

campus.euniv.eu © Universitas Europaea IMF  
Juan Ulises PÉREZ VISAIRAS

# **Diseño responsivo y ágil**

## **© Universitas Europaea IMF**

campus.euniv.eu © Universitas Europaea IMF  
Juan Ulises PÉREZ VISAIRAS

campus.euniv.eu © Universitas Europaea IMF  
Juan Ulises PÉREZ VISAIRAS

# Indice

<b>Diseño responsivo y ágil</b>	<b>3</b>
1. Principios de Diseño Responsivo	3
1.1 Adaptabilidad: Interfaces flexibles para todos los dispositivos	3
1.2 Rejillas Flexibles (Fluid Grids)	4
Ventajas del uso de rejillas flexibles:	4
1.3 Imágenes Flexibles: Adaptación sin distorsión	4
1.4 Media Queries y Puntos de Interrupción (Breakpoints)	4
Definición de puntos de interrupción recomendados:	5
1.5 Contenido Primero (Content First)	5
Beneficios de este enfoque:	5
1.6 Optimización de Desempeño y Velocidad	5
Conclusión	6
2. Metodologías Ágiles en el Diseño	6
2.1 Iteración Rápida y Feedback Continuo	6
2.2 Colaboración entre Diseñadores y Desarrolladores	6
2.3 Flexibilidad y Priorización de Funcionalidades	7
Priorización de funcionalidades	7
2.4 Sprints de Diseño en Scrum	7
Estructura de un sprint de diseño en Scrum	7
2.5 Pruebas de Usabilidad en un Entorno Ágil	8
Ventajas de hacer pruebas frecuentes en diseño ágil	8
Conclusión	8
3. Implementación del Diseño Responsivo y Ágil	8
3.1 Evaluación de Dispositivos y Contextos de Uso	9
3.2 Herramientas y Frameworks para Diseño Responsivo	9
Frameworks de CSS	9
Sistemas de Diseño	9
3.3 Prototipado Rápido y Pruebas en Múltiples Dispositivos	10
Herramientas para prototipado	10
3.4 Pruebas de Usabilidad y Optimización Continua	10
Métodos de prueba de usabilidad	10
Herramientas para pruebas de usabilidad	10
3.5 Integración del Diseño con el Desarrollo Frontend	10
Uso de herramientas de colaboración	10
Buenas prácticas en desarrollo frontend	11
Conclusión	11
<b>Actividades prácticas</b>	<b>12</b>

# Diseño responsivo y ágil

En la actualidad, el diseño de interfaces enfrenta el reto de adaptarse a una gran variedad de dispositivos, desde teléfonos móviles hasta pantallas de escritorio. La evolución tecnológica ha llevado a un cambio en la forma en que las personas acceden a la información, lo que hace necesario que las interfaces sean **responsivas**, es decir, capaces de ajustarse dinámicamente a distintos tamaños de pantalla sin perder funcionalidad ni estética.



El **diseño responsivo** (*Responsive Web Design, RWD*) es un enfoque que permite que los sitios web y aplicaciones se visualicen correctamente en cualquier dispositivo. Este método se basa en **rejillas flexibles, imágenes adaptables y media queries**, permitiendo que el contenido fluya de manera natural según el espacio disponible. El objetivo es garantizar que la experiencia del usuario (*User Experience, UX*) sea óptima sin importar la plataforma utilizada.

Por otro lado, el proceso de desarrollo de interfaces también ha evolucionado, adoptando enfoques más **dinámicos y colaborativos**. Las **metodologías ágiles**, como **Scrum y Kanban**, han demostrado ser eficaces en la creación de productos digitales, ya que permiten una **iteración rápida, integración del feedback de los usuarios y flexibilidad ante cambios**. En el contexto del diseño de interfaces, la combinación del **diseño responsivo con metodologías ágiles** es clave para el desarrollo de productos exitosos.

Esta unidad explorará los **principios fundamentales del diseño responsivo**, las **metodologías ágiles aplicadas al diseño de interfaces** y cómo estas técnicas pueden implementarse en un entorno de desarrollo moderno.

