

# Ingeniería de Requisitos

**Sesión 10**

**Requisitos de Desarrollo**

***Prof. Ciro Rodriguez***  
*crodriguezro@unmsm.edu.pe*

# Tema: Requisitos de Dominio

- Identificación y Análisis de Requisitos de Desarrollo
- Especificación de Requisitos de Desarrollo
- Verificación de Requisitos de Desarrollo
- Gestión de Cambios en los Requisitos de Desarrollo

# Requisitos del Desarrollo

Se refieren a las **necesidades** y **expectativas** relacionadas con el **proceso de desarrollo de software**, asegurando que las **técnicas, herramientas y prácticas** utilizadas sean efectivas y alineadas con los **objetivos del proyecto**. Estos requisitos garantizan que el **entorno de desarrollo**, las **metodologías** y los **estándares** se ajusten para proporcionar un **software de alta calidad**.



# Identificación y Análisis de Requisitos de Desarrollo

## 1. Identificación de requisitos de desarrollo

- **Fuentes de Información:** Incluye la revisión de las políticas y normativas internas del departamento de TI, entrevistas con equipos de desarrollo, retrospectivas de proyectos anteriores, y consulta de estándares de la industria.
- **Entrevistas y Talleres:** Realizar entrevistas y talleres con desarrolladores, testers, arquitectos de software y otros stakeholders para entender las necesidades específicas.
- **Análisis de Proyectos Anteriores:** Evaluar proyectos anteriores para identificar qué técnicas y herramientas fueron exitosas y cuáles presentaron problemas.
- **Análisis del Producto:** Considerar las características y la complejidad del producto que se va a desarrollar para determinar los requisitos específicos.

# Identificación y Análisis de Requisitos de Desarrollo

## 2. Análisis de requisitos de desarrollo

- **Evaluación de Herramientas y Tecnologías:** Identificar y analizar las herramientas de desarrollo, plataformas, lenguajes de programación, y marcos de trabajo (frameworks) que mejor se adaptan a las necesidades del proyecto.
- **Evaluación de Metodologías:** Evaluar metodologías y prácticas de desarrollo (ágil, SCRUM, Kanban, etc.) que sean adecuadas para el equipo y el tipo de proyecto.
- **Evaluación de Competencias:** Analizar las habilidades y competencias del equipo de desarrollo para asegurar que coincidan con los requisitos del proyecto.
- **Análisis de Riesgos:** Identificar y analizar riesgos específicos relacionados con el desarrollo y determinar estrategias de mitigación.

# Especificación de Requisitos de Desarrollo

## 1.Documentación Clara y Precisa:

- **Plantillas Estándar:**

Utilizar plantillas estandarizadas para documentar los requisitos de desarrollo, garantizando uniformidad y claridad.

- **Descripciones Detalladas:**

Proporcionar descripciones detalladas de cada requisito, incluyendo el contexto, la justificación, y los criterios de aceptación.

Plantillas de Documentación de IEEE  
El estándar IEEE 830-1998,  
Engineering Board (IREB):  
IREB metodologías Project Management  
Institute (PMI):  
PMI plantillas y guías  
PMBOK Guide  
Plantillas de Requirements Engineering  
(REQUISITEPro):  
Herramientas como IBM Rational  
RequisitePro  
IBM Rational RequisitePro  
Herramientas de Colaboración PM  
Confluence (Atlassian):  
Google Docs, Microsoft Word

# Especificación de Requisitos de Desarrollo

## 2. Secciones Detalladas

- 1. Entorno de Desarrollo:** Especificar detalladamente el entorno de desarrollo, incluyendo sistemas operativos, versiones de software, herramientas de desarrollo, y configuraciones de hardware necesarias.
- 2. Metodologías de Desarrollo:** Definir las metodologías de desarrollo a utilizar, incluyendo prácticas específicas como revisiones de código, integración continua, y despliegue continuo.
- 3. Estándares de Codificación:** Describir los estándares y guías de codificación que deben seguirse para mantener la coherencia y calidad del código.
- 4. Prácticas de Seguridad:** Documentar las prácticas de seguridad que deben ser adoptadas durante el desarrollo para garantizar la protección del software y los datos.
- 5. Requisitos de Pruebas:** Especificar los tipos de pruebas (unitarias, de integración, funcionales, de rendimiento, etc.) que deben realizarse, así como las herramientas y entornos de prueba necesarios.

