



Carrera

Analista Programador Computacional

Semana 2

Ingeniería de Software (PRY3211)

Formato de respuesta

Nombre estudiante: Camilo Arriagada, Pamela Azúa, Katherin Gastelumendi.	
Asignatura: Ingeniería de Software	Carrera: Analista Programador Computacional
Profesor: Patricio Aranedi Oliva	Fecha:

Descripción de la actividad

En esta segunda semana, deberás realizar la actividad formativa grupal con encargo llamada "Ajustando la visión del producto a un ciclo de desarrollo ágil", donde utilizarás el proyecto definido por medio del ERS para traspasarlo a modalidad ágil, definiendo épicas e historias de usuarios, creando el productbacklog del caso de Proyecto de la asignatura.

Instrucciones específicas

En este formato de respuesta deberás trabajar el Documento Project Burndown.

Primera parte: Documento Project Burndown

Sistema RoomBooker

Descripción de la metodología de trabajo (Scrum)

Versión 1.0

Historial de Revisiones

Fecha	Versión	Descripción	Autores
12/08/2024	1.0	Primera versión definición de la Visión del Proyecto con los apartados y contenidos asociados	

Tabla de Contenidos

Contenido

1. Introducción	4
1.1.1 Propósito de este documento	4
1.1.2 Problemática que resolver	4
1.1.3 Objetivo del Proyecto	5
1.1.4 Alcances	5
2. Descripción General de la Metodología ágil a adoptar.	5
2.1. Fundamentación	5
2.2. Valores de trabajo	6
3. Personas y roles del proyecto.	6
4. Product Backlog. Lista de Componentes y artefactos para Construir.	6
4.1 Épicas e historias de usuarios.	6
5. Definición del Done	7
5. Definición tecnologías de Desarrollo a utilizar	8

1. Introducción

Este documento describe la implementación de la metodología de trabajo Scrum para el desarrollo del proyecto **RoomBooker** destinado a **HOTEL PACIFIC REEF**. Este proyecto tiene como objetivo resolver una serie de desafíos relacionados con la gestión de reservas y la experiencia del cliente en el proceso de hospedaje. **HAPPY CODERS**, la empresa encargada del desarrollo, se compromete a entregar un producto de software que no solo cumpla con los requisitos técnicos y funcionales, sino que también mejore significativamente la eficiencia operativa y la satisfacción del cliente.

