Anteproyecto – Eduardo Merino Fernández

Aplicación para cadena de hoteles

1. INTRODUCCIÓN

Este proyecto va a tratar de una plataforma web que se encarga de gestionar reservas para una cadena de hoteles. Va a constar de 2 partes, una parte de Front que será toda la interfaz web a la que puede acceder el usuario y realizar las funciones y otra parte back encargada de realizar operaciones y comunicarse con la base de datos SQL. Habrá un apartado para los diferentes administradores de los hoteles que puedan realizar diferentes gestiones.

2. UBICACIÓN Y DESTINATARIOS

La aplicación va dirigida a empresas hoteleras que requieran un sistema de reservas eficiente y una interfaz para el usuario clara y con numerosas funcionalidades como el sistema de puntos de fidelización. El sistema está diseñado para una cadena de hoteles, permitiendo la gestión de reservas de habitaciones y servicios adicionales como desayunos o comidas, a parte de esta gestión la aplicación será usada por los clientes para que puedan reservar online su habitación y esta reserva será registrada en la base de datos del hotel.

**Supuesto Práctico Real:** Una cadena hotelera con presencia en varias ciudades necesita una solución centralizada para gestionar las reservas de sus habitaciones. El sistema permitirá que los clientes realicen reservas en línea y que los administradores del hotel puedan gestionar la disponibilidad de las habitaciones, modificar reservas y generar informes…

4. OBJETIVOS

¿Qué se intenta conseguir con el desarrollo de tu proyecto? Defínelos en infinitivo (al menos

8-10 objetivos).

1. Desarrollar una API REST en Spring Boot para la gestión de reservas hoteleras.
2. Implementar un sistema de autenticación y autorización para administradores y clientes.
3. Permitir a los clientes consultar disponibilidad y realizar reservas en tiempo real.
4. Desarrollar un frontend intuitivo que consuma la API REST.
5. Gestionar cancelaciones y modificaciones de reservas.
6. Implementar un sistema de notificación para confirmar reservas y recordar check-in/check-out.
7. Garantizar la seguridad de los datos de los usuarios y transacciones.
8. Optimizar la API para garantizar un rendimiento eficiente con carga alta.
9. Proporcionar reportes y estadísticas a los administradores sobre la ocupación del hotel.
10. Diseñar la arquitectura del software para futuras ampliaciones e integraciones con otros sistemas.

5. ESTUDIO DE VIABILIDAD.

1. REQUISITOS HW y SW

Qué herramientas vas a utilizar para el desarrollo del mismo ¿Máquinas reales o

virtuales? ¿SW? ¿HW específico? Indica la versión del software a utilizar, si es

gratuito, de evaluación o de pago. En caso de utilización de hardware, ¿qué

requisitos técnicos tiene? ¿Arquitectura (Windows, Linux, Mac, ARM, …)?

* **Software:**
  + Spring Boot 3.x (Backend – API REST)
  + Java 17
  + PostgreSQL o MySQL (gestión de bases de datos)
  + React.js o Angular (Frontend)
  + Docker (para despliegue y gestión de contenedores)
  + Postman (para pruebas de la API)
  + Swagger (para documentación de la API)
  + Git y GitHub (control de versiones)
  + Junit y Mockito (para pruebas unitarias)
* **Hardware:**
  + Servidores con arquitectura Linux (Ubuntu Server o Debian)
  + CPU: 4 cores o superior
  + RAM: 8GB o más
  + Espacio en disco: 50GB SSD (para base de datos y logs)
  + Cloud Hosting (AWS, Azure, o DigitalOcean para el despliegue)

1. TEMPORALIZACIÓN PLANIFICACIÓN

¿Cuándo se va a desarrollar? Planifica las semanas que tienes para analizar las

opciones de mercado, los procesos de instalación, configuración, testeo, desarrollo

de memoria, página web (si existe), preparación de la defensa (presentación,

videotutoriales, o cualquier otro elemento). Ten en cuenta el número de

componentes, las tareas que puedes realizar a la par y las que requieres tener

acabadas para poder continuar, etc. Para ello debes de realizar un diagrama de

Gantt. (Estructuración del cronograma de actividades)

Se planifica el desarrollo en 12 semanas dividiendo las tareas en fases.

| **Semana** | **Tarea** |
| --- | --- |
| 1-2 | Análisis de requisitos y diseño de arquitectura |
| 3-4 | Desarrollo del backend (API REST en Spring Boot) |
| 5-6 | Desarrollo del frontend (React/Angular) |
| 7-8 | Integración API – Frontend |
| 9-10 | Pruebas y depuración |
| 11 | Despliegue en entorno real |
| 12 | Documentación y preparación de la defensa |

Se realizará un diagrama de Gantt para estructurar las actividades detalladamente.

6. INTEGRANTES DEL PROYECTO Y REPARTO DE FUNCIONES

Debe indicarse el nombre y apellido de cada miembro del proyecto y especificar el reparto

de tareas realizado entre los mismos

| **Nombre** | **Funciones** |
| --- | --- |
| Eduardo Merino | Desarrollo del backend (Spring Boot, API REST)  Desarrollo del frontend (React.js o Angular)  Gestion de la base de datos y despliegue  Pruebas, documentación y control de versiones. |