Texto, Logotipo

Descripción generada automáticamente

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto Sistema Web Integrado con Chat Bot para la Recomendación de Componentes de Hardware Basado en el Presupuesto del Usuario.**

Curso: *Construcción de Software I*

Docente: Ing. Ricardo Eduardo Valcarcel Alvarado

|  |  |
| --- | --- |
| Integrantes: |  |
| ***Ccalli Chata, Joel Robert*** | ***(2017057528)*** |
| ***Poma Chura, Jhon Romario*** | ***(2019064022)*** |

**Tacna – Perú**

***2024***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | JCC | JRPC | JRPC | 30/10/2024 | Versión Original |

**Sistema Web Para la Recomendación de Componentes de Hardware Basado en el Presupuesto del Usuario Documento de Arquitectura de Software**

**Versión 1.0**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | JRCC JRPC | JRCC JRPC | JRCC | 21/09/2024 | Versión Original |

ÍNDICE GENERAL

1. [INTRODUCCIÓN 4](#_bookmark0)
   1. [Propósito (Diagrama 4+1) 4](#_bookmark1)
   2. [Alcance 4](#_bookmark2)
   3. [Definición, siglas y abreviaturas 4](#_bookmark3)
   4. [Organización del documento 5](#_bookmark4)
2. [OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS 5](#_bookmark5)

[Requerimientos Funcionales 5](#_bookmark6)

[1.1.1. Requerimientos No Funcionales – Atributos de Calidad 7](#_bookmark7)

1. [REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 7](#_bookmark8)
   1. [Vista de Caso de uso 7](#_bookmark9)

[1.1.1. Diagramas de Casos de uso 9](#_bookmark10)

* 1. [Vista Lógica 9](#_bookmark11)
     1. [Diagrama de Subsistemas (paquetes) 10](#_bookmark12)
     2. [Diagrama de Secuencia (vista de diseño) 10](#_bookmark13)
     3. [Diagrama de Colaboración (vista de diseño) 13](#_bookmark14)
     4. [Diagrama de Objetos 14](#_bookmark15)
     5. [Diagrama de Clases 14](#_bookmark16)
     6. [Diagrama de Base de datos (relacional o no relacional) 15](#_bookmark17)
  2. [Vista de Implementación (vista de desarrollo) 15](#_bookmark18)
     1. [Diagrama de arquitectura software (paquetes) 15](#_bookmark19)
     2. [Diagrama de arquitectura del sistema (Diagrama de componentes) 16](#_bookmark20)
  3. [Vista de procesos 16](#_bookmark21)
     1. [Diagrama de Procesos del sistema (diagrama de actividad) 16](#_bookmark22)
  4. [Vista de Despliegue (vista física) 16](#_bookmark23)
     1. [Diagrama de despliegue 16](#_bookmark24)

1. [Diagrama de Arquitectura 17](#_bookmark25)
2. [Diagrama de Componentes 18](#_bookmark26)
3. [ATRIBUTOS DE CALIDAD DEL SOFTWARE 18](#_bookmark27)

[Escenario de Funcionalidad 19](#_bookmark28)

[Escenario de Usabilidad 21](#_bookmark29)

[Escenario de confiabilidad 21](#_bookmark30)

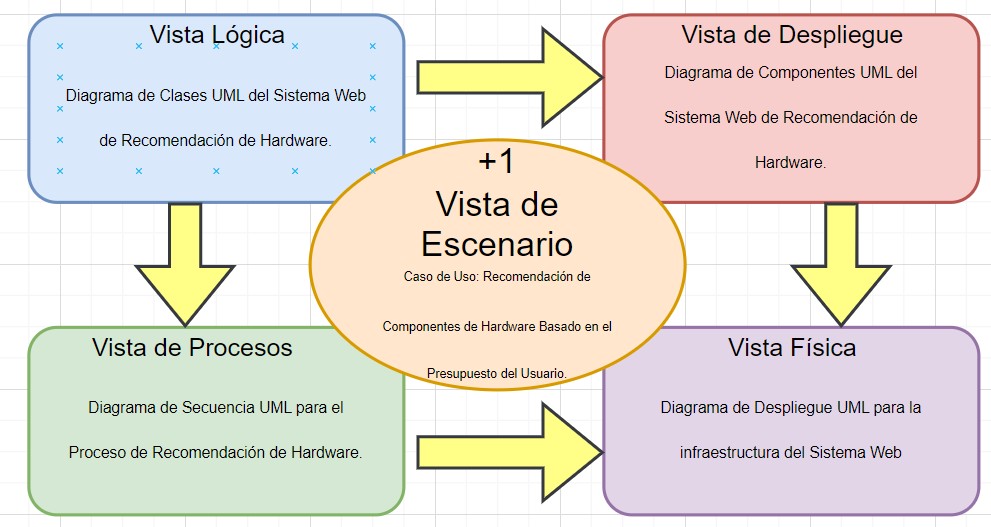
[Escenario de rendimiento 21](#_bookmark31)

[Escenario de mantenibilidad 23](#_bookmark32)

##### INTRODUCCIÓN

### Propósito (Diagrama 4+1)

*Ilustración 1*



*Fuente: elaboración propia*

*Descripción: La imagen nos muestra una orden de como se lleva a cabo el propósito del sistema.*

### Alcance

El sistema abarcará desde la recolección de datos de los usuarios, como su presupuesto y preferencias, hasta la generación de recomendaciones de hardware que cumplan con sus requisitos. Se integrará con bases de datos de componentes de hardware y ofrecerá una interfaz de usuario intuitiva para facilitar la selección y comparación de opciones.

### Definición, siglas y abreviaturas

•HW: Hardware.

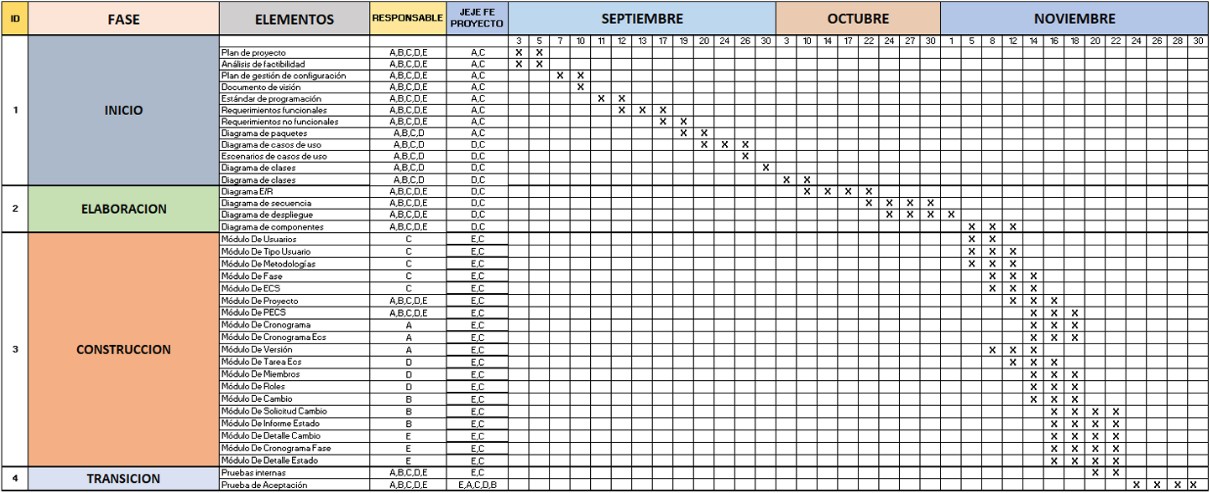
•SWR: Sistema Web de Recomendación.

