



Contenido

[**1. OBJETIVOS GENERALES 2**](#_heading=h.612tutavcebb)

[**2. OBJETIVOS ESPECÍFICOS 3**](#_heading=h.tldygc1hnpcu)

[**3. ROLES 3**](#_heading=h.32f5hu9x2xzm)

[**4. Recolección de los requerimientos 5**](#_heading=h.6yo7hondacp2)

[**ESTABLECER ETAPAS DE PLANEACIÓN DE PROYECTO DE TI 7**](#_heading=h.g1jyv93h6y7a)

[**METODOLOGÍAS ÁGILES (FASES O ETAPAS) 7**](#_heading=h.b1o5crxjg9s1)

[**RAD - Rapid Application Development 7**](#_heading=h.iydawvi2khpr)

[**Definición 7**](#_heading=h.uhifhrhrvbhu)

[**Objetivo 7**](#_heading=h.t0mmd0rv9s6w)

[**CARACTERÍSTICA 7**](#_heading=h.k58eq5qxbk8n)

[**FASES 7**](#_heading=h.y0u70n3h7ygb)

[1. Recolección de requisitos. 7](#_heading=h.v3exp0subee4)

[2. Diseño 7](#_heading=h.80nl19lck2pc)

[3. Feedback del usuario 8](#_heading=h.fxsagcbtj6eg)

[4. Prueba del Producto 8](#_heading=h.wzzsacsq8rmf)

[**VENTAJAS 8**](#_heading=h.26ksbmbwzyk)

[1. Velocidad de desarrollo: 8](#_heading=h.lrf2u5lzghbz)

[2. Alta flexibilidad y adaptabilidad: 8](#_heading=h.ncje4k4egqp1)

[3. Retroalimentación continua del usuario: 8](#_heading=h.2184l1j5pmn6)

[4. Mejor alineación entre el producto y las necesidades del cliente: 9](#_heading=h.t3ij37wrm3ph)

[5. Reducción de errores y retrabajo: 9](#_heading=h.2hi0j4bayglw)

[6. Ahorro de costos en ciertos contextos: 9](#_heading=h.1q991o7m410n)

[7. Incremento en la motivación del equipo: 9](#_heading=h.zhnpzv1g5flb)

[**DESVENTAJAS 9**](#_heading=h.ypu9jd4twy6l)

[1. No apto para proyectos grandes o complejos: 9](#_heading=h.5ofkj0l42qkl)

[2. Menor énfasis en la documentación: 9](#_heading=h.4t971fpivzqd)

[3. Riesgo de comprometer la calidad: 9](#_heading=h.s5m4a656i5vb)

[**EJEMPLO DE USO EN TI 9**](#_heading=h.8inb9nmum3wo)

[**CONCLUSIÓN 10**](#_heading=h.r5yhpuve886d)

[**6. HERRAMIENTAS PARA LA GESTIÓN DE PROYECTO 10**](#_heading=h.h85etnu42685)

[¿Qué es Trello? 10](#_heading=h.gx8or9cwzmyz)

[¿Cómo funciona Trello? 10](#_heading=h.uw31a83v6ngm)

[¿Trello es gratuito? 10](#_heading=h.lax6k7vg6rgm)

[¿Para qué puedes usarlo? 11](#_heading=h.ahfyeje3m2ex)

# OBJETIVOS GENERALES

Se busca desarrollar una aplicación web innovadora cuyo propósito principal sea facilitar a los clientes la solicitud de mandados de forma rápida, eficiente y confiable, a través del servicio de **Mandadito Express**. Esta plataforma busca optimizar al máximo el tiempo de los clientes, permitiéndoles delegar tareas como compras, u otros recados cotidianos, para que puedan enfocarse en lo que realmente importa en su día a día. **Mandadito Express** será el encargado de llevar a cabo estas actividades, asegurándose de ofrecer una atención completamente personalizada. Esto no solo permitirá cubrir las necesidades específicas de cada cliente, sino que también fomentará una relación de confianza, basada en la responsabilidad, la puntualidad y el compromiso con un servicio de calidad. Además, se garantizará que los productos o encargos solicitados lleguen en perfecto estado y de manera segura hasta que lleguen a manos del cliente.

