



**Semana 3**

Ingeniería de Software (PRY3211)

Formato de respuesta

| **Nombre estudiante:** Javiera Banda, Danitza Dominguez, Ayline Rios | |
| --- | --- |
| **Asignatura: Ingeniería de Software** | **Carrera: Analista Programador Computacional** |
| **Profesor: Jorge Canales** | **Fecha:** |

# 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, entregarán 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 influyen en el ámbito profesional y consecución de tus metas.

**Documento Proyect Burndown (versión 2)**

Sistema Reserva Pacific Reef

Descripción de la metodología de trabajo (Scrum)

Versión 1.0

Historial de Revisiones

| **Fecha** | **Versión** | **Descripción** | **Autores** |
| --- | --- | --- | --- |
| 26/08/2025 | 1.0 | Primera versión del proyecto, revisión del inicio y marcha del sistema de reservas. | Javiera Banda |
| 29/08/2025 | 1.0 | Revisión del proyecto, integración de Definición del Done y RoadMap | Danitza Dominguez |

Tabla de Contenidos

Contenido

[1. Introducción 3](#_heading=h.c4x5wve51q75)

[1.1.1 Propósito de este documento 4](#_heading=h.w2p1k0yf9871)

[1.1.2 Problemática que resolver 4](#_heading=h.w9j4ntcwvyhd)

[1.1.3 Objetivo del Proyecto 4](#_heading=h.6h3moylnvc9p)

[1.1.4 Alcances 5](#_heading=h.x4mbc33hcbwm)

[2. Descripción General de la Metodología ágil a adoptar. 5](#_heading=h.pz7mp958ckvy)

[2.1. Fundamentación 6](#_heading=h.xdngfw9z0jkc)

[2.2. Valores de trabajo 6](#_heading=h.6hkq8quwcke7)

[3. Personas y roles del proyecto. 6](#_heading=h.u6xtkc40m6id)

[4. Product Backlog. Lista de Componentes y artefactos para Construir. 7](#_heading=h.itopm3n08kb3)

[4.1 Épicas e historias de usuarios. 8](#_heading=h.cylsh3n7x9rk)

[5. Definición del Done 9](#_heading=h.p22yb5wju6t6)

6[. Definición tecnologías de Desarrollo a utilizar 9](#_heading=h.6nkei0fup0v1)

[7. Descripción de herramientas de gestión en modalidad Scrum. 11](#_heading=h.8e8m9q3ratuf)

# Introducción

Este documento presenta cómo vamos a trabajar en nuestro proyecto aplicando la metodología Scrum. El proyecto, llamado “Sistema de Reservas de Hotel”, tiene como idea principal facilitar a los turistas la búsqueda y reserva de habitaciones de manera rápida y sencilla, mientras que a los administradores y empleados del hotel les permitirá organizar mejor la información y llevar un control más claro de las reservas.

Lo que buscamos resolver es la falta de un sistema ágil que permita consultar disponibilidad en tiempo real, evitar confusiones con reservas duplicadas y reducir los errores que suelen ocurrir cuando todo se hace de forma manual. Con esta propuesta, tanto los clientes como el personal del hotel tendrán una herramienta moderna que mejora la experiencia de uso y la gestión interna.

Para lograrlo, decidimos trabajar con Scrum, porque nos permite avanzar paso a paso, organizando las tareas en Sprints y teniendo siempre claro qué debemos priorizar. También usaremos herramientas digitales como Trello y Miro para dar seguimiento al avance del equipo de una manera visual y colaborativa.

## **Propósito de este documento**

El objetivo de este documento es dar una guía clara a todas las personas que participan en el desarrollo del Sistema de reservas de Pacific Reef. La idea es que todos tengan la misma información de referencia y sepan cómo vamos a trabajar durante el proyecto.

