir entre distintas alternativas permitiéndole elegir las más adecuadas a las condiciones de un proyecto de desarrollo específico.
- Cuento con elementos para aprender una tecnología vinculada a la construcción de interfaces de usuario/a desconocida por el/ella e incluirla en el marco de los conocimientos vistos en la materia.
- Sea capaz de construir aplicaciones sencillas aplicando los conceptos vistos en la materia y utilizando distintas tecnologías.
- Reconozca y adquiera algunas buenas prácticas de aplicación particular al desarrollo de la interfaz de usuario de una aplicación.

Contenidos mínimos:

- Variantes en arquitecturas de sistema respecto de la interfaz de usuario: aplicación centralizada, cliente-servidor o distribuida; ejecución en un cliente de aplicación (browser, flash, otros) o mediante un programa específico; concepto

de RIA.

- Arquitecturas web, protocolos y tecnologías asociados.
- Modelos de interacción de la interfaz de usuario/a con su entorno: interfaces orientadas a eventos, pedido-respuesta, basadas en continuations. Aplicaciones client-initiative y application-initiative.
- Componentes gráficos usuales en 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, descripción basada en componentes, descripciones declarativas. Problemas característicos de cada estrategia; herramientas que las soportan.
- Vinculación entre la interfaz de usuario/a y el modelo de dominio subyacente. Problemática asociada a transformaciones, validaciones, manejo de errores, excepciones, transacciones e identidad. Distintos enfoques: generación automática de la interfaz de usuario/a a partir del modelo, vínculos explícitos entre elementos de interfaz de usuario/a y de modelo, DAOs, servicios.
- Adaptaciones de un modelo de dominio a las necesidades de dinamismo, navegación y distintos niveles de discriminación/agregación de la interfaz de usuario. Objetos de nivel de aplicación, casos de uso, concepto de modelo de la vista. Patrones de interacción, mvc.
- Análisis de tecnologías de presentación de acuerdo a los conceptos presentados en esta materia; evaluación de características, selección de opciones tecnológicas teniendo en cuenta el proyecto de desarrollo a realizar. Nociones sobre desarrollo propio de complementos a tecnologías desarrolladas por otros.
- Impacto de la distribución de aplicaciones en la interfaz de usuario/a, comunicación sincrónica y asincrónica.
- Navegación y manejo del estado conversacional. REST, estado en sesión.
- Nociones de usabilidad: concepto, pertinencia, conveniencia de definir y mantener standards.

Carga horaria semanal: 6 hs

Programa analítico:

Unidad 1: Introducción a las interfaces de usuario

Diferentes tipos de interfaces de usuario/a (interfaces de línea de comando, gráficas, web, móviles, otras). Diferentes tipos de aplicaciones cliente: clientes pesados, clientes web. Aplicaciones distribuidas. Características y aplicabilidad de cada uno. Fenómeno de la interacción, nociones de usabilidad y diseño centrado en el usuario.

Unidad 2: Diseño de interfaces de usuario

Programación de interfaces de usuario, concepto de componente, elementos comunes a las interfaces, eventos y binding. Diferentes mecanismos de interacción entre vista y modelo: MVC, MMVC (MVVM/MVB), otras 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/a. Validaciones y manejo de errores. Estado conversacional. Ciclo de vida de los objetos.

Unidad 3: Tecnologías Web

Características de los sistemas web: modelo pedido-respuesta, asíncrono y sin estado. Tecnologías web básicas: HTTP, HTML, CSS, modelo cliente servidor, generación de HTML basado en templates. Manejo del estado en aplicaciones web. Aplicaciones REST. Manejo de la navegación, formularios. Patrones de diseño web. Comportamiento client-side. Javascript. Actualizaciones parciales y asíncronas (AJAX). Single-page application. Aplicaciones centradas en el servidor. Servidor de aplicaciones. Extensión de los conceptos de binding y eventos a la programación web. Idea de callback y continuation. Arquitectura orientada a servicios. Manejo de estado, estado conversacional en la web.

Unidad 4: Tecnologías Móviles

Introducción al desarrollo de aplicaciones mobile. Aplicaciones web e híbridas. Aplicaciones Responsive. Arquitectura de una aplicación Android: aplicaciones, frameworks, librerías, intents y services. Diseño de la interfaz visual. Layout. Navegación. Binding manual contra objetos de dominio. Manejo de eventos. Recursos. Comunicación con el server. Push Notifications.