## 1. Principios de Diseño Responsivo

El diseño responsivo (*Responsive Web Design, RWD*) es un enfoque que permite que los sitios web y aplicaciones se adapten automáticamente a diferentes dispositivos y tamaños de pantalla sin comprometer la usabilidad ni la experiencia del usuario. A medida que el acceso a internet se ha diversificado, con una gran cantidad de usuarios navegando desde teléfonos móviles, tablets, portátiles y monitores de gran tamaño, la necesidad de interfaces flexibles y adaptables se ha convertido en un estándar de la industria.

A continuación, se detallan los principios fundamentales del diseño responsivo y cómo pueden aplicarse para crear interfaces que sean visualmente atractivas y funcionalmente eficientes en cualquier dispositivo.

### 1.1 Adaptabilidad: Interfaces flexibles para todos los dispositivos

La esencia del diseño responsivo es la **adaptabilidad**, es decir, la capacidad de la interfaz para ajustarse automáticamente al entorno en el que se presenta. En lugar de diseñar versiones separadas para distintos dispositivos, el diseño responsivo permite que una sola estructura se reconfigure según el tamaño y orientación de la pantalla.

Para lograr esta adaptabilidad, los diseñadores deben considerar varios factores clave:

- **Escalabilidad de los elementos de la interfaz:** Botones, menús y tipografías deben mantenerse funcionales y legibles en cualquier tamaño de pantalla.

- **Optimización de interacciones táctiles:** Asegurar que los elementos interactivos sean accesibles tanto con ratón como con pantalla táctil.
- **Compatibilidad con distintos sistemas operativos y navegadores:** Probar la interfaz en múltiples plataformas para garantizar consistencia.

## 1.2 Rejillas Flexibles (*Fluid Grids*)

Uno de los principios más importantes en el diseño responsivo es el uso de **rejillas flexibles** en lugar de estructuras basadas en unidades fijas de medida, como píxeles. Las rejillas flexibles permiten que los elementos de la página se redimensionen proporcionalmente en función del ancho disponible en pantalla.

### Ventajas del uso de rejillas flexibles:

1. **Escalabilidad automática:** El contenido fluye de manera natural en pantallas grandes y pequeñas.
2. **Menos trabajo manual:** No es necesario crear versiones separadas del diseño para distintos dispositivos.
3. **Mejor experiencia de usuario:** Se evita la necesidad de hacer zoom o desplazarse horizontalmente.

Las rejillas flexibles pueden implementarse con **CSS Grid** o **Flexbox**, dos herramientas poderosas en el desarrollo web moderno que permiten distribuir los elementos de manera adaptable y eficiente.

## 1.3 Imágenes Flexibles: Adaptación sin distorsión

Las imágenes juegan un papel crucial en el diseño de interfaces, y deben poder adaptarse sin perder calidad ni causar problemas de carga. Para lograrlo, se aplican varias técnicas:

- **Uso de valores relativos en CSS:** Definir imágenes con `max-width: 100%` para que escalen automáticamente dentro de su contenedor.
- **Carga adaptable con `srcset` y `sizes`:** Permite que los navegadores elijan la versión óptima de la imagen según el ancho de la pantalla.
- **Compresión y formatos optimizados:** Formatos como **WebP** y **AVIF** reducen el peso sin sacrificar calidad.

Optimizar las imágenes no solo mejora la experiencia del usuario, sino que también contribuye a un **mejor rendimiento y menor consumo de datos en dispositivos móviles**.

## 1.4 Media Queries y Puntos de Interrupción (*Breakpoints*)

Las **media queries** permiten aplicar diferentes estilos CSS según las características del dispositivo, como el ancho de la pantalla, la resolución o la orientación. Gracias a ellas, se pueden definir **puntos de interrupción (breakpoints)**, que indican en qué momento la interfaz debe cambiar su estructura para ofrecer una experiencia óptima.

Ejemplo de **media query** en CSS:

```
css
@media (max-width: 768px) {
  body {
    background-color: lightgray;
  }
}
```

En este caso, el fondo del sitio cambiará a gris en pantallas menores a **768 píxeles**, optimizando la apariencia para dispositivos móviles.

### Definición de puntos de interrupción recomendados:

- **320px - 480px** → Smartphones pequeños.
- **481px - 768px** → Tablets en modo vertical.
- **769px - 1024px** → Tablets en modo horizontal o portátiles pequeños.
- **1025px - 1200px** → Escritorios y monitores medianos.
- **1201px en adelante** → Monitores grandes y pantallas extendidas.

