



# Universidad Pontificia de Salamanca

Trabajo práctico:

## **Aplicación de eventos y lugares de interés en CyL.**

**Asignatura:** Diseño, evaluación y desarrollo de interfaces

**Alumnos:** Barreras Sierra, Gaspar y Egido Vaquero, Bruno

**Fecha:** 05 de diciembre de 2025

Nota:

El trabajo aún no está finalizado. Falta una pequeña parte de la implementación y revisar la redacción de la documentación.

Se ha usado el chat para redactar algunas partes de esta documentación.

También se ha usado en el proyecto, en ciertas partes del JavaScript y en algún caso concreto para el CSS.

## Introducción y definición de objetivos

### Introducción

Este trabajo tiene como objetivo aplicar los conocimientos y procesos vistos en la asignatura de Diseño, Evaluación y Desarrollo de Interfaces. En concreto, debemos aplicar al proceso de diseño el modelo de **Elementos de Experiencia del Usuario** y la metodología de **Diseño Centrado en el Usuario** (DCU). Además, siempre que sea posible debemos tratar de desarrollar un producto que como mínimo cumpla con los estándares WCAG, asegurándonos así de que se trata de un producto accesible. Se nos pide desarrollar un producto digital de nuestra elección –aplicación o web–. En nuestro caso, hemos decidido crear una aplicación ya que consideramos que este formato se corresponde mejor con las funcionalidades que vamos a ofrecer.

### Definición de objetivos

La finalidad de nuestra aplicación será mostrar los **lugares de interés** y eventos cercanos a la ubicación del usuario, facilitando así la exploración de lugares y la participación en actividades sociales. Los datos se obtendrán a partir del Portal de Datos Abiertos de la Junta de Castilla y León.

## Fase de análisis de requisitos

El objetivo de la aplicación, como se explicó antes, es proporcionar a los usuarios la información y localización de los eventos y lugares de interés turístico próximos a su ubicación para facilitar su exploración y la participación en ellos. (**Plano de estrategia**)

Para ajustarnos debidamente a las necesidades de los usuarios de la aplicación aplicaremos la metodología de Diseño Centrado en el usuario (DCU). Así, **guiaremos nuestras decisiones de diseño a partir de las observaciones realizadas durante esta fase del desarrollo.**

Nosotros hemos hecho uso de la técnica *User Personas*, vista en clase. **Hemos avanzado directamente hasta la creación de los personajes ya que en nuestro caso no era posible realizar entrevistas o pruebas de usuario.**

Hemos identificado dos arquetipos de usuario para la aplicación:

## Personajes

Hemos identificado dos personajes para nuestra aplicación



**Yao Ming**

55 años

Contable en un banco

Características ¿PRINCIPAL? -> sería mas principal q el otro???

- Poco tiempo para hacer turismo ya que viene en una excursión grupal.
- No se ubica en la ciudad, es la primera vez que la visita.
- Muy interesado en la historia.

### Necesidades

- Encontrar monumentos cerca de su ubicación.
- Consultar las indicaciones para llegar a la ubicación del monumento.
- Buscar monumentos de la época romana.



**Daniel Rodríguez**

25 años

Estudiante de Filología  
Hispanica

### Características

- Tardes libres, solo tiene clases por la mañana.
- Solo se ubica por el centro, lleva poco en la ciudad.
- Aficionado a la lectura.

### Necesidades

- Descubrir actividades con la que ocupar sus tardes.
- Consultar las direcciones para llegar a la ubicación.
- Buscar clubs de lectura.

## Escenarios

Yao dispone de 1 hora libre antes de tener que regresar con el grupo. Abre la aplicación y ajusta el filtro de la pantalla “Monumentos” para que se muestren únicamente las iglesias y catedrales. Echa un vistazo a las más cercanas y descubre que hay varias de la época romana. Selecciona una de ellas, clic en el botón “Direcciones” de esta y comienza a caminar hacia su ubicación.

Mientras desayuna, Daniel quiere encontrar un plan para la tarde. Abre la aplicación de ToroTour y cambia a la pantalla “Eventos”. Ajusta el filtro para que se muestren únicamente los clubs de lectura y se fija si hay alguno ese día. Ve que sí, por lo que abre los detalles del evento para ver la hora y el lugar exactos.

