

Resumenparaelexamenfinal.pdf



user_2883399



Interfaces Persona Máquina



3º Grado en Ingeniería Informática



Facultad de Informática
Universidad de A Coruña

CONSIGUE HASTA
100€ DE DESCUENTO
CON TU CUPÓN WUORACC!!!!

RACC **START**



Resumen para el examen final

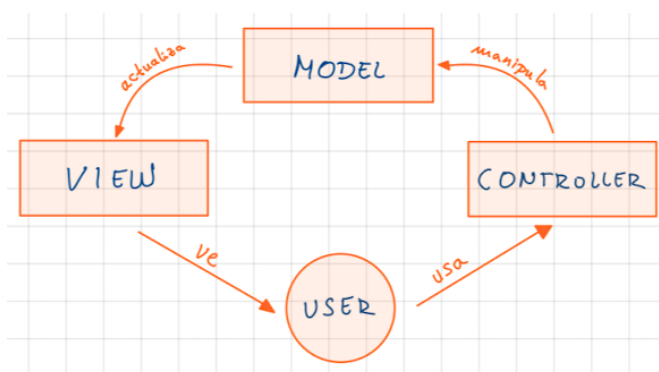


Created	@January 7, 2023 11:10 AM
Tags	
Files & Media	

Model View Controller (MVC)

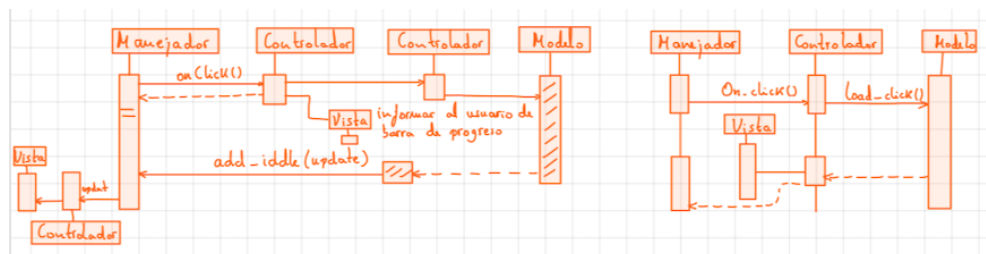
El modelo-vista-controlador es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una app de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos.

- **Modelo:** define que datos debe contener la app. Si el estado de los datos cambia, el modelo notifica a la vista y, a veces, al controlador.
- **Vista:** define cómo debe verse los datos de la app y recibe los datos del modelo.
- **Controlador:** contiene la lógica que actualiza el modelo y el modelo y/o la vista en respuesta a los inputs del usuario.



Concurrencia en las interfaces de usuario. ¿Por qué es importante?

La concurrencia permite la creación de interfaces que respondan mejor a las órdenes del usuario. Cuando una app tiene que realizar alguna tarea larga, su interfaz debería seguir respondiendo y no bloquearse. Por ejemplo, la ventana de la app debería refrescarse y no quedarse en blanco. Otro ejemplo, los botones existentes para cancelar una operación deberían cancelar la operación de un modo inmediato.



Usabilidad

Es lo bien o mal que los usuarios son capaces de usar la funcionalidad del sistema. Tiene varias DIMENSIONES:

- **Aprendizaje** → Define si es fácil de aprender a usar.

- **Eficiencia** → Una vez aprendido, cómo de rápido es de usar.
- **Recuerdo** → ¿es fácil retener lo aprendido?
- **Errores** → ¿Son escasos los errores y el sistema fácilmente recuperable?
- **Satisfacción** → ¿Es disfrutable su uso?

Dimensiones desarrollo del software

- Funcionalidad
- Rendimiento
- Coste
- Seguridad
- Usabilidad
- Fiabilidad
- etc.

Modelo en cascada para IUs

Proceso arriesgado y menos predecible de lo normal, en el que usuario no interviene hasta la fase de aceptación.

Desarrollo en espiral para IUs

La dimensión radial corresponde al coste de cada iteración. Hay un mayor control del proceso. Los errores serían más baratos de detectar en una etapa inicial (prototipado). Iteraciones más avanzadas poseen más detalles. Cada prototipo es evaluado por el usuario.

Etapas del proceso de diseño

- Análisis de usuarios / tareas.
- Diseño de la solución propuesta.
- Implementación de la solución.
- Testeo de la solución.

Pruebas de la solución implementada

- **Funcionalidad** → ¿Hace lo que tiene que hacer?
- **Usabilidad**: extraer feedback de los usuarios.
- **Pruebas de stress**: simulaciones de carga reales.
- **Recuperación antes errores**.
- **Seguridad**.
- **Alpha testing** → pruebas en las etapas iniciales del desarrollo.
- **Beta Testing** → etapas más avanzadas del desarrollo, pero aún no en producción.

Niveles de accesibilidad

- **A** → esencial
- **AA** → soporte ideal
- **AAA** → soporte especializado

**¿Hacer de mi hobby
mi trabajo no me va
a sacar de pobre?**

**ya
mi
qué!**

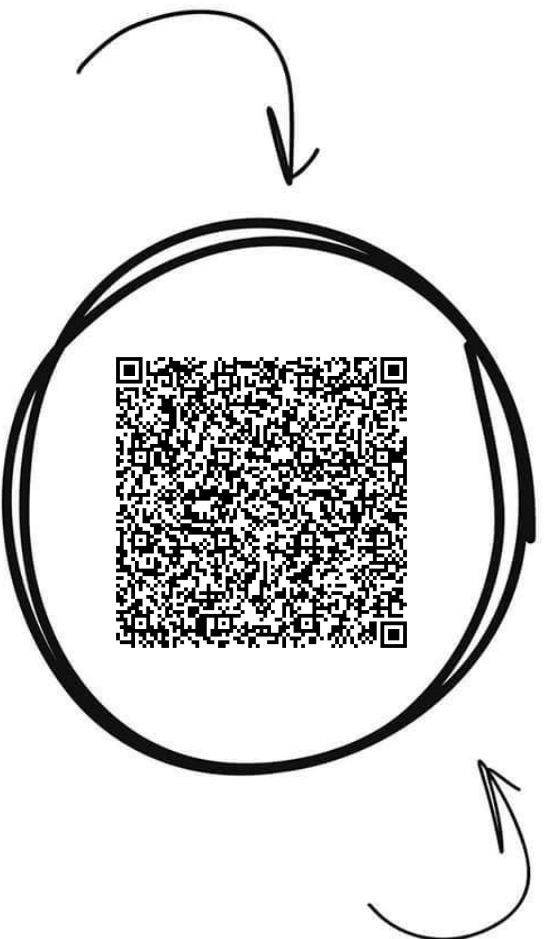
**Seguir tu propio
camino te hace
GRANDE**

Qué

Interfaces Persona Máquina



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Banco de apuntes de la

WUOLAH

- 1** Imprime esta hoja
- 2** Recorta por la mitad
- 3** Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

- 4** Llévate dinero por cada descarga de los documentos descargados a través de tu QR



Sección 508

Estándares de accesibilidad requeridos para software público en USA.

¿Cómo hacer que una web sea más accesible?

1. Navegable por teclado.
2. Agregar texto alternativo a imágenes.

```
<img alt="Texto alternativo">
```

3. Elección de los colores con cuidado → la información no puede depender del color.
4. Usar encabezados para estructurar tu contenido correctamente.
5. Formularios accesibles agrupando elementos relacionados.
6. No producir un cambio de contexto cuando el usuario selecciona o introduce info, sólo con botones o enlaces.
7. Proporcionar una descripción clara del objetivo de cada enlace/botón.
8. Cada web debe proporcionar un título que describa su propósito.
9. Comprobar que al redimensionar el texto no se rompe la estructura de la web.
10. No usar tablas para diseñar → dan problemas con lectores de pantalla.
11. No reproducir archivos multimedia automáticamente, hacer que el audio tenga descripción y que el vídeo tenga subtítulos.
12. Evitar scrolls y parpadeo de elementos.
13. CAPTCHA no solo basado en imágenes.
14. Incluir idioma general de la página.
15. Validar código HTML.

WAI-ARIA

Iniciativa de accesibilidad web que define como hacer accesibles contenidos y webs.

Roles WAI-ARIA

- Abstract roles
- Landmark roles → identifican regiones en el documento, establecen marcos de navegación para lectores de pantalla.
- Document structure roles → describen estructuras que organizan el contenido de una página.
- Widget roles → se usan para definir objetos no estándar de HTML. Un widget es un objeto de la interfaz con el cual el usuario puede interactuar.
- Live region roles → indican que el contenido de esta región puede cambiar de forma dinámica sin que el usuario interaccione con ellas.

Estados y propiedades WAI-ARIA

- Atributos de widgets.
- Atributos de relación.
- Atributos de regiones activas.

Responsive web design

Es adaptar los contenidos del sitio web a distintos dispositivos. Hay 2 posibilidades de hacerlo:



QUE LOS EXÁMENES NO TE ASUSTEN

ESCAQUÉATE CON CHEETOS



1. **Media-queries:** mismo código HTML para todos los dispositivos.

```
@media screen and (min-width: 1024px) {  
  /*Ajustes menores*/  
  section ul li { width: 20%; }  
}
```

2. Adaptaciones a dispositivos concretos en hojas de estilos diferentes para cada uno.

```
<link href="default.css" type="text/css" rel="stylesheet" media="screen"/>  
  
<link href="desktop.css" type="text/css" rel="stylesheet" media="only screen and (min-device-width: 1024px) and (max width: 989px)"/>
```

Tratamiento de algunos elementos para que sean adaptativos

- **Tipografías flexibles:** establecer tamaño de fuente al tamaño por defecto del navegador. Usar unidades relativas para escalar el tamaño de la fuente (em/rem).
- **Imágenes flexibles:** escalar imagen a medida que escalamos el grid.

```
img { max-width: 100% }
```

- **Viewport:** define el área del dispositivo y proporciona instrucciones al navegador para controlar las dimensiones de la página y el escalado.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
```

Características de elementos para web móvil

- botones y enlaces grandes (se usa la web con los dedos no con el ratón)
- Eventos como `on mouse over/move` no existen.
- Simplificar formularios y establecer su disposición en vertical, usar además el atributo "type" más adecuado.
- Reducir tamaño y usar peticiones AJAX cuando se pueda para mejorar velocidad y consumo de datos.
- Simplificar el código, reducir el número de peticiones HTTP, usar librerías mobile de Javascript...

AJAX

Combinación de objetos y funciones del navegador para realizar peticiones de datos con Javascript y HTML DOM para mostrar dichos datos. Permite actualizar páginas web de forma asíncrona sin recargar la página completa.

Técnicas de investigación para conocer que hacen los usuarios (Diseño UI)

- **Observación en el trabajo (Job Shadowing):** la idea del job shadowing es poder visitar a los usuarios en su entorno y poder extraer directamente información útil. Cumple la máxima "lo que ves es más importante que lo que te cuentan".
- **Entrevista en su contexto:** mucho conocimiento del dominio solo puede extraerse hablando con los expertos. Es conveniente, tras observar, pasar un tiempo con el usuario preguntándole sobre aspectos observados y otros que quizás no lo fueran. En la entrevista debemos evitar opiniones personales y no forzar al usuario a diseñar por nosotros.
- **Observación remota (Remote shadowing):** como job shadowing pero remotamente mediante grabaciones.

Personas

Representaciones ficticias de ciertos grupos objetivo. Permiten resumir información en una persona que un "segmento". Ayuda a comunicarnos de forma más sencilla y a focalizar el producto identificando categorías.

Diseño centrado en actividades

Aproximación más intuitiva que la centrada en usuarios, se centra en las actividades que se van a llevar a cabo en lugar de centrarse en los usuarios. Debemos identificar tareas individuales que el programa debe hacer, siendo cada tarea un objetivo.

Card sorting

Permite obtener información sobre lo que el usuario piensa de las partes individuales de nuestro producto. La idea es usar etiquetas de papel para representar los distintos índices o elementos de la estructura de la aplicación. Aclarar que se pretende que agrupen cartas similares pero desde el punto de vista de la app.

Modelo mental

Un usuario tiene ideas preconcebidas de cómo deben funcionar ciertas cosas. Ese concepto que el usuario se forma de cómo funciona algo se llama modelo mental.

Principios para ayudar al usuario a formarse modelos mentales

- Simplicidad
- Familiaridad
- Reconocimiento
- Flexibilidad
- Feedback
- Seguridad
- Affordances (potencial, adecuación)

Prototipado en papel

Diseño de la estructura

- **Diagrama de flujo:** preferiblemente sencillos y que cubran necesidades básicas.
- **Soryboards:** desenvuelven el camino de un usuario para una cierta acción en una serie de snapshots.

Diseño de pantallas

- **Wireframes:** estructura exacta de una pantalla sin detalles de decoración. Se centran en el contenido que se quiere mostrar.
- **Mock-ups:** añaden decoración a los wireframes. No es solo apariencia, también da pistas o facilita al usuario el acceso a funcionalidades.

Testing prototipado en papel

Pruebas de usabilidad

- **Testing informal (Guerrilla testing):** usamos los bocetos para mostrarle al usuario el acceso a la funcionalidad.
- **Testing más completo o formal:** nos centraremos en un subconjunto de la funcionalidad del sistema. Basado en tareas. Le pedimos al usuario que realice una tarea concreta. No estamos recabando opiniones, observamos como el usuario realiza la tarea. Decimos qué queremos que el usuario haga, no como debe hacerlo (Intuitividad).

Preparación del test

- reclutar gente.

- mejor gente poco familiarizada.
- Dividir en grupos y dejar espacio entre las pruebas de cada grupo.

Ejecución del test

No influenciar al usuario ni resolverle dudas.

Análisis de los resultados

Queremos información cualitativa, no cuantitativa.

Aspectos del diseño de interfaces

Realismo

Buscamos realismo en las interfaces, es buena idea utilizar iconos o símbolos realistas como una cámara. Simular un interfaz gráfico como un elemento del mundo real con el que comparte funcionalidad.

- **Skeuomorfismo:** estilo de diseño de interfaces, que simula objetos típicos para crear una interfaz más realista y que al usuario le parezca familiar. Versiones nuevas de objetos que retienen elementos de versiones antiguas que ya no aportan funcionalidad.

Interacción natural

Dar feedback al usuario de la manera más inmediata posible. Cuando tenga sentido, usar gestos que activen comandos, no que manipulen objetos de la pantalla. Evitar aprendizaje de gestos muy complejos. Controlar entradas accidentales.

Ley de Fitt

Es un modelo aplicado al movimiento humano que mide el tiempo necesario para alcanzar un objetivo con un movimiento rápido, es una función del tamaño de dicho objetivo y la distancia que hay que recorrer hasta él.

$$T = a + b \log_2 \left(1 + \frac{D}{W} \right)$$

a y b son constantes que dependen del dispositivo: tiempo de arranque/parada y factor inverso a la velocidad de movimiento.

- Conviene separar elementos pequeños, elementos más grandes son más fáciles de usar.

Interrupciones

Deben ser evitadas, lastran la productividad a la hora de realizar una tarea. Debe interrumpirse solo para decisiones urgentes, no para informar. Es importante dejar al usuario deshacer acciones.

Funcionalidades

Demasiadas funcionalidades pueden evitar el acceso a nuevos usuarios. Si queremos eliminar funcionalidades debemos informar a usuarios y facilitar alternativas cuando sea posible.

Animaciones

Evitar animaciones innecesarias, aunque bien implementadas pueden ayudar al usuario a formar modelos mentales adecuados.

Consistencia

armonía o lógica entre las diferentes partes del producto. Muchas veces se usa referida a la apariencia. Consistencia de comportamiento o funcionalidad.

Modos

¿Que pillo apuntes
de Wuolah porque
los míos no
los entiendo?

ya
mi
qê!

seguir tu propio
camino te hace
GRANDE

Parte de la app de la que hay que entrar o salir y que restringe o acota las operaciones que se puedan llevar a cabo mientras esté activo.

Velocidad de una app

Característica más importante, en caso de apps lentas los usuarios pueden abandonarlas. La capacidad de respuesta de un sistema es crucial. Si una operación va a durar más de lo esperado es bueno dar feedback al usuario (barra de progreso, por ejemplo). Más importante que la velocidad real es la **velocidad percibida**. Para mejorar la velocidad percibida:

- Mostrar resultados parciales cuando sea posible.
- No bloquear la interfaz con operaciones lentas → **Concurrencia**.
- Decoraciones, efectos, animaciones...