**PROPUESTA DE PROYECTO DE ESTANCIA**

**Fecha: 28/10/2025**

**INFORMACIÓN GENERAL DEL PROYECTO**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| **Programa Educativo:** | | | Ing. En Software | | |
| **Título del proyecto:** | | | Sistema de Gestión de Servicios Operativa de Mantenimiento de Habitaciones y Espacios Comunes SGSOM (BackEnd) | | |
| **Nombre del alumno:** | | | Cruz Flores Juan Leonardo | | |
| **Matrícula:** | 202300097 | | | **Teléfono:** | 9985555000 |
| **Correo electrónico:** | | leonardo.cfjl@gmail.com | | | |
| **Asesor Académico** | | Vaitiare Moreno G. Cantón | | | |
| **Proceso** | | Estancia I | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Empresa donde realizará el proyecto:** | JW MARRIOTT RESORT & SPA | | |
| **Dirección:** | Puerto, Fraccion Hotelera Fh5-C1 Subdelegacion De La  Playita, 23403 San José del Cabo, BCS Dirección | | |
| **Asesor empresarial:** | Ing. Fidel Cruz Lozada | | |
| **Cargo:** | Gerente de Ingenieria y Mantenimiento | | |
| **Teléfono:** | 624-105-6671 Ext 2320 | **Correo:** | [fcruz@grupodiestra.com](mailto:fcruz@grupodiestra.com) |
| **Área donde se realizará el proyecto:** | Departamento de ingeniería y mantenimiento del hotel | | |

**DEFINICIÓN DEL PROYECTO**

1. Planteamiento del Problema: exponer los aspectos, elementos y relaciones del problema.

|  |
| --- |
| El Hotel enfrenta diversos desafíos en la gestión eficiente de incidencias, quejas y mantenimientos de los habitaciones y espacios comunes. Actualmente, las actividades del área se realizan mediante métodos manuales o sistemas desactualizados, lo que provoca pérdida de información, falta de seguimiento y duplicidad de esfuerzos. Estas deficiencias impactan directamente en la operatividad, la coordinación del personal y la satisfacción del huésped. Los problemas identificados incluyen:  Aspectos:   * Gestión Ineficiente: El registro de mantenimientos, averías y quejas se realiza en papel o mediante sistemas no centralizados, lo que ocasiona pérdida de información, inconsistencias y dificultad para acceder al historial de servicios. * Falta de Priorización: No existe un mecanismo que clasifique las incidencias por nivel de urgencia o importancia, lo que genera retrasos y una atención inadecuada de las averías. * Control de Estado Deficiente: La organización carece de un sistema que permita visualizar en tiempo real el estado actual de las habitaciones (ocupado, vacío, en mantenimiento o fuera de servicio), dificultando la coordinación entre áreas. * Falta de Notificaciones: No se cuenta con un sistema automatizado que informe sobre mantenimientos programados, averías o movimientos futuros, afectando la planificación preventiva y la comunicación entre el personal. * Bitácora Desorganizada: No existe un registro centralizado que almacene de forma estructurada las observaciones, resultados de inspección y detalles de mantenimiento de cada cuarto, lo que limita la trazabilidad y el control histórico de las actividades.   Elementos del Sistema:   * Habitaciones/Espacios Comunes: Entidades con estados de ocupación (ocupado/vacío/mantenimiento/fuera de servicio), niveles de prioridad y el registro de sus mantenimientos e inspecciones. * Notificaciones de alertas: Sistema de notificaciones automáticas que avisa sobre mantenimientos programados o pendientes, utilizando un semáforo de prioridad (alta, media, baja). * Mantenimientos con averías o alertas: Registro detallado de cada tipo de mantenimiento (preventivo, correctivo, urgente) con su respectiva prioridad y fecha programada. * Buscador: Herramienta que permitirá filtrar por habitaciones, edificios y espacios comunes, con una carga eficiente y rápida para localizar información específica. * Bitácora Digital: Registro centralizado y cronológico de todas las actividades de mantenimiento e inspección, con observaciones, evidencias, fecha automática y checklist, garantizando trazabilidad y control histórico.   Relaciones:   * Las habitaciones se encuentran asociadas a mantenimientos y averías con prioridad de atención. * Las alertas y notificaciones se adaptan según el nivel de prioridad y el tipo de mantenimiento programado. * Las inspecciones alimentan la bitácora digital, actualizando automáticamente el historial y estado del cuarto. |