La clave en el uso de media queries es **no abusar de ellas**, sino establecer puntos de interrupción en base a la estructura del contenido en lugar de dispositivos específicos.

## 1.5 Contenido Primero (*Content First*)

Un error común en el diseño web es adaptar primero la estética y luego el contenido. En el enfoque **"Content First"**, el contenido es la prioridad y el diseño se ajusta en función de su importancia y relevancia.

### Beneficios de este enfoque:

- **Mejor experiencia de usuario:** Se prioriza la información clave, evitando que quede oculta o mal estructurada en pantallas pequeñas.
- **Mayor eficiencia en carga y rendimiento:** Se optimiza la jerarquía del contenido, reduciendo elementos innecesarios.
- **Mayor accesibilidad:** El contenido es el foco principal, asegurando que sea comprensible en cualquier contexto.

Un diseño centrado en el contenido se enfoca en **legibilidad, accesibilidad y navegación eficiente**, lo que es especialmente importante en dispositivos móviles donde el espacio es limitado.

## 1.6 Optimización de Desempeño y Velocidad

El rendimiento es un factor determinante en la experiencia del usuario. Un sitio web puede ser visualmente atractivo, pero si tarda demasiado en cargar, la tasa de abandono aumentará significativamente. Algunas estrategias clave incluyen:

- **Minificación de archivos CSS y JavaScript:** Reducir el tamaño de los archivos para acelerar el tiempo de carga.
- **Carga diferida (*Lazy Loading*):** Permite que las imágenes y otros recursos se carguen solo cuando son necesarios.
- **Uso de CDN (*Content Delivery Network*):** Distribuye los archivos estáticos en servidores cercanos a los usuarios para mejorar la velocidad de acceso.
- **Eliminación de recursos innecesarios:** Evitar el uso de scripts y estilos que no sean esenciales para la funcionalidad del sitio.

El diseño responsivo no solo implica adaptar visualmente la interfaz, sino también optimizar su desempeño para garantizar una experiencia fluida en cualquier dispositivo.

## Conclusión

El diseño responsivo es un estándar esencial en el desarrollo web moderno, garantizando que las interfaces sean funcionales y accesibles en cualquier pantalla. Aplicar principios como **rejillas flexibles, imágenes adaptativas, media queries, contenido primero y optimización del rendimiento** no solo mejora la usabilidad, sino que también contribuye a una mejor indexación en buscadores y a una mayor retención de usuarios.

Al implementar estos principios, los diseñadores pueden crear experiencias digitales **eficientes, accesibles y escalables**, adaptadas a las necesidades de una audiencia cada vez más diversa.

## 2. Metodologías Ágiles en el Diseño

En el desarrollo de interfaces y experiencias digitales, las metodologías ágiles han revolucionado la manera en que los equipos de diseño trabajan. Tradicionalmente, los proyectos de diseño seguían un enfoque secuencial o en cascada (*Waterfall*), donde cada fase (análisis, diseño, desarrollo y prueba) se completaba antes de pasar a la siguiente. Sin embargo, este método no siempre es eficiente en entornos donde los requisitos pueden cambiar rápidamente.

Las **metodologías ágiles**, originalmente desarrolladas para la ingeniería de software, han demostrado ser altamente efectivas en el diseño de interfaces, ya que permiten una **iteración rápida, integración continua del feedback y adaptación a nuevas necesidades**. A continuación, se detallan los principios clave y las metodologías más utilizadas en el diseño ágil.

### 2.1 Iteración Rápida y Feedback Continuo

Uno de los principios fundamentales del diseño ágil es la **iteración rápida**, donde los productos no se desarrollan en su versión final de una sola vez, sino a través de ciclos cortos de trabajo llamados **iteraciones** o **sprints**.

Cada iteración permite:

- **Realizar pruebas tempranas** con usuarios para identificar mejoras necesarias.
- **Reducir errores** al hacer pequeños cambios de manera frecuente.
- **Asegurar que el producto evolucione** según las necesidades del usuario.

El **feedback continuo** es otro pilar del diseño ágil. En lugar de esperar a que el diseño esté completamente finalizado, se recopilan opiniones de usuarios, clientes y desarrolladores en cada fase del proceso. Herramientas como **Figma, InVision y Adobe XD** facilitan la colaboración y permiten compartir prototipos interactivos con los equipos para recibir retroalimentación en tiempo real.