Unidad 5: Arquitectura de las interfaces de usuario

Comunicación sincrónica y asíncrona. Interfaces de Manipulación directa. Herramientas de automatización de tests de interfaces de usuario/a. Nociones básicas de performance. Problemas típicos de seguridad en las interfaces de usuario/a. Nociones de usabilidad: affordance, modelo mental, mapeo con la realidad, uso de convenciones, metáforas, consistencia, feedback, restricciones, modos. Testing: tipos de tests (integración, carga, stress, usabilidad, unitarios). Pruebas de usabilidad. Consecuencias de la automatización de tests de interface.

Bibliografía obligatoria:

- Jenifer Tidwell, Designing Interfaces: Patterns for Effective Interaction Design, O'Reilly Media, 2nd edition, 2011
- Don Norman, The Design of Every Day Things, Basic Books, Inc., New York, NY, USA, 2013.
- Steve Krug, Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability, New Riders, 3 edition, 2014
- Martin Fowler, Patterns of Enterprise Application Architecture. Particularmente: Web Presentation Patterns y Session State Patterns. Addison Wesley Longman Publishing Co., Inc., Boston MA, USA, 2002.
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley Longman Publishing Co., Inc., Boston, MA, USA. 1994.
- Dmitry Jemerov, Kotlin in Action, Manning Publications, 1st Edición, 2017.
- Leonard Richardson & Mike Amundsen & Sam Ruby, RESTful Web APIs: Services for a Changing World, O'Reilly Media, 1st Edition, 2013.
- Jon Duckett, Web Design with HTML, CSS, JavaScript and jQuery Set, Wiley, 1st edition, 2014.
- Douglas Crockford, JavaScript: The Good Parts (Inglés), O'Reilly Media, 1st edition, 2008.

- Alex Banks & Eve Porcello, Learning React: Functional Web Development with React and Redux, O'Reilly Media, 1st edition, 2017.

Bibliografía de consulta:

- Jonathan Anderson, John McRee, Robb Wilson, Effective UI: The Art of Building Great User Experience in Software, O'Reilly Media, 2010.
- Carolyn Snyder, Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces (Interactive Technologies), Morgan Kaufmann, 2003.
- Eric Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison-Wesley Professional, 2003

Organización de las clases

Actividades de aprendizaje

Para facilitar el aprendizaje se propone reforzar los conceptos a través de actividades prácticas.

Realizar prototipos de interfaces: presentar la necesidad de interacción con un sistema describiendolo desde el punto de vista del usuario/a que desea usarlo. Prototipar al menos dos vistas que cumplan con los requerimientos y realizar una prueba de uso con cada opción. Dependiendo del tamaño del curso y el equipo docente docentes en el aula la actividad puede desarrollarse en forma conjunta o en grupos de trabajo más pequeños.

Resolución de problemas: planteo de una situación problemática a resolver, especificando ejemplos de uso concretos de la aplicación dando un perfil del usuario/a que interactúa con la misma. Durante la actividad, fomentar la proposición de resoluciones y realizar al menos una prueba de concepto en papel de las mismas. Determinar fortalezas y debilidades de la alternativa seleccionada según el perfil de uso de la aplicación. Codificar y analizar los resultados con la interfaz real.

Análisis de soluciones: plantear una situación problemática y presentar dos soluciones a todo el curso. En las soluciones una debe aplicar los conceptos vistos en la materia y la otra no. Contrastar las características de ambas soluciones y destacar las consecuencias de las decisiones tomadas y ventajas/desventajas de cada una. Extender el problema adicionando una vista similar a la requerida en primer lugar y otra completamente distinta, analizar la dificultad de implementarlas siguiendo cada una de las alternativas propuestas.

Comparación de tecnologías: realizar una comparación en base a los conceptos vistos en la materia de cómo permite cada tecnología resolver cada tipo de problema (por ejemplo: layout, navegación, manejo de estado, comunicación modelo-vista). Determinar si los conceptos vistos en la materia son posible de ser implementados y que tan costoso sería en caso de ser posibles.

Introducción de cambios: proponer modificaciones que produzcan cambios en lo construido detectando consecuencias de las decisiones tomadas durante el desarrollo.

Trabajo práctico grupal: desarrollar en su totalidad un programa en diferentes tecnologías (desktop, web centrada en el cliente, web centrado en el servidor y

mobile). Utilizar siempre el mismo dominio y realizar pantallas similares de forma de terminar construyendo un único dominio del problema con cuatro interfaces de usuario alternativas. Contrastar resultados, justificar las decisiones tomadas, el código construido y presentar modelos alternativos a lo desarrollado.