1. Definir los objetivos generales y específicos.

|  |
| --- |
| Objetivo General:  Desarrollar un sistema de levantamiento de servicios enfocado en el área de mantenimiento del Hotel, que permita la gestión integral de los servicios de mantenimiento de habitación y espacios comunes del hotel; identificando los estados, los registros del servicio (inspecciones), historial de las actividades a través de la bitácora digital y reforzando las alertas automáticas para dar el servicio correcto al cliente.  Objetivos Específicos:   1. Diseñar una interfaz web intuitiva y responsive que permita la gestión completa de las habitaciones por edificio y los respectivos mantenimientos. 2. Implementar el sistema CRUD (Crear, Leer, Actualizar y Eliminar) para los registros, filtros, búsqueda en tiempo real y edición inline. 3. Desarrollar un sistema de alertas programables con notificaciones push, alertas sonoras y sincronización automática, orientado a los mantenimientos rutinarios o prioritarios. 4. Integrar funcionalidades PWA que permitan la disponibilidad y operación offline del sistema. 5. Implementar un control de estados de habitaciones que permita actualizar en tiempo real las condiciones de “ocupado”, “vacío”, “en mantenimiento” y “fuera de servicio”. 6. Incorporar un registro detallado de mantenimientos, incluyendo responsable, fecha, hora, descripción de actividades, herramientas y materiales utilizados, con la opción de exportar la información a formato Excel. |

1. Establecer los alcances de su proyecto y los entregables finales. (Metas a la que apunta)

|  |
| --- |
| Módulo de Gestión Base   * Sistema CRUD completo para mantenimientos, averías e incidencias. * Interfaz responsive optimizada para móviles y desktop * Filtrado multicriteria y búsqueda en tiempo real * Edición inline con actualización automática * Lazy loading para optimización de rendimiento   Sistema de Alertas   * Alertas programables por fecha y hora específica * Notificaciones push del navegador con fallbacks * Sonidos para las notificaciones * Historial de alertas * Persistencia local de configuraciones de usuario   Progressive Web App (PWA)   * Instalación en dispositivos desktop * Manifest personalizable con branding * Actualizaciones automáticas en segundo plano   Registro de Mantenimientos   * Descripción detallada del mantenimiento * Identificación del responsable (nombre y área) * Fecha y hora exacta de realización * Descripción detallada de la avería y acciones tomadas * Herramientas y recursos utilizados para la solución * Formato exportable a tabla de Excel (CSV) para análisis y reportes * Observaciones, evidencias y fecha automática.   Control de Estados de Habitaciones   * Estados dinámicos: vacío, ocupado, mantenimiento, fuera de servicio * Interfaz visual intuitiva con códigos de color de los mantenimientos * Historial de cambios de estado con timestamp y observaciones * Priorización de mantenimientos   Entregables Finales   * Aplicación Web Completa - Sistema funcional con todas las características implementadas * Base de Datos Optimizada - Estructura completa con datos de prueba y procedimientos almacenados * API REST Completa - Endpoints documentados para todas las funcionalidades * Progressive Web App - PWA instalable con funcionalidades offline * Documentación Técnica - Arquitectura, APIs y guías de desarrollo * Sistema Productivo - Aplicación deployada y configurada en servidor |

1. Metodología de desarrollo a utilizar (justificar el uso de la metodología a usar)

|  |
| --- |
| **Metodología Seleccionada:** Desarrollo Ágil con Scrum Adaptado  **Justificación de la Metodología:**   * Entrega Incremental: Funcionalidades utilizables desde sprints tempranos * Feedback Continuo: Validación constante con usuarios finales * Flexibilidad ante Cambios: Adaptación a nuevos requerimientos durante desarrollo * Riesgo Controlado: Detección temprana de problemas técnicos * Calidad Integrada: Testing y revisión en cada iteración * Transparencia: Visibilidad completa del progreso del proyecto   **Estructura de Sprints:**   * Sprint 0: Análisis y diseño (1 semana) * Sprint 1: Sistema base con PWA y control de estados (4 semanas) * Sprint 2: Alertas, notificaciones e inspecciones (4 semanas) * Sprint 3: Finalización, testing, documentación y deployment (3 semanas)   **Entregables por Sprint:**   * Sprint 0: Documentación técnica y configuración del entorno. * Sprint 1: Sistema base con CRUD, control de estados y PWA funcional. * Sprint 2: Módulo de alertas, notificaciones y sistema de inspecciones. * Sprint 3: Pruebas integrales, documentación de usuario y deployment final. |

