



# Construcción de interfaces de usuario

**Carrera:** Licenciatura en Informática

**Actividad curricular:** Construcción de Interfaces de Usuario

**Área:** Ingeniería de software, bases de datos y sistemas de información

**Prerrequisitos:** Programación con Objetos II

**Carga Horaria:**

- Carga horaria total 96 horas
- Carga horaria práctica: 72 hs
  - Formación Experimental: 24 hs
  - Resolución de problemas: 16 hs
  - Proyecto y diseño: 32 hs
- Carga horaria semanal: 6 horas por semana

**Objetivos:**

Que el estudiante:

- 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 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 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.



## Instituto de Tecnología e Ingeniería

- 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.
- Cuente con elementos para aprender una tecnología vinculada a la construcción de interfaces de usuario 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 (IU): 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 IU 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. Vinculación entre la IU y el modelo de dominio subyacente. Problemática asociada a transformaciones, validaciones, manejo de errores, excepciones, transacciones e identidad. Impacto de la distribución de aplicaciones en la IU, comunicación sincrónica y asincrónica. Navegación y manejo del estado conversacional. REST, estado en sesión. Nociones de usabilidad. Diseño centrado en el usuario.

### Programa analítico:

- UNIDAD 1: Introducción a las interfaces de usuario

Diferentes tipos de interfaces de usuario (interfaces de línea de comando, gráficas, web, móviles, otras). Diferentes tipos de aplicaciones cliente: clientes pesados, clientes web. Aplicaciones distribuidas. Nociones de usabilidad y diseño centrado en el usuario.

- UNIDAD 2: Interacción con el usuario

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). Estado del sistema, feedback, visibilidad, relación con el mundo real. Errores, recuperación ante errores. Heurísticas de usabilidad. Prototipos.



- UNIDAD 3: 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), otros. Distribución de componentes. Navegación. Técnicas y herramientas para diseño del comportamiento de una aplicación. Patrones de diseño de interfaces de usuario. Ciclo de vida de los objetos.

- UNIDAD 4: Tecnologías WEB

Conceptos de las tecnologías web. HTTP, HTML, CSS, HTML5. Gestión del estado en aplicaciones web. REST. Patrones web: navegación, formularios. Javascript. Actualizaciones parciales y asíncronas (AJAX). Aplicaciones centradas en el servidor. Servidor de aplicaciones. Extensión de los conceptos de binding y eventos a la programación web. Callback y continuation. Arquitectura orientada a servicios.

- UNIDAD 5: Tecnologías móviles

Introducción. Tipos: web e híbridas, Responsive. Arquitectura Android: aplicaciones, frameworks, librerías, intents y services. Diseño de la interfaz visual. Layout. Navegación. Eventos. Recursos. Comunicación con el server. Push Notifications.

- UNIDAD 6: Test de interfaz de usuarios

Manipulación directa. Herramientas para generación de tests de usuario. Conceptos de performance. Seguridad en las interfaces de usuario. Distintos tipos de test. Pruebas de usabilidad. Consecuencias de la automatización de tests de interface.

### **Bibliografía obligatoria:**

- Julie A. Jacko (Ed.) - Human-Computer Interaction Handbook (3rd Edition). CRC Press. ISBN 1-4398-2943-8., 2012
- Designing Interfaces –Jenifer Tidwell, 2011.
- The Design of Every Day Things –Don Norman, 2013.
- Patterns of Enterprise Application Architecture. Particularmente: Web Presentation Patterns y Session State Patterns – Martin Fowler, 2002.



**Instituto de  
Tecnología e Ingeniería****Bibliografía de consulta:**

- Don't Make me Think: A Common Sense Approach to Web Usability –Steve Krug, 2005.
- Effective UI: The Art of Building Great User Experience in Software – Jonathan Anderson, John McRee, Robb Wilson, 2010.
- Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces (Interactive Technologies) – Carolyn Snyder, 2003.
- Eric Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software, Addison-Wesley Professional, 2003.
- MODELS - VIEWS – CONTROLLERS - Trygve Reenskaug, 1979.
- Designing Object-Oriented User Interfaces- Dave Collins, Addison-Wesley Object Technology Series, 1995.

**Organización de las clases**

Las actividades de aprendizaje se establecen a partir de definir requerimientos simples que se integren en el diseño de una aplicación web o móvil simple pero que se corresponda con un Proyecto de final de cursada.

Experimentación: Realizar prototipos de interfaces. Prototipar al menos dos vistas que cumplan con los requerimientos y realizar una prueba de uso con cada opción.

Resolución de problemas: planteo de una situación problemática a resolver, especificando ejemplos de uso concretos de la aplicación dando un perfil del usuario que interactuará con la misma. Durante la actividad, fomentar la proposición de resoluciones y realizar al menos una prueba de concepto en papel de las mismas.

Análisis casos y soluciones: plantear una situación problemática y presentar alternativas para que se apliquen los conceptos analizados en la materia. Comparar y evaluar situaciones. Modificar la propuesta, y volver a analizar los resultados.

Proyecto integrador grupal: desarrollar una aplicación eligiendo una tecnología (desktop, web centrada en el cliente, web centrada en el servidor o móvil. Documentar y justificar las decisiones tomadas para la definición de las interfaces, y realizar una presentación final.

Trabajos prácticos: el objetivo es presentar un problema real. Las entregas se dividen en:

- Resolución del problema: apunta a la revisión de los conceptos.
- Construcción de interfaz en distintos modelos:

**Instituto de  
Tecnología e Ingeniería**

1. De escritorio.
2. Interfaces web. Se recomienda utilizar Angular JS y framework que permita generar JSON y Wicket como tecnología centrada en el servidor.
3. Interfaz móvil: Uso de los mismos servicios construidos para la primera interfaz web. Como tecnología se propone usar Android ya que no presenta restricciones de hardware propietario para desarrollar en tecnologías móviles. Uso de emuladores.

**Uso del campus virtual.**

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía así como también el programa y cronograma de la asignatura y las guías de Trabajos Prácticos y ejercicios.

**Modalidad de evaluación**

Se organiza a partir de 4 entregas parciales del proyecto a realizar y una presentación final del mismo que permitan determinar el grado de aprendizaje del alumno. En las entregas parciales se hará también una evaluación de conceptos teóricos.

La calificación de cada entrega se determinará en la escala 0 a 10

La materia podrá aprobarse mediante: régimen de promoción directa, exámenes finales regulares y exámenes libres.

- Régimen de promoción directa (sin examen final): los/las estudiantes deberán aprobar las con siete (7) o más puntos de promedio entre todas las instancias evaluativas, sean éstas parciales o sus recuperatorios, debiendo tener una nota igual o mayor a seis (6) puntos en cada una de éstas. Todas las instancias evaluativas tendrán una posibilidad de recuperación. En el caso de los ausentes en la fecha original, el recuperatorio operará como única fecha de examen.
- Exámenes finales regulares: para aquellos/as estudiantes que hayan obtenido una calificación de al menos de 4 (cuatro) y no se encuentren en las condiciones de promoción, deberán rendir un examen final que se aprobará con una nota no inferior a 4 (cuatro) puntos.

La asistencia no debe ser inferior al 75% en las clases presenciales.