## Especificación de requisitos

Este proyecto estará centraremos exclusivamente en la **interfaz** –experiencia de usuario, arquitectura de la información, diseño visual, usabilidad y accesibilidad–. La implementación de la lógica funcional se pospondrá hasta el siguiente cuatrimestre. A continuación, hemos listado los requisitos funcionales (RF), requisitos no funcionales (RNF) y requisitos de contenido (RC) de nuestra aplicación (**Plano de alcance**):

### Requisitos funcionales (RF):

#### →RF1 – Localizar puntos de interés cercanos (M):

El sistema permitirá mostrar los monumentos cercanos a la ubicación actual del usuario, utilizando geolocalización y representación cartográfica.

#### →RF2 – Visualizar monumentos en distintos modos (M):

El sistema permitirá visualizar los monumentos tanto en formato mapa como en lista, pudiendo alternar entre ambas representaciones.

#### →RF3 – Consultar información resumida de un elemento (M):

El sistema permitirá acceder a información básica de un monumento o evento al seleccionarlo desde el mapa, el calendario o una lista.

#### →RF4 – Consultar información detallada de un elemento (M):

El sistema permitirá acceder a una vista de detalle con información ampliada de un monumento o evento seleccionado.

#### →RF5 – Filtrar y buscar elementos (M):

El sistema permitirá buscar y filtrar monumentos y eventos por distintos criterios como nombre, tipo o categoría.

#### →RF6 – Obtener indicaciones hasta un monumento o evento (S):

El sistema permitirá al usuario consultar indicaciones para llegar a la ubicación de un monumento o evento seleccionado.

#### →RF7 – Consultar eventos mediante calendario (M):

El sistema permitirá consultar los eventos disponibles mediante una vista de calendario, destacando los días con eventos y el día actual.

#### →RF8 – Gestionar elementos favoritos (M):

El sistema permitirá añadir y eliminar monumentos y eventos a una lista de favoritos, accesible desde distintas partes de la aplicación.

#### →RF9 – Consultar la lista de favoritos (M):

El sistema permitirá consultar un listado con los monumentos y eventos marcados como favoritos.

#### →RF10 – Acceder a información externa de un evento (M):

El sistema permitirá acceder a la página web oficial de un evento cuando esta esté disponible.

→RF11 – Configurar preferencias básicas de la aplicación (S):

El sistema permitirá configurar preferencias básicas, como la pantalla principal que se mostrará al iniciar la aplicación.

→RF12 – Navegar entre las secciones principales de la aplicación (M):

El sistema permitirá navegar entre las distintas secciones principales de la aplicación (Monumentos, Eventos, Favoritos).

**Requisitos no funcionales (RNF):**

→RNF1 – Usabilidad general del sistema (M):

La aplicación deberá ofrecer una experiencia de uso clara e intuitiva, proporcionando retroalimentación visible ante las acciones del usuario y mostrando mensajes de error comprensibles que indiquen tanto la causa del problema como la posible solución.

→RNF2 – Accesibilidad del contenido y la interacción (M):

La aplicación deberá cumplir los principios POUR de accesibilidad y ajustarse, en la medida de lo posible, a las recomendaciones WCAG, garantizando contraste suficiente, tamaños de texto legibles, foco visible, navegación operable y evitando el uso exclusivo del color para transmitir información.

→RNF3 – Consistencia visual y coherencia gráfica (S):

La interfaz deberá mantener una consistencia visual en todas las pantallas, aplicando los principios de diseño CRAP y Gestalt en la disposición de componentes, listas, tarjetas y marcadores, con el objetivo de facilitar el reconocimiento y reducir la carga cognitiva del usuario.

→RNF4 – Compatibilidad y contexto de uso (S):

La aplicación deberá diseñarse siguiendo un enfoque mobile-first, adaptándose correctamente a diferentes tamaños de pantalla y contextos de uso, priorizando la interacción táctil.

→RNF5 – Robustez semántica y soporte técnico (S):

Cuando sea necesario, se utilizarán atributos WAI-ARIA y marcado semántico adecuado para mejorar la robustez del sistema y su compatibilidad con tecnologías de asistencia.

**Requisitos de contenido (RC):**

→Vista Monumentos – Mapa:

- RC1.1 (M): Mapa de la zona con la ubicación actual del usuario claramente destacada.
- RC1.2 (M): Marcadores de los monumentos cercanos, mostrando su localización en el mapa.
- RC1.3 (M): Campo de búsqueda con texto descriptivo para localizar monumentos por nombre.
- RC1.4 (S): Mensaje contextual o texto de ayuda breve para guiar el uso de la búsqueda y los filtros.

- RC1.5 (M): Panel informativo inferior con información resumida del monumento seleccionado.
- RC1.6 (M): En la tarjeta inferior: nombre del monumento y tipo de construcción.