Aquí explicaremos cómo se desarrollará el sistema siguiendo un ciclo de vida adaptativo e incremental, es decir, que lo iremos construyendo por etapas y mejorando en el camino según las necesidades que vayan surgiendo.

También mostraremos las Épicas y sus Historias de Usuario, además de los componentes y artefactos que forman parte del sistema. Esto ayudará a entender de manera sencilla qué se va a crear y cuál será el aporte de cada parte.

Por último, se presentarán los roles del equipo Scrum, explicando de forma simple qué hace cada integrante, cuáles son sus responsabilidades principales y las habilidades que necesitan para cumplir su función en el proyecto.

En pocas palabras, este documento busca que todos los que forman parte del equipo tengan la misma visión y claridad sobre lo que queremos construir, cómo lo vamos a organizar y qué esperamos entregar en cada etapa.

## **Problemática que resolver**

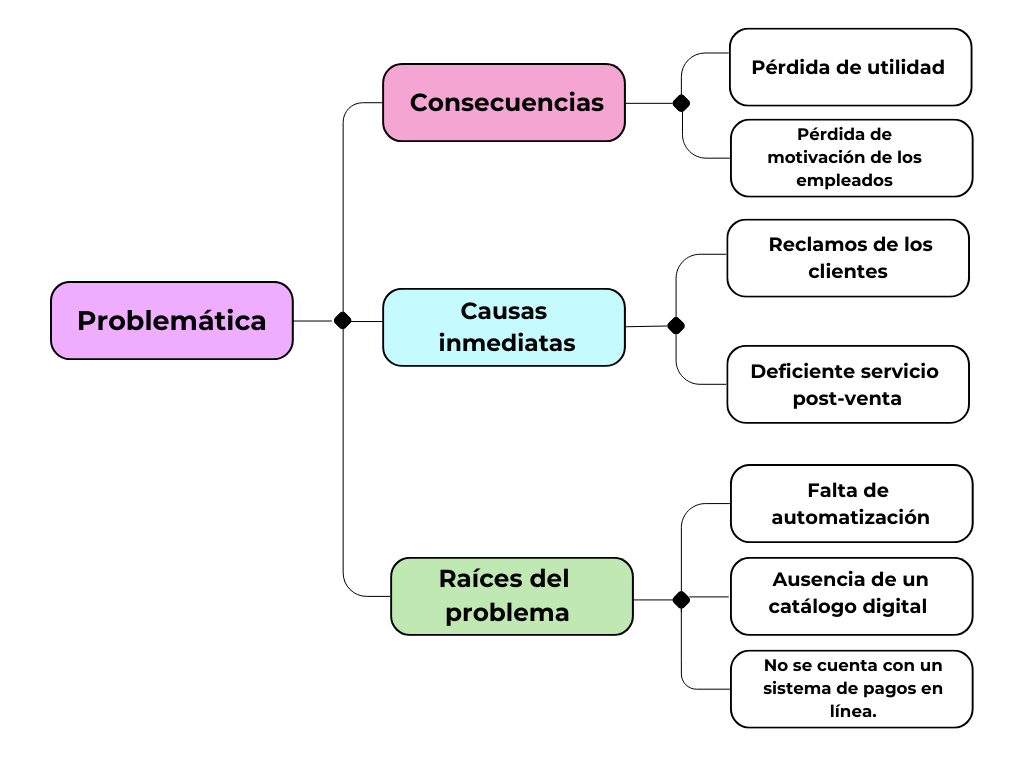
Actualmente, los hoteles y hospedajes suelen enfrentar dificultades al gestionar sus reservas, ya que gran parte del proceso se hace de manera manual o con sistemas poco integrados. Esto genera problemas frecuentes como reservas duplicadas, falta de actualización en tiempo real, errores en el registro de clientes y demoras en la confirmación. Estas falencias no solo afectan a la administración, sino que también perjudican la experiencia del turista, quien no siempre puede acceder a información confiable sobre la disponibilidad de habitaciones.