# Verificación de Requisitos de Desarrollo

## 1. Revisiones y Auditorías:

**Revisiones de Parejas:** Realizar revisiones de requisitos por parte de múltiples stakeholders para asegurar que los requisitos están completos y precisos.

**Auditorías Internas:** Llevar a cabo auditorías internas para verificar que los requisitos de desarrollo cumplen con las normativas y estándares internos y externos.

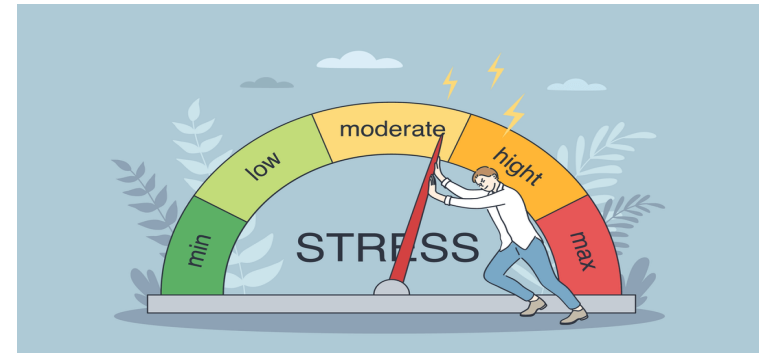
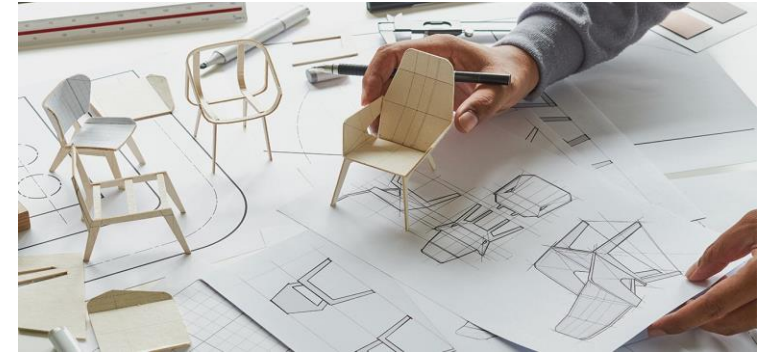




# Verificación de Requisitos de Desarrollo

## 2. Pruebas y Validaciones:

1. **Prototipos y Pilotos:** Crear y evaluar prototipos o pilotos para validar que las herramientas y metodologías elegidas son adecuadas y efectivas.
2. **Pruebas de Estrés y Rendimiento:** Implementar pruebas de estrés y rendimiento en el entorno de desarrollo para asegurar que cumple con los requisitos establecidos.
3. **Pruebas de Seguridad:** Realizar pruebas de seguridad para verificar que las prácticas de desarrollo garantizan la protección adecuada del software y los datos.



# Verificación de Requisitos de Desarrollo

## 3. Reuniones de Revisión: Reuniones de Revisión

**Periódica:** Establecer reuniones de revisión periódica para evaluar el progreso y hacer ajustes necesarios en los requisitos de desarrollo.



# Gestion de cambios en los Requisitos de Desarrollo

## 1. Proceso Formal de Gestión de Cambios:

- **Solicitud de Cambios:** Establecer un proceso estructurado para la solicitud de cambios en los requisitos de desarrollo, incluyendo la identificación, evaluación, y aprobación de cambios.
- **Registro de Cambios:** Utilizar herramientas de seguimiento para documentar y gestionar los cambios en los requisitos, incluyendo el motivo, el impacto esperado, y la decisión tomada.
- **Flujos de Trabajo:** Definir flujos de trabajo claros para la gestión de cambios que aseguren la evaluación, aprobación y comunicación oportuna de los cambios.

# Gestión de cambios en los Requisitos de Desarrollo

## 2. Evaluación de Impacto:

**Análisis de Impacto:** Realizar análisis detallados para evaluar el impacto de los cambios en el proyecto, incluyendo costos, tiempo, riesgos y recursos necesarios.