1.1.1 Propósito de este documento

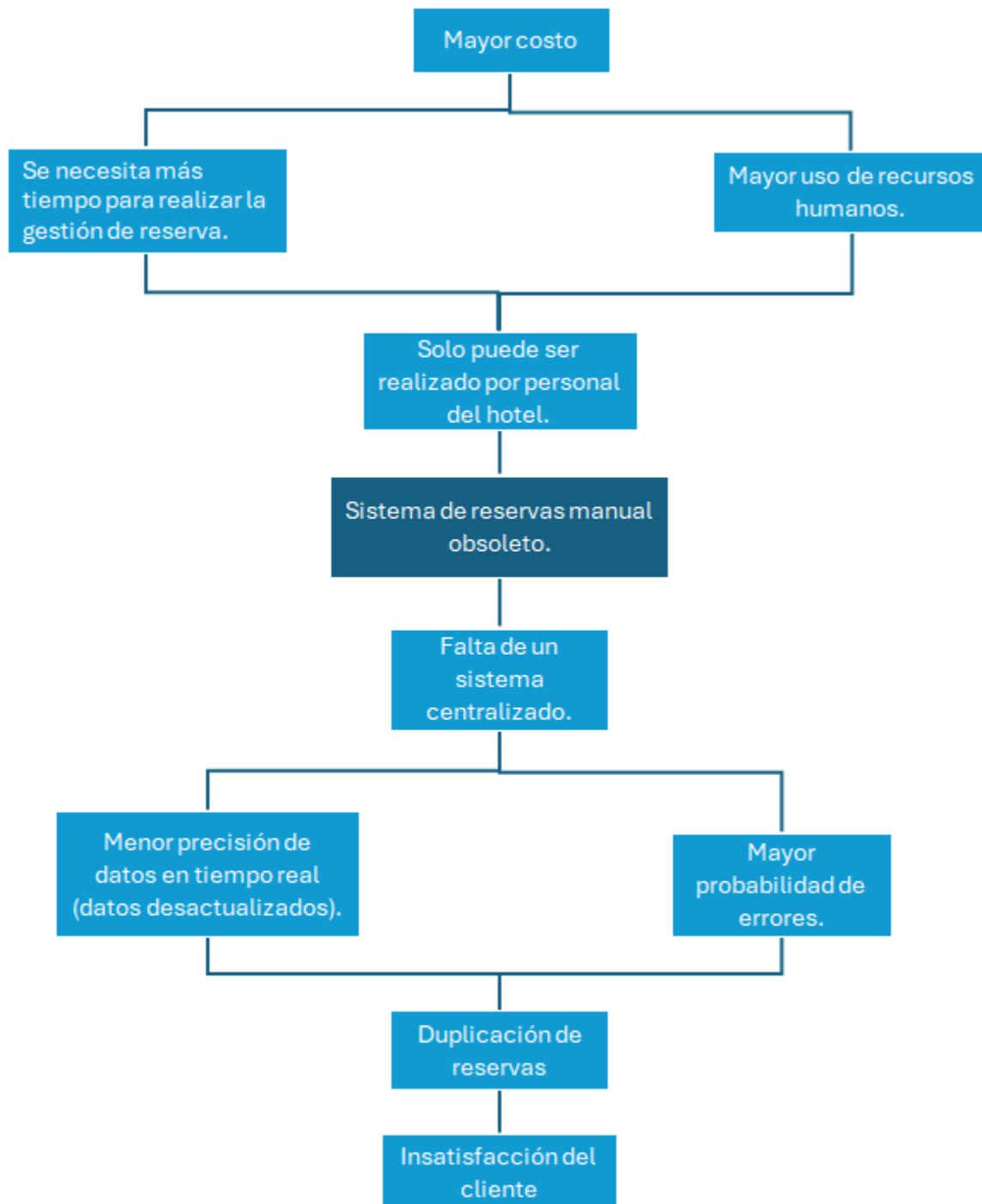
El propósito de este documento es facilitar la información de referencia necesaria a las personas involucradas en el desarrollo del sistema **RoomBooker**. Se explicará cómo se ejecutará el desarrollo del producto de software utilizando un ciclo de vida adaptativo e integrativo, siguiendo la metodología Scrum.

Además, se presentarán las Épicas y sus Historias de Usuarios, junto con los componentes y artefactos que se construirán. Se detallarán los roles dentro del equipo Scrum, especificando las competencias técnicas necesarias y las funciones que cada rol desempeñará en el proyecto.

1.1.2 Problemática que resolver

El **HOTEL PACIFIC REEF** enfrenta varios desafíos en la gestión de sus reservas y la administración general de su operación hotelera. Actualmente, el proceso de reserva es manual y requiere de una gran cantidad de tiempo y esfuerzo tanto por parte del personal como de los clientes. Esto no solo reduce la eficiencia del hotel, sino que también puede generar errores en la asignación de habitaciones y en la gestión de pagos, afectando negativamente la experiencia del cliente.

Además, la falta de un sistema centralizado de gestión de reservas dificulta la toma de decisiones basada en datos. Sin un sistema adecuado, es complicado para la administración obtener informes precisos y en tiempo real sobre la ocupación del hotel, las tendencias de reservas y la rotación de habitaciones. Esta falta de visibilidad impide que el hotel optimice sus operaciones y mejore la satisfacción del cliente, especialmente en un entorno tan competitivo como el de la industria hotelera.