# **OBJETIVOS ESPECÍFICOS**

La aplicación será capaz de:

* Registrar a los clientes que desean ocupar la aplicación web.
* Permitir que el cliente inicie y cierre sesión de forma segura.
* Mostrar el historial de los pedidos o mandados que ha hecho el cliente.
* Calificar el servicio y la satisfacción del cliente dependiendo al servicio recibido.
* El cliente podrá ver el estado en tiempo real de su pedido.
* Atender a más de un pedido a la vez.
* Cambiar el estado del mandado a pendiente, en camino y entregado según el administrador de la aplicación web.
* Seleccionar el método de pago.

| ROLES | | FUNCIONES |
| --- | --- | --- |
| Tester | Gloria y Lucio. | Identificar errores, defectos y problemas en el código, validando que el software cumpla con los requisitos y funcione de manera óptima |
| Analista | David. | Se enfoca en entender las necesidades del negocio y los requerimientos de los usuarios para luego definir la arquitectura y los detalles técnicos del software |
| Programadores | Gloria y Lucio. | Se refiere a las tareas y responsabilidades específicas de las personas involucradas en la creación y el mantenimiento del software. |
| Diseñadores | David, Gloria y Lucio. | Son responsables de crear la estructura, apariencia y experiencia del usuario de un software, web o aplicación |

# Recolección de los requerimientos

FUNCIONALES.

1. ¿Te gustaría poder registrarte y guardar tus datos personales para facilitar futuras o solicitudes de mandados?

* Si.
* No.

2. ¿Consideras importante poder iniciar sesión con tu cuenta para acceder a tus servicios?

* Si.
* No.

**3. ¿Qué información te gustaría visualizar en tu historial de pedidos?** *(Puedes marcar más de una opción)*

* *Fecha del mandado.*
* *Estado del mandado.*
* *productos encargados.*
* *Calificación del servicio.*
* *Notas*

*4. ¿Te gustaría poder calificar el servicio una vez finalizado el mandado?*

* *Si.*
* *No.*

*5. ¿Te parece útil poder ver el estado del mandado (pendiente, en camino, entregado)?*

* *Si.*
* *No.*

*6. ¿Sería valioso para ti poder solicitar varios mandados al mismo tiempo?*

* *Si.*
* *No.*

*7. ¿Debería existir una opción para cerrar sesión cuando no se use la app?*

* *Si.*
* *No.*

*8. ¿Estás de acuerdo con que el administrador de la app actualice el estado del mandado en cada fase?*

* *Si.*
* *No.*

NO FUNCIONALES

*1. ¿Qué tan importante es para ti que la aplicación cargue rápido (menos de 3 segundos)?*

* *Muy importante.*
* *Mas o menos.*
* *No importa.*

*2. ¿Desde qué dispositivo prefieres usar este tipo de servicio?*

* *Navegador web celular.*
* *Aplicación móvil.*
* *No me importa.*

*3. ¿Qué tan importante es que la aplicación tenga una interfaz sencilla e intuitiva?*

* *Muy importante.*
* *Más o menos.*
* *No importa.*

*4. ¿Confías en una aplicación web para guardar tus datos personales de forma segura?*

* *Si.*
* *No.*

*5. ¿Te molestaría recibir actualizaciones frecuentes de la plataforma?*

* *Si.*
* *No.*

*6. ¿Esperas que haya soporte técnico o atención al cliente si algo sale mal?*

* *Si.*
* *No.*

*7. ¿Usas más de un navegador? ¿Esperas que la app funcione correctamente en todos (Chrome, Safari, Firefox, etc.)?*

* *Si.*
* *No.*

1. Metodología

# ESTABLECER ETAPAS DE PLANEACIÓN DE PROYECTO DE TI

# METODOLOGÍAS ÁGILES (FASES O ETAPAS)

# RAD - Rapid Application Development

## Definición

RAD (Desarrollo Rápido de Aplicaciones) es una metodología ágil enfocada en desarrollar software de forma rápida y con alta calidad, utilizando prototipos funcionales, retroalimentación continua del cliente y equipos de trabajo pequeños y eficientes.