**Priorización y Planeación:** Reevaluar la prioridad de los requisitos y ajustar el plan del proyecto para incorporar los cambios de manera efectiva.

# Gestion de cambios en los Requisitos de Desarrollo

## 3. Comunicación y Aprobación:

### **Involucrar Stakeholders:**

Asegurar la participación de todos los stakeholders relevantes en la evaluación y aprobación de cambios para asegurar que todas las perspectivas son consideradas.

### **Documentación y Comunicación:**

Documentar todas las decisiones de cambio y comunicar claramente a todos los miembros del equipo y stakeholders los cambios aprobados.

# Gestion de cambios en los Requisitos de Desarrollo

## 3. Implementación y Verificación de Cambios:

### **Implementación Controlada**

Controlar y gestionar la implementación de cambios de manera estructurada, asegurando que se siguen los procedimientos establecidos y que la calidad no se ve comprometida.

### **Revalidación de Requisitos**

Realizar revalidaciones de los requisitos después de la implementación de cambios para asegurar que se siguen cumpliendo las expectativas.

# Caso Práctico :Sistema de Gestión de Salud:

## Identificación y Análisis

Durante la fase de planificación para un Sistema de Gestión de Salud en un hospital, se identifican los siguientes requisitos de desarrollo:

- El entorno de desarrollo debe ser compatible con sistemas operativos Windows y Linux.
- Se utilizará la metodología ágil SCRUM para gestionar el desarrollo del proyecto.
- Las herramientas de desarrollo serán Visual Studio Code y Git.
- El equipo de desarrollo requiere formación en prácticas de seguridad de software.
- Se adoptarán estándares de codificación basados en las guías de OWASP.

# Caso Práctico :Sistema de Gestión de Salud:

## Especificación

En la fase de especificación, se documentan los requisitos detallados:

- El entorno de desarrollo especificado incluye Windows Server 2019 y Ubuntu 20.04 LTS.
- SCRUM se implementará con sprints de dos semanas y reuniones diarias de seguimiento.
- Las herramientas de desarrollo serán Visual Studio Code v1.60 y Git v2.33.
- Los desarrolladores recibirán formación en seguridad basada en las guías de OWASP.
- Se seguirán las guías de estilo de codificación de C# y se utilizarán linters y formatters automáticos.



# Caso Práctico :Sistema de Gestión de Salud:

## Verificación:

Durante la verificación:

- Se realizan revisiones de pares para validar los requisitos de desarrollo documentados.
- Se crea un prototipo de la aplicación en el entorno de desarrollo especificado para validar la compatibilidad.
- Se ejecutan pruebas de rendimiento en el entorno de desarrollo para asegurar tiempos de respuesta aceptables.
- Se realizan pruebas de seguridad para validar la aplicación de las guías de OWASP.

# Caso Práctico :Sistema de Gestión de Salud:

## Gestión de Cambios:

Si se requiere un cambio, como la actualización de Visual Studio Code a una nueva versión, se sigue el procedimiento:

- Se solicita el cambio y se analiza el impacto.
- Se realiza una reunión con stakeholders para discutir y aprobar el cambio.
- Se documenta el cambio y se comunica a todo el equipo.
- Se implementa el cambio y se verifica mediante pruebas para asegurar que no afecta negativamente al proyecto.

## Caso Práctico

# Desarrollo de una Plataforma de Comercio Electrónico

Desarrollar una plataforma de comercio electrónico que permita a los usuarios comprar y vender productos en línea. La plataforma debe ser escalable, segura y fácil de usar, y debe integrarse con diversos servicios de pago y envío.

