PLAN DE DESARROLLO E

IMPLEMENTACIÓN DEL SOFTWARE

**Curso:** Administración de Proyectos de Software

**Sesión:** 07

**Docente:** Ing. José Vásquez Pereyra

# Introducción

Este documento detalla las actividades planificadas para el desarrollo y la implementación del software. Incluye fases, tareas, cronograma, responsables, estrategias de despliegue y consideraciones técnicas y operativas para su ejecución.

# Objetivo del Plan

# El objetivo de este plan es definir la estrategia técnica para el desarrollo e implementación del sistema web con chatbot, orientado a mejorar la experiencia de los huéspedes en el Hotel Casa Andina Premium San Isidro.

# Este documento establece las directrices para construir, revisar y probar el sistema, garantizando su correcto funcionamiento antes de su puesta en marcha. Se detallan las herramientas que se utilizarán, los recursos disponibles y los tiempos estimados para cada etapa, asegurando que la solución sea accesible desde cualquier navegador web y cumpla con los estándares de calidad requeridos.

# Asimismo, se especificará cómo se realizará la implementación en un entorno de pruebas y, posteriormente, su despliegue en producción.

# Enfoque de Desarrollo

# Se adoptará el marco de trabajo Scrum, un enfoque ágil que permite desarrollar el sistema de manera iterativa e incremental, a través de módulos funcionales entregables en ciclos cortos (sprints). Este método facilita la adaptación a cambios y la retroalimentación continua, mejorando la calidad del producto final.

# Para el desarrollo se utilizarán herramientas de código abierto, entre ellas:

# Visual Studio Code: como entorno de desarrollo integrado (IDE).

# GitHub: para el control de versiones, gestión de ramas y colaboración en el código fuente.

# El flujo de trabajo propuesto seguirá las siguientes etapas por cada incremento funcional:

1. **Planificación**: definición de objetivos del sprint y asignación de tareas.
2. **Desarrollo**: implementación de funcionalidades conforme a los requerimientos.
3. **Revisión**: análisis del código y evaluación entre pares.
4. **Integración**: unión de los módulos desarrollados al sistema principal.
5. **Pruebas**: verificación funcional y detección de errores.
6. **Validación**: aprobación del incremento por parte del cliente o equipo evaluador.

# Fases del Desarrollo

|  |  |  |
| --- | --- | --- |
| **Fase del Desarrollo** | **Actividades Clave** | **Entregables** |
| **Análisis de Requisitos** | - Reunión con el cliente  - Identificación y documentación de requerimientos funcionales y no funcionales | - Especificación de Requisitos del Software (SRS) |
| **Diseño del Sistema** | - Elaboración del diseño arquitectónico  - Creación de prototipos y mockups  - Diseño de base de datos y reportes | - Diseño Arquitectónico  - Prototipo funcional / Mockups  - Diseño de Base de Datos y Reportes |
| **Planificación del Desarrollo** | - Definición de cronograma y asignación de tareas  - Selección de herramientas y tecnologías  - Establecimiento del flujo de trabajo | - Plan de Desarrollo e Implementación |
| **Codificación e Integración** | - Desarrollo de módulos según los requisitos  - Control de versiones y pruebas unitarias  - Integración continua del sistema | - Repositorio de código fuente actualizado |
| **Pruebas y Validación** | - Ejecución de pruebas funcionales, de integración y de aceptación  - Registro de resultados y corrección de errores | - Registro de Pruebas (plan de pruebas + evidencias) |
| **Documentación y Capacitación** | - Elaboración de manuales de usuario y técnico  - Capacitación al personal responsable del uso y mantenimiento del sistema | - Manual del Usuario  - Manual Técnico |
| **Implementación y Despliegue** | - Instalación del sistema en entorno de pruebas  - Validación final con el cliente  - Puesta en producción | - Sistema operativo en entorno de producción  - Documentación técnica final |

# Plan de Implementación

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea de Implementación** | **Responsable** | **Entorno Objetivo** | **Fecha Estimada** |
| Configuración del entorno de pruebas (hosting y base de datos simulada) | Programador/Tester | Pruebas | Semana 10 |
| Despliegue del sistema web con chatbot | Programador/Tester | Pruebas | Semana 12 |
| Pruebas de aceptación y validación del sistema | Encargado de Calidad | Pruebas | Semana 13 |
| Documentación de instalación y uso | Analista/Diseñador | Pruebas | Semana 14 |
| Revisión y retroalimentación final | Cliente | Pruebas | Semana 14 |
| Entrega final y cierre del proyecto | Director de Proyecto | Pruebas | Semana 15 |

# Plan de Control de Versiones

Para la gestión de versiones del software se utilizará **Git** como sistema de control de versiones distribuido, con un **repositorio central alojado en GitHub**. Esto permitirá llevar un historial claro de los cambios, facilitar la colaboración entre desarrolladores y asegurar la trazabilidad del código fuente.

