

Semana 3

Ingeniería de Software (PRY3211)

Formato de respuesta

|  |  |
| --- | --- |
| **Nombre estudiante:** Mauricio Rojas (Grupo 1) | |
| **Asignatura: Ingeniería de Software** | **Carrera: Analista Programador** |
| **Profesor: Victor Rosendo** | **Fecha: 31-03-25** |

# Descripción de la actividad

En esta semana realizarás la actividad sumativa grupal con encargo de la Experiencia 1, llamada "Definiendo la visión del producto inicial bajo una propuesta de planificación ágil", la cual estará dividida en dos partes. En la primera parte, entregaran de manera grupal la propuesta del tipo de software a desarrollar con un documento Project Burndown basado en la lista de Requisitos del Sistema y aportando un Diagrama de Caso de Usos para representar la vista de escenario principal. Además, deberán aportar una Planificación ágil basada en Scrum basado en una Planilla Product Backlog con principales épicas e historias de usuarios y sus sprint, aportando una Planilla de Planificación del tiempo RoadMap distribuidas en semanas y la implementación de la herramienta Trello para la organización de principales tareas y entregables.

En la segunda parte, realizarán una presentación a través de un video de no más de 7 minutos, usando la herramienta de Teams.

En la tercera parte y de manera individual, deberás escribir una reflexión personal sobre los aprendizajes obtenidos en la Experiencia 1; cómo estos impactarán en tu desarrollo personal e influirán en el ámbito profesional y consecución de tus metas.

**Documento Proyect Burndown (versión 2)**

Sistema de Gestión de Reserva Hotelera

Descripción de la metodología de trabajo (Scrum)

Versión 1.0

Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autores** |
| 00/00/2023 | 1.0 | Primera versión definición de la Visión del Proyecto con los apartados y contenidos asociados | Mauricio Rojas |
| 08/05/2025 | 2.0 | Proyecto definido con su respectivo MVP | Mauricio Rojas |
|  |  |  |  |

Tabla de Contenidos

Contenido

[1. Introducción 3](#_Toc1700174074)

[1.1.1 Propósito de este documento 4](#_Toc1242438244)

[1.1.2 Problemática que resolver 4](#_Toc2023862465)

[1.1.3 Objetivo del Proyecto 4](#_Toc1858285031)

[1.1.4 Alcances 5](#_Toc1946355572)

[2. Descripción General de la Metodología ágil a adoptar. 5](#_Toc1950648504)

[2.1. Fundamentación 6](#_Toc431988535)

[2.2. Valores de trabajo 6](#_Toc447856862)

[3. Personas y roles del proyecto. 6](#_Toc265027249)

[4. Product Backlog. Lista de Componentes y artefactos para Construir. 7](#_Toc67129394)

[4.1 Épicas e historias de usuarios. 8](#_Toc1571379472)

[5. Definición del Done 9](#_Toc1899544251)

[5. Definición tecnologías de Desarrollo a utilizar 9](#_Toc42934262)

[7. Descripción de herramientas de gestión en modalidad Scrum. 11](#_Toc442726364)

# Introducción

Este documento describe la implementación de la metodología de trabajo **Scrum** para el desarrollo del proyecto **"Sistema de Gestión de Reservas de Habitaciones"** destinado a optimizar y automatizar el proceso de reserva de habitaciones en un sistema hotelero. El proyecto busca mejorar la experiencia tanto para los clientes como para los administradores, asegurando una interfaz sencilla y fluida para la consulta, reserva y gestión de habitaciones, además de la integración con pasarelas de pago y la gestión eficiente de perfiles de usuario.

El sistema tiene como objetivo principal agilizar el proceso de reservas, proporcionando una solución robusta, segura y eficiente para la gestión de la disponibilidad de habitaciones, las reservas y las transacciones de pago. De igual manera, se pretende que el sistema sea escalable y eficiente para manejar grandes cantidades de usuarios y reservas, mientras se asegura de mantener altos estándares de rendimiento, seguridad y disponibilidad.

## **Propósito de este documento**

El propósito de este documento es poder facilitar la información de referencia necesaria a las personas involucradas en el desarrollo del **Sistema de Gestión de Reservas de Habitaciones**. Este documento proporcionará una descripción general sobre cómo se ejecutará el desarrollo del producto de software utilizando un ciclo de vida adaptativo e integrativo basado en la metodología Scrum.