## 1. Identificación y Análisis de Requisitos de Desarrollo

### Contexto:

Durante las fases iniciales del proyecto, el equipo de desarrollo se reúne para identificar y analizar los requisitos necesarios para asegurar que la plataforma se desarrolle de manera efectiva y eficiente.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Requisitos de Desarrollo:

- El entorno de desarrollo debe soportar los sistemas operativos Windows y macOS.
- El lenguaje de programación principal será JavaScript con Node.js para el backend y React.js para el frontend.
- Utilización de Git y GitHub para el control de versiones y colaboración del equipo.
- Implementación de la metodología ágil SCRUM con sprints de dos semanas.
- El equipo de desarrollo debe recibir formación en seguridad de aplicaciones web.
- La plataforma debe ser compatible con bases de datos SQL y NoSQL.
- Integración continua (CI) y despliegue continuo (CD) utilizando Jenkins.
- Estándares de codificación deben seguir la guía de estilo de Airbnb para JavaScript.
- Herramientas de seguimiento de errores y gestión de tareas como JIRA.
- Automatización de pruebas utilizando Jest para el frontend y Mocha para el backend.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Requisitos de Dominio:

- 1.RD1: La plataforma debe cumplir con las normativas locales de protección de datos.
- 2.RD2: El sistema debe permitir la integración con múltiples proveedores de servicios de pago y envío.
- 3.RD3: Los vendedores deben poder gestionar inventarios y precios.
- 4.RD4: La plataforma debe soportar descuentos y cupones promocionales.
- 5.RD5: Los usuarios deben poder dejar comentarios y calificaciones de productos.
- 6.RD6: La plataforma debe ofrecer recomendaciones de productos basadas en el comportamiento del usuario.
- 7.RD7: Soporte para múltiples idiomas y monedas.
- 8.RD8: Gestión de cuentas de usuario con roles y permisos.
- 9.RD9: Registro detallado de transacciones para auditorías.
- 10.RD10: Reportes de ventas y análisis de datos para los vendedores.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Requisitos Funcionales

1. Los usuarios deben poder buscar y filtrar productos.
2. Registro y gestión de cuentas de usuario.
3. Los usuarios deben poder añadir productos al carrito de compras y realizar pedidos.
4. Gestión de inventarios por parte de los vendedores.
5. Procesamiento de pagos en línea.
6. Seguimiento de pedidos y notificaciones de estado.
7. Soporte para devoluciones y reembolsos.
8. Los vendedores deben poder visualizar sus ventas y análisis.
9. Envío automático de emails de confirmación tras compras.
10. Los usuarios deben poder generar listas de deseos.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### 2. Especificación de Requisitos de Desarrollo

**Contexto:** Después de la identificación y análisis de requisitos, el equipo documenta detalladamente cada requisito de desarrollo para proporcionar una guía clara y precisa durante el desarrollo del proyecto.

#### Requisitos de Desarrollo:

1. Entorno de desarrollo específicamente configurado para Windows 10 y macOS Catalina.
2. El back-end se desarrollará en Node.js v14 y el front-end en React.js v17.
3. Uso de Git para el control de versiones y GitHub para la colaboración, con un flujo de trabajo GitFlow.
4. SCRUM se implementará con sprints de dos semanas y reuniones diarias de stand-up.
5. Los desarrolladores recibirán una formación de 3 días en seguridad web basada en OWASP.
6. La base de datos principal será PostgreSQL, con MongoDB para manejar grandes volúmenes de datos no estructurados.
7. Jenkins se utilizará para CI/CD, con integración a través de pipelines automatizados.
8. Se adoptarán los estándares de codificación de Airbnb para JavaScript y se usarán ESLint y Prettier para la autoformateo.
9. JIRA se utilizará para la gestión de tareas y seguimiento de errores, con tableros Kanban para la visualización del flujo de trabajo.
10. Se utilizará Jest para pruebas unitarias y de integración en el frontend, y Mocha para el backend.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### 3. Verificación de Requisitos de Desarrollo

**Contexto:** Durante la fase de verificación, el equipo se asegura de que los requisitos de desarrollo se están cumpliendo y que el entorno de desarrollo está correctamente configurado y operativo.