Por otro lado, los administradores y empleados deben manejar grandes cantidades de datos de forma dispersa, lo que aumenta la carga de trabajo y el riesgo de cometer errores humanos. La ausencia de un sistema digital centralizado limita la capacidad del hotel para responder de forma rápida y ordenada a las necesidades de los clientes. En consecuencia, existe una clara necesidad de contar con una solución tecnológica que facilite el proceso de reservas, brinde mayor transparencia y seguridad en los pagos, y entregue una mejor experiencia tanto a turistas como al personal interno.

Ejemplo:

**Figura 1:**

*Mapa del Problema*

**

## **Objetivo del Proyecto**

* **Implementar:** un sistema digital de reservas en línea para el hotel que permita a los turistas consultar disponibilidad en tiempo real y evitar duplicidad de reservas.
* **Desarrollar:** un módulo de pagos seguro e integrado que agilice las transacciones y brinde mayor confianza al cliente.
* **Construir:** una plataforma administrativa que centralice la información de reservas, usuarios y reportes, optimizando la gestión interna del personal.
* **Optimizar:** la experiencia del cliente mediante una interfaz intuitiva que facilite el proceso de búsqueda, reserva y confirmación de servicios.
* **Reducir**: los errores humanos y la sobrecarga de trabajo administrativo mediante la automatización de procesos clave.

## **Alcances**

* Lo que sí hará el sistema:

El sistema permitirá a los turistas consultar disponibilidad en tiempo real, realizar reservas en línea y efectuar pagos digitales de forma segura.

Incluirá un calendario de reservas y un catálogo virtual de habitaciones con información básica (tipo de habitación, precio y estado).

Los administradores podrán modificar o cancelar reservas, gestionar precios, visualizar reportes y administrar cuentas de usuarios internos.

El sistema será desarrollado bajo la metodología Scrum, en ciclos cortos (Sprints) que permitirán mejorar y ajustar funcionalidades de manera progresiva.

Se integrará con una base de datos centralizada para almacenar información de clientes, reservas y transacciones.

* Limitaciones o restricciones que podría tener el sistema:

En su primera versión, el sistema solo estará disponible en entorno web y no contará con aplicación móvil.

El idioma inicial será español, quedando la traducción a otros idiomas como una posible mejora futura.

El sistema no incluye pasarelas de pago internacionales, solo pagos locales predefinidos.

La seguridad se implementará mediante buenas prácticas y autenticación básica, pero no contará con certificaciones avanzadas (ejemplo: PCI DSS) en esta primera etapa.

La integración con sistemas externos (contabilidad, marketing, etc.) queda fuera del alcance inicial y se plantea como una evolución futura.

# Descripción General de la Metodología ágil a adoptar.

## **Fundamentación**

Para este proyecto decidimos trabajar con un ciclo de vida adaptativo e incremental, porque nos permite avanzar de a poco, entregando resultados visibles en cada etapa en lugar de esperar hasta el final para mostrar el sistema completo. La idea es ir construyendo el producto en partes más pequeñas, que podamos probar, revisar y mejorar con el feedback del equipo y de los usuarios.

Esto encaja muy bien con la forma de trabajar de Scrum, ya que se organiza el desarrollo en Sprints, donde cada uno tiene un objetivo claro y termina con un avance concreto del sistema. De esta manera, si aparece un cambio en los requisitos o surge una nueva necesidad, podemos adaptarnos sin que todo el proyecto se vea afectado. Además, así aseguramos que en cada entrega se aporte un valor real a quienes van a usar el sistema.

## **Valores de trabajo**

Para que el equipo Scrum pueda trabajar de manera efectiva en este proyecto, no basta solo con tener conocimientos técnicos, también son muy importantes las habilidades blandas que nos permiten coordinarnos y mantener una buena comunicación.