## Objetivo

Reducir el tiempo de desarrollo al mínimo sin sacrificar la calidad del producto, mediante el uso de herramientas visuales, desarrollo modular y la constante colaboración con el cliente.

## CARACTERÍSTICA

Un tipo de metodología ágil, ofrece resultados en tiempo real y funciona bien cuando debe entregar rápidamente un producto y actualizar las características según sea necesario. Aunque se hace hincapié en la velocidad, no se basa en un período de tiempo específico.

## FASES

# Recolección de requisitos.

* En esta primera fase, lo que habrá que hacer será sentar las bases de las necesidades del proyecto, tanto de necesidades de la aplicación como de alcance del proyecto y de esta manera comenzar a trabajar en la creación de prototipos.

## Diseño

* La primera etapa es totalmente necesaria para comenzar con el diseño de los prototipos, ya que nos permite medir el alcance y las condiciones necesarias para su debida puesta en marcha. Con ello, los profesionales tendrán que trabajar junto a los clientes para llevar un buen desarrollo de los prototipos, buscando siempre ofrecer la mejor solución final posible.

## Feedback del usuario

* En cuanto se haya terminado de definir el alcance del proyecto se puede comenzar la fase de desarrollo. Los diseñadores y desarrolladores trabajarán estrechamente con los clientes con el objetivo de crear y mejorar los prototipos que ya están en marcha hasta que el producto final esté preparado.

## Prueba del Producto

* Antes de dar el proyecto por finalizado, es necesaria una fase exigente de pruebas, para comprobar que todas las exigencias y requisitos planteados en el primer punto se cumplen y la calidad del producto es óptima. No debemos dejar de tener en cuenta las opiniones del cliente, ya que estas nos seguirán sirviendo para optimizar el funcionamiento y ajustarlo a sus necesidades.

1. **Presentación de la aplicación y lanzamiento**

* Tras finalizar las comprobaciones y asegurarnos de que el producto cumple con los requisitos, es el momento de presentarlo, convirtiendo los datos y formato al usuario para que tenga todas las capacidades necesarias a la hora de utilizarla, y, posteriormente, lanzarla definitivamente.

## VENTAJAS

### Velocidad de desarrollo:

* Permite desarrollar software mucho más rápido que los modelos tradicionales.

### Alta flexibilidad y adaptabilidad:

* Se aceptan cambios en los requisitos en cualquier etapa del desarrollo sin afectar drásticamente el proyecto. Esto lo hace ideal para proyectos donde los requerimientos no están completamente definidos desde el inicio.

### Retroalimentación continua del usuario:

* Los usuarios participan activamente en el diseño, desarrollo y prueba.Esto garantiza que el producto final cumpla exactamente con las expectativas del cliente.

### Mejor alineación entre el producto y las necesidades del cliente:

* Al trabajar con prototipos que evolucionan, los usuarios pueden visualizar el sistema real y dar retroalimentación desde etapas tempranas

### Reducción de errores y retrabajo:

* Al detectar errores durante las iteraciones, se corrigen de inmediato.Esto evita la acumulación de errores graves al final del desarrollo.

### Ahorro de costos en ciertos contextos:

* Al reducir el tiempo de desarrollo y evitar cambios costosos al final, puede ser una opción más económica para proyectos pequeños y medianos.

### Incremento en la motivación del equipo:

* Ver avances funcionales constantes y recibir feedback positivo puede elevar el compromiso y productividad del equipo.