### 2.2 Colaboración entre Diseñadores y Desarrolladores

El diseño ágil fomenta la **colaboración constante** entre diseñadores, desarrolladores y otros miembros del equipo. En proyectos tradicionales, los diseñadores crean una interfaz y luego la entregan a los desarrolladores para su implementación. Esto puede generar problemas cuando los diseñadores no consideran limitaciones técnicas o los desarrolladores no interpretan correctamente la intención del diseño.

Para evitar esto, las metodologías ágiles promueven:

- **Trabajo en conjunto desde el inicio del proyecto**, permitiendo que los desarrolladores contribuyan con soluciones técnicas viables.
- **Uso de sistemas de diseño** como **Material UI**, **Tailwind CSS** o **Bootstrap**, que garantizan coherencia visual y facilitan la implementación del diseño.
- **Reuniones diarias de sincronización** (*daily stand-ups*) para discutir avances, bloqueos y próximos pasos.

Esta integración temprana y constante mejora la eficiencia y reduce la necesidad de realizar grandes correcciones en etapas avanzadas del proyecto.

## 2.3 Flexibilidad y Priorización de Funcionalidades

Uno de los mayores beneficios del diseño ágil es la capacidad de **adaptarse a cambios** en los requisitos del proyecto. En lugar de planificar toda la interfaz de antemano, los equipos trabajan en **módulos funcionales** que pueden ajustarse según las necesidades del usuario.

### Priorización de funcionalidades

En un entorno ágil, no todas las características tienen la misma importancia. Se utiliza el concepto de **MVP (Minimum Viable Product)**, que se enfoca en desarrollar las funcionalidades esenciales primero y luego expandirlas en iteraciones futuras.

Para priorizar de manera efectiva, se pueden usar técnicas como:

- **Matriz de Eisenhower**: Clasifica tareas según urgencia e importancia.
- **MoSCoW** (*Must-have, Should-have, Could-have, Won't-have*) para definir qué elementos son imprescindibles y cuáles pueden esperar.
- **Roadmaps dinámicos**: Listas de funcionalidades que pueden cambiar según la retroalimentación del usuario.

Este enfoque evita desperdiciar tiempo en desarrollar características innecesarias y asegura que el producto evolucione de manera eficiente.

## 2.4 Sprints de Diseño en Scrum

Scrum es una de las metodologías ágiles más populares y está basada en **ciclos de trabajo cortos llamados sprints**, que suelen durar entre una y cuatro semanas. Durante un sprint de diseño, el equipo se enfoca en desarrollar una parte específica de la interfaz, probarla y mejorarla antes de avanzar al siguiente sprint.

### Estructura de un sprint de diseño en Scrum

1. **Planificación del Sprint**: Se definen objetivos, tareas y prioridades.
2. **Diseño y Prototipado**: Se crean bocetos, wireframes o prototipos interactivos.

3. **Pruebas y Feedback:** Se recopila retroalimentación de usuarios o del equipo.
4. **Iteración y Ajustes:** Se mejoran los diseños basados en el feedback.
5. **Revisión y Cierre:** Se evalúa el trabajo realizado y se planea el siguiente sprint.

Este método permite una evolución progresiva del diseño y garantiza que el producto final esté alineado con las expectativas del usuario.

## 2.5 Pruebas de Usabilidad en un Entorno Ágil

Las pruebas de usabilidad son esenciales en cualquier proyecto de diseño, pero en un entorno ágil se realizan de manera **continua y sistemática**. En lugar de esperar hasta que el diseño esté completamente terminado, se realizan pruebas desde las primeras iteraciones.

Algunas técnicas clave incluyen:

- **Pruebas A/B:** Comparar dos versiones de un diseño para determinar cuál funciona mejor.
- **Heatmaps:** Analizar las áreas donde los usuarios hacen más clics o interactúan más.
- **Pruebas de recorrido:** Se asignan tareas a los usuarios para evaluar la facilidad de uso del diseño.
- **Encuestas y entrevistas:** Obtener comentarios directos sobre la experiencia del usuario.

### Ventajas de hacer pruebas frecuentes en diseño ágil

- Detectar errores tempranos y corregirlos antes de que se vuelvan costosos.
- Mejorar la experiencia del usuario a lo largo del desarrollo.
- Validar hipótesis de diseño con datos reales.

Las pruebas de usabilidad garantizan que la interfaz no solo sea visualmente atractiva, sino también funcional y efectiva.