En lo técnico, el equipo debe manejar programación, bases de datos, control de versiones (como Git), diseño de interfaces y pruebas de software. Estas competencias son necesarias para que cada integrante pueda aportar en la construcción de los componentes del sistema, ya sea en el frontend, backend o en la integración con la base de datos.

Por otro lado, también necesitamos habilidades blandas que nos ayuden a trabajar de forma ágil y en equipo. Algunas de las más importantes son la comunicación clara, la colaboración, la responsabilidad personal y grupal, además de la adaptabilidad al cambio, porque en Scrum los requisitos pueden ajustarse en cualquier momento. Otro valor clave es la transparencia, ya que todos deben estar al tanto del avance real y de las dificultades que se presenten.

En pocas palabras, un equipo Scrum efectivo combina lo mejor de las competencias técnicas con valores de trabajo como el respeto, el compromiso y la apertura al cambio, lo que nos permitirá cumplir con los objetivos del proyecto de manera ordenada y colaborativa.

# Personas y roles del proyecto.

| **Persona** | **Rol** | **Función** |
| --- | --- | --- |
| Ayline Ríos | Stakeholder/s | Representa los intereses de los usuarios y aporta visión de lo que se quiere entregar como producto final. |
| Danitza Domínguez | Product Owner | Define y prioriza el backlog, asegurando que se cumplan los objetivos del proyecto. |
| Javiera Banda | Scrum Master | Facilita la aplicación de Scrum, resuelve impedimentos y fomenta la mejora continua. |
| Javiera Banda | Developer 1 | Actúa como *Developer 1*, participando en el desarrollo técnico junto a sus compañeras. |
| Ayline Ríos | Developer 2 | Participa activamente como *Developer 2*, contribuyendo en el desarrollo técnico del proyecto |
| Danitza Domínguez | Developer 3 | Cumple funciones de *Developer 3*, colaborando directamente en la programación y pruebas. |