Se dará a conocer el enfoque del proyecto, que incluye las **Épicas** y sus respectivas **historias de usuarios**, los componentes y artefactos a construir, y los principales pasos a seguir para garantizar el éxito del desarrollo del sistema. También se describirán los **roles del equipo Scrum** y las competencias técnicas necesarias para cada rol dentro del equipo, asegurando que las funciones y responsabilidades estén bien definidas para cada miembro del equipo.

## **Problemática que resolver**

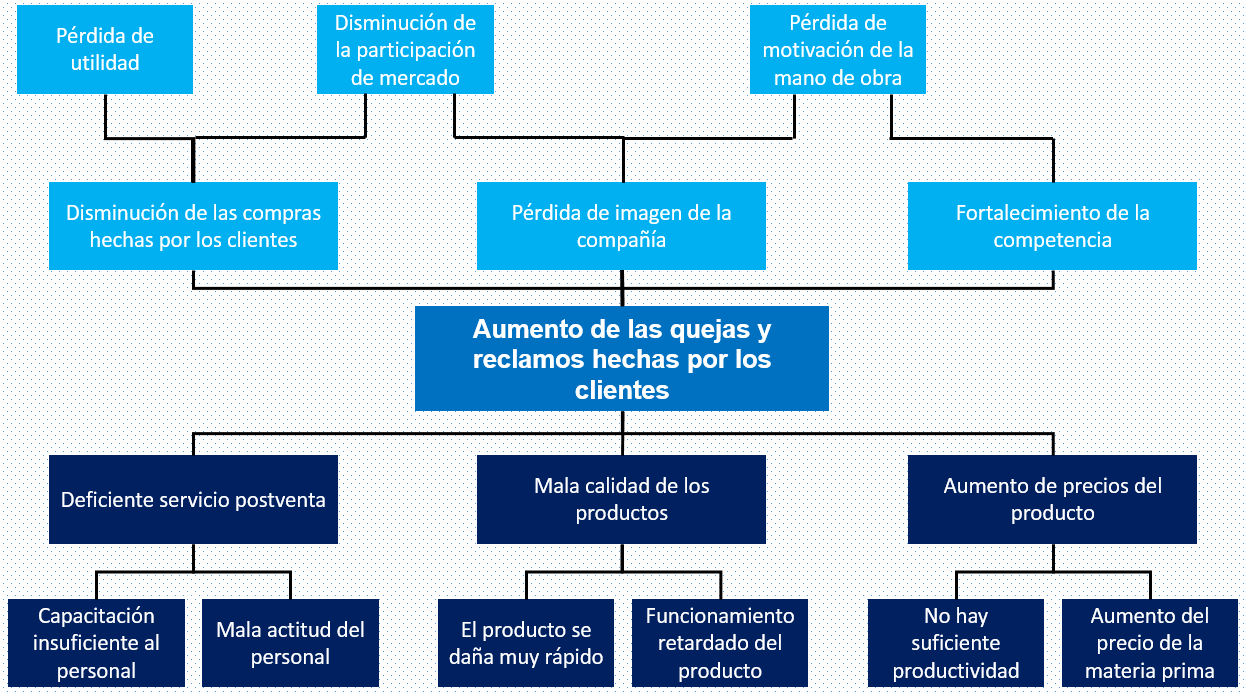
La problemática que se busca resolver con el desarrollo de este sistema es la **ineficiencia** y **complejidad** que enfrentan tanto los clientes como los administradores al gestionar las reservas de habitaciones de un hotel. Actualmente, muchos sistemas de reservas no ofrecen una experiencia de usuario fluida y accesible, lo que lleva a frustraciones en los clientes que desean realizar reservas rápidas y eficientes. Además, los administradores del hotel deben lidiar con sistemas dispersos, dificultando la gestión y seguimiento de las reservas, lo que afecta la toma de decisiones sobre la disponibilidad de habitaciones y la organización de pagos.

Otro problema importante que se busca resolver es la **gestión inadecuada de los datos** de los clientes y las reservas. Muchos sistemas no ofrecen un control de disponibilidad efectivo, lo que puede llevar a la doble reserva de habitaciones o incluso a la pérdida de reservas importantes. Además, la falta de integración con sistemas de pago modernos puede generar errores en las transacciones, afectando la experiencia de pago del cliente. Este proyecto tiene como objetivo central resolver estas dificultades, proporcionando una solución integral que optimice tanto la consulta de disponibilidad de habitaciones como el procesamiento de pagos, todo mientras asegura la integridad y seguridad de los datos.

Ejemplo:

**Figura 1:**

*Mapa del Problema*



Nota: Ejemplo de un mapa de problemas. Betancourt, D. (2016). *Cómo hacer un árbol de problemas: Ejemplo práctico*. Recuperado el 26 de mayo de 2023, de Ingenio Empresa: [www.ingenioempresa.com/arbol-de-problemas](http://www.ingenioempresa.com/arbol-de-problemas)

## **Objetivo del Proyecto**

El objetivo principal de este proyecto fue **desarrollar un sistema de gestión de reservas de habitaciones** que permita a los usuarios consultar la disponibilidad de habitaciones, realizar reservas de manera eficiente, gestionar pagos y recibir confirmaciones, mientras que los administradores puedan gestionar y modificar reservas, así como controlar la disponibilidad de las habitaciones en tiempo real. Este sistema se integrará con pasarelas de pago, gestionará perfiles de usuario y brindará notificaciones automáticas a los clientes, garantizando una experiencia fluida tanto para el cliente como para el administrador.

A la fecha de esta versión (v2.0), se ha logrado implementar un MVP funcional que cumple con todos los requerimientos definidos en la planilla Product Backlog. El sistema permite la consulta de disponibilidad, reserva, gestión de pagos mediante integración con Transbank, y gestión de usuarios, entre otras funcionalidades. Estas funciones están disponibles de forma local mediante una interfaz desarrollada en Python y Django.

## **Alcances**

**Alcances del Proyecto**:

* **Funcionalidades desarrolladas e implementadas**:
  + Consulta de disponibilidad de habitaciones con interfaz amigable y validación de fechas.
  + Registro de usuario personalizado y autenticación (login).
  + Reserva de habitaciones desde el panel del cliente.
  + Integración con **API Transbank (Webpay)** para simulación de pagos en entorno local.
  + Panel de administración para la gestión de habitaciones y reservas.
  + Visualización de historial de reservas del usuario.
  + Notificaciones internas del sistema en interfaz (a futuro: envío por email).
  + Protección de rutas según tipo de usuario (cliente o administrador).
* **Tecnologías Utilizadas**:
  + Backend: Python, Django, Django ORM, conexión local a base de datos SQLite.
  + Frontend: HTML, CSS, JavaScript (en Django Templates).
  + Integración de pagos: API Transbank Webpay (modo integración).
  + Entorno de desarrollo: Visual Studio Code.
  + Sistema operativo: desarrollo local en Windows 10/11.
  + Control de versiones: Git (GitHub/Trello para gestión Scrum).

**Limitaciones y Restricciones**:

* + **Plataforma de pago**: se encuentra funcional en entorno local usando la API de Transbank en modo integración. No está habilitado en producción.
  + **Infraestructura de despliegue**: el MVP se ejecuta de manera local, aún no está desplegado en un servidor (por ejemplo, Heroku, AWS o Railway).
  + **Notificaciones automáticas**: por ahora solo se visualizan en interfaz, no se incluye el envío de correos electrónicos u otros canales de comunicación externa.
  + **Escalabilidad**: el MVP está optimizado para funcionamiento local, y no se ha probado en ambientes de alta concurrencia (más de 100 usuarios simultáneos).
  + **Compatibilidad multiplataforma**: si bien es accesible desde navegador web, no se ha probado exhaustivamente en todos los dispositivos móviles o navegadores.
  + **Gestión de usuarios**: se usa un modelo personalizado (Usuario) en lugar del sistema de usuarios estándar de Django, lo que implica ciertas restricciones en compatibilidad con librerías externas.
  + **Seguridad**: incluye validaciones básicas y separación de roles, pero no se ha auditado el sistema contra ataques como XSS, CSRF o inyecciones SQL.
  + **Recuperación ante fallos y respaldo de datos**: no se implementó en el MVP una política de backup ni mecanismos automáticos de recuperación.

# Descripción General de la Metodología ágil a adoptar.

## Fundamentación

El ciclo de vida de desarrollo adoptado en este proyecto fue el **ciclo iterativo e incremental** de Scrum. Se utilizó Trello para la gestión del Product Backlog, y se definieron roles de Product Owner, Scrum Master y Development Team. Se generó un burndown chart para seguir el avance de las tareas, logrando un cumplimiento progresivo y verificable de funcionalidades durante los sprints.

Esto permitió la entrega continua de funcionalidades de acuerdo con las prioridades del negocio y de los usuarios, tal como se describió en el ERS de la primera versión (v1.0). A través de este enfoque ágil, se aseguró que cada requisito del sistema sea entregado y probado de forma progresiva en sprints (ciclos cortos de desarrollo).

La metodología Scrum fue adecuada para este proyecto ya que permitió gestionar tanto los requisitos funcionales (como la gestión de reservas, procesamiento de pagos, y administración de perfiles de usuarios) como los no funcionales (seguridad, fiabilidad, y rendimiento del sistema) de manera flexible y adaptativa. Además, el uso de Scrum facilitó la continua interacción con los interesados (clientes y administradores) para obtener retroalimentación sobre las funcionalidades a implementar en cada sprint.

Cada sprint se desarrolló, probó y entregó todas las partes funcionales del sistema, siguiendo las prioridades indicadas en el **ERS** y la **planilla de requerimientos definidos en la ERS (v1.0)**.

## Valores de trabajo

Para trabajar de manera efectiva en un entorno ágil como Scrum, el equipo contó con un conjunto de competencias técnicas y habilidades blandas que favorecieron la colaboración, la innovación y la entrega de valor continuo. A continuación, se detallan las competencias necesarias:

**Competencias Técnicas:**

1. **Conocimientos en desarrollo de software multiplataforma:** El equipo debe tener experiencia en el desarrollo de sistemas accesibles desde diversas plataformas (web, móvil, etc.), ya que el **ERS** indica que el sistema debe ser accesible en estos entornos.
2. **Experiencia en bases de datos:** Debido a que el sistema requiere integrar una base de datos para gestionar la información de reservas, usuarios y pagos, es fundamental contar con desarrolladores con conocimientos avanzados en bases de datos tales como SQL, y django.
3. **Conocimiento en seguridad de sistemas:** La implementación de seguridad debe ser uno de los pilares del sistema (según los requisitos del **ERS**), lo que requiere que el equipo tenga conocimientos sobre autenticación segura, protección contra vulnerabilidades (inyecciones SQL, XSS, etc.), y medidas para garantizar la confidencialidad de los datos.
4. **Desarrollo de APIs y servicios externos:** El sistema integró una API para procesamiento de pagos (según **R-6** de la planilla), por lo que los desarrolladores son expertos en el diseño, consumo e integración de APIs externas.

**Habilidades Blandas:**

1. **Comunicación efectiva:** El equipo fue capaz de comunicarse de manera fluida y constante durante todas las fases del proyecto, tanto dentro del equipo como con los stakeholders. Esto es clave en la metodología Scrum, donde las interacciones diarias en las reuniones (Daily Scrum) son esenciales para asegurar que todos estén alineados.
2. **Adaptabilidad y flexibilidad:** El proyecto, siendo de naturaleza ágil, tuvo cambios continuos basados en la retroalimentación de los usuarios y en la evolución de las prioridades del negocio. Los miembros del equipo deben ser capaces de adaptarse rápidamente a estos cambios y ajustar su trabajo de manera eficiente.
3. **Trabajo colaborativo:** Scrum enfatiza la colaboración entre los miembros del equipo. Los desarrolladores deben estar dispuestos a colaborar entre sí y con los roles de Scrum, como el Product Owner y Scrum Master, para alcanzar los objetivos de cada sprint.
4. **Resolución de problemas:** El equipo estuvo preparado para abordar problemas técnicos y no técnicos de forma efectiva durante el proceso de desarrollo, asegurando que el sistema funcione correctamente y cumpla con los requisitos establecidos en el **ERS**.

# Personas y roles del proyecto.

|  |  |  |
| --- | --- | --- |
| **Persona** | **Rol** | **Función** |
| Ana González | Stakeholder/s | Proveer retroalimentación continua sobre los requisitos del sistema. Asegurar que el sistema cumpla con las necesidades del cliente final. |
| Javier Pérez | Product Owner | Representar los intereses del cliente y asegurarse de que los requerimientos se alineen con las expectativas del mercado. |
| Mauricio Rojas | Scrum Master | Facilitar el proceso Scrum, eliminar impedimentos que el equipo enfrente y asegurar que se sigan las prácticas ágiles correctamente. |
| Roberto Sánchez | Developer 1 | Desarrollar funcionalidades del sistema como la consulta de disponibilidad de habitaciones, la gestión de reservas y la implementación de la seguridad (según **R-1**, **R-5**, **R-15**). |
| Marta López | Developer 2 | Implementar funcionalidades relacionadas con la base de datos, integraciones con el sistema de pagos, y la generación de confirmaciones y tickets (**R-6**, **R-7**, **R-13**). |
| Pablo Martinez | Developer 3 | Desarrollar interfaces de usuario y gestionar la experiencia del cliente en el sistema de reservas. Asegurarse de que el sistema sea intuitivo y fácil de usar (**R-8**, **R-9**). |

# Product Backlog. Lista de Componentes y artefactos para Construir.

## **Épicas e historias de usuarios.**

* Tabla resumen con Principales Épicas e Historias de Usuarios v/s Componentes.

[Insertar Tabla Product Backlog ampliada a Sprintbacklog y describir]

**(Insertar link de archivo planilla ubicado en su repositorio)**

**Figura 2:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req.** | **Enunciado de la Historia (Usuarios Clientes)** | **USUARIO** | **Artefacto** | **EPICA Componente** | **PTS por Historia Complejidad** | **Esfuerzo Tiempo**  **Días** | **Duración Sprint Semanas** | **Iteración (Sprint)** | **Prioridad** | **Estado** |
| **HU-01** | Como Scrum team, queremos implementar un servidor de aplicaciones y base de datos para cargar los elementos funcionales del sistema, para asegurar que el entorno de desarrollo y producción esté listo para el proyecto. | Scrum Team | Servidor de Aplicaciones y Base de Datos | E0: Implementar Ambiente de desarrollo y producción para el sistema | 14 | 3 | 2 | 1 | Alta | Hecho |
| **HU-02** | Como Scrum team, queremos implementar ambientes de prueba para realizar testing funcional y validar los Sprint, para garantizar que el sistema cumple con los requisitos y expectativas antes de avanzar. | Ambiente de Pruebas | 12 | 3 | Alta | Hecho |
| **HU-03** | Como Scrum team, queremos implementar aplicaciones de control de versiones y colaboración, para gestionar el código de forma eficiente y asegurar que el equipo trabaje de manera sincronizada. | Control de Versiones | 10 | 2 | Alta | Hecho |
| **HU-04** | Como Scrum team, queremos modelar y prototipar el producto mínimo viable de la Épica 1 de reserva de habitaciones, para determinar la visión inicial del producto y validarla con el Product Owner y el Cliente. | Prototipo MVP | 16 | 4 | Alta | Hecho |
| **HU-05** | Como administrador, quiero poder ver y modificar las reservas realizadas por los usuarios, para gestionar eficazmente las reservas y realizar cambios cuando sea necesario. | Administrador | Vista Administrativa de Reservas | E1: Gestión Administrativa | 20 | 5 | 3 | 2 | Alta | Hecho |
| **HU-06** | Como administrador, quiero poder gestionar el catálogo de habitaciones (añadir, modificar o eliminar), para mantener la información de las habitaciones actualizada y permitir que los usuarios vean opciones disponibles. | Vista Administrativa de Habitaciones | 18 | 5 | Alta | Hecho |
| **HU-07** | Como administrador, quiero poder gestionar el estado de las reservas (confirmadas, pendientes, canceladas), para tener control sobre el estado de cada reserva y asegurarme de que se gestionen correctamente. | Vista Administrativa de Estado de Reservas | 15 | 5 | Media | Hecho |
| **HU-08** | Como usuario, quiero poder consultar la disponibilidad de habitaciones para un rango de fechas específico, para poder tomar decisiones informadas al momento de planificar mi reserva. | Usuario (Cliente) | Interfaz de Consulta de Habitaciones | E2: Funcionalidades de Reservas y Pagos | 13 | 5 | 1 | 1 | Alta | Hecho |
| **HU-09** | Como usuario, quiero poder registrar una reserva para las habitaciones disponibles, para asegurarme de que mi estancia esté reservada y pueda disfrutar de la habitación que elijo. | Formulario de Reserva | 21 | 5 | Alta | Hecho |
| **HU-10** | Como usuario, quiero que el sistema calcule automáticamente el monto a pagar según la duración de la estancia y el tipo de habitación seleccionada, para poder conocer el precio total de manera clara y rápida. | Sistema de Cálculo de Pago | 16 | 5 | 2 | 3 | Alta | Hecho |
| **HU-11** | Como usuario, quiero poder realizar el pago de la reserva a través de una API de pago integrada, para completar mi reserva de manera segura y eficiente sin tener que salir del sistema. | API de Pagos | 18 | 5 | Alta | Hecho |
| **HU-12** | Como usuario, quiero recibir una confirmación de la reserva por correo electrónico, para tener una evidencia clara y oficial de mi reserva y poder acceder a la información cuando la necesite. | Usuario (Cliente) | Sistema de Envío de Confirmación | E3: Funcionalidades de Usuario | 10 | 4 | 2 | 4 | Media | Hecho |
| **HU-13** | Como usuario, quiero poder gestionar mi perfil personal dentro del sistema, para actualizar mi información de contacto, preferencias y detalles de reserva de manera autónoma. | Interfaz de Gestión de Perfil | 12 | 5 | Media | Hecho |
| **HU-14** | Como usuario, quiero poder recuperar mi contraseña en caso de olvido, para poder acceder a mi cuenta nuevamente sin depender del soporte técnico. | Formulario de Recuperación de Contraseña | 8 | 3 | 1 | 5 | Baja | Hecho |
| **HU-15** | Como usuario, quiero ser notificado automáticamente por correo cuando mi reserva se confirme o se cancele, para estar al tanto del estado de mi reserva sin tener que verificar manualmente. | Sistema de Notificación de Reservas | 12 | 4 | 2 | 6 | Alta | Hecho |
| **HU-16** | Como usuario, quiero poder ver un historial de todas mis reservas pasadas y futuras, para tener un registro completo de mis actividades de reserva en el sistema y hacer un seguimiento de mis estancias anteriores. | Vista de Historial de Reservas | 14 | 5 | Media | Hecho |
| **HU-17** | Como usuario, quiero estar seguro de que mis datos personales y financieros estén protegidos (autenticación con JWT, protección contra XSS y SQL Injection), para garantizar la seguridad de mi información dentro del sistema. | Sistema de Autenticación y Seguridad | 18 | 5 | 1 | 8 | Alta | Hecho |
| **HU-18** | Como administrador, quiero poder asegurarme de que el sistema funcione de manera rápida y eficiente, incluso con 100 usuarios simultáneos, para garantizar un buen desempeño del sistema bajo carga. | Sistema | Pruebas de Carga y Rendimiento | E4: Requisitos de Rendimiento y Fiabilidad | 13 | 5 | 1 | 7 | Alta | Hecho |
| **HU-19** | Como administrador, quiero poder asegurarme de que el sistema se recupere rápidamente en caso de fallo, con un tiempo de inactividad menor a 5 minutos, para minimizar las interrupciones del servicio y asegurar la continuidad del negocio. | Administrador | Sistema de Recuperación ante Fallos | 15 | 5 | 1 | 8 | Alta | Hecho |
| **HU-20** | Como usuario, quiero tener acceso al sistema durante un 99% del tiempo sin interrupciones, para poder realizar mis reservas y gestionar mi perfil sin problemas de disponibilidad. | Usuario (Cliente) | Sistema de Disponibilidad | E5: Disponibilidad | 10 | 4 | 1 | 9 | Alta | Hecho |
| **HU-21** | Como equipo de desarrollo, quiero garantizar que el sistema sea fácilmente mantenible y escalable en el futuro, para poder adaptarlo a nuevas necesidades y mantener su eficiencia a lo largo del tiempo. | Equipo de Desarrollo | Sistema de Mantenimiento y Escalabilidad | E6: Mantenibilidad y Portabilidad | 14 | 3 | 1 | 9 | Media | Hecho |
| **HU-22** | Como equipo de desarrollo, quiero asegurarme de que el sistema sea compatible con diferentes plataformas y entornos, para asegurar su accesibilidad y funcionalidad en diversas configuraciones de hardware y software. | Sistema Multiplataforma | 12 | 3 | Baja | Hecho |

# Definición tecnologías de Desarrollo a utilizar

Para el desarrollo del sistema, se utilizarán las siguientes tecnologías, las cuales se alinean con los requisitos del proyecto y se justifican por su robustez, escalabilidad y eficiencia:

**Tecnologías de Desarrollo:**

1. **Lenguaje Python**
   * **Versión recomendada**: Python 3.2 o superior.
   * **Justificación**:

* Versatilidad para el desarrollo de back-end, lógica de negocio y automatización.
* Excelente compatibilidad con el framework Django y soporte para conexión con Oracle Database.
* Amplia comunidad y abundante documentación.

1. **Framework Django**
   * **Versión recomendada**: Django 3.2 o superior.
   * **Justificación**:
     + Utilizado para construir la aplicación web, incluyendo vistas para cliente y administrador.
     + Estructura basada en MVT (Modelo-Vista-Template), que promueve una arquitectura limpia y escalable.
     + Soporte para autenticación personalizada (modelo Usuario) y middleware de seguridad.
     + Integra ORM que facilita la interacción con la base de datos Oracle.
2. **Lenguajes y Tecnologías de front-end:**

**HTML5, CSS3, JavaScript (ES6+)**

* **Justificación:**
  + Utilizados en el desarrollo de las vistas del cliente y administrador.
  + Aseguran compatibilidad con todos los navegadores modernos.
  + JavaScript proporciona dinamismo e interactividad (uso actual con funciones nativas; posible integración futura con frameworks como React o Vue.js).

1. **Oracle (Base de Datos)**

**- Versión recomendada: Oracle 19c o superior.**

**- Justificación:**

* **Base de datos robusta, usada en entornos empresariales.**
* **Compatible con procedimientos almacenados, funciones, triggers y packages (ya implementados en el sistema).**
* **Integración funcional mediante el backend en Django utilizando cx\_Oracle.**

1. **Herramientas de Gestión de Proyecto**

* **Git (control de versiones):**
  + Para gestión del código fuente en equipo, seguimiento de cambios y despliegue.
* **Trello (gestión ágil del proyecto):**
  + Uso de tableros para la planificación y seguimiento de tareas mediante el enfoque Scrum.
  + Integración del Product Backlog y control de avances de Sprint.

# Definición de tecnologías de implementación.

Para la implementación y despliegue del sistema, se consideran las siguientes tecnologías de infraestructura y servicios:

**Servidor de Aplicaciones:**

* **Tipo:** VPS (Servidor Privado Virtual) o Servidor en la Nube.
* **Proveedor sugerido:** DigitalOcean, Linode, AWS EC2 o PythonAnywhere.
* **Justificación:**
  + Permiten desplegar la aplicación Django con configuración personalizada (Gunicorn + Nginx).
  + Escalables en función de la demanda del sistema.
  + Soporte para entornos virtuales y gestión segura del código y datos.

**Servidor de Base de Datos:**

* **Tipo:** Instancia dedicada de Oracle en servidor propio o en la nube.
* **Proveedor sugerido:** Oracle Cloud Free Tier o despliegue local para fines académicos.
* **Justificación:**
  + Alta disponibilidad, rendimiento y herramientas administrativas avanzadas.
  + Configuración de usuarios, backups, y monitoreo del rendimiento.

**Otros Servicios de Soporte:**

* **Correo electrónico:** Uso de servicios SMTP para notificaciones automatizadas al usuario.
* **Repositorio Git remoto:** GitHub o GitLab para control de versiones y colaboración.
* **Despliegue CI/CD (opcional):** Posibilidad de integrar GitHub Actions para automatizar pruebas y despliegues futuros.
* **Backup y recuperación:** Configuración de respaldo periódico de la base de datos y archivos del sistema.

# Resultados del prototipado y validación

**Resultados del prototipado y pruebas**

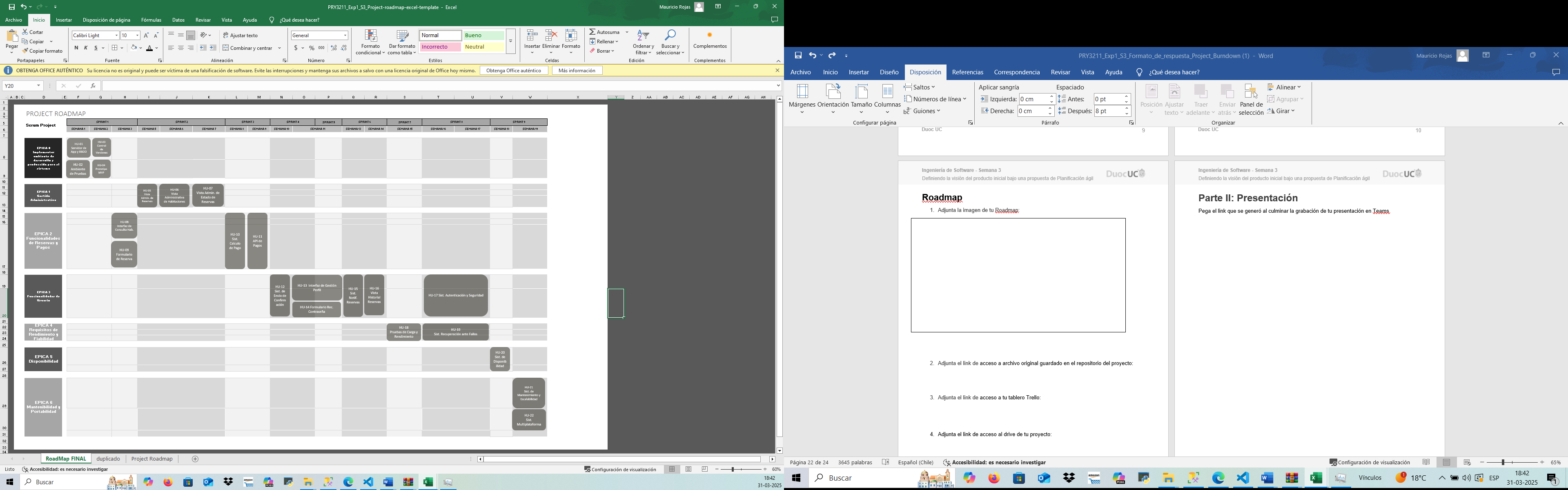
* Se desarrollaron vistas front-end para cliente y administrador, validadas durante reuniones de revisión de sprint.
* El sistema fue probado con casos de prueba definidos en el manual de usuario, alcanzando un 100% de cumplimiento de requisitos.
* Las funcionalidades fueron verificadas mediante pruebas funcionales exploratorias por parte del equipo.

**Roadmap**

1. Adjunta la imagen de tu Roadmap:

**Profesor, adjuntare el enlace del Excel donde elaboré el ROADMAP, ya que en imagen no se ve bien.**

**https://docs.google.com/spreadsheets/d/1DyhaaU6BWZIwz0UqWNMMV9MlR\_M83xcL/edit?usp=sharing&ouid=105796062867078251691&rtpof=true&sd=true**



1. Adjunta el link de acceso a archivo original guardado en el repositorio del proyecto:

https://github.com/maurojasaros/Ing.Software.git

1. Adjunta el link de acceso a tu tablero Trello:

<https://trello.com/invite/b/67eaf4da94966d32fe413437/ATTI2e108d6c4df17c248eadac3e03b3b8ad6315E53C/mi-tablero-de-trello>

[Mi tablero de Trello](https://trello.com/b/Mq294B5V/mi-tablero-de-trello)

1. Adjunta el link de acceso al drive de tu proyecto:

<https://drive.google.com/drive/folders/16MO-xhbKfF757Y1Npv7wQ_OxIroKu6IN?usp=sharing>

**Parte II: Presentación**

Pega el link que se generó al culminar la grabación de tu presentación en Teams

https://drive.google.com/file/d/1lHNlYZMgYlRsR\_qX\_UVkRegYrB2bC4hv/view?usp=sharing





**Duoc UC**