



## PROPUESTA TRABAJO FINAL DE ASIGNATURA SOFTWARE ENGINEERING MANAGEMENT

### Integrantes:

- Juan Carlos Armijos
- Luis Alfredo Jumbo

**Fecha de entrega:** 07/11/2022

**Tema:** Sistema de control de reservas y cobros en un hotel

### Objetivo general

- Diseñar e implementar un sistema web de uso genérico que permita realizar reservas de habitaciones y efectuar cobros al huésped por los servicios brindados por el hotel.

### Objetivos específicos

- Desarrollar un módulo que permita la autenticación y autorización de usuarios dentro del sistema hotelero.
- Desarrollar un módulo que permita listar las habitaciones disponibles con su respectivo precio, tipo y características, en donde el huésped pueda reservarlas según sus necesidades.
- Desarrollar un módulo que permita a la recepcionista aprobar o rechazar la solicitud de reservación del huésped para las habitaciones solicitadas, realizando así, el cobro de su costo correspondiente y costos adicionales de los servicios brindados por el hotel.
- Desarrollar un módulo que permita al administrador gestionar las cuentas de usuario (crear, actualizar, eliminar) del sistema hotelero, teniendo en consideración roles y niveles de autorización (huésped y recepcionista).
- Desarrollar un módulo que permita al administrador gestionar la información general del hotel (nombre, dirección, contactos, características del hotel, formas de pago).
- Desarrollar un módulo que permita al administrador observar un reporte de las reservaciones y cobros realizados (historial).

### Alcance

- El sistema puede ser utilizado por cualquier hotelera que busque controlar las reservas de sus habitaciones y gestionar los cobros, tanto de las reservas como de los costos adicionales.
- El sistema será soportado por cualquier navegador web con acceso a internet.
- El sistema permitirá validar y autorizar las cuentas de los usuarios según se haya especificado al momento de su creación.
- El huésped podrá reservar las habitaciones disponibles según su necesidad cuando se cancele su costo. Además, podrá hacer usos de los servicios brindados por el hotel, generando así costos adicionales.
- La recepcionista podrá aceptar o rechazar las solicitudes para la reservación de las habitaciones según la lista de habitaciones disponibles que se tenga.
- El administrador podrá gestionar (crear, actualizar, eliminar) las cuentas de usuario existentes en el sistema, manteniendo un control sobre los distintos niveles de autorización.
- El administrador podrá gestionar (crear, actualizar, eliminar) la información general del hotel, detallando: (nombre, dirección, contactos, características del hotel, formas de pago).
- El administrador podrá observar un reporte de todas las reservaciones y cobros realizados hasta el momento (historial).

### Tecnologías (Lenguajes, metodologías, plataformas)

- JavaScript.
- React.
- Firebase.
- GitLab.
- Fork.

- Postman.
- Node JS.
- Metodología iterativo incremental.
- Visual Studio Code.
- Navegador Web.

### Metodología

La metodología a implementar es la metodología iterativo incremental donde se realizan un conjunto de tareas en etapas pequeñas de forma repetida dónde consiste en ir complejizando y perfeccionando ese producto en cada una de sus entregas (incrementarlo) a la luz del feedback que se obtiene de su testeo (iterarlo) [1].

Se planifica un proyecto en distintos bloques temporales que se le denomina iteración. En una iteración se repite un determinado proceso de trabajo que brinda un resultado más completo para un producto final, de forma de que quien lo utilice reciba beneficios de este proyecto de manera creciente.

Para llegar a lograr esto, cada requerimiento debe tener un completo desarrollo en una única iteración que debe de incluir pruebas y una documentación para que el equipo pueda cumplir con todos los objetivos que sean necesarios y esté listo para ser dado al cliente. Así se evita tener riesgosas actividades en el proyecto finalizado.

Lo que se busca es que en cada iteración los componentes logren evolucionar el producto dependiendo de los completados de las iteraciones antecesoras, agregando más opciones de requisitos y logrando así un mejoramiento mucho más completo [2].

