# Gestión y planificación de proyectos.

El propósito de este acta es recopilar toda documentación relacionada con el desarrollo del Sistema de Gestión de Habitaciones.

A lo largo de diez secciones se mostrará cómo se utilizaron diversas herramientas y tecnologías para poder lleva a cabo el desarrollo completo del sistema, abarcando desde el alcance, planificación, desarrollo, pruebas e implementación, es decir, a lo largo de todo el ciclo de vida del software.

## Herramientas automáticas de estimación.

Este tipo de herramientas permiten calcular la cantidad de esfuerzo, tiempo y recursos necesarios para poder completar un proyecto.

En el caso del Sistema de Gestión de Habitaciones, al no tener bien definida la cantidad de líneas de código finales, se procedió a utilizar como herramienta de estimación los puntos de función.

Los puntos de función es una técnica de medición utilizada para estimar el tamaño y la complejidad de un sistema de software. Esta técnica se basa en la funcionalidad que el sistema proporciona al usuario, en lugar de enfocarse en las líneas de código o el esfuerzo de desarrollo.

Los puntos de función ayudan a evaluar el esfuerzo y el costo de desarrollo de un proyecto, así como estimar plazos y recursos necesarios.

### Puntos de función – Determinar entradas.

Antes de pasar al análisis del cálculo de complejidad veamos cuales son los componentes clave de

nuestro sistema:

**1.- Entradas externas.**

**Formulario de reservación:** Permite a los usuarios ingresar datos para realizar una reservación,

(nombre, fechas, tipo de habitación, etc.).

Registro del usuario: Permite a los usuarios crear una cuenta en el sistema y acceder a él.

**Formulario de cancelación:** Permite a los usuarios cancelar una reservación.

**2.- Salidas externas.**

**Confirmación de reservación:** Mensaje o correo electrónico que confirma la reservación.

**Informe de disponibilidad:** Muestra las habitaciones disponibles en un rango de fechas.

**Resumen de reservación:** Detalle o consulta de las reservaciones activas.

**3.- Consultas externas.**

**Búsqueda de disponibilidad:** Permite a los usuarios buscar habitaciones disponibles según

criterios específicos (fecha, tipo de habitación).

**Consulta de Información de la habitación:** Muestra características de una habitación

específica.

**4.- Archivos internos del sistema.**

**Base de datos de reservaciones:** Almacena la información de las reservaciones.

**Base de datos de usuarios:** Almacena información de los usuarios.

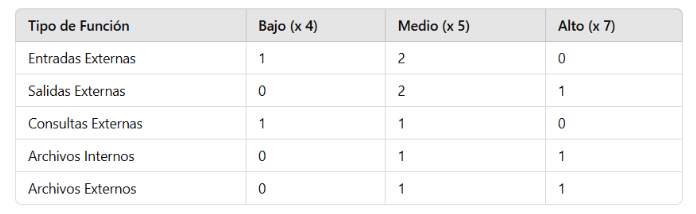
*(Ambas tablas van en una misma base de datos).*

**5.- Archivos externos.**

**Integración con el sistema de facturación:** Accede a información de facturación.

**Integración con sistema de limpieza:** Accede a información sobre el estado de las habitaciones.

Muy bien ahora, veamos de manera gráfica como quedarían las funcionalidades o los puntos de función de nuestro sistema para la gestión de reservaciones de un hotel, donde se categorizan los puntos por medio de su nivel de complejidad.



### Puntos de función – Conclusiones.

Como podemos ver, tenemos un total de 64 puntos de función por lo que la realizar el cálculo

correspondiente, tenemos que, mediante puntos de función, el esfuerzo total es de 640 horas-hombre, tomando en cuenta la siguiente formula; 64 puntos × 10 horas.

## Estimación de costes.

Es claro que para poder emprender un proyecto de esta naturaleza hay que tener bien definidos los costos directos e indirectos que involucran su desarrollo y posterior finalización.

### Costes directos. (Salarios)

Respecto a los costos directos del proyecto, tenemos como personal involucrado;

*Cuatro desarrolladores o programadores, un arquitecto de software, un analista de requerimientos y un proyect manager.*

Cada uno de ellos tiene su propio salario en pesos mexicanos, los cuales se calculan por hora, así que tomando en cuento el nivel de complejidad del proyecto, tenemos un total de 640 horas – hombre.