→ Vista Monumentos – Lista:

- RC2.1 (M): Campo de búsqueda con texto descriptivo para localizar monumentos por nombre.
- RC2.2 (M): Listado de monumentos disponibles en la zona.
- RC2.3 (M): Para cada monumento: nombre, tipo de construcción y localidad.
- RC2.4 (S): Representación visual consistente mediante tarjetas.
- RC2.5 (S): Indicador visual de monumentos marcados como favoritos.

→ Vista Detalle Monumento:

- RC3.1 (M): Nombre del monumento.
- RC3.2 (M): Tipo y estilo de construcción.
- RC3.3 (M): Localización (localidad / provincia).
- RC3.4 (M): Periodo histórico.
- RC3.5 (M): Descripción textual del monumento.
- RC3.6 (S): Imagen representativa del monumento (cuando esté disponible).
- RC3.7 (S): Indicador visual de si el monumento está marcado como favorito.

→ Vista Eventos – Calendario:

- RC4.1 (M): Campo de búsqueda con texto descriptivo para localizar eventos por nombre.
- RC4.2 (M): Nombre del mes y año actualmente visualizados (ej. diciembre 2025).
- RC4.3 (M): Calendario mensual con los días claramente delimitados.
- RC4.4 (M): Destacado visual del día actual.
- RC4.5 (M): Indicadores visuales en los días que contienen eventos.
- RC4.6 (M): Nombre del evento mostrado en el día correspondiente.
- RC4.7 (M): En la tarjeta inferior: nombre y clasificación del evento.

→ Vista Eventos – Lista:

- RC5.1 (M): Campo de búsqueda con texto descriptivo para localizar eventos por nombre.
- RC5.2 (M): Listado de eventos disponibles.
- RC5.3 (M): Para cada evento: nombre, categoría y fecha.
- RC5.4 (S): Representación mediante tarjetas con jerarquía visual clara.
- RC5.5 (S): Indicador visual de eventos marcados como favoritos.

→ Vista Detalle Evento:

- RC6.1 (M): Nombre del evento.
- RC6.2 (M): Categoría y temática del evento.
- RC6.3 (M): Fecha y horario del evento.
- RC6.4 (M): Ubicación del evento (localidad y provincia).
- RC6.5 (M): Público destinatario.
- RC6.6 (S): Precio o indicación de gratuidad.

- RC6.7 (M): Descripción textual del evento.
- RC6.8 (S): Enlace a la página oficial del evento.
- RC6.9 (S): Indicador visual de si el evento está marcado como favorito.

#### →Vista Favoritos:

- RC7.1 (M): Campo de búsqueda con texto descriptivo para localizar elementos favoritos por nombre.
- RC7.2 (M): Título de la vista con el texto “Mis favoritos”.
- RC7.3 (S): Selector visual mediante chips que indica y permite alternar el tipo de elemento mostrado en la lista de favoritos (monumentos / eventos).
- RC7.4 (M): Listado de monumentos/eventos marcados como favoritos.
- RC7.5 (S): Representación consistente de los elementos favoritos mediante tarjetas.
- RC7.6 (S): Indicador visual de eventos marcados como favoritos.
- RC7.7 (S): Mensaje informativo visible cuando no existan elementos marcados como favoritos.

#### →Navegación global:

- RC8.1 (M): Barra de navegación inferior visible en todas las pantallas principales.
- RC8.2 (M): Etiquetas textuales claras para cada sección: Monumentos, Eventos y Favoritos.
- RC8.3 (S): Iconografía coherente y consistente con el contenido representado.
- RC8.4 (M): Acceso y retorno a las vistas principales desde vistas secundarias (detalle de monumento y evento) mediante controles de navegación visibles y convenciones de la plataforma.

/\*\*/

## Fase de diseño

Una vez hemos definido la estrategia y el alcance de nuestro producto, la siguiente capa que debemos definir será el **Plano de estructura**. Es decir, debemos definir la organización y jerarquía de la información que se muestra en la aplicación, así como el flujo de navegación para llegar hasta ella.

## Arquitectura de la información

La arquitectura de la información de nuestra aplicación se organiza en torno a los dos tipos de contenidos que se proporcionan a partir del Portal de Datos Abiertos de la Junta de Castilla y León: **monumentos** y **eventos**. Estos dos grupos estructuran la navegación general de la aplicación y permiten que el usuario acceda de manera clara y directa a la información que busca.