## Conclusión

Las metodologías ágiles han cambiado la forma en que se diseñan interfaces, permitiendo iteraciones rápidas, colaboración efectiva y flexibilidad para adaptarse a los cambios. A través de **sprints de diseño, feedback continuo y priorización de funcionalidades**, los equipos pueden desarrollar productos más alineados con las necesidades de los usuarios.

El diseño ágil no solo mejora la eficiencia del proceso, sino que también permite una mejor integración entre diseñadores y desarrolladores, asegurando que las interfaces sean funcionales y adaptables desde el primer momento. Aplicar estas metodologías en el diseño responsivo es clave para crear experiencias digitales dinámicas y optimizadas.

## 3. Implementación del Diseño Responsivo y Ágil



La implementación del diseño responsivo y ágil requiere un enfoque estructurado que combine herramientas, frameworks y metodologías para garantizar que las interfaces sean funcionales, accesibles y optimizadas para todos los dispositivos. A continuación, se presentan las mejores prácticas y técnicas utilizadas en el desarrollo de interfaces que cumplen con estos principios.

## 3.1 Evaluación de Dispositivos y Contextos de Uso

Antes de comenzar con el diseño, es fundamental comprender la diversidad de dispositivos y contextos de uso en los que se visualizará la interfaz. Un diseño bien implementado debe considerar:

- **Diferentes tamaños de pantalla:** Desde teléfonos móviles y tablets hasta monitores de escritorio y pantallas ultraanchas.
- **Condiciones de uso:** Uso en interiores y exteriores, accesibilidad con poca iluminación, conexiones de internet limitadas.
- **Interacción del usuario:** Navegación táctil, ratón, teclado y asistentes de voz.
- **Requisitos de accesibilidad:** Compatibilidad con tecnologías asistivas como lectores de pantalla y navegación por teclado.

Para realizar esta evaluación, se pueden utilizar herramientas como **Google Analytics**, que permite conocer los dispositivos más utilizados por los usuarios, y **BrowserStack**, que facilita la prueba en múltiples plataformas.

## 3.2 Herramientas y Frameworks para Diseño Responsivo

El uso de herramientas y frameworks ayuda a acelerar el desarrollo de interfaces responsivas y a garantizar consistencia en el diseño. Algunas de las opciones más utilizadas incluyen:

### Frameworks de CSS

- **Bootstrap:** Framework popular basado en un sistema de rejillas flexible ( *grid system*) y componentes reutilizables.
- **Tailwind CSS:** Framework de utilidad que permite un diseño altamente personalizable sin necesidad de escribir CSS extenso.
- **Material UI:** Basado en los principios de diseño de Google, ideal para interfaces modernas y escalables.

## Sistemas de Diseño

Los sistemas de diseño ayudan a garantizar consistencia visual en la interfaz. Algunos de los más usados incluyen:

- **Google Material Design:** Proporciona directrices y componentes para el diseño de interfaces intuitivas.
- **IBM Carbon Design System:** Útil para aplicaciones empresariales con interfaces avanzadas.
- **Fluent UI (Microsoft):** Ideal para aplicaciones que se integran con el ecosistema de Microsoft.

Estos frameworks y sistemas facilitan la creación de interfaces que sean coherentes en diferentes dispositivos y plataformas.

### 3.3 Prototipado Rápido y Pruebas en Múltiples Dispositivos

El **prototipado rápido** es una técnica clave en el diseño ágil, ya que permite visualizar y probar la interfaz antes de su desarrollo completo. Esto ayuda a detectar problemas y realizar ajustes de manera temprana, evitando retrabajos costosos.

#### Herramientas para prototipado

- **Figma**: Permite colaboración en tiempo real, ideal para equipos de diseño distribuidos.
- **Adobe XD**: Facilita la creación de prototipos interactivos con transiciones avanzadas.
- **InVision**: Herramienta para crear prototipos clicables que pueden ser probados por usuarios y clientes.

Además de la creación de prototipos, es crucial realizar **pruebas en múltiples dispositivos** para asegurar que el diseño responsivo funcione correctamente. Herramientas como **Responsive Design Mode en Chrome DevTools** o **LambdaTest** permiten simular el diseño en diferentes resoluciones.