## DESVENTAJAS

### No apto para proyectos grandes o complejos:

* Sistemas que requieren mucha integración, arquitecturas muy complejas o dependencias múltiples no son adecuados para RAD.

### Menor énfasis en la documentación:

* Se prioriza el desarrollo rápido sobre la documentación técnica.Esto puede afectar el mantenimiento y escalabilidad del sistema a largo plazo.

### Riesgo de comprometer la calidad:

* La rapidez puede llevar a tomar decisiones técnicas apresuradas o usar código poco estructurado.

## EJEMPLO DE USO EN TI

Desarrollar una aplicación web para ventas, donde el cliente va probando prototipos semanales y sugiere cambios que se implementan de forma inmediata para la siguiente versión

## CONCLUSIÓN

**RAD** es una metodología poderosa cuando se busca rapidez, colaboración constante y adaptabilidad. Sin embargo, necesita un entorno favorable, como equipos comprometidos, usuarios activos y un alcance de proyecto bien acotado.

# 

# 6. HERRAMIENTAS PARA LA GESTIÓN DE PROYECTO

## ¿Qué es Trello?

**Trello** es una herramienta de gestión de proyectos basada en el sistema **Kanban**, que te permite **organizar tareas de forma visual** mediante tableros, listas y tarjetas. Es muy usada para proyectos personales, académicos o de equipo, y se adapta fácilmente a metodologías ágiles como **Scrum**, **Kanban** o incluso **RAD** (Desarrollo Rápido de Aplicaciones).

## ¿Cómo funciona Trello?

* Un **Tablero** representa un proyecto (por ejemplo: “Desarrollo de App”).
* Las **Listas** son etapas o columnas (como: “Por hacer”, “En progreso”, “Terminado”).
* Las **Tarjetas** son tareas específicas (como: “Diseñar prototipo”, “Programar login”, “Subir a GitHub”).

**Puedes añadir**:

* Fechas límite ⏰
* Etiquetas de colores 🔖
* Adjuntos 📎
* Comentarios 💬
* Checklists ✅

## ¿Trello es gratuito?

**Sí, Trello tiene una versión gratuita muy completa.** Incluye:

* Tableros ilimitados por usuario.
* Hasta **10 tableros por espacio de trabajo**.
* Tarjetas, listas, etiquetas, fechas, comentarios, checklists.
* **Automatizaciones básicas** (con Butler).
* Colaboración en tiempo real con hasta 10 personas por tablero.

## ¿Para qué puedes usarlo?

* Proyectos escolares o universitarios.
* Gestión de tareas personales o del hogar.
* Desarrollo de software ágil.
* Organización de emprendimientos.
* Equipos de trabajo remoto.

metodologia rad.

**1.- Planificación y recolección de requisitos.**

* Reunirse con el cliente para saber qué esperan de la app.
* Hacer una lista simple de funciones.

1.-Registro de usuarios.

2.-Iniciar y cerrar sesiones.

3.- Hacer pedidos.

4.-Ver estado de pedido.

5.-Calificar servicio.

6.-El administrador puede cambiar el pedido.

**2.- Diseño del usuario (prototipo).**

* Dibujar las pantallas principales.

1.-Pantalla de inicio de sesión.

2.-Pantalla de pedidos.

3.-Pantalla para ver estado de pedidos.

4.-Pantalla para calificar.

5.-Pantalla del administrador.

* Mostrar prototipos a los usuarios.
* Ajustar el diseño según lo que digan los usuarios.

**3.-Fase de construcción rápida.**

1.-Programar el registro y login del usuario.

2.-Permitir crear un pedido y guardarlo.

3.-Mostrar el historial de pedidos.

4.-Ver el estado del pedido en tiempo real (por ejemplo: pendiente, en camino, entregado).

5.-Añadir opción para calificar el servicio.

6.-Hacer que el administrador pueda ver todos los pedidos y cambiar el estado.