Salario del desarrollador por hora: 975 pesos MXN.

Salario del proyect manager por hora: 1092 pesos MXN.

Salario del analista de requerimientos por hora: 936 pesos MXN.

Salario del arquitecto de software por hora: 1404 pesos MXN.

Por lo tanto, sólo contando el salario de los desarrolladores, estaríamos hablando de un total de 2496000 pesos MXN.

Sumando los salarios del proyect manager, que es un total de 69880 pesos MXN, con el salario del analista de requerimientos de 598240 pesos MXN y el salario del aquitecto de software con 897600 pesos MXN, tenemos un total referente a los costes directos de 4690720 pesos MXN.

### Costes indirectos.

Los costes indirectos se calcularán a partir del 10% del costo total de los costos directos, por lo que tenemos la siguiente formula; 4690720 pesos MXN \* 0.10 es igual a 469072 pesos MXN.

Estos gastos refieren a gastos de infraestructura, gastos administrativos, herramientas de software general, costos operacionales, seguros e impuestos.

### Coste total.

El costo total del proyecto se obtiene sumando tanto los gastos directos con los gastos indirectos, por lo que tenemos un total de 5159792 pesos MXN.

## Planificación.

El objetivo de planificar el proyecto es poder visualizar cada etapa del mismo, cual podría ser la duración de cada una, los recursos necesarios.

Qué tareas se deben de realizar para finalizar cada fase o etapa del proyecto.

A continuación se detalla un plan que contempla varias fases para completar el Sistema de Gestión de Habitaciones.

### Fase 1 – Análisis de requerimientos.

*Fechas: Del 15/08/2024 al 21/08/2024 (1 semana)*

|  |  |
| --- | --- |
| **Actividad** | **Responsable** |
| Reuniones con el cliente | Project Manager y Analista de Requerimientos |
| Definición de requisitos del sistema | Analista de Requerimientos |
| Definición de requisitos no funcionales | Analista de Requerimientos y Arquitecto de Software |
| Recolección de información necesaria | Analista de Requerimientos |
| Diseño de la arquitectura del sistema | Arquitecto de Software |
| Elaboración de diagramas de flujo y modelos | Arquitecto de Software |

### Fase 2 – Diseño del sistema.

*Fechas: Del 22/08/2024 al 28/08/2024 (1 semana)*

|  |  |
| --- | --- |
| **Actividad** | **Responsable** |
| Diseño detallado del sistema (diagramas, arquitectura, interfaces, etc.) | Arquitecto de Software y Desarrolladores |

### Fase 3 – Desarrollo e integración.

*Fechas: Del 29/08/2024 al 09/10/2024 (5 semanas)*

|  |  |
| --- | --- |
| **Actividad** | **Responsable** |
| Desarrollo del módulo de reservas | Desarrolladores |
| Desarrollo del módulo de cancelaciones | Desarrolladores |
| Desarrollo del módulo para mostrar habitaciones | Desarrolladores |
| Integración de sistemas externos (facturación y limpieza) | Desarrolladores y Arquitecto de Software |

### Fase 4 – Capacitación y pruebas.

*Fechas: Del 10/10/2024 al 11/12/2024 (9 semanas)*

|  |  |
| --- | --- |
| **Actividad** | **Responsable** |
| Capacitación del personal en el sistema | Project Manager y Desarrolladores |
| Pruebas unitarias | Desarrolladores |
| Pruebas de integración | Desarrolladores y Arquitecto de Software |

### Fase 5 – Implementación final.

*Fechas: Del 12/12/2024 al 19/12/2024 (1 semana)*

|  |  |
| --- | --- |
| **Actividad** | **Responsable** |
| Implementación del sistema en el entorno real | Desarrolladores y Project Manager |
| Revisión y ajustes post-implementación | Desarrolladores y Project Manager |

## Planificación de recursos.

El objetivo de realizar una planificación de los recursos es poder identificar los recursos que se utilizarán en cada fase del proyecto hasta su finalización.