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

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

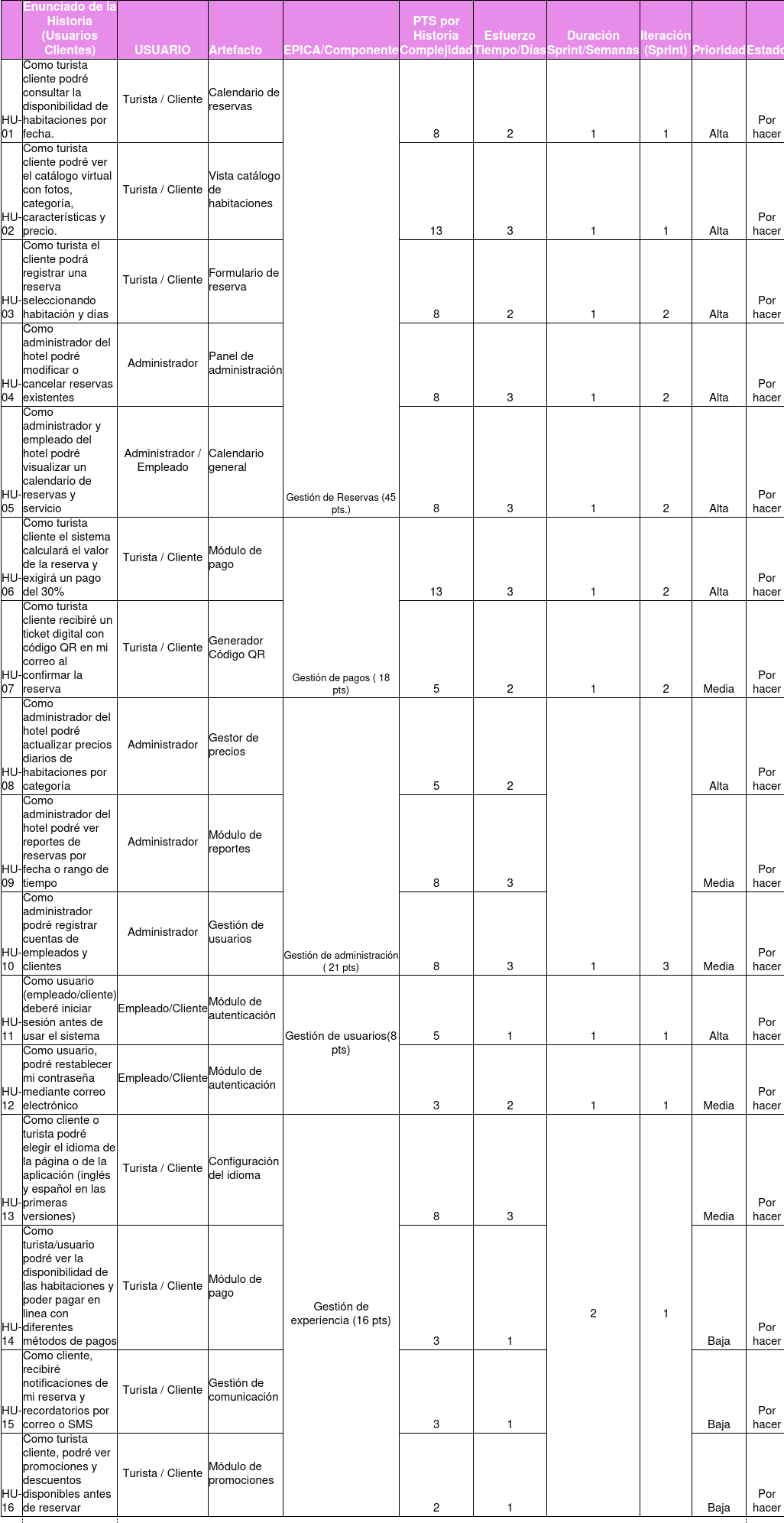
La siguiente tabla muestra la relación entre las Épicas, sus Historias de Usuario y los componentes o artefactos que se desarrollarán en el proyecto. Cada tarjeta está organizada en un tablero Trello bajo la metodología Scrum, lo que permite visualizar en qué estado se encuentra cada tarea: Por Hacer, En Progreso, En Revisión o Finalizada.

De esta manera, se logra una vista clara del avance del equipo. Por ejemplo, en este Sprint ya se encuentra finalizado el Calendario de Reservas, mientras que otros elementos como el Catálogo de Habitaciones y el Formulario de Reservas están en revisión. El resto de las funcionalidades se encuentran en desarrollo o pendientes de iniciar, lo que refleja un progreso ordenado y consistente con la planificación ágil.

**Link repositorio planilla product backlog: https://github.com/javbanda/Pacific\_reef/blob/main/e1\_Planilla\_product\_backlog\_grupo5\_s2.xlsx**

**Figura 2:**

*Planilla Product Backlog*



# Definición del Done

La Definición de Done (DoD) establece los criterios mínimos que cada historia de usuario o tarea debe cumplir para considerarse finalizada dentro del proyecto *Sistema de Reservas Pacific Reef*. Esta definición asegura que el equipo entregue funcionalidades de calidad, listas para ser utilizadas y que aporten valor real en cada sprint.

Para este proyecto, una tarea se considera Done cuando:

* El código ha sido desarrollado siguiendo las prácticas acordadas por el equipo.
* La funcionalidad ha sido probada localmente y pasa las pruebas unitarias (JUnit) y de integración.
* Se ha realizado la integración con la base de datos y servicios externos (ejemplo: pagos, correo).
* La interfaz ha sido validada respecto al diseño en Figma y cumple con criterios de usabilidad.
* La tarea está documentada en GitHub con commits claros y descripciones del cambio realizado.
* La funcionalidad está disponible en el entorno de pruebas (Docker + AWS) y es visible para el Product Owner.
* Se revisa en la reunión de sprint y recibe la aprobación final del Product Owner.

