Revisión de arquitectura de software

[Nombre proyecto]

|  |  |  |
| --- | --- | --- |
| **Código documento** | : | Informe de evaluación de arquitecturas de software |
| **Versión** | : | R001 |
| **Fecha inicio vigencia** | : |  |
| **Aprobado por** | : | Carlos Alberto Ortiz |
| **Fecha aprobación** | : |  |
| **Preparado por** | : | Dirección de Informática UC |
| **Autor** | : | Karl Wennerstrom |
| **Fecha última modificación** | : |  |
| **Fecha creación** | : | Julio 2.024 |

Tabla de contenidos

[1 **PROPÓSITO** 3](#_Toc173410888)

[2 **ÁMBITO** 3](#_Toc173410889)

[3 **INFORMACIÓN DEL PROYECTO** 3](#_Toc173410890)

[4 **DESCRIPCIÓN GENERAL DE LA ARQUITECTURA** 3](#_Toc173410891)

[4.1 **VISIÓN GENERAL DEL SISTEMA** 3](#_Toc173410892)

[4.2 **RELACIÓN ENTRE COMPONENTES** 5](#_Toc173410893)

[4.2.1 DIAGRAMA DE CONTEXTO DEL SISTEMA 5](#_Toc173410894)

[4.2.2 DIAGRAMA DE ARQUITECTURA DE INFORMACIÓN (DATOS) 5](#_Toc173410895)

[4.2.3 DIAGRAMA DE ARQUITECTURA DE APLICACIÓN 5](#_Toc173410896)

[4.2.4 DIAGRAMA DE TECNOLOGÍA 5](#_Toc173410897)

[4.2.5 DIAGRAMA DE NEGOCIO (PROCESOS) 5](#_Toc173410898)

[4.2.6 TECNOLOGÍAS UTILIZADAS 6](#_Toc173410899)

[5 **REQUISITOS DE CALIDAD DEL PROYECTO** 6](#_Toc173410900)

[6 **INCIDENCIAS DE ARQUITECTURA ENCONTRADAS EN REVISIÓN** 7](#_Toc173410901)

[7 **RECOMENDACIONES** 7](#_Toc173410902)

[8 **DOCUMENTOS RELACIONADOS** 7](#_Toc173410903)

[9 **DEFINICIONES Y ACRONIMOS** 8](#_Toc173410904)

[10 **HISTORIAL DE REVISIÓN** 8](#_Toc173410905)

# PROPÓSITO

El propósito de este documento es proporcionar una guía estructurada para la revisión de la arquitectura de software. Esta plantilla tiene como objetivo estandarizar el proceso de evaluación, asegurando que todos los aspectos críticos de la arquitectura sean considerados y documentados de manera consistente. Al utilizar esta plantilla, los equipos de desarrollo y arquitectura podrán identificar fortalezas, debilidades, riesgos y oportunidades de mejora en sus sistemas de software, facilitando la toma de decisiones informadas y promoviendo la calidad, escalabilidad, rendimiento y seguridad del software desarrollado.

# ÁMBITO

Este documento aplica a todos los desarrollos de software cuyo producto final sea alojado en data center de la universidad.

# INFORMACIÓN DEL PROYECTO

[Para ser llenado por JP de proyecto]

Fecha de revisión de la arquitectura: 08/07/2024

Responsable del proyecto: Ejemplo Karl Wennerstrom

Entidad que entrega: Dirección de Transformación Digital (DTDF)

Entidad que recibe: Dirección de Informática (DI)

# DESCRIPCIÓN GENERAL DE LA ARQUITECTURA

[Para ser llenado por JP de proyecto]

## VISIÓN GENERAL DEL SISTEMA

Descripción General: Proporcione una descripción concisa del sistema, incluyendo su propósito y características principales. Por ejemplo:

"El Sistema de Gestión de Inventarios en Tiempo Real es una aplicación distribuida basada en microservicios que permite el seguimiento, actualización y análisis de inventarios para múltiples ubicaciones en tiempo real."

### COMPONENTES PRINCIPALES

Liste y describa brevemente los componentes principales del sistema. Cada componente debe ser explicado en términos de su función y su papel dentro del sistema global. Esto ayuda a entender cómo se descompone el sistema y cómo interactúan sus partes.

Ejemplo:

**Servicio de Autenticación y Autorización**: Describa el servicio responsable de autenticar y autorizar a los usuarios.

**Servicio de Gestión de Inventario**: Describa el servicio encargado de la gestión y actualización de inventarios.

**Servicio de Análisis y Reportes**: Describa el servicio que proporciona capacidades de análisis y generación de reportes.

**Servicio de Notificaciones**: Describa el servicio que gestiona el envío de notificaciones, como alertas de stock bajo.

**API Gateway**: Describa el componente que actúa como puerta de enlace para todas las solicitudes de API, asegurando la correcta enrutación y gestión de tráfico.

**Base de Datos Distribuida**: Describa la base de datos utilizada para almacenar y gestionar los datos del sistema de manera distribuida.

### FUNCIONALIDADES PRINCIPALES

Describa las funcionalidades clave del sistema. Cada funcionalidad debe ser explicada de manera que se entienda qué hace y cómo beneficia a los usuarios. Esto debe incluir ejemplos específicos de cómo se usa cada funcionalidad en la práctica.

Ejemplo:

**Registro y Seguimiento de Productos**: escriba la funcionalidad que permite registrar y hacer un seguimiento detallado de los productos en el inventario.