~~Dado que las categorías asociadas tanto a los monumentos como a los eventos ya vienen definidas en las bases de datos oficiales, nuestra labor se ha reducido a presentar esos contenidos, sin necesidad de redefinir taxonomías ni realizar técnicas de investigación adicionales como *card sorting*, simplificando así el proceso de diseño de la Arquitectura de la Información de nuestra aplicación.~~

Dado que las categorías asociadas tanto a los monumentos como a los eventos ya vienen definidas en las bases de datos que alimentan nuestra aplicación, nuestro impacto en la taxonomía ha sido mínimo.

Nos hemos reducido a analizar los datos que presentaban las bases de datos para cada objeto y a seleccionar de entre estos aquellos que aportasen valor informativo a los usuarios. Por ejemplo, los monumentos presentan un campo “Código postal”, pero estos datos no se muestran en la app.

Además, para el campo enumerado periodo histórico de monumento, si ha sido necesario hacer una pequeña redefinición de los miembros. Esto ha sido necesario ya que, a diferencia de los eventos, la base de datos de los monumentos no cuenta con un campo imagen. Por ello, y con el fin de poder añadir algún elemento visual a la pantalla de información de un monumento, hemos redefinido y reducido el número de miembros de la enumeración.

Antes de llevar a cabo la redefinición, es positivo hacer un “card-sorting” para identificar la forma en la que los usuarios agrupan los miembros en grupos más genéricos. En nuestro caso hemos simulado la prueba usando tres modelos de IA — ChatGPT, Gemini y Grok—.

→ ¿¿Habría que meter imágenes la respuesta de cada IA??

Así quedaron los grupos finalmente:

#### Prehistoria

- └─ Paleolítico
- └─ Postpaleolítico
- └─ Neolítico
- └─ Edad del Cobre
- └─ Edad del Bronce
- └─ Edad del Hierro
- └─ Prehistoria

#### Pueblos prerromanos

- └─ Prerromano
- └─ Celta (Vacceo)
- └─ Celta (Vetón)
- └─ Celtibérico (Arévaco)



Época romana

- └─ Romano
- └─ Tardorromano
- └─ Paleocristiano

Edad Media

- └─ Visigodo
- └─ Califal
- └─ Medieval
- └─ Edad Media

Edad Moderna

- └─ Edad Moderna

Edad Contemporánea

- └─ Edad Contemporánea

## Flujo de interacción

Estos son los flujos de interacción que los usuarios llevarán a cabo dentro de la aplicación. Estos flujos reflejan los **casos de uso esenciales** derivados de los requisitos.

### Flujo 1: Consultar información de un monumento

1. Usuario abre la aplicación (Vista Mapa si así lo configuró).
2. Se muestran los marcadores de monumentos.
3. El usuario selecciona un marcador.
4. Se abre la card view con la información del monumento.
5. Puede pulsar “*Cómo llegar*” para abrir indicaciones externas.

### Flujo 2: Encontrar un evento por fecha

1. Usuario accede a la pestaña **Calendario**.
2. Selecciona un día del calendario.
3. Aparece la lista de eventos para ese día.
4. Selecciona un evento.
5. La card view muestra sus detalles (categoría, fecha, lugar, precio).

### Flujo 3: Filtrar eventos por categoría

1. Usuario abre la pestaña **Calendario**.
2. Pulsa el botón “*Filtrar*”.
3. Selecciona una categoría del dataset.
4. El calendario se actualiza mostrando solo los eventos filtrados.

### Flujo 4: Fijar un evento

1. Usuario abre la card view de un evento.
2. Pulsa “*Fijar evento*”.
3. El evento se añade a la sección *Eventos fijados*.

En esta fase también se realizaron los *wireframes* y *moqups* del proyecto, quedando así definido el **Plano de Esqueleto**.

Tanto los *wireframes* como los *moqups* están incluidos en la carpeta documentación del proyecto.

/\*Redactar documentación. \*/

Por último, tenemos el **plano de superficie**, en el que se incluyen aquellos elementos que dan estilo visual a la aplicación. Es decir, colores, tipografía e iconos.

Aunque muchos de estos ya están implementados, ninguno es definitivo todavía, ya que hemos decidido dejar la implementación de este aspecto de la aplicación para el final del proceso.

## Fase de desarrollo

Actualmente seguimos con el desarrollo. Nos quedan únicamente las pantallas de información de cada objeto.

Como comentamos hemos aplicado un enfoque *mobile-first*, y en caso de necesitar ajustar la UI a otros dispositivos esto se haría con *media queries*.

/\*  
-Documentar proceso implementación  
-Indicar para que partes se uso chat  
\*/

## Fase de pruebas y evaluación

N/A