7.- Se hacen pruebas con usuarios cada semana.

**4.-Fase de entrega y lanzamiento.**

* Probar toda la aplicación completa como si fuera el usuario.
* Corregir detalles finales.
* Explicar a los usuarios como usarla.
* Escuchar sugerencias para futuras mejoras.

### Resultado Final:

Una aplicación web sencilla que permite:

* Registrarse e iniciar sesión.
* Hacer y ver pedidos.
* Ver el estado del pedido.
* el administrador puede manejar todos los pedidos.

## 7. HERRAMIENTA DE TRELLO

#### 

## COMO SE USA LA HERRAMIENTA

## GITHUB

¿QUÉ ES GITHUB?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

* Presentar o compartir el trabajo.
* Seguir y administrar los cambios en el código a lo largo del tiempo.
* Dejar que otros usuarios revisen el código y realicen sugerencias para mejorarlo.
* Colaborar en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub.

**¿QUÉ ES GIT?**

Git es un sistema de control de versiones que realiza un seguimiento de los cambios en los archivos. Git es especialmente útil cuando un grupo de personas y tú estáis haciendo cambios en los mismos archivos al mismo tiempo.

Normalmente, para hacerlo en un flujo de trabajo basado en Git, harías lo siguiente:

* **Crear una rama** a partir de la copia principal de archivos en los que tú (y tus colaboradores) estáis trabajando.
* **Realizar modificaciones** en los archivos de forma independiente y segura en tu propia rama personal.
* Dejar que Git **fusione mediante combinación** y de forma inteligente los cambios específicos en la copia principal de archivos, de modo que los cambios no afecten a las actualizaciones de otras personas.
* Dejar que Git **realice un seguimiento** de tus cambios y los de otras personas, por lo que todos siguen trabajando en la versión más actualizada del proyecto.

**¿Cómo funcionan Git y GitHub de manera conjunta?**

Al cargar archivos en GitHub, lo almacenarán en un "repositorio de Git". Esto significa que al realizar cambios (o "confirmaciones") en los archivos de GitHub, Git se iniciará automáticamente para realizar el seguimiento de los cambios y administrarlos.

Hay muchas acciones relacionadas con Git que puedes completar en GitHub directamente en el navegador, como crear un repositorio de Git, crear ramas y cargar y editar archivos.

Pero la mayoría de los usuarios trabajan en sus archivos localmente (en su propio ordenador), luego sincroniza continuamente estos cambios locales y todos los datos de Git relacionados, con el repositorio central "remoto" en GitHub. Hay muchas herramientas que puedes usar para hacerlo, como GitHub Desktop.

Cuando empieces a colaborar con otros y todos necesitas trabajar en el mismo repositorio al mismo tiempo, haréis lo siguiente continuamente:

* **Extraer** todos los cambios más recientes realizados por los colaboradores del repositorio remoto en GitHub.
* **Insertar** tus propios cambios en el mismo repositorio remoto en GitHub.

Git determina cómo combinar inteligentemente este flujo de cambios y GitHub te ayuda a administrarlo mediante características como las "solicitudes de cambios".

**¿Dónde empezar?**

Si no tienes experiencia con GitHub y Git, te recomendamos consultar los artículos de la categoría Póngase en marcha. Los artículos se centran en las tareas que puedes completar directamente en el explorador en GitHub y te ayudarán a lo siguiente:

* Crear una cuenta en GitHub.
* Aprender el "flujo de GitHub" y los principios clave del trabajo colaborativo (ramas, confirmaciones, solicitudes de cambios, fusiones mediante combinación).
* Personaliza tu perfil para compartir tus intereses y aptitudes.
* Explorar GitHub para encontrar inspiración para tus propios proyectos y conectar con otros usuarios.
* Obtén información sobre cómo descargar código interesante para tu propio uso.
* Obtener información sobre cómo cargar algo en lo que trabajas en un repositorio de GitHub.