Mapa del Problema

1.1.3 Objetivo del Proyecto

Desarrollar un sistema de gestión de reservas hoteleras para **HOTEL PACIFIC REEF** que **optimice** el proceso de reserva y registro de hospedaje, permitiendo una **gestión eficiente** de las habitaciones disponibles y **mejorando** la experiencia del cliente a través de un sistema intuitivo y accesible tanto para turistas como para el personal del hotel. Este sistema permitirá **automatizar** la consulta de disponibilidad, **facilitar** el registro de reservas, **calcular** y **procesar** pagos, y **generar** reportes detallados sobre la ocupación y gestión hotelera.

1.1.4 Alcances

Limitaciones del Sistema:

- **Idiomas Soportados:** El sistema estará disponible en inglés y español únicamente.
- **Plataformas:** La aplicación será accesible mediante dispositivos móviles y computadoras de escritorio, pero no se desarrollarán aplicaciones nativas específicas para iOS o Android en esta fase.
- **Métodos de Pago:** Solo se implementarán métodos de pago en línea comunes como tarjetas de crédito y débito. Otros métodos de pago, como transferencias bancarias o pagos en efectivo, no serán soportados en esta versión.
- **Integración con Sistemas Externos:** No se realizará integración con otros sistemas de gestión hotelera existentes en esta fase del proyecto.

Restricciones en la Implementación Futura:

- **Capacidad de Escalabilidad:** La versión inicial del sistema está diseñada para manejar la capacidad actual del **HOTEL PACIFIC REEF** (38 habitaciones). La escalabilidad para soportar un mayor número de habitaciones o múltiples establecimientos requerirá una futura reingeniería del sistema.
- **Funcionalidades Avanzadas:** Funcionalidades como la integración con sistemas de marketing digital, personalización avanzada para clientes VIP, o soporte para múltiples monedas, no están contempladas en la versión inicial y se considerarán en futuras fases de desarrollo.

- **Soporte Multilenguaje:** Cualquier expansión para incluir otros idiomas o configuraciones culturales será parte de fases de implementación futuras.

2. Descripción General de la Metodología ágil a adoptar.

2.1. Fundamentación

Ciclo de Vida del Desarrollo

El ciclo de vida adoptado para este proyecto es **iterativo e incremental**. Este enfoque permite la construcción del sistema a través de múltiples iteraciones, cada una de las cuales agrega nuevas funcionalidades al producto, con base en las prioridades establecidas por el Product Owner. Al final de cada Sprint, se entregará un incremento funcional del software, que será revisado y evaluado por el equipo y los stakeholders.

Sprints Cortos: Cada Sprint tendrá una duración de 2 semanas, permitiendo revisiones frecuentes y la posibilidad de realizar ajustes rápidos según los comentarios del cliente.

Revisión y Retroalimentación Continua: Al final de cada Sprint, se presentará un incremento del producto al cliente y stakeholders para obtener retroalimentación inmediata, lo que permitirá ajustar el desarrollo en función de las necesidades reales del negocio.

Flexibilidad en los Requisitos: Dado que los requisitos pueden evolucionar durante el proyecto, Scrum permite la re-priorización de las tareas en el backlog para asegurarse de que se trabaje en las funcionalidades más importantes primero.

Transparencia y Colaboración: Scrum promueve la transparencia en el proceso de desarrollo, con reuniones diarias (Daily Standups) donde el equipo discute el progreso, identifica obstáculos, y coordina los próximos pasos.

2.2. Valores de trabajo

- **Desarrollo de Software:** Conocimiento en lenguajes de programación, frameworks, y metodologías de desarrollo que sean relevantes para el proyecto.
- **Gestión de Bases de Datos:** Habilidad para diseñar, implementar y mantener bases de datos que aseguren la integridad y disponibilidad de la información.
- **Despliegue en la Nube:** Experiencia en el uso de plataformas de computación en la nube para el despliegue y la gestión del sistema.
- **Control de Versiones:** Competencia en el uso de herramientas de control de versiones para gestionar y colaborar en el código fuente del proyecto.
- **Seguridad Informática:** Conocimiento de las mejores prácticas de seguridad para proteger los datos y garantizar la integridad del sistema.
- **Pruebas y Validación:** Capacidad para desarrollar y ejecutar pruebas que aseguren el correcto funcionamiento del sistema.