Con esta definición, garantizamos que cada entrega parcial del sistema no solo sea un avance técnico, sino también una funcionalidad usable, estable y alineada con las necesidades del usuario final.

1. [**Definición tecnologías de Desarrollo a utilizar**](#_heading=h.6nkei0fup0v1)
2. **Backend:**

El sistema utilizará Java 17 en conjunto con el framework Spring Boot 3.x, ya que permite construir aplicaciones web seguras, modulares y rápidas de implementar. Esta tecnología facilita la creación de APIs REST, el manejo de transacciones ACID, la integración con bases de datos y la implementación de medidas de seguridad, lo que resulta fundamental para gestionar las reservas, pagos y la lógica de negocio sin inconsistencias.

1. **Frontend:**

La interfaz gráfica se desarrollará con React 18, una librería que permite construir aplicaciones web dinámicas, responsivas y fáciles de mantener. Gracias a su enfoque basado en componentes, facilita la creación de módulos reutilizables como el catálogo de habitaciones, el calendario de disponibilidad y los formularios de reserva, ofreciendo una experiencia de usuario moderna, llamativa y fluida.

1. **Base de datos:**

Se trabajará con MySQL 8.0 en la nube, dado que es un sistema de gestión de bases de datos relacional confiable y de alto rendimiento. Compatible con operaciones transaccionales ACID y con soporte para JSON, resulta versátil tanto para proyectos pequeños como para aplicaciones empresariales de gran escala (ejemplo: Netflix). Será la encargada de almacenar y organizar los datos de usuarios, habitaciones, reservas y transacciones, garantizando la integridad y consistencia de la información.

1. **Pasarelas de pagos:**

El sistema contará con una pasarela de pagos externa (WebPay o PayPal) para procesar el anticipo obligatorio del 30% de la reserva. La comunicación se realizará mediante API REST en formato JSON bajo HTTPS, utilizando API keys o tokens OAuth para la autenticación. La confirmación de transacciones se recibirá a través de webhooks, los cuales serán verificados con firma digital antes de actualizar el estado en la base de datos. Esta implementación facilitará las pruebas del prototipo sin exponer información sensible ni realizar transacciones reales.

1. **Correo electrónico:**

Se integrará un servicio externo de mensajería, como SendGrid o AWS SES, para el envío automático de correos de confirmación de reserva. Estos correos incluirán un ticket digital con código QR como comprobante para el cliente. La comunicación se realizará mediante protocolo SMTP seguro (TLS) o API REST, según el proveedor elegido. Esta solución asegura una alta tasa de entrega, personalización de plantillas y mayor confiabilidad que un servidor propio.

1. **Seguridad y autenticación:**

La autenticación de usuarios se maneja con tokens JWT, lo que permite un control seguro de sesiones en aplicaciones web y su fácil integración con dispositivos móviles. Las contraseñas se almacenarán en la base de datos utilizando algoritmos de hashing como bcrypt, y toda la comunicación entre cliente y servidor se realizará bajo HTTPS, garantizando la confidencialidad e integridad de los datos.

1. **Herramientas de desarrollo:**

* **Control de versiones:** Git con repositorio en GitHub, para asegurar un trabajo colaborativo y controlado del código fuente.
* **Pruebas**: Postman para validar las APIs REST y JUnit 5 para pruebas unitarias en el backend.
* **Diseño y prototipado**: : Figma, para crear prototipos de la interfaz de usuario antes del desarrollo.
* **Contenedores**: Docker, que permite un entorno de desarrollo consistente y facilita el despliegue en la nube.

En conjunto, estas tecnologías han sido seleccionadas porque cumplen con los requisitos establecidos en el ERS, ofrecen facilidad de integración con servicios externos y permiten construir un sistema seguro, modular y adaptable, alineándose con las necesidades de un prototipo funcional en un plazo reducido.