•BD: Base de Datos.

•UI: Interfaz de Usuario.

### Organización del documento

*Ilustración 2*



*Fuente: elaboración propia*

*Descripción: La imagen nos muestra el cronograma que se llevara a cabo el llevado del sistema.*

# OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS

* 1. Priorización de requerimientos

#### Requerimientos Funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Nombre** | **Descripción** | **Prioridad** |
| RF-001 | Ingreso de  Presupuesto | El sistema permitirá que el usuario ingrese un  presupuesto inicial para la recomendación. | Alta |
| RF-002 | Selección de Componentes | El sistema mostrará los componentes recomendados basados en el presupuesto  ingresado. | Alta |
| RF-003 | Filtros  Personalizados | El usuario podrá aplicar filtros personalizados  como marca y tipo de uso. | Media |
| RF-004 | Actualización de  Precios | El sistema actualizará automáticamente los  precios de los componentes. | Alta |
| RF-005 | Chatbot de Recomendaciones | El sistema incluirá un chatbot que interactuará  con el usuario y proporcionará recomendaciones. | Alta |

|  |  |  |  |
| --- | --- | --- | --- |
| RF-006 | Visualización de Componentes | El usuario podrá visualizar detalles de cada  componente recomendado, incluyendo especificaciones. | Alta |
| RF-007 | Guardar Historial de  Recomendaciones | El sistema registrará las recomendaciones  dadas al usuario para futuras consultas. | Media |
| RF-008 | Comparación de  Productos | El usuario podrá comparar dos o más  componentes recomendados. | Media |
| RF-009 | Validación de  Presupuesto | El sistema validará que el presupuesto  ingresado sea un número válido. | Alta |
| RF-010 | Filtrar por Tipo de  Uso | El usuario podrá filtrar recomendaciones según  su tipo de uso (gaming, oficina, etc.). | Media |
| RF-011 | Mostrar Reseñas de  Productos | El sistema mostrará reseñas de otros usuarios  sobre los productos recomendados. | Media |
| RF-012 | Enviar Consultas al  Soporte | El usuario podrá enviar consultas o  comentarios al soporte a través del chatbot. | Media |
| RF-013 | Estadísticas de Uso | El sistema podrá mostrar estadísticas sobre las  recomendaciones más populares. | Baja |
| RF-014 | Soporte para Múltiples Monedas | El sistema podrá manejar presupuestos en diferentes monedas, actualizando tasas  automáticamente. | Media |
| RF-015 | Respuestas Personalizadas del  Chatbot | El chatbot responderá de manera personalizada según las interacciones previas  del usuario. | Alta |

##### Requerimientos No Funcionales

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| RNF-001 | Seguridad | El sistema garantizará la protección de los datos del  usuario mediante protocolos de seguridad SSL. |
| RNF-002 | Escalabilidad | El sistema será escalable para soportar un gran número de  usuarios simultáneos. |
| RNF-003 | Mantenibilidad | La arquitectura del sistema permitirá un fácil  mantenimiento y actualizaciones regulares. |
| RNF-004 | Rendimiento | El sistema garantizará tiempos de respuesta rápidos,  menores a 2 segundos por cada consulta. |
| RNF-005 | Usabilidad | La interfaz será intuitiva y fácil de usar, orientada a  usuarios sin conocimientos técnicos. |
| RNF-006 | Compatibilidad | El sistema será compatible con los principales navegadores  web. |
| RNF-007 | Accesibilidad | El sistema cumplirá con las pautas de accesibilidad web  para usuarios con discapacidades. |
| RNF-008 | Documentación | Se proporcionará documentación clara y detallada para el  usuario final y para desarrolladores. |
| RNF-009 | Respaldo de Datos | Se implementarán medidas de respaldo de datos  automáticas para prevenir pérdida de información. |

|  |  |  |
| --- | --- | --- |
| RNF-010 | Monitoreo y Logs | El sistema llevará un registro de logs de eventos para  facilitar la auditoría y el monitoreo. |

* 1. Requerimientos No Funcionales – Atributos de Calidad

|  |  |  |
| --- | --- | --- |
| **Código** | **Nombre** | **Descripción** |
| RNF-001 | Escalabilidad | El sistema será escalable para soportar un gran número de usuarios simultáneos. |
| RNF-002 | Mantenibilidad | La arquitectura del sistema permitirá un fácil mantenimiento y actualizaciones regulares. |

* 1. Restricciones

Recursos Financieros Limitados: El proyecto se llevará a cabo dentro de un presupuesto definido, sin acceso a recursos adicionales. Límites de Tiempo: El proyecto debe completarse dentro de un plazo específico. Disponibilidad de Personal Clave: La participación del equipo es crucial para el éxito del proyecto.

# REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

### Vista de Caso de uso

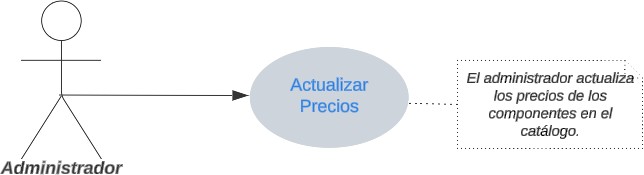
*Ilustración 3*



*Fuente: elaboración propia*

*Descripción: La imagen nos muestra el caso de uso Gestionar Catalogo de Componentes donde muestra como el administrador debe gestionarla el catálogo de ventas de los componentes de software.*

*Ilustración 4*



*Fuente: elaboración propia*

*Descripción: La imagen nos muestra el caso de uso de como el administrador debe gestionarla el catálogo de ventas de los componentes de software.*