3. Personas y roles del proyecto.

Persona	Rol	Función
	Stakeholder/s	
	Product Owner	
	Scrum Master	
Katherine Gastelumendi	Developer 1	Desarrolladora Back end
Pamela Azua	Developer 2	Desarrolladora Front end
Camilo Arriagada	Developer 3	Desarrollador Full Stack

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

4.1 Épicas e historias de usuarios.

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

Epicas

1. Gestión de Reservas

Descripción: Esta épica abarca todo lo relacionado con la creación, modificación, y cancelación de reservas en el sistema.

- **Historia de Usuario 1:** Como usuario, quiero consultar la disponibilidad de habitaciones en fechas específicas para poder hacer una reserva.
- **Historia de Usuario 2:** Como usuario, quiero registrar una reserva de una habitación para asegurar mi estadía en el hotel.
- **Historia de Usuario 3:** Como usuario, quiero cancelar o modificar mi reserva para ajustar mi estadía según mis necesidades.
- **Historia de Usuario 4:** Como administrador, quiero ver y gestionar todas las reservas realizadas para mantener el control de la ocupación del hotel.

2. Catálogo de Habitaciones

Descripción: Esta épica se enfoca en la visualización y administración del catálogo de habitaciones, incluyendo detalles y características de cada una.

- **Historia de Usuario 1:** Como usuario, quiero ver un catálogo visual de las habitaciones disponibles con fotos y descripciones detalladas para elegir la que mejor se adapte a mis necesidades.
- **Historia de Usuario 2:** Como administrador, quiero actualizar la información del catálogo de habitaciones para asegurarme que los datos estén siempre al día.

- **Historia de Usuario 3:** Como usuario, quiero filtrar las habitaciones por características específicas (como categoría, precio, etc.) para encontrar la opción que más me conviene.

3. Gestión de Pagos

Descripción: Esta épica incluye todas las funcionalidades necesarias para el cálculo, procesamiento, y confirmación de pagos en el sistema.

- **Historia de Usuario 1:** Como usuario, quiero calcular el costo total de mi reserva en función del número de días seleccionados y el tipo de habitación.
- **Historia de Usuario 2:** Como usuario, quiero realizar el pago de mi reserva de manera segura para confirmar mi estadía.
- **Historia de Usuario 3:** Como usuario, quiero recibir un comprobante de pago y un ticket de reserva con un código QR para presentarlo al hacer check-in.

4. Gestión de Usuarios

Descripción: Esta épica se refiere a la creación, administración, y autenticación de usuarios, tanto clientes como empleados del hotel.

- **Historia de Usuario 1:** Como administrador, quiero crear y gestionar cuentas de usuarios para controlar el acceso al sistema.
- **Historia de Usuario 2:** Como usuario, quiero registrarme en el sistema para acceder a mis reservas y gestionar mis datos personales.
- **Historia de Usuario 3:** Como usuario, quiero iniciar sesión en el sistema para acceder a las funcionalidades personalizadas.
- **Historia de Usuario 4:** Como administrador, quiero asignar roles y permisos a los empleados del hotel para que puedan realizar sus tareas dentro del sistema.

5. Gestión de Reportes

Descripción: Esta épica abarca la generación y visualización de reportes sobre las reservas y la ocupación del hotel.

- **Historia de Usuario 1:** Como administrador, quiero generar reportes diarios, semanales, o mensuales sobre las reservas realizadas para analizar la ocupación del hotel.
- **Historia de Usuario 2:** Como administrador, quiero visualizar estadísticas de ocupación por categoría de habitación para optimizar la gestión de recursos.
- **Historia de Usuario 3:** Como administrador, quiero exportar los reportes en diferentes formatos (PDF, Excel) para compartirlos con otros departamentos.