# Definición de tecnologías de implementación.

Para la implementación del sistema de Gestión de Reserva Hotelera Pacific Reef se decidió trabajar en la nube, usando principalmente los servicios de Amazon Web Services (AWS). Esta elección se debe a que la nube facilita mucho el despliegue de un prototipo, permite escalar en caso de que el sistema crezca y ofrece servicios administrados que reducen el trabajo técnico del equipo. Además, AWS cuenta con documentación abundante y un plan gratuito que resulta útil para el desarrollo en etapa inicial.

El backend de la aplicación se ejecutará en una instancia EC2, donde se levantará el sistema desarrollado con Spring Boot (Tomcat embebido). Para mantener consistencia entre lo que se prueba en local y lo que corre en el servidor, se usará Docker, de manera que todo funcione igual en cualquier entorno. Sobre este mismo servidor se instalará Nginx, que actuará como proxy inverso, mejorará el rendimiento y permitirá gestionar los certificados digitales para que toda comunicación se haga bajo HTTPS. Con esto se asegura la protección de los datos sensibles, como información de usuarios y transacciones.

En el caso del frontend, durante el prototipo los archivos estáticos de React podrán servirse desde el mismo servidor Nginx en EC2, lo que simplifica el despliegue inicial. Sin embargo, pensando en un paso a producción, sería mejor considerar el uso de S3 junto con CloudFront, ya que esta combinación permite distribuir los archivos de manera más rápida y confiable a los usuarios cuando hay mayor volumen de tráfico.

Para la base de datos, se optará por MySQL 8.0 desplegado en AWS RDS, ya que este servicio es estable, seguro y tiene la ventaja de ofrecer copias de seguridad automáticas. Otro punto importante es que la base de datos no estará expuesta a internet, sino que sólo podrá ser accedida por el backend desde la misma red privada, lo que agrega una capa extra de seguridad. En cuanto al almacenamiento de imágenes del catálogo de habitaciones, se utilizará AWS S3, que es un servicio económico, escalable y con gran durabilidad, ideal para guardar este tipo de archivos sin recargar el servidor principal.

El sistema también necesita integraciones externas. Para los pagos en línea, se conectará a una pasarela como WebPay o PayPal. Esta integración funcionará con APIs REST en formato JSON, bajo comunicación segura con HTTPS y autenticación mediante tokens o API keys. Cuando el pago se complete, la pasarela enviará un webhook de confirmación que el sistema verificará antes de actualizar el estado en la base de datos, lo que ayuda a prevenir fraudes o registros duplicados. De forma similar, el envío de correos electrónicos de confirmación de reserva se hará con un servicio externo como SendGrid o AWS SES, ya que estos garantizan una alta tasa de entrega y permiten usar plantillas, evitando los problemas que tendría montar un servidor de correo propio.

En el tema de seguridad, toda la comunicación del sistema se realizará con TLS 1.2 o superior para que los datos viajen cifrados. Además, las credenciales y llaves privadas necesarias para conectarse con la base de datos o con servicios externos no se guardarán en el código, sino que se manejan con herramientas como AWS Secrets Manager, lo que da más seguridad en la gestión de secretos.

Para los recursos mínimos del prototipo, se considera suficiente una instancia EC2 de 1 vCPU, 2 GB de RAM y 20 GB de almacenamiento, junto con una base de datos RDS db.t3.micro. Esta configuración es pequeña pero suficiente para pruebas y demostraciones. Más adelante, si el sistema pasa a producción, será fácil aumentar la capacidad sin necesidad de cambiar toda la arquitectura.

Finalmente, para hacer más sencillo y seguro el proceso de despliegue, se usará GitHub Actions como herramienta de integración y entrega continua. Esto permitirá automatizar pasos cómo construir la imagen Docker, ejecutar pruebas y desplegar la aplicación en EC2. Al mismo tiempo, se conservarán las versiones anteriores del sistema, lo que hace posible hacer un rollback rápido en caso de errores, evitando interrupciones largas y manteniendo la estabilidad del prototipo.