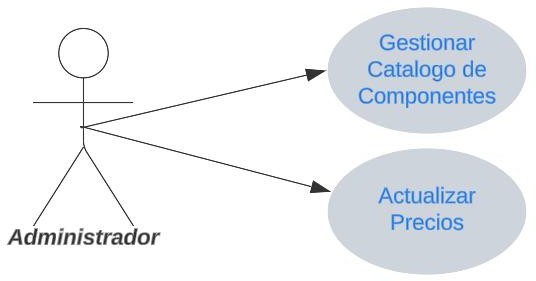


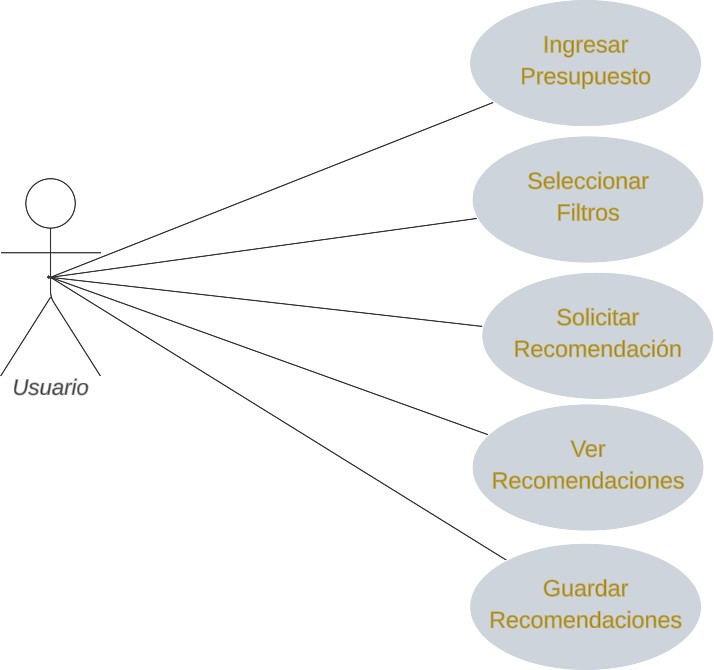






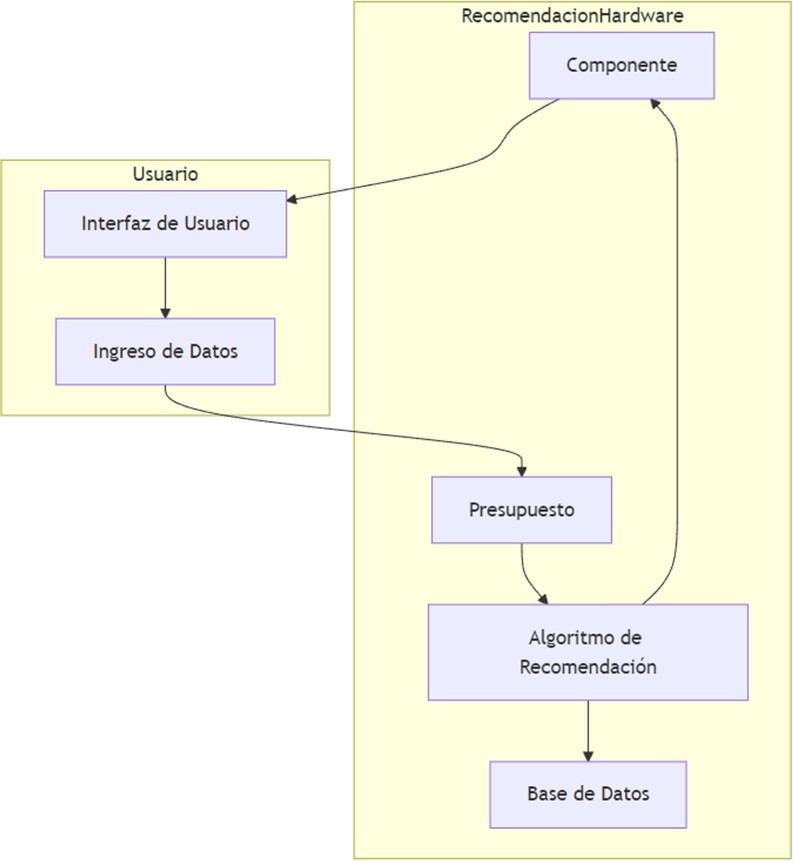


* + 1. Diagramas de Casos de uso



### Vista Lógica

##### Diagrama de Subsistemas (paquetes)



* + 1. Diagrama de Secuencia (vista de diseño)