1. Cronograma de trabajo

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Cronograma de actividades | | | | | | | | | | | | | | | | | | Etapa | **Actividad** | **Mes1** | | | | **Mes2** | | | | **Mes3** | | | | **Mes4** | | | Sprint 0: Fundación | Análisis de requerimientos detallado | ✔️ | | | |  |  |  |  |  |  |  |  |  |  | | Setup de ambiente de desarrollo | ✔️ | | | |  |  |  |  |  |  |  |  |  |  | | Diseño de arquitectura sistema | ✔️ | | | |  |  |  |  |  |  |  |  |  |  | | Diseño de base de datos | ✔️ | | | |  |  |  |  |  |  |  |  |  |  | |  | Prototipos de interfaz | ✔️ | | | |  |  |  |  |  |  |  |  |  |  | | Sprint 1: Sistema Base | CRUD completo con interfaz responsive |  | | | | ✔️ | | | |  |  |  |  |  |  | | Control de estados de habitaciones (4 niveles) |  | | | | ✔️ | | | |  |  |  |  |  |  | | Filtrado avanzado y búsqueda |  |  |  |  | ✔️ | | | |  |  |  |  |  |  | | PWA funcional |  |  |  |  | ✔️ | | | |  |  |  |  |  |  | |  | Desarrollo del módulo de mantenimientos con formato exportable y rastreabilidad. |  |  |  |  | ✔️ | | | |  |  |  |  |  |  | | Sprint 2: Alertas y Estados | Sistema de alertas programable con historial |  |  |  |  |  |  | |  | ✔️ | | | |  |  | | Sistema de asignación de averías |  |  |  |  |  |  | |  | ✔️ | | | |  |  | | Notificaciones push y audio |  |  |  |  |  |  |  |  | ✔️ | | | |  |  | |  |  |  |  |  |  |  |  |  |  | | | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | |  | | | | Sprint 3: Finalización | Documentación de usuario |  |  |  |  |  |  |  |  |  |  |  | ✔️ | | | | Testing integral del sistema |  |  |  |  |  |  |  |  |  |  |  | ✔️ | | | | Deployment y documentación |  |  |  |  |  |  |  |  |  |  |  | ✔️ | | | |

1. Asignaturas y temas aplicables al proyecto (mínimo 3 asignaturas y 5 temas)

|  |
| --- |
| **1. PROGRAMACIÓN WEB**   1. Desarrollo Frontent: Uso de JavaScript y programación HTML modular. 2. APIs REST Complejas: Endpoints para IA y predicciones 3. WebSockets: Comunicación en tiempo real para estados de habitaciones 4. Progressive Web Apps: Service Workers para IA offline 5. Desarrollo Backend: Uso de php y modulos node para arrancar servidores y funciones avanzadas.   **2. BASES DE DATOS**   1. Diseño de BD Compleja: Normalización para edificios, mantenimientos, incidencias, habitaciones y espacios comunes. 2. Data Warehousing: Almacenamiento histórico de la bitácora. 3. Triggers: Automatización de acciones en la base de datos de forma anónima al código.   **3. PROGRAMACIÓN CLIENTE SERVIDOR**   1. WebSockets Avanzados: Comunicación en tiempo real para las habitaciones y espacios comunes. 2. Comunicación Asíncrona: Implementa Fetch API para peticiones, utilizando Promesas y async/await para manejar respuestas JSON y validación de datos 3. Optimización: Utiliza técnicas de lazy loading, caching, Service Workers y compresión de datos para mejorar el rendimiento y la experiencia del usuario. 4. Seguridad: El sistema implementa headers de seguridad, escape de HTML, validación de datos y manejo de sesiones para proteger la aplicación 5. Capas de protección: Protección de datos sensibles de la BD.   **4. DISEÑO DE INTERFACES**   1. Diseño Responsive: El proyecto implementa un diseño responsive que se adapta a diferentes dispositivos, utilizando layouts flexibles y media queries para asegurar una experiencia de usuario óptima en cualquier dispositivo. 2. Interactividad: Implementa eventos de usuario, animaciones, feedback visual y notificaciones para crear una interfaz interactiva y atractiva para el usuario. 3. Accesibilidad: El sistema sigue principios de accesibilidad con estructura semántica, contraste de colores, navegación por teclado y mensajes de error claros, asegurando que la aplicación sea usable para todos los usuarios. 4. UX: Interfaces intuitivas con acciones cómodas para el usuario. 5. UI: Diseños de interfaz moderno, con estética minimalista y suave. |