1. Revisiones de pares para validar que los requisitos de desarrollo están correctamente documentados.
2. Creación y pruebas de un prototipo de la plataforma en el entorno de desarrollo especificado.
3. Realización de pruebas de rendimiento en el entorno de desarrollo para asegurar que se cumplen los tiempos de respuesta.
4. Realización de pruebas de carga para asegurar que el entorno de desarrollo soporta el número esperado de usuarios concurrentes.
5. Ejecución de pruebas de seguridad para validar la implementación de prácticas de desarrollo seguro.
6. Auditar el cumplimiento de los estándares de codificación mediante herramientas automáticas.
7. Validación de la configuración de CI/CD mediante pruebas de despliegue continuo automatizado con Jenkins.
8. Verificación de la integración correcta de bases de datos SQL y NoSQL.
9. Pruebas de automatización de tareas con JIRA para asegurar el correcto seguimiento y gestión de tareas y errores.
10. Validación de la herramienta de control de versiones (GitHub) para asegurar la integridad y coordinación del trabajo en el equipo.



# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### 3. Gestión de Cambios en los Requisitos de Desarrollo

**Contexto:** Durante el desarrollo del proyecto, pueden surgir necesidades de cambio en los requisitos de desarrollo. La gestión de estos cambios es crucial para mantener la consistencia y calidad del proyecto.

#### Requisitos de Desarrollo:

1. Establecer un proceso formal de solicitud de cambios mediante JIRA, incluyendo la evaluación y aprobación por los stakeholders pertinentes.
2. Documentar y registrar todos los cambios solicitados en el sistema de gestión de cambios.
3. Evaluar el impacto de cada cambio en el cronograma, costos y calidad del proyecto.
4. Priorización de los cambios en función del impacto y la urgencia.
5. Aprobación de los cambios por el equipo de desarrollo y stakeholders.
6. Comunicación de los cambios aprobados a todo el equipo.
7. Implementación controlada de los cambios a través de ramas específicas en Git.
8. Pruebas de verificación de los cambios implementados para asegurar que cumplen con las expectativas.
9. Revisión de las métricas y resultados de las pruebas para evaluar el éxito de los cambios.
10. Revalidación de los requisitos de desarrollo y ajuste del plan del proyecto según sea necesario.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Requisitos de Dominio, Funcionales y No Funcionales:

- Se incluyen según el caso y se documentan en secciones separadas para mantener claridad y precisión. Se utiliza una plantilla estándar que divide claramente cada tipo de requisito.
- Los requisitos son validados mediante pruebas funcionales (funcionales), pruebas de usabilidad (no funcionales), y revisiones con expertos para garantizar cumplimiento regulatorio y operacional (dominio).
- Los cambios son gestionados mediante un proceso similar, asegurando que todos los tipos de requisitos se ajustan y actualizan de manera coherente. Las sesiones de revisión y validación con stakeholders para cada tipo de requisito aseguran que no se comprometan las expectativas iniciales ni la calidad del proyecto.
- A través de los casos detallados se identifican, especifican, verifican y gestionan los requisitos de desarrollo diferenciándolos claramente de los requisitos funcionales, no funcionales y de dominio. Esta práctica les proporcionará una comprensión profunda y habilidades prácticas necesarias para llevar a cabo proyectos complejos de desarrollo de software de manera efectiva y eficiente.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Requisitos de Dominio, Funcionales y No Funcionales:

- Se incluyen según el caso y se documentan en secciones separadas para mantener claridad y precisión. Se utiliza una plantilla estándar que divide claramente cada tipo de requisito.
- Los requisitos son validados mediante pruebas funcionales (funcionales), pruebas de usabilidad (no funcionales), y revisiones con expertos para garantizar cumplimiento regulatorio y operacional (dominio).
- Los cambios son gestionados mediante un proceso similar, asegurando que todos los tipos de requisitos se ajustan y actualizan de manera coherente. Las sesiones de revisión y validación con stakeholders para cada tipo de requisito aseguran que no se comprometan las expectativas iniciales ni la calidad del proyecto.
- A través de los casos detallados se identifican, especifican, verifican y gestionan los requisitos de desarrollo diferenciándolos claramente de los requisitos funcionales, no funcionales y de dominio. Esta práctica les proporcionará una comprensión profunda y habilidades prácticas necesarias para llevar a cabo proyectos complejos de desarrollo de software de manera efectiva y eficiente.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Requisitos de Desarrollo (RD):

- **Desarrollo (RD):** Requisitos relacionados con el entorno, herramientas, metodologías y prácticas de desarrollo.
- **Dependencias:** Los requisitos de desarrollo suelen depender de varios requisitos funcionales y no funcionales, ya que proporcionan el marco necesario para realizar el trabajo de desarrollo efectivo.
- **Impacto:** Tienen un impacto directo en la eficiencia y eficacia del proceso de desarrollo.