Los recursos implican recursos humanos, materiales y técnicos.

A continuación se presentan todo los recursos requeridos para cada fase del proyecto.

### Fase 1 – Análisis de requerimientos.

**Responsables:** Analista de Requerimientos (principal), Arquitecto de Software, Project Manager.

**Recurso, herramientas:** Visual Studio Code para documentación, PgAdmin para consultas iniciales si hay datos previos.

Hardware: 2 máquinas.

**Actividades:** Recopilación de requisitos con el cliente, análisis de factibilidad, documentación detallada de requisitos.

### 

### Fase 2 – Diseño del sistema.

**Responsables:** Arquitecto de Software (principal), Project Manager, Analista de Requerimientos.

**Recursos, herramientas:** Visual Studio Code, herramientas de diseño de software, PyQt 6 y Qt Designer.

**Hardware:** 2 máquinas.

**Actividades:** Diseño de arquitectura y bases de datos en PostgreSQL, especificaciones de interfaces (PyQt para GUI), revisión y aprobación del diseño.

### Fase 3 – Desarrollo e integración.

**Responsables:** 4 Desarrolladores, Arquitecto de Software, Project Manager.

**Recursos, herramientas:** Visual Studio Code, Python 3, PostgreSQL, PgAdmin, librerías de PostgreSQL, PyQt.

**Hardware:** 7 máquinas de trabajo y 1 para el sistema de pruebas.

**Actividades:** Codificación de módulos principale, integración con la base de datos, desarrollo de la interfaz gráfica, integración continua y revisiones periódicas.

### Fase 4 – Capacitación y pruebas.

**Responsables:** Project Manager, Analista de Requerimientos.

**Recursos, herramientas:** Entorno de pruebas en PostgreSQL, PyQt para interfaz.

**Hardware:** 7 máquinas de trabajo.

**Actividades:** Pruebas de integración y unitarias, capacitación para usuarios clave y simulación de casos de uso.

### Fase 5 – Implementación final.

**Responsables:** Todo el equipo (coordinado por el Project Manager).

**Recursos, herramientas:** PostgreSQL en el servidor de producción.Hardware: 1 máquina para alojar el sistema.

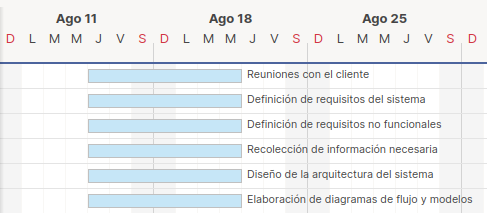
**Actividades:** Instalación del sistema en el entorno final, monitoreo de la implementación inicial, documentación de la entrega final.

## Planificación temporal.

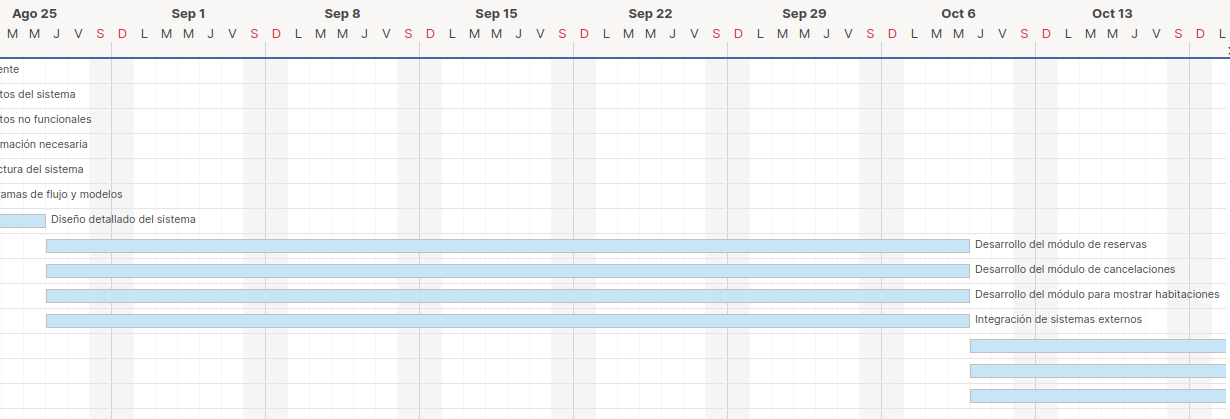
Para poder desarrollar nuestro diagrama de Gantt utilizamos la plataforma en linea de Smatsheet, en la que podemos básicamente crear un cronograma y visualizarlo de manera más sencilla y entendible.