- **Fases [3].**
  - **Planificación del plan de iteraciones:** En esta fase se planifica un período estrictamente ligado el uno a continuación del otro; donde se hace constar cada una de las iteraciones que cronológicamente serán desarrolladas
  - **Análisis:** En esta fase se detalla las necesidades que debe llegar a satisfacer cada uno de los módulos en base a las necesidades planteadas por el usuario al inicio de la planificación del plan de iteraciones, al culminar el diseño e integrar cada módulo que previamente se estableció se debe tener en cuenta el tiempo estimado que no debe sobrepasar de un período máximo de 30 días.
  - **Diseño:** En esta fase se especifica el diseño de cada módulo (iteración) para este caso mostraremos el diseño de los diagramas de la base de datos así como los diagramas de clase y actividad, además de la especificación de los diagramas de arquitectura del sistema.
  - **Implementación:** Esta fase consiste en realizar el diseño de acuerdo al plan de iteraciones establecidas al inicio del proyecto se realizará el diseño en la plataforma de programación o específicamente el diseño y codificación en el lenguaje de programación que se vaya a desarrollar.
  - **Pruebas:** En esta fase es esencial el riguroso control sobre la programación y los eventos aquí realizados ya que de esto depende mucho verificar si lo codificado cumple con lo requerido por el usuario ya que los datos que han sido utilizados por los desarrolladores deben ser acordes a lo planteado para el desarrollo de la iteración en el periodo que se le ha otorgado.
  - **Integración:** En esta fase el cumplimiento del funcionamiento del diseño de los módulos en cada una de las iteraciones deben ser estrictamente funcionales, es aquí donde se completa el desarrollo por módulos y se los integra al sistema una vez realizado esto el sistema debe ser capaz de funcionar adecuadamente y responder eficientemente a las tareas que el usuario requiera asignar como tareas para verificar su funcionamiento y que cumpla con las expectativas que se plantearon al inicio del proyecto de desarrollo de la aplicación.
- **Roles [4].**
  - **Cliente:** Es la persona que solicita el desarrollo del sistema en base a sus necesidades. En este caso, al ser un software genérico, no existe un cliente como tal, por tanto, dicho cliente será el propio equipo involucrado en el desarrollo del mismo.
  - **Gerente de proyecto:** Se trata de la persona responsable de gerenciar el proyecto de desarrollo, siendo el principal responsable de negociar con el cliente antes, durante y después de ser contratados para desarrollar un programa. El gerente de proyecto también es responsable de establecer y hacer cumplir los plazos de tiempo, así como también debe regular el presupuesto y tomar a cabo las decisiones necesarias para solventar problemas que surjan en el proceso.
  - **Analista de requerimientos:** Es una de las personas más importantes durante la fase de planificación y análisis de un nuevo software, ya que se encarga de documentar, precisar establecer todas las funcionalidades que debe tener el software a desarrollar según las necesidades y consentimiento del cliente. El analista debe presentar toda esta información de forma lógica, minuciosa y entendible

para el resto del equipo de desarrollo.

- **Desarrollador:** El desarrollador es el responsable de escribir y confeccionar el código del software, así como es la persona encargada de llevar a cabo el mantenimiento y depuración del mismo.
- **Testeador:** Establece y lleva a cabo las pruebas de control de calidad y detección de errores dentro del software. El testeador es responsable de probar el producto según el uso que le darán los usuarios al software y además, es quien realiza un informe con los errores que deben ser depurados antes de que el producto pase a la fase de despliegue y mantenimiento.
- **Arquitecto:** El arquitecto es responsable de establecer la estructura del código, el lenguaje de programación a utilizar y además es quien gestiona los diferentes aspectos relacionados al desarrollo del software que no tienen relación directa con la funcionalidad del mismo.