6. Cambio de Idioma

Descripción: Esta épica se centra en la capacidad del sistema para ofrecer soporte multilingüe y cambiar entre idiomas según las preferencias del usuario.

- **Historia de Usuario 1:** Como usuario, quiero cambiar el idioma del sistema entre inglés y español para utilizar la interfaz en mi idioma preferido.

7. Notificaciones y Recordatorios

Descripción: Esta épica abarca la funcionalidad para enviar notificaciones automáticas a los usuarios sobre sus reservas y recordatorios de próximos eventos.

- **Historia de Usuario 1:** Como usuario, quiero recibir una notificación por correo electrónico cuando mi reserva sea confirmada o cancelada.
- **Historia de Usuario 2:** Como usuario, quiero recibir un recordatorio antes de mi fecha de check-in para prepararme adecuadamente.
- **Historia de Usuario 3:** Como administrador, quiero poder configurar notificaciones automáticas para mantener a los clientes informados sobre sus reservas.

[Insertar Tabla Product Backlog y describir]

Va adjunto el archivo Excel.

(Insertar link de archivo planilla ubicado en su repositorio)

5. Definición tecnologías de Desarrollo a utilizar

Arquitectura del Sistema

1. Arquitectura General

El sistema se basa en una arquitectura de **cliente-servidor** dividida en tres capas principales:

- **Capa de Presentación (Frontend):** Utiliza **React** para construir una interfaz de usuario interactiva y responsiva. Esta capa se encarga de presentar la información al usuario y recoger sus entradas, interactuando con el servidor a través de APIs.
- **Capa de Aplicación (Backend):** **FastAPI** se encarga de proporcionar la lógica del servidor y exponer APIs para que el frontend pueda interactuar con el sistema. Esta capa maneja la lógica de negocio, la autenticación, y la comunicación con la base de datos.
- **Capa de Datos (Base de Datos):** **PostgreSQL** actúa como el sistema de gestión de bases de datos, almacenando toda la información relevante sobre reservas, habitaciones, usuarios, y pagos. Esta capa asegura la integridad y la disponibilidad de los datos.

2. Flujo de Datos

- **Interacción del Usuario:** Los usuarios interactúan con la aplicación a través del frontend en React. La información ingresada o solicitada por los usuarios se envía al backend mediante llamadas a las APIs expuestas por FastAPI.
- **Procesamiento en el Backend:** FastAPI procesa las solicitudes del frontend, realiza la lógica de negocio necesaria y accede a la base de datos PostgreSQL para recuperar o almacenar información. Luego, el backend devuelve los resultados al frontend.
- **Almacenamiento y Recuperación de Datos:** PostgreSQL maneja todas las operaciones relacionadas con la persistencia de datos, asegurando que la información esté correctamente almacenada y sea accesible de manera eficiente para el backend.

3. Beneficios de la Arquitectura

- **Modularidad:** La separación en capas permite una clara división de responsabilidades y facilita el mantenimiento y la escalabilidad del sistema.
- **Desempeño:** La combinación de React para el frontend y FastAPI para el backend garantiza una experiencia de usuario rápida y eficiente, mientras que PostgreSQL asegura un manejo de datos robusto y confiable.
- **Flexibilidad:** La arquitectura permite actualizar o reemplazar componentes individuales sin afectar el sistema en su totalidad, facilitando adaptaciones futuras o mejoras.

6. Definición de tecnologías de implementación.

1. Servidor de Aplicaciones

- **Descripción:** El servidor de aplicaciones es el entorno donde se desplegará la aplicación web. Es responsable de ejecutar el código de la aplicación y servir las solicitudes de los usuarios.
- **Tipo: Nube** (Cloud)
- **Google Cloud Platform (GCP):** Proporciona **Google Compute Engine** para máquinas virtuales y **Google App Engine** para despliegues gestionados.

7. Adjunta el link de acceso al archivo original guardado en el repositorio del proyecto aquí:



DuocUC[®] ONLINE

Duoc UC