Veamos cada una de las fases en sus lapsos de tiempo determinados:

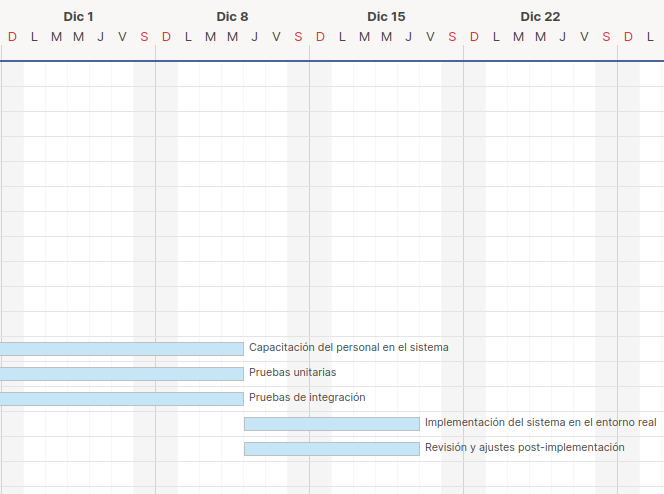
### Primera fase.



### Segunda y tercera fase.



### Cuarta y quinta fase.



## Equipo de desarrollo.

Los roles del equipo de desarrollo son básicamente los integrantes que lo componen, en este caso tenemos cuatro desarrolladores, un proyect manager, un analista de requerimientos y un arquitecto de software.

A continuación veremos cuales son las actividades o responsabilidades de cada uno de ellos;

****Project Manager**:** Encargado de coordinar el proyecto, gestionar el tiempo y los recursos, y garantizar que se cumplan los objetivos.

****Analista de Requerimientos**:** Responsable de trabajar con el cliente para recopilar los requisitos y asegurar que se cumplan las expectativas del sistema.

****Arquitecto de Software**:** Diseña la arquitectura del sistema, toma decisiones técnicas clave y asegura que las soluciones sean escalables y eficientes.

****Desarrolladores**:** Son los encargados de programar los módulos del sistema, realizar la integración con sistemas externos y realizar las pruebas unitarias e integradas.

Incluye conocimientos de programación orientada a objetos en Python, conocimientos en sistemas gestores de bases de datos con tecnologías como Postgresql y su administración. Conocimientos en el desarrollo front end con PyQt y QtDesigner.

## Perfil del planificador.

**Nombre del puesto:** Planificador, gerente de proyecto (Luis Alberto Reyes Cruz).

**Descripción del puesto:**

El planificador es el encargado de liderar el equipo y gestionar los recursos, tiempo, y presupuesto del proyecto. Su función principal es coordinar a los distintos actores involucrados y asegurar que cada fase del proyecto se cumpla conforme al cronograma establecido.

Para esto, se enfoca en la supervisión de tareas, gestión de riesgos, y la comunicación eficiente entre el equipo y los stakeholders.

**Habilidades Requeridas:**

* **Liderazgo y motivación:** Capacidad para influir positivamente en el equipo y motivarlos a alcanzar las metas establecidas.
* **Comunicación eficaz:** Habilidad para transmitir ideas y estrategias de forma clara, mantener informado al equipo y a los stakeholders.
* **Gestión del tiempo:** Organización del cronograma para que las tareas se realicen dentro de los plazos establecidos.
* **Capacidad de análisis y resolución de problemas:** Identificación de problemas potenciales y toma de decisiones basada en datos y el contexto del proyecto.
* **Adaptabilidad y flexibilidad:** Capacidad para ajustar el proyecto ante cambios o imprevistos sin comprometer los objetivos.