## Arquitectura de software

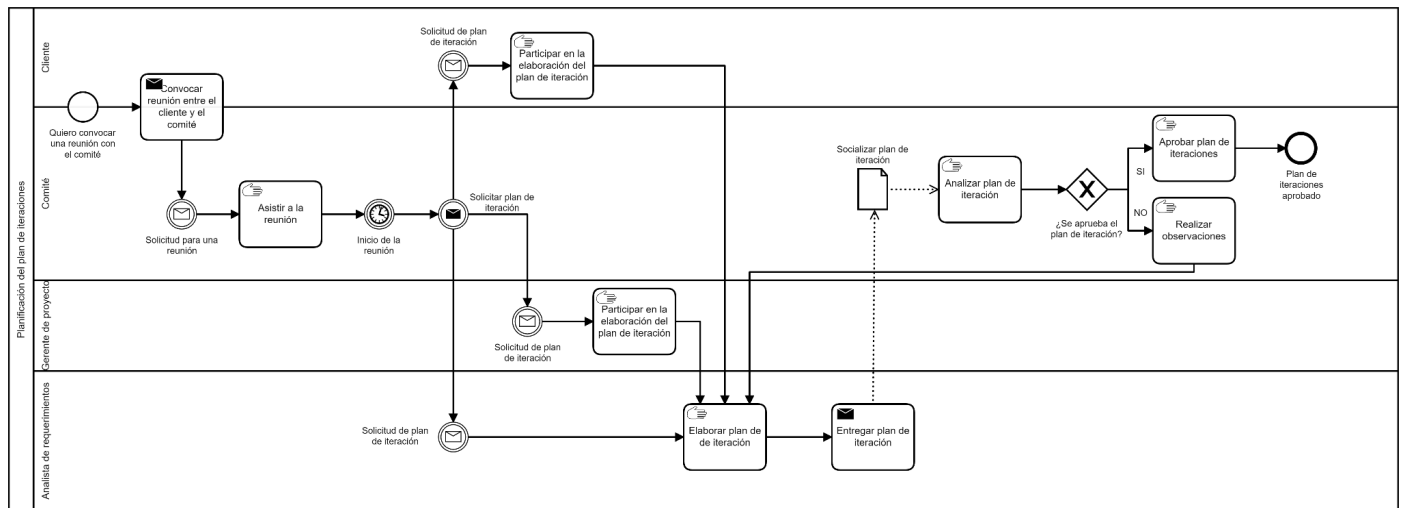
La arquitectura a utilizar para el sistema es la arquitectura de tres capas donde la programación por capas tiene como objetivo la separación de la lógica de negocios de la lógica de diseño, se podría tomar como el patrón de diseño Modelo Vista Controlador (MVC).

## Ventajas de la arquitectura tres capas [5].

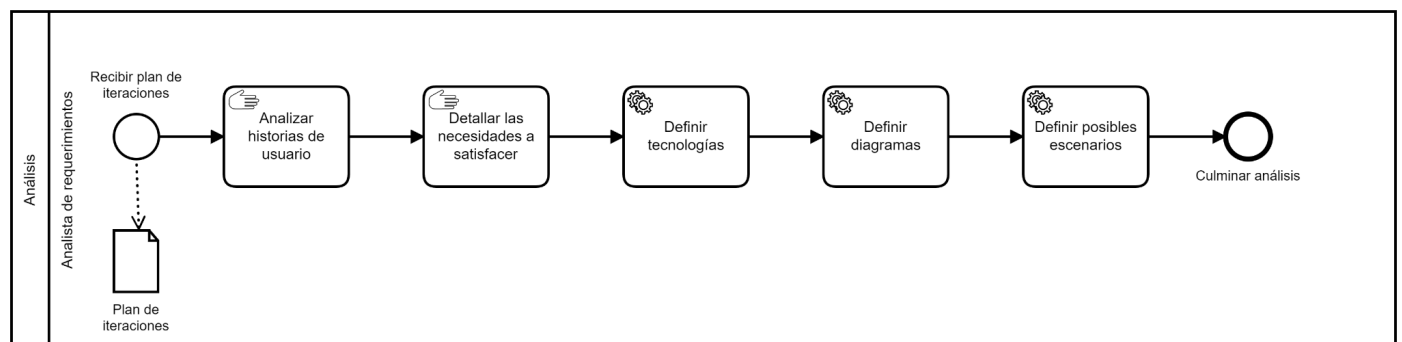
- Simplifica la comprensión y la organización del desarrollo de sistemas complejos
- Reduce las dependencias de forma que las capas más bajas no son conscientes de ningún detalle de las superiores
- Esta separación añade una enorme flexibilidad al diseño de la aplicación.

## BPMN

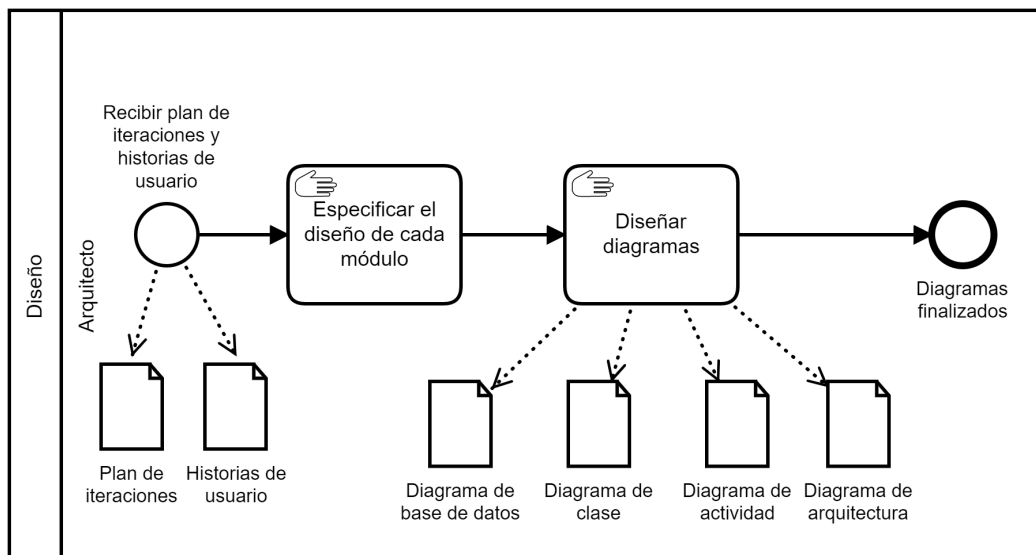
### Planificación



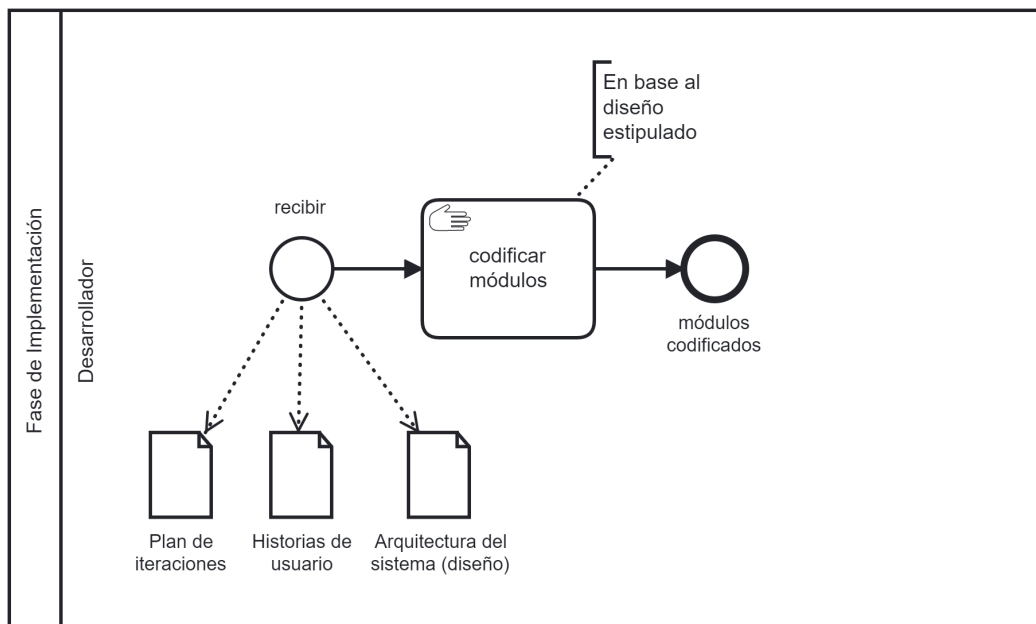
### Análisis



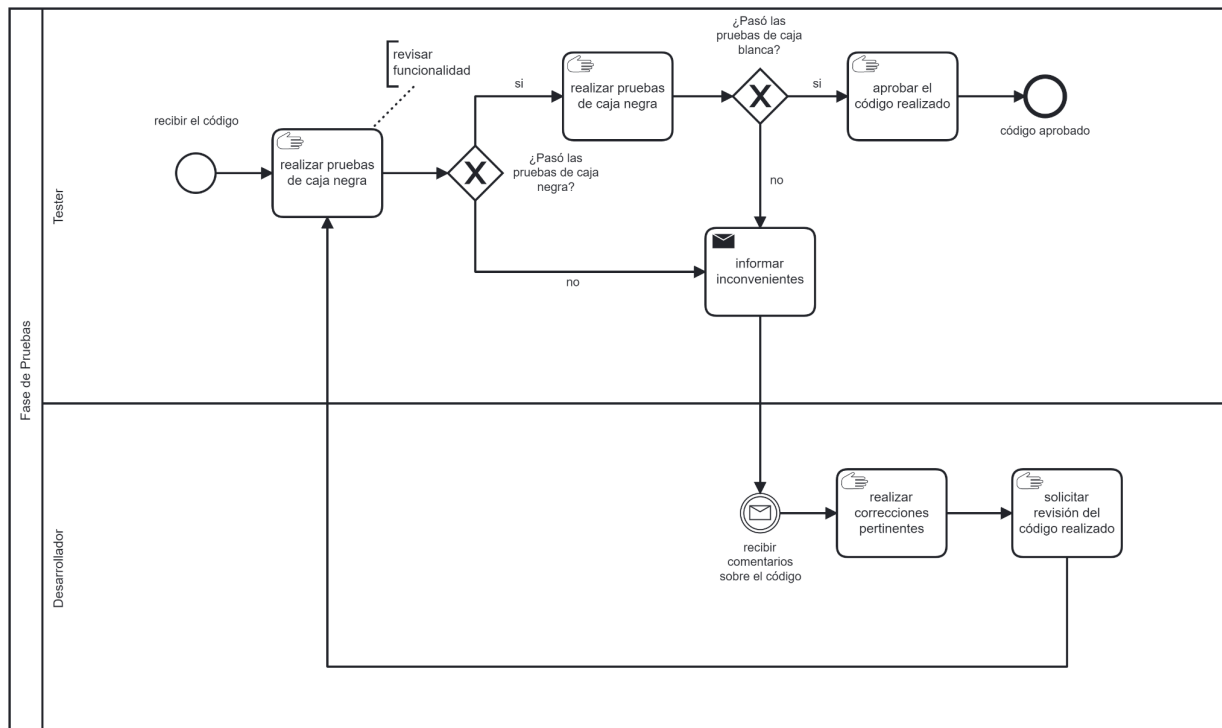
## Diseño



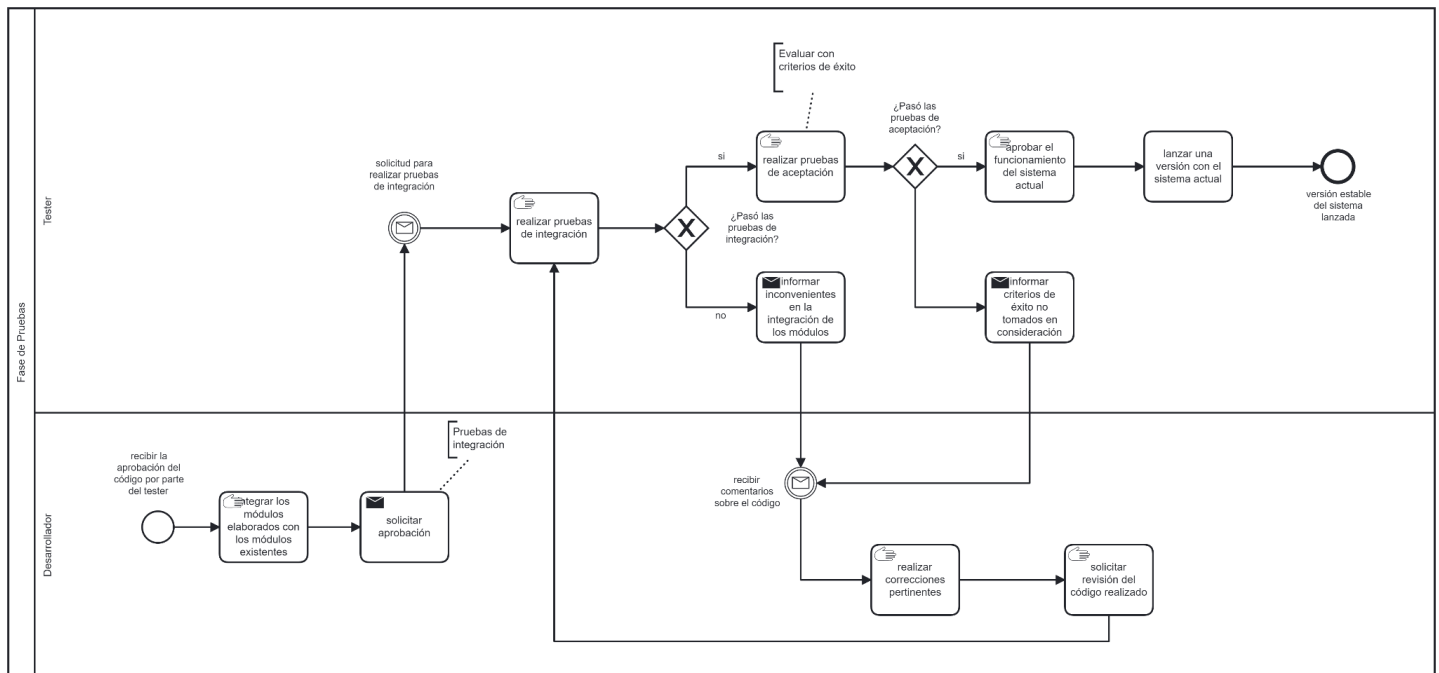
## Implementación



## Pruebas



## Integración



## Bibliografía.

- [1] . Runroom. (2022). "Desarrollo iterativo incremental". [En línea]. Disponible en: <https://www.runroom.com/realworld/desarrollo-iterativo-incremental>
- [2]. Proyectos Ágiles.(2017). "Desarrollo iterativo e incremental". [En línea]. Disponible en: <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- [3].M. Lema, J. Ortiz (2016). "DESARROLLO DE UN SISTEMA DE GESTIÓN INTEGRADO UTILIZANDO SOFTWARE LIBRE CON EL MODELO ITERATIVO INCREMENTAL PARA LLEVAR EL CONTROL DE LOS PROCESOS EN LA EMPRESA SOFTWARE Y HARDWARE". [En línea]. Disponible en: <http://repositorio.utc.edu.ec/bitstream/27000/3685/1/T-UTC-000036.pdf>
- [4]. ConceptoABC. (2022). "Ciclo de vida del software". [En línea]. Disponible en: [https://conceptoabc.com/ciclo-de-vida-del-software/#Desarrollo\\_iterativo\\_y\\_creciente](https://conceptoabc.com/ciclo-de-vida-del-software/#Desarrollo_iterativo_y_creciente)
- [5]. D. Peña. (21, Abril, 2014). "Tipos de Arquitecturas de Software". [En línea]. Disponible en: <https://prezi.com/2uraxobilh0t/tipos-de-arquitecturas-de-software/>