### Requisitos de Dominio (DD):

- **Dominio (DD):** Requisitos específicos del negocio y regulaciones que el sistema debe cumplir.
- **Dependencias:** Los requisitos de dominio pueden depender tanto de los requisitos funcionales como no funcionales.
- **Impacto:** Aportan especificaciones clave que deben cumplirse por la naturaleza del campo de negocio o área de especialidad.

### Requisitos Funcionales (RF):

- **Funcionales (RF):** Describen lo que el sistema debe hacer. Incluyen características y comportamientos específicos del sistema.
- **Dependencias:** Pueden depender de los requisitos de desarrollo para la implementación correcta y de los requisitos de dominio para la validez.
- **Impacto:** Afectan directamente las funcionalidades y operaciones del sistema.

### Requisitos No Funcionales (NF):

- **No Funcionales (NF):** Relacionados con el rendimiento, seguridad, escalabilidad, y usabilidad del sistema.
- **Dependencias:** Pueden depender de los requisitos de desarrollo para garantizar su implementación adecuada.
- **Impacto:** Afectan la calidad del servicio proporcionado por el sistema.

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

Los actores y casos de uso de la plataforma de comercio electrónico, identificamos los actores involucrados y luego listamos los casos de uso correspondientes a cada actor. Resumen textual a ser transformado en un diagrama de casos de uso en UML.

### Actores:

- 1.Comprador
- 2.Vendedor
- 3.Administrador del Sistema
- 4.Sistema de Pago
- 5.Sistema de Envío
- 6.Visitante (Usuario No Registrado)

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Casos de Uso:

#### Comprador:

- 1.Registrarse en la plataforma
- 2.Iniciar sesión
- 3.Buscar productos
- 4.Filtrar productos
- 5.Agregar productos al carrito
- 6.Realizar pedidos
- 7.Ver historial de pedidos
- 8.Dejar comentarios y calificaciones
- 9.Solicitar devoluciones y reembolsos
- 10.Gestionar perfiles

## **Caso Práctico**

### **Desarrollo de una Plataforma de Comercio Electrónico**

#### **Casos de Uso:**

##### **Vendedor:**

- 1.Registrarse como vendedor
- 2.Iniciar sesión
- 3.Agregar y actualizar productos
- 4.Gestionar inventarios
- 5.Gestionar precios y descuentos
- 6.Ver y gestionar pedidos recibidos
- 7.Generar reportes de ventas
- 8.Responder a comentarios y calificaciones

# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Casos de Uso:

#### Administrador del Sistema:

1. Gestionar usuarios
2. Configurar parámetros del sistema
3. Generar informes administrativos
4. Gestionar recursos del sistema
5. Realizar auditorías de seguridad
6. Gestionar contenido del portal

### Casos de Uso:

#### Sistema de Pago:

1. Procesar pagos
2. Verificar transacciones
3. Emitir reembolsos

#### Sistema de Envío:

1. Gestionar envíos
2. Actualizar estado de envío
3. Proveer servicios de rastreo



# Caso Práctico

## Desarrollo de una Plataforma de Comercio Electrónico

### Casos de Uso:

Visitante (Usuario No Registrado):

- 1.Explorar la plataforma
- 2.Buscar productos
- 3.Ver detalles de productos
- 4.Registrarse
- 5.Ver comentarios y calificaciones de productos

### Descripción:

- **Comprador:** Realiza funciones de búsqueda y compra en la plataforma.
- **Vendedor:** Gestiona sus productos, inventario, pedidos y precios.
- **Administrador del Sistema:** Configura, gestiona y audita el sistema y sus usuarios.

**Sistema de Pago:** Tareas de iniciar y procesar pagos y transacciones.

**Sistema de Envío:** Monitorea y actualiza estado de envío.

*iiiA trabajar!!!*