Proceso de Intervención pedagógica

Para el dictado de las clases se propone:

Dictado de clase teórica: Cada clase comienza con un repaso (breve) de los temas vistos con anterioridad y que son relevantes al tema de la clase actual. En todos los casos se plantea la necesidad del tema a desarrollar, el tema en sí y su relación con la práctica directa. Como disparador del tema a desarrollar se usan ejemplos de situaciones cercanas a la vida cotidiana de los/las estudiantes. Toda clase teórica debe terminar con una recapitulación de lo visto.

Discusiones grupales: fomentar la participación de quienes cursen, al incentivarlos a dar su opinión sobre las soluciones presentadas en clase, mostrando variantes y sugiriendo modificaciones. Para ello el objetivo es que participen en la construcción del concepto y presentación mostrando primero su utilidad práctica. Luego, formalizar lo visto y sugerir material de referencia si es necesario.

Trabajo en grupo: es muy importante fomentar la discusión entre pares, con lo que se propone realizar la mayoría de las actividades de forma grupal, de esta manera se facilita la presentación de distintas opiniones.

Tutor o tutora asignada al grupo: para ordenar la discusión grupal y poder dar seguimiento más personalizado se propone asignar una/un tutor a cada grupo para que quienes cursan la materia, puedan consultar e intercambiar opiniones de forma más directa. El tutor/ra detectará posibles problemas y brindará soporte para resolverlos así también sugiriendo herramientas tecnológicas que se adapten a las necesidades de cada grupo.

Trabajos Prácticos

El estudiantado cuentan con guías prácticas de acuerdo a cada una de las unidades de la materia. Las guías cuentan con recomendaciones y uso de material básico como así también explicación de las actividades. Además se cuenta con una serie de TPs a realizar en forma grupal. El objetivo del TP es presentar un problema real que debe solucionarse bajo el paradigma orientado a objetos. El problema se presenta de forma incremental. Si bien en cada TPs se trabaja sobre una o más Unidades se mantiene un único dominio común que es necesario seguir utilizando en cada trabajo. Dado que cada TP plantea la interfaz de usuario de una tecnología diferente, es por ello que el dominio es único y compartido entre TPs, de manera de poder analizar y comparar los distintos aspectos de cada interfaz. El estudiantado tiene la libertad de elegir sus ambientes de desarrollo (IDEs) pero deben utilizar la tecnología de interfaz planteada en cada TP. Cada TP cuenta con una presentación del cuerpo docente donde se explican los objetivos y alcances de cada trabajo. El estudiantado trabaja en grupos (de 2 o 3) todos y todas, deben demostrar conocimiento acerca del trabajo realizado. Durante las clases prácticas quienes cursen la materia cuentan con la guía de los/ las docentes del curso.

Los/las estudiantes reciben un dominio ya programado, que puede ser un trabajo desarrollado en una materia previa o bien ha sido provisto por el equipo docente. En base a ese dominio deben generar diferentes interfaces de usuario. Los trabajo

prácticos a desarrollar son:

- ☛ Trabajo Práctico 1. Interfaz Desktop: se genera una aplicación desktop que conecta directamente con el dominio. Se plantean conceptos de MVC/MVVM, Binding bidireccional y se presentan los componentes clásicos de las interfaces de usuario.
- ☛ Trabajo Práctico 2. Interfaz Web Server: se agrega funcionalidad al trabajo anterior generando una interfaz con tecnología web-server con cliente liviano. Se incorporan los conceptos de HTTP y server rendering.
- ☛ Trabajo Práctico 3. Interfaz Web Client: se genera una aplicación web complementaria para introducir los conceptos de arquitectura cliente-servidor con cliente pesado. Se utiliza también un modelo REST. Se plantea el concepto de SPA (Single Page Application).
- ☛ Trabajo Práctico 4. Interfaz Mobile: se complementa la aplicación con una interfaz Mobile. Se analizan las formas de generar interfaces mobile en un ambiente con multiplicidad de resoluciones potenciales.

Seguimiento del estado de la cursada: siempre accesible para quienes cursen la materia, con información detallada del estado de su cursada: trabajos presentados, resultados obtenidos, calificaciones pendientes, próximas fechas de presentación y evaluación a nivel global de su cursada.

Material de consulta: ordenado por temas vistos en la materia y con ejemplos prácticos de referencia, principalmente en la tecnología usada durante la cursada y contando con variantes para que el estudiantado pueda contrastar con lo visto en clase.

Respecto actividades extra-áulicas obligatorias, el trabajo práctico integrador se desarrollará a lo largo de toda la cursada fuera de horario de clase. Es por eso que dictamos temas involucrados con el trabajo en equipo, y así fomentar la participación y el intercambio de ideas y facilitar la tarea por parte de los/las estudiantes.