**Responsabilidades:**

* **Desarrollo del cronograma y asignación de recursos:** Crear un cronograma detallado que incluya todas las tareas del proyecto, tiempos y recursos necesarios.
* **Supervisión y control del progreso:** Monitorear continuamente el estado de cada fase del proyecto, tomando acciones correctivas si es necesario.
* **Gestión de riesgos:** Identificar riesgos potenciales y planificar estrategias para mitigarlos.
* **Comunicación de avances y resultados:** Realizar reportes regulares sobre el estado del proyecto y comunicar a los stakeholders de manera efectiva.
* **Control presupuestario:** Administrar el presupuesto, asegurándose de que los costos se mantengan dentro del límite establecido.

**Metodología de Supervisión y Manejo de Problemas:**

* **Supervisión:** El planificador utiliza herramientas de gestión de proyectos, como Microsoft Project o JIRA, para dar seguimiento detallado a cada actividad y medir el progreso.
* **Gestión de problemas:** En caso de que surja un problema, el planificador lleva a cabo una evaluación rápida, define una estrategia para abordarlo y asegura la participación del equipo en la resolución.

**Impacto en el proyecto:**

Un planificador con estas habilidades y responsabilidades asegura que el proyecto avance sin demoras significativas, minimizando costos adicionales y evitando desviaciones en los objetivos. Su capacidad de adaptación y resolución de problemas es clave para mantener el proyecto en el buen camino, incluso ante circunstancias inesperadas.

## Errores típicos.

Dentro de los equipos no sólo de desarrollo siempre existen problemas o inconvenientes que pueden retrasar la entrega de tareas e incluso demorar la entrega final del proyecto total, por lo que es vital, podes acatar dichos conflictos de raíz en pro de mejorar la comunicación del equipo, la sinergia y la productividad.

### Plan de acción.

Poder aumentar la calidad de la comunicación entre los programadores en pro de mejorar los procesos de desarrollo haciéndolos más eficientes, rápidos y modulares.

Esto no sólo para evitar retrasos en los planes de entrega sino para optimizar el código respecto a su calidad y consistencia.

**1.- Canales de comunicación: Discord.**

Se utilizará Discord para realizar reuniones periódicas para discutir temas generales como soporte, retroalimentación, revisión de código, manejo de versiones, etc.

**2.- Documentación compartida.**

Mediante Teams, se definirá el espacio para poder almacenar toda la documentación del proyecto.

Esencialmente, el tipo de arquitectura, la metodología de desarrollo y claro, las practicas involucradas para llevar a cabo la codificación.

**3.- Revisiones de código.**

Se realizarán revisiones de código obligatorias con el fin de asegurar la calidad y compartir el conocimiento de los integrantes del equipo. La plataforma elegida será Github y Visual Studio Code en caso de trabajar de modo presencial.

**4.- Parejas de programación.**

Se implementará un programa de programación en parejas en pro de poder mejorar la calidad del código fomentando no sólo la comunicación y fomentar el aprendizaje ente los desarrolladores.

**5.- Comunicación más detallada respecto a los commits.**

Las descripciones deben de ser ampliadas cada que un equipo de desarrollo realice un commit, esto para poder seguir de manera correcta el hilo del proyecto y el hilo del desarrollo en todas sus fases.

**6.- Retroalimentación colectiva.**

Los miembros del equipo deben de sentirse cómodos al momento de poder externar sus problemas o preocupaciones, siempre de manera respetuosa.

Las respuestas deben de ser siempre constructivas, en pro de la mejora continua, el aprendizaje y la corrección de los problemas presentados.

**7.- Retrospectivas.**

Al final del ciclo, se deben realizar retrospectivas para discutir qué funcionó bien y qué se puede mejorar en términos de comunicación y trabajo en equipo.

**8.- Celebración de logros y avances.**

Cuando una entrega se da a tiempo, es decir que una tarea se completa con éxito, se debe de reconocer y celebrar estos logros para poder generar un ambiente positivo y fomenta el trabajo en equipo.

## Documentación – Plan de proyecto.

Mediante el siguiente anexo, se puede acceder al documento que recopila toda la información relacionada con el proyecto para la gestión de reservaciones de habitaciones en hoteles.

Contiene lo relacionado a los objetivos, requerimientos, cronogramas, recursos, costos, etc.