El flujo de versiones seguirá esta estructura:

* **Rama principal (main)**: versión estable.
* **Rama de desarrollo (develop)**: integración de nuevas funciones.
* **Ramas de características (feature/)**: desarrollo de módulos específicos.

# Estrategia de Despliegue

Se adoptará una estrategia de despliegue del tipo Big Bang, en la cual todo el sistema será implementado de forma simultánea en un entorno de pruebas. Esta modalidad es adecuada dado que se trata de un sistema nuevo que no reemplaza uno anterior, lo que minimiza el riesgo de interrupciones operativas.

Para asegurar una transición segura y controlada hacia el nuevo sistema, se contemplan las siguientes acciones:

* **Validación previa**: Antes del despliegue, se realizará una validación funcional completa del sistema para asegurar que cumple con todos los requerimientos y no presenta errores críticos.
* **Documentación técnica**: Se elaborará documentación detallada con los procedimientos de instalación, configuración y uso del sistema.

# Gestión de Cambios en Producción

Aunque el sistema no será implementado en un entorno de producción real, se establecerá un procedimiento formal para gestionar los cambios posteriores al despliegue en el entorno de pruebas, con el objetivo de mantener el control, trazabilidad y calidad del software.

El proceso para la gestión de cambios será el siguiente:

1. **Solicitud de cambio**El cliente o usuario autorizado presenta la solicitud formal de modificación al Director de Proyecto, especificando el motivo, alcance y urgencia del cambio.
2. **Evaluación técnica**El Analista/Diseñador y el Programador/Tester analizan la viabilidad del cambio, su impacto en el sistema y el tiempo estimado de implementación.
3. **Autorización**El Director de Proyecto revisa la evaluación técnica y decide aprobar o rechazar la solicitud de cambio, considerando prioridades, riesgos y recursos disponibles.
4. **Implementación**Una vez aprobado, el Programador/Tester aplica el cambio en el entorno de pruebas, siguiendo los estándares de desarrollo definidos.
5. **Validación** El Encargado de Calidad verifica que el cambio haya sido implementado correctamente y que no afecte negativamente otras funcionalidades del sistema.
6. **Comunicación y documentación**El Director de Proyecto informa al equipo sobre el cambio realizado, y se actualiza la documentación técnica y funcional correspondiente.

Este procedimiento asegura que todos los cambios post-despliegue estén debidamente controlados, validados y comunicados, evitando inconsistencias y manteniendo la estabilidad del sistema.

1. **Aprobaciones y Validaciones del Documento**

Director del Proyecto

Nombre: Martinez Rodriguez, Nilver Jefferson

Diagrama

El contenido generado por IA puede ser incorrecto.Firma:

Fecha: 12/05/2025

Responsable de calidad

Nombre: Sánchez Saavedra, Brayan José

Imagen que contiene Carta

El contenido generado por IA puede ser incorrecto.Firma:

Fecha: 12/05/2025

1. **Conclusiones**

El presente documento establece una hoja de ruta clara para el desarrollo, despliegue y control del sistema web con chatbot para el hotel Casa Andina Premium San Isidro. Se definieron las fases, responsables, entregables, estrategias de despliegue y mecanismos de control de cambios, lo que garantiza una gestión organizada, trazable y alineada con los objetivos del proyecto.

1. **Aprendizaje del Equipo**

El equipo comprendió la importancia de una planificación estructurada para evitar improvisaciones. Definir roles, fases y procedimientos facilita una implementación ordenada y reduce riesgos de errores críticos. Además, el uso de control de versiones y una estrategia de despliegue clara asegura un flujo de trabajo eficiente y una comunicación efectiva entre los involucrados.