Diagrama de Secuencia para CU01: **Ingreso de Presupuesto**

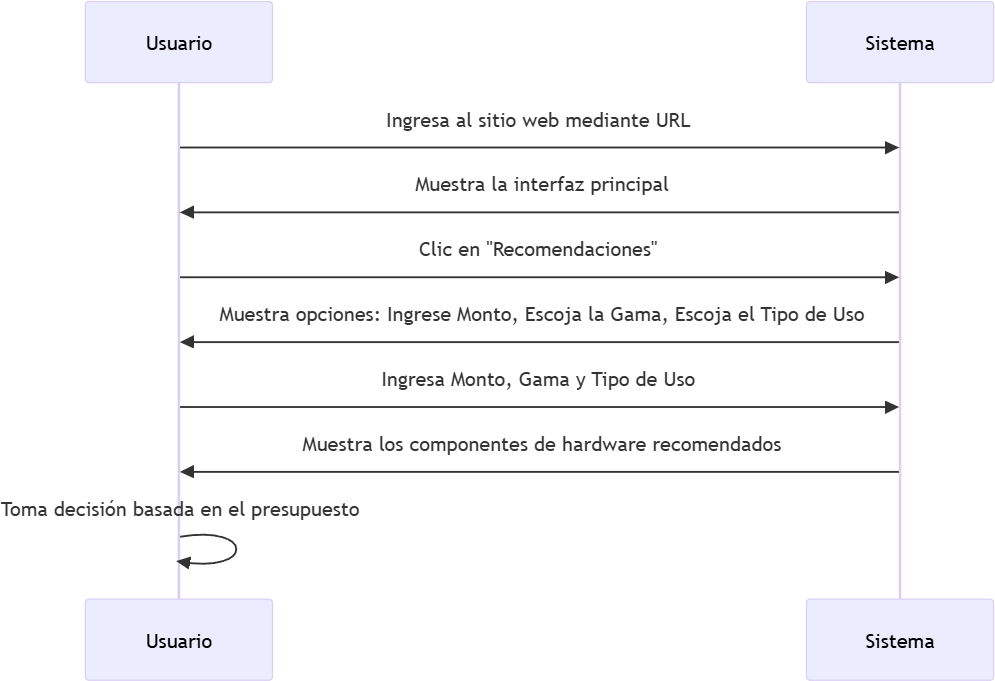


Diagrama de Secuencia para CU02: **Filtrar Recomendaciones por Categoría**

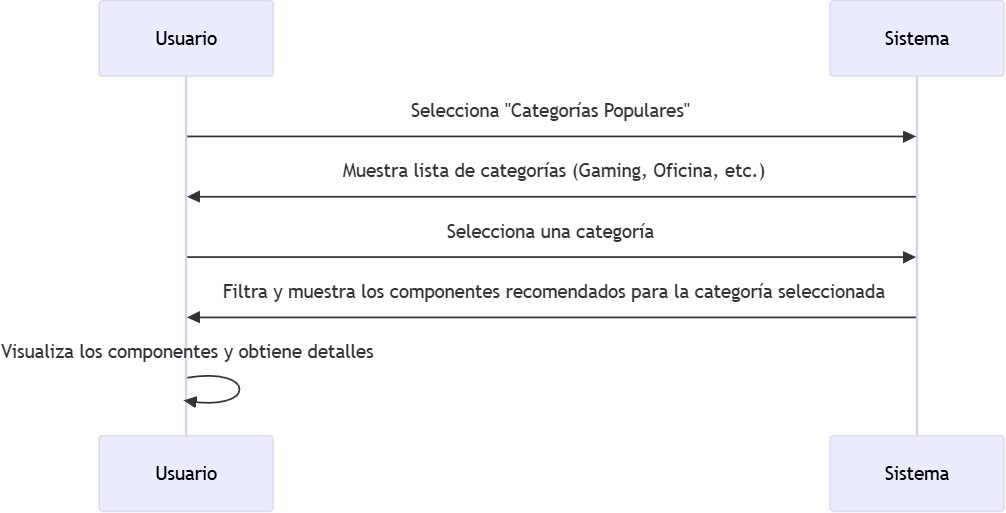


Diagrama de Secuencia para CU03: **Guardar Recomendaciones Favoritas**

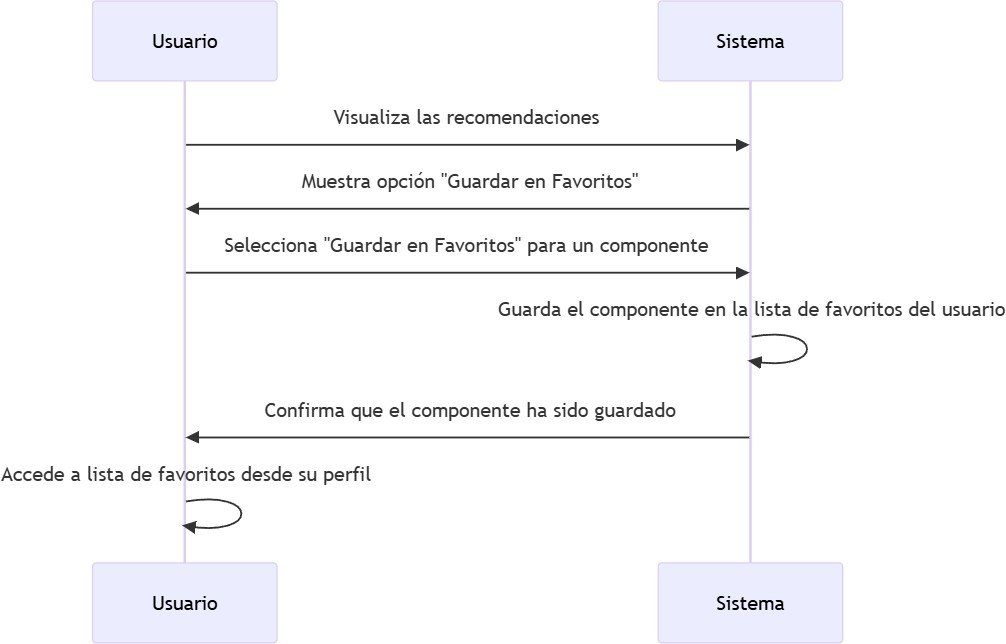


Diagrama de Secuencia para CU04: **Comparar Componentes de Hardware**