**Roadmap**

1. Adjunta la imagen de tu Roadmap:

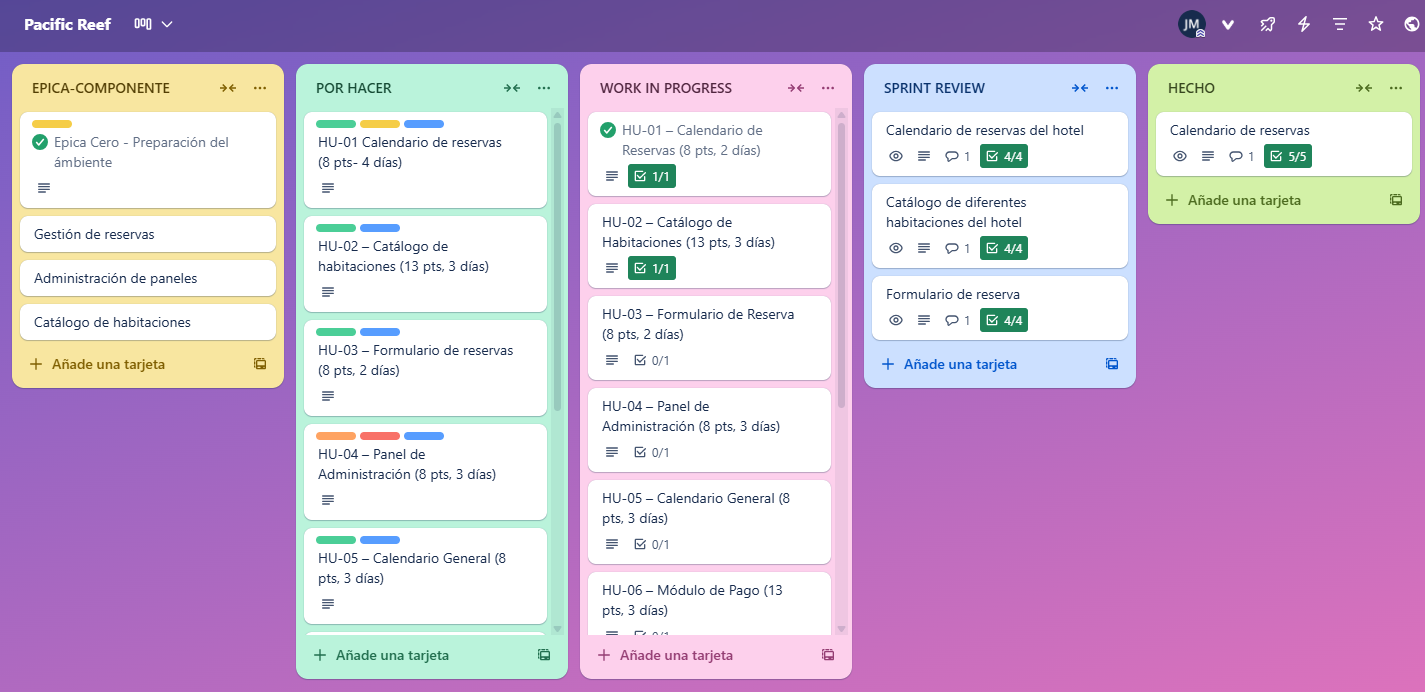


1. Adjunta el link de acceso a archivo original guardado en el repositorio del proyecto:

https://github.com/javbanda/Pacific\_reef

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

https://trello.com/b/Z9n0i8Dy/pacific-reef



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

https://drive.google.com/drive/u/1/folders/1bgxT1rkLeHo3XLeTAATyshAszd6tPVKl

**Parte II: Presentación**

https://drive.google.com/file/d/14EewVhDFB6WEQWIKEiw93DHxkbkoi02L/view?usp=drive\_link





**Duoc UC**