### 3.4 Pruebas de Usabilidad y Optimización Continua

Una vez implementado el diseño responsivo, es fundamental realizar pruebas de usabilidad para garantizar que la interfaz sea intuitiva y fácil de usar en todas las plataformas. Existen varias metodologías para evaluar la experiencia del usuario:

#### Métodos de prueba de usabilidad

- **Pruebas A/B**: Comparar dos versiones del diseño para determinar cuál funciona mejor.
- **Mapas de calor (Heatmaps)**: Analizar las áreas donde los usuarios hacen más clics.
- **Pruebas de recorrido**: Pedir a los usuarios que completen tareas específicas para identificar dificultades en la navegación.

#### Herramientas para pruebas de usabilidad

- **Hotjar**: Proporciona mapas de calor y grabaciones de sesiones de usuario.
- **Crazy Egg**: Permite visualizar cómo los usuarios interactúan con la interfaz.
- **UserTesting**: Plataforma para obtener retroalimentación directa de usuarios reales.

Las pruebas de usabilidad deben realizarse de manera continua, especialmente después de cada sprint de diseño, para asegurarse de que el producto evoluciona según las necesidades de los usuarios.

### 3.5 Integración del Diseño con el Desarrollo Frontend

Para que un diseño responsivo y ágil se implemente correctamente, es crucial una estrecha colaboración entre diseñadores y desarrolladores. Algunas prácticas recomendadas incluyen:

#### Uso de herramientas de colaboración

- **Zeplin**: Permite a los diseñadores entregar especificaciones claras a los desarrolladores.
- **Storybook**: Framework que ayuda a documentar y probar componentes de UI en entornos de desarrollo frontend.

- **Design Tokens:** Uso de variables predefinidas en código para mantener la coherencia en colores, tipografías y espaciados.

## Buenas prácticas en desarrollo frontend

- **Uso de Flexbox y CSS Grid:** Permiten estructurar el diseño de manera flexible sin necesidad de usar medidas fijas.
- **Carga diferida (*Lazy Loading*):** Optimiza el rendimiento al cargar imágenes y recursos solo cuando son necesarios.
- **Minificación de archivos:** Reducir el peso de los archivos CSS y JavaScript mejora la velocidad de carga.

El trabajo conjunto entre diseñadores y desarrolladores es fundamental para que el diseño responsivo no solo sea visualmente atractivo, sino también eficiente en términos de rendimiento y accesibilidad.

## Conclusión

La implementación del diseño responsivo y ágil requiere una combinación de **evaluación de dispositivos, herramientas especializadas, prototipado rápido, pruebas de usabilidad y colaboración con desarrollo frontend**. Utilizar frameworks como **Bootstrap y Tailwind CSS**, junto con metodologías ágiles como **Scrum**, facilita la creación de interfaces escalables y flexibles.

Realizar **pruebas continuas** y optimizar el rendimiento garantiza que la interfaz funcione correctamente en todos los dispositivos, proporcionando una experiencia de usuario óptima. Integrar el diseño con el desarrollo desde las primeras etapas del proyecto permite que el producto final sea funcional, accesible y alineado con las necesidades del usuario.

En un mundo digital en constante evolución, el diseño responsivo y ágil es una **estrategia esencial** para el éxito de cualquier producto digital.

## Actividades prácticas

### Caso Práctico 15

Estás desarrollando un sitio web de noticias que debe ser accesible tanto en dispositivos móviles como en computadoras de escritorio. Describe cómo aplicarías principios de diseño responsivo para asegurar una experiencia consistente y agradable en todas las plataformas. ¿Cuáles son las prácticas clave del diseño responsivo que implementarías en este sitio de noticias?

1. ¿Cuáles son las prácticas clave del diseño responsivo que implementarías en este sitio de noticias?

Procesando respuesta, no cierres el navegador, este proceso podría tardar unos segundos

### Caso Práctico 16

Tu equipo está desarrollando una aplicación web para gestionar proyectos utilizando metodologías ágiles. Describe cómo integrarías principios de diseño ágil en el proceso de desarrollo y diseño de esta aplicación. ¿Qué metodologías ágiles aplicarías para el diseño y desarrollo de tu aplicación de gestión de proyectos?

1. ¿Qué metodologías ágiles aplicarías para el diseño y desarrollo de tu aplicación de gestión de proyectos?

Procesando respuesta, no cierres el navegador, este proceso podría tardar unos segundos