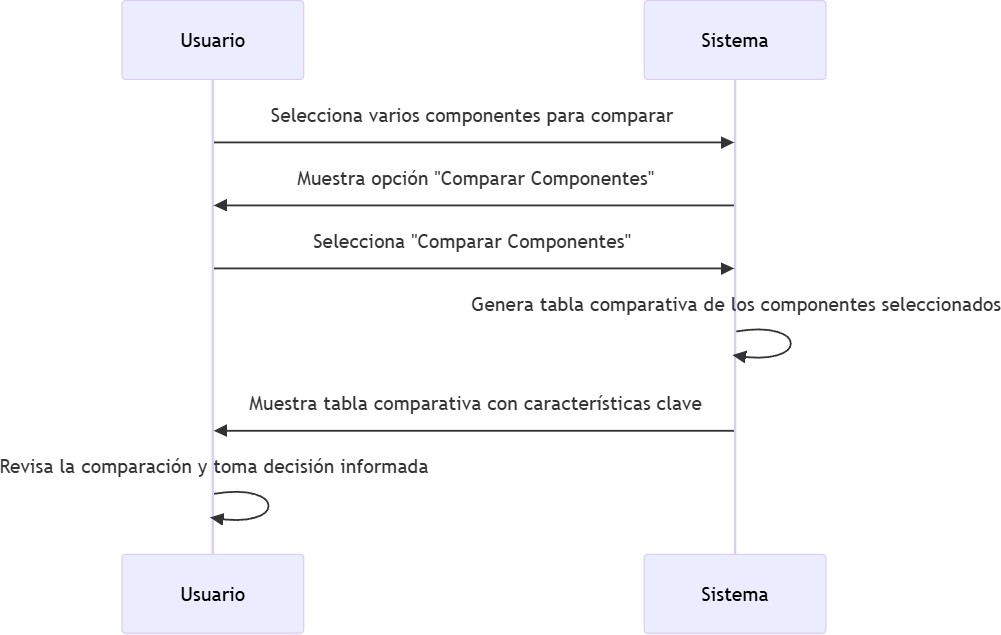
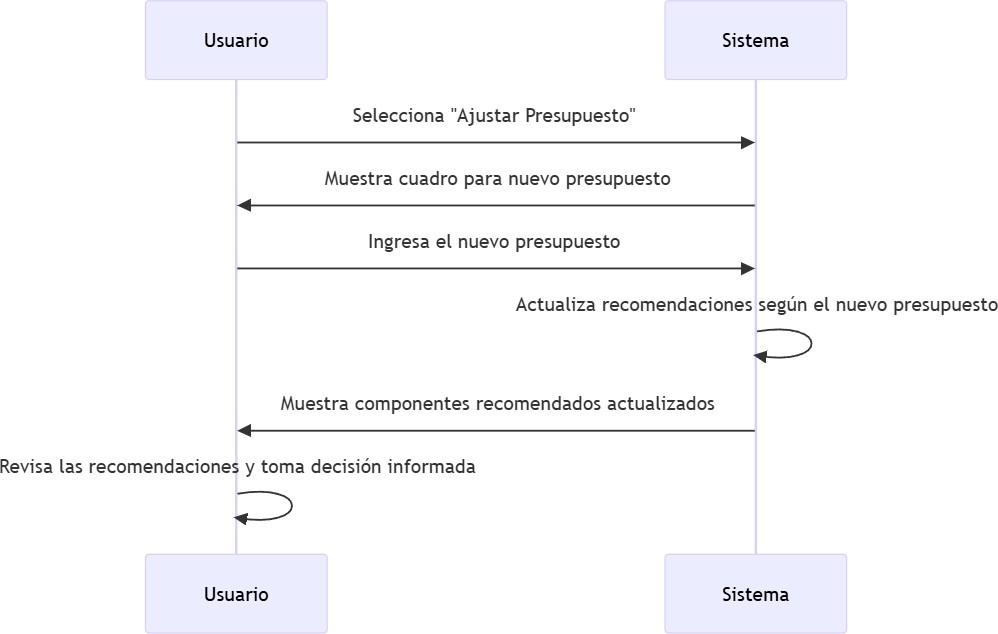
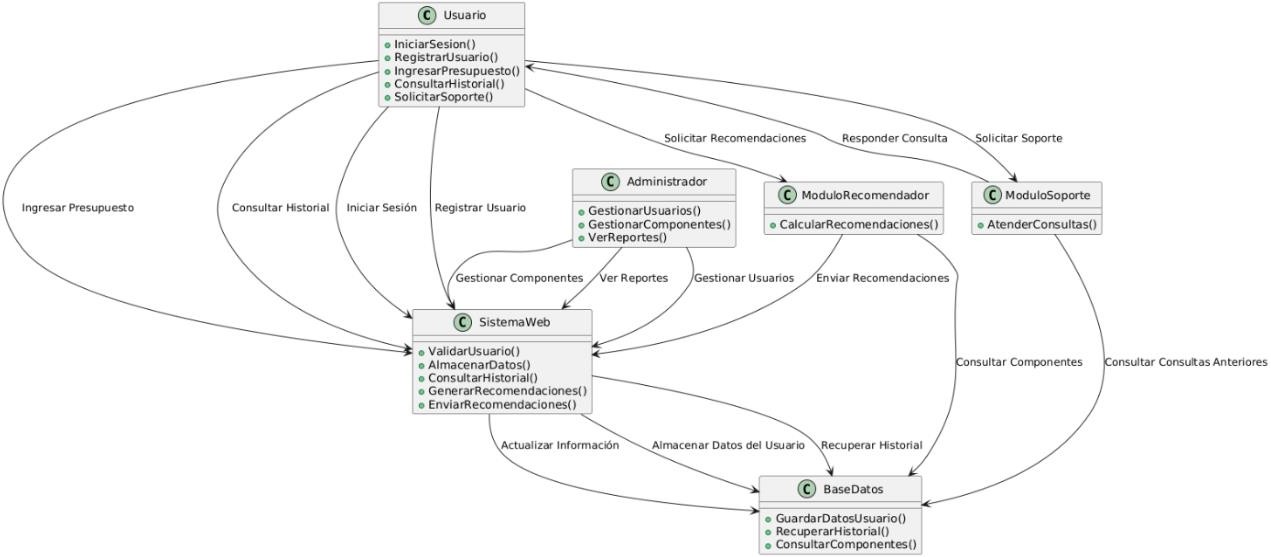


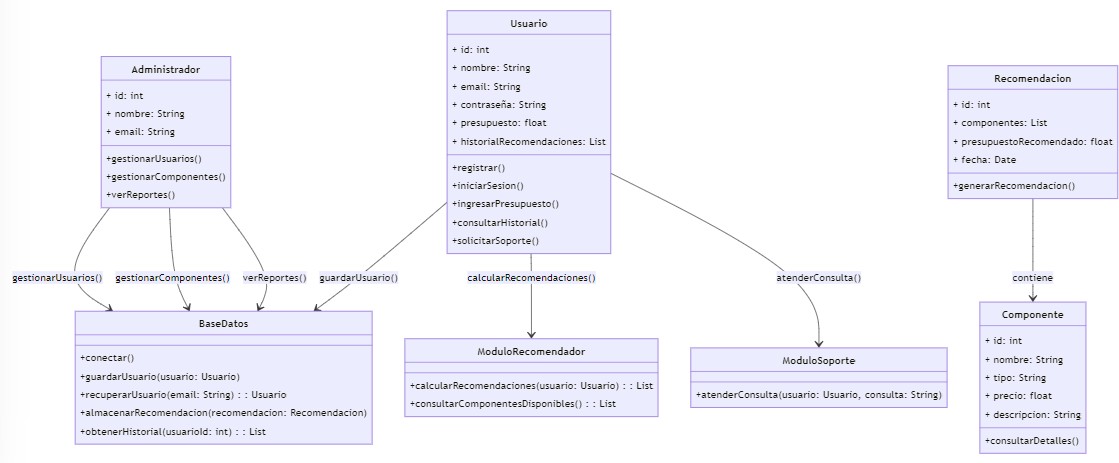
Diagrama de Secuencia para CU05: **Ajustar Presupuesto y Ver Nuevas Recomendaciones**



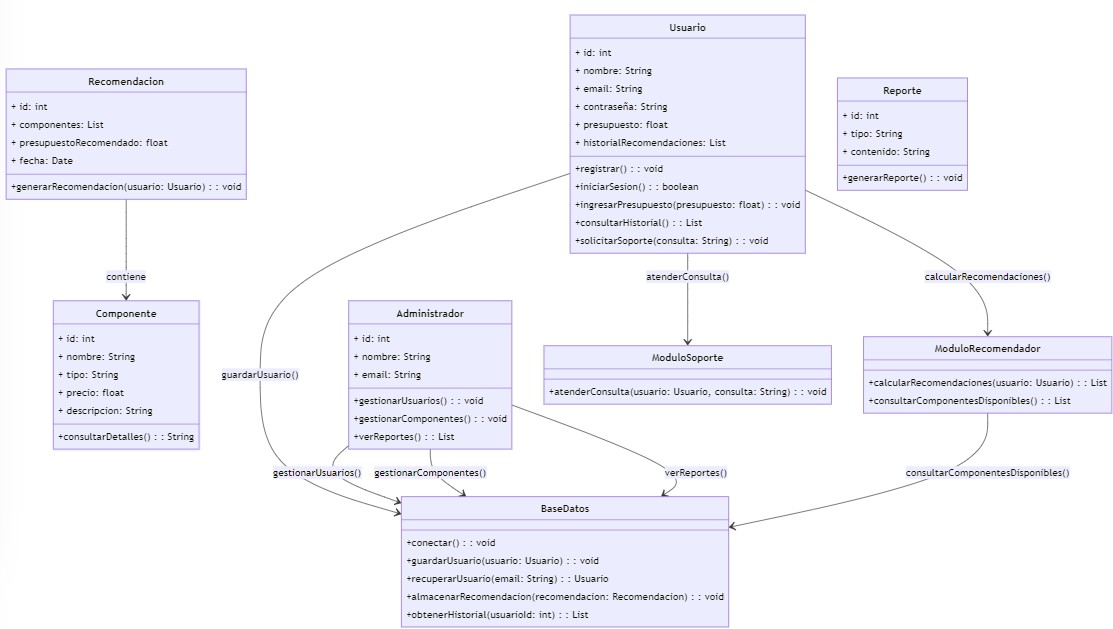
##### Diagrama de Colaboración (vista de diseño)