Tiene como objetivos:

- ☛ Demostrar que quienes cursan pueden comprender los temas dictados en clase y ser capaz de aplicarlos en un proyecto "real".
- ☛ Acercarlos/las al mundo de la industria, en donde deberán trabajar con otros/otras profesionales, implementar ideas de negocio y aplicar los modelos planteados en la materia con criterio.
- ☛ El uso de tecnologías industrialmente aceptadas, claves para el desarrollo de software, en general.

La nota del trabajo práctico es grupal y se conforma el 50% de la nota final, junto con el otro 50% proveniente del parcial.

Parte de la carga horaria de esta asignatura forma parte del grupo de espacios curriculares de la carrera que implementan horas de Instancias de Formación de Prácticas Profesionales Supervisadas (IFPPS), que se regulan a través del Reglamento aprobado por Res. (CDCyT) N.º 034/21, o cualquier otra resolución que la modifique o la reemplace. En estas instancias quienes cursen deben cumplir con la evaluación por la aplicación de los conceptos teóricos/prácticos aprendidos en relación con los objetivos de aprendizaje. Así mismo se formulan otros objetivos vinculados a la

solución de un problema del mercado laboral -en el cual se insertarán como profesionales. De esta manera, en lo que respecta a las horas totales de la materia, se dedican 10 horas totales para las IFPPS y las 98 horas restantes para clases teóricas y prácticas (trabajos prácticos usuales) necesarias para el aprendizaje de los conceptos.

Modalidad de evaluación

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18)

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

CRONOGRAMA TENTATIVO

Se ma na	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Introducción a la materia (condiciones de aprobación, repaso de dudas, tps, parcial) / UNIDAD 1	Introducción a interfaces de usuario	Resolución Escrita	Setup ambiente		
2	Fenómeno de interacción / UNIDAD 5	UX – HCI -UCD - Paper Prototyping	Escrito	TP1 – consulta de dudas	Construcción de prototipo con roles de usuario	
3	MVC / UNIDAD 2	MVC – Binding – Navegación – Validaciones – Adapters y Transformers	Resolución en PC	TP1 – consulta de dudas	Corregir programa y adicionar comportamiento	
4	Modelo de Aplicación / UNIDAD 2	MMVC – Distribución de Responsabilidades	Resolución en PC	TP1 – consulta de dudas	Modificar programa funcionando (agregar componentes)	
5	Clase de consulta por trabajo práctico	--	Resolución en PC	TP1 – Clase de consulta		
6	Correcciones al trabajo práctico	--				Entrega TP1
7	Parcial – Introducción a WEB / UNIDAD 3	Intro web + Server side. HTTP Server + Comunicación cliente-Servidor.	Resolución en PC	Publicación servicio REST		Parcial

		Aplicación RIA				
8	Tecnologías Web / UNIDAD 3	Tecnologías web básicas: HTTP, HTML, CSS, modelo cliente servidor, generación de HTML basado en templates. Manejo del estado en aplicaciones web. Aplicaciones REST. Servidor de aplicaciones. Extensión de los conceptos de binding y eventos a la programación		TP 2 - consulta de dudas		
9	Tecnologías Web / UNIDAD 3	Manejo de la navegación, formularios. Patrones de diseño web. Comportamiento client-side. Javascript. Actualizaciones parciales y asincrónicas (AJAX). Single-page application. Aplicaciones centradas en el servidor. web. Idea de callback y continuation. Arquitectura orientada a servicios. Manejo de estado, estado conversacional en la web.		TP 2 - consulta de dudas		
10	Testing de interfaces de usuario / UNIDAD 5	Pruebas de usabilidad. Consecuencias de la	Resolución en PC	TP 2 - consulta de dudas	Ejemplo práctico de prueba automatizada	

		automatización de tests de interface.				
11	Correcciones al trabajo práctico					Entrega TP 2
12	Intro Aplicaciones móviles / UNIDAD 4	Aplicaciones móviles. Que es mobile? Arquitectura y ambiente de desarrollo		TP 3 - consulta de dudas		
13	Navegación – Manejo de estado / UNIDAD 4	MVC en arquitecturas móviles – nociones de aplicaciones híbridas		TP 3 - consulta de dudas		
14	Comunicación / UNIDAD 4	Diseño de la interfaz visual. Layout. Navegación. Binding manual contra objetos de dominio. Manejo de eventos. Recursos. Comunicación con el server. Push Notifications		TP 3 - consulta de dudas		
15	Correcciones al trabajo práctico					Entrega TP 3
16	Recuperatorio Parcial					Rec. Parcial
17	Recuperatorio TPS					Rec. TPS
18	Examen Integrador					Integrador