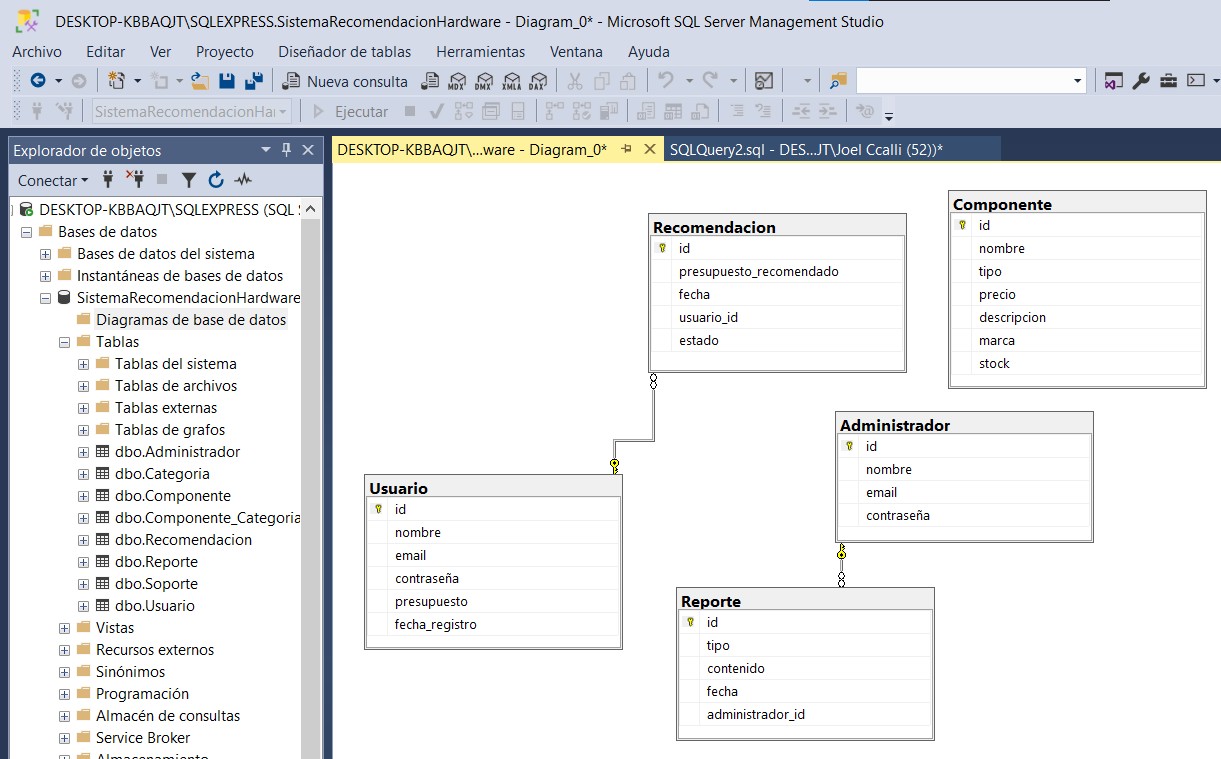
* + 1. Diagrama de Objetos



##### Diagrama de Clases

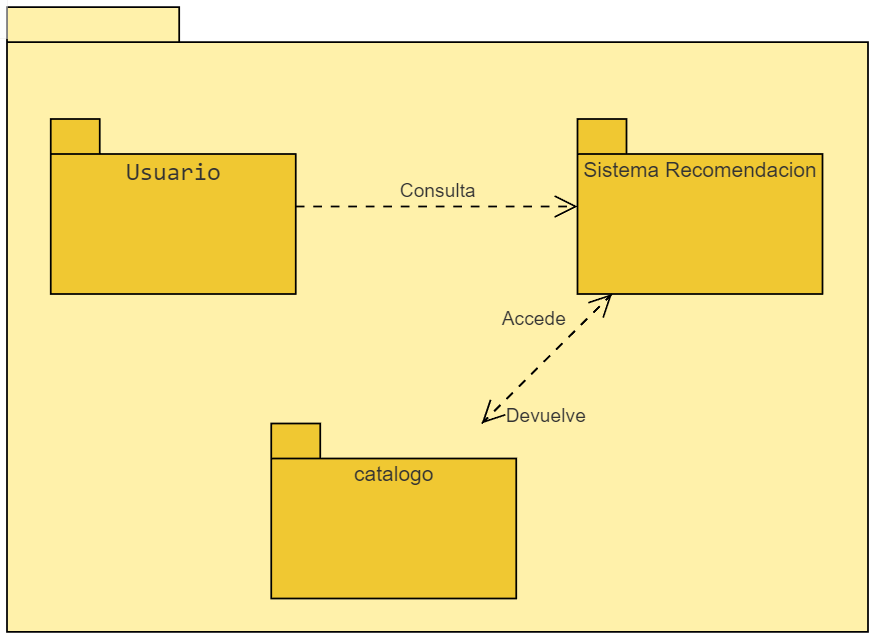


* + 1. Diagrama de Base de datos (relacional o no relacional)

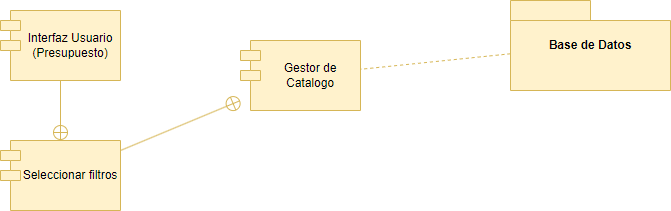


### Vista de Implementación (vista de desarrollo)

##### Diagrama de arquitectura software (paquetes)

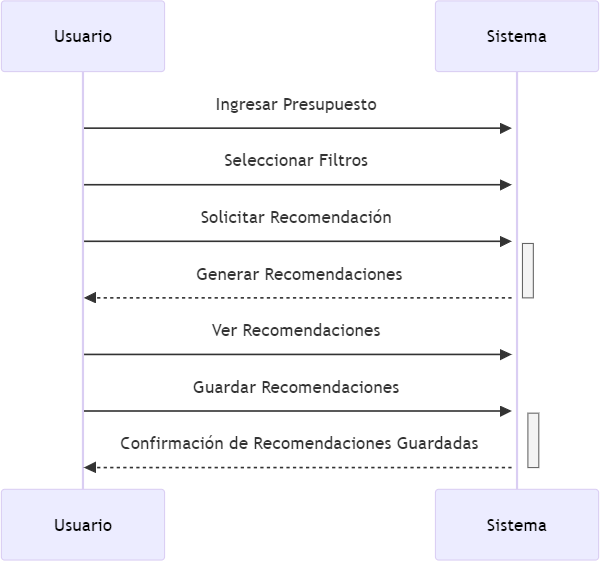


* + 1. Diagrama de arquitectura del sistema (Diagrama de componentes)



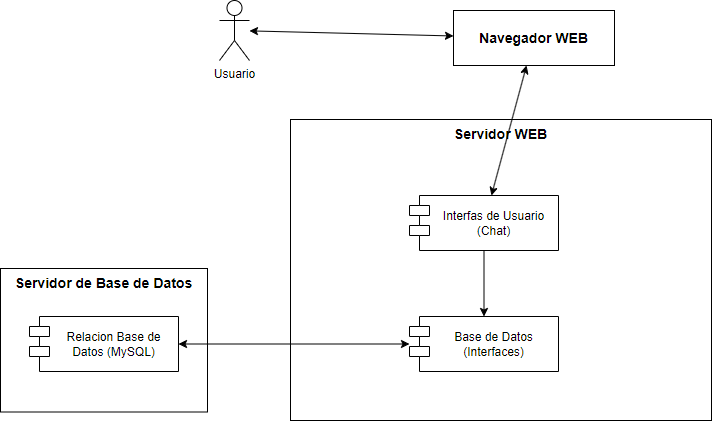
### Vista de procesos

* + 1. Diagrama de Procesos del sistema (diagrama de actividad)

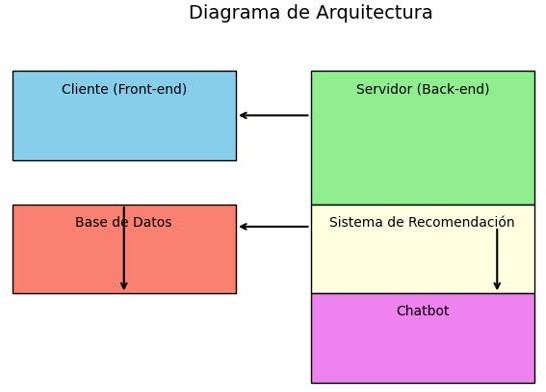


### Vista de Despliegue (vista física)

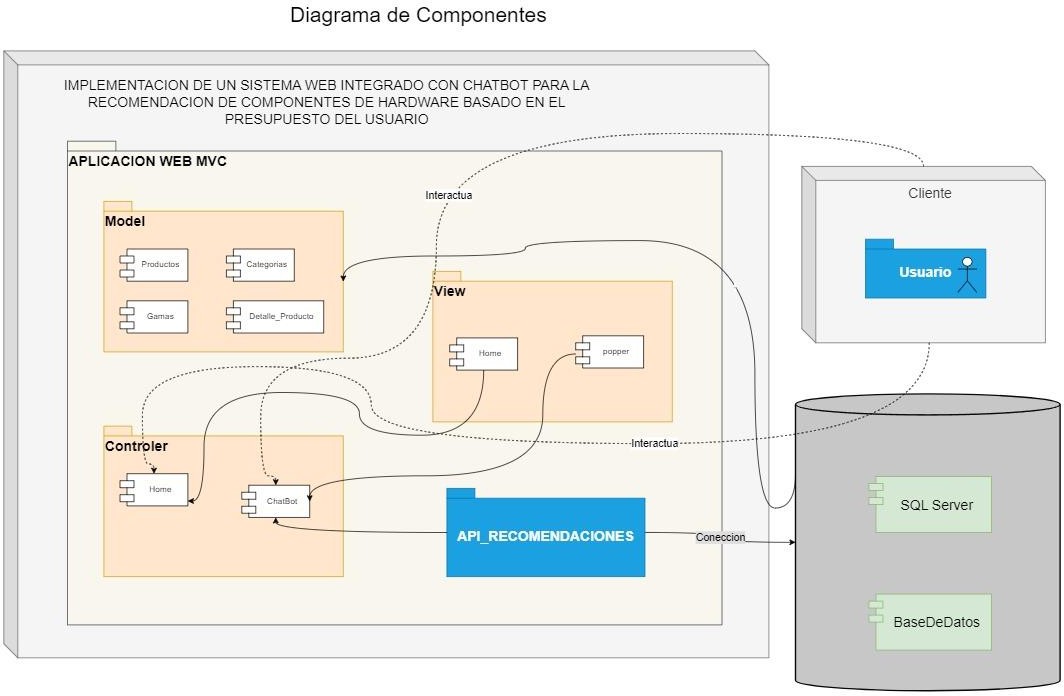
* + 1. Diagrama de despliegue



### Diagrama de Arquitectura



1. Diagrama de Componentes



# ATRIBUTOS DE CALIDAD DEL SOFTWARE

Los QAs se consideran como requerimientos no funcionales, lo que significa que se centran en cómo el software realiza sus funciones, más que en las funciones mismas. La funcionalidad se refiere a la capacidad del sistema para llevar a cabo tareas específicas, mientras que los atributos de calidad evalúan la efectividad y eficiencia con la que se realizan estas tareas. Ignorar estos atributos puede resultar en software que, aunque funcional, sea disfuncional o insatisfactorio para los usuarios.

Los atributos de calidad más comúnmente reconocidos incluyen:

* Funcionalidad: Capacidad del software para satisfacer necesidades específicas de los usuarios.
* Fiabilidad: Consistencia del software para funcionar sin fallos bajo condiciones específicas.
* Usabilidad: Facilidad con la que los usuarios pueden interactuar con el software.
* Rendimiento: Eficiencia del software en términos de velocidad y recursos utilizados.
* Mantenibilidad: Facilidad con la que el software puede ser modificado o actualizado

Modelos de Calidad

El modelo ISO/IEC 25010 es uno de los estándares más utilizados para evaluar la calidad del software. Este modelo clasifica los atributos en varias categorías, incluyendo:

* Calidad funcional: Completa, correcta y apropiada.
* Calidad en uso: Efectividad, eficiencia y satisfacción del usuario durante el uso del software

## Escenario de Funcionalidad

Características y Capacidades

* Ingreso de Presupuesto: El usuario puede ingresar su presupuesto disponible para la compra de hardware.
* Selección de Tipo de Uso: Opciones para especificar el propósito principal del hardware (por ejemplo, gaming, diseño gráfico, ofimática, etc.).
* Recomendación de Componentes: El sistema genera una lista de componentes compatibles dentro del presupuesto y adecuados para el uso especificado.
* Visualización Detallada: Cada componente recomendado incluye especificaciones técnicas, precio y razón de la recomendación.
* Alternativas: El sistema ofrece opciones para cada componente, permitiendo al usuario ajustar la configuración.
* Comparación de Configuraciones: Capacidad para comparar diferentes configuraciones lado a lado.
* Guardado de Configuraciones: Los usuarios pueden guardar y recuperar configuraciones para futuras referencias.
* Actualización de Precios: El sistema actualiza regularmente los precios de los componentes para mantener la precisión de las recomendaciones.

Generalidad de las Funciones

* Adaptabilidad: El sistema se adapta a diferentes rangos de presupuesto y tipos de uso.
* Escalabilidad: Capacidad para añadir nuevos componentes y categorías de hardware sin modificar la estructura principal del sistema.
* Personalización: Los usuarios pueden ajustar la importancia de ciertos factores en las recomendaciones (por ejemplo, priorizar rendimiento sobre precio).

Seguridad General del Sistema

* Autenticación de Usuarios: Sistema seguro de login para proteger las configuraciones guardadas de los usuarios.
* Encriptación de Datos: Toda la información sensible del usuario se almacena de forma encriptada.
* HTTPS: Uso de conexiones seguras para todas las interacciones entre el cliente y el servidor.
* Validación de Entradas: Verificación rigurosa de todas las entradas del usuario para prevenir ataques maliciosos.
* Auditoría: Registro de actividades importantes del sistema para detectar y responder a posibles amenazas de seguridad.

## Escenario de Usabilidad

Utilización Eficiente del Sistema

* Diseño Intuitivo: Interfaz clara con una estructura lógica que sigue el flujo natural del proceso de selección de componentes.
* Filtros Rápidos: Permiten refinar las recomendaciones con pocos clics.

Minimización del Impacto de Errores

* Validación en Tiempo Real: Feedback inmediato sobre errores en la entrada de datos.
* Función de Deshacer: Permite revertir acciones fácilmente.

Accesibilidad

* Ajuste de Contraste y Tamaño de Fuente: Mejora la legibilidad para usuarios con dificultades visuales.

## Escenario de confiabilidad

Prevención

* Control de versiones para todas las actualizaciones de la base de datos de componentes.
* Registro detallado de todas las acciones de los usuarios.
* Arquitectura de alta disponibilidad para Celulares y Escritorio

Precaución

* Copias de seguridad a la Base de datos regulares.
* Almacenamiento seguro y redundante de logs de actividad.
* Monitoreo continuo del rendimiento y la capacidad del sistema

## Escenario de rendimiento

Velocidad de Procesamiento Objetivos:

* Generar recomendaciones de componentes en menos de 2 segundos para configuraciones estándar.
* Procesar actualizaciones de la base de datos de componentes en tiempo real sin afectar el rendimiento del sistema.

Estrategias:

* Implementar algoritmos de recomendación optimizados utilizando técnicas de machine learning.
* Utilizar procesamiento paralelo para manejar múltiples solicitudes simultáneas.
* Emplear técnicas de caching para almacenar resultados de cálculos frecuentes.

Tiempo de Respuesta Objetivos:

* Cargar la página principal en menos de 1 segundo para el 95% de los usuarios.
* Responder a las interacciones del usuario (clicks, selecciones) en menos de 200 ms.

Estrategias:

* Implementar una arquitectura de microservicios para reducir la latencia.
* Optimizar consultas a la base de datos mediante indexación y caching.

Uso de Recursos Objetivos:

* Mantener el uso de CPU por debajo del 70% en condiciones de carga normal.
* Limitar el uso de memoria RAM a un máximo de 4GB por instancia de servidor.
* Optimizar el almacenamiento para no exceder 1TB de datos incluyendo backups.

Estrategias:

* Implementar auto-escalado basado en el uso de recursos.
* Utilizar compresión de datos para reducir el uso de almacenamiento y ancho de banda.

Conjunto (Throughput)

Objetivos:

* Soportar hasta 10,000 usuarios concurrentes sin degradación del rendimiento.
* Procesar un mínimo de 1,000 recomendaciones por minuto en horas pico. Estrategias:
* Implementar balanceo de carga para distribuir el tráfico equitativamente entre servidores.
* Utilizar bases de datos NoSQL para manejar grandes volúmenes de datos no estructurados.
* Emplear técnicas de cola de mensajes para gestionar picos de demanda.

## Escenario de mantenibilidad

Extensibilidad

La extensibilidad se refiere a la facilidad con la que el sistema puede ser ampliado con nuevas funcionalidades o capacidades.

Objetivos:

* Permitir la adición de nuevas categorías de componentes sin modificar el código base.
* Facilitar la integración de nuevos algoritmos de recomendación.
* Posibilitar la expansión a nuevos mercados o regiones geográficas.

Adaptabilidad

La adaptabilidad se refiere a la capacidad del sistema para ajustarse a nuevos entornos o requisitos cambiantes.

Objetivos:

* Permitir la personalización de la interfaz de usuario sin cambios en el backend.
* Facilitar la adaptación a diferentes dispositivos y tamaños de pantalla.
* Posibilitar cambios en la lógica de negocio sin afectar la estructura del sistema.

Capacidad de Servicio (Serviceability)

La capacidad de servicio se refiere a la facilidad con la que el sistema puede ser mantenido, reparado o mejorado.

Objetivos:

* Minimizar el tiempo de inactividad durante las actualizaciones del sistema.
* Facilitar la identificación y resolución de problemas.
* Permitir la realización de backups y restauraciones sin interrumpir el servicio.

Prácticas de Desarrollo Control de Versiones:

* Utilizar Git con una estrategia de branching.
* Implementar revisiones de código obligatorias antes de fusionar cambios.

Documentación:

* Mantener documentación actualizada de la arquitectura del sistema.

Análisis:

* Realizar revisiones periódicas de código para identificar áreas de mejora.
* Analizar logs y métricas para identificar patrones de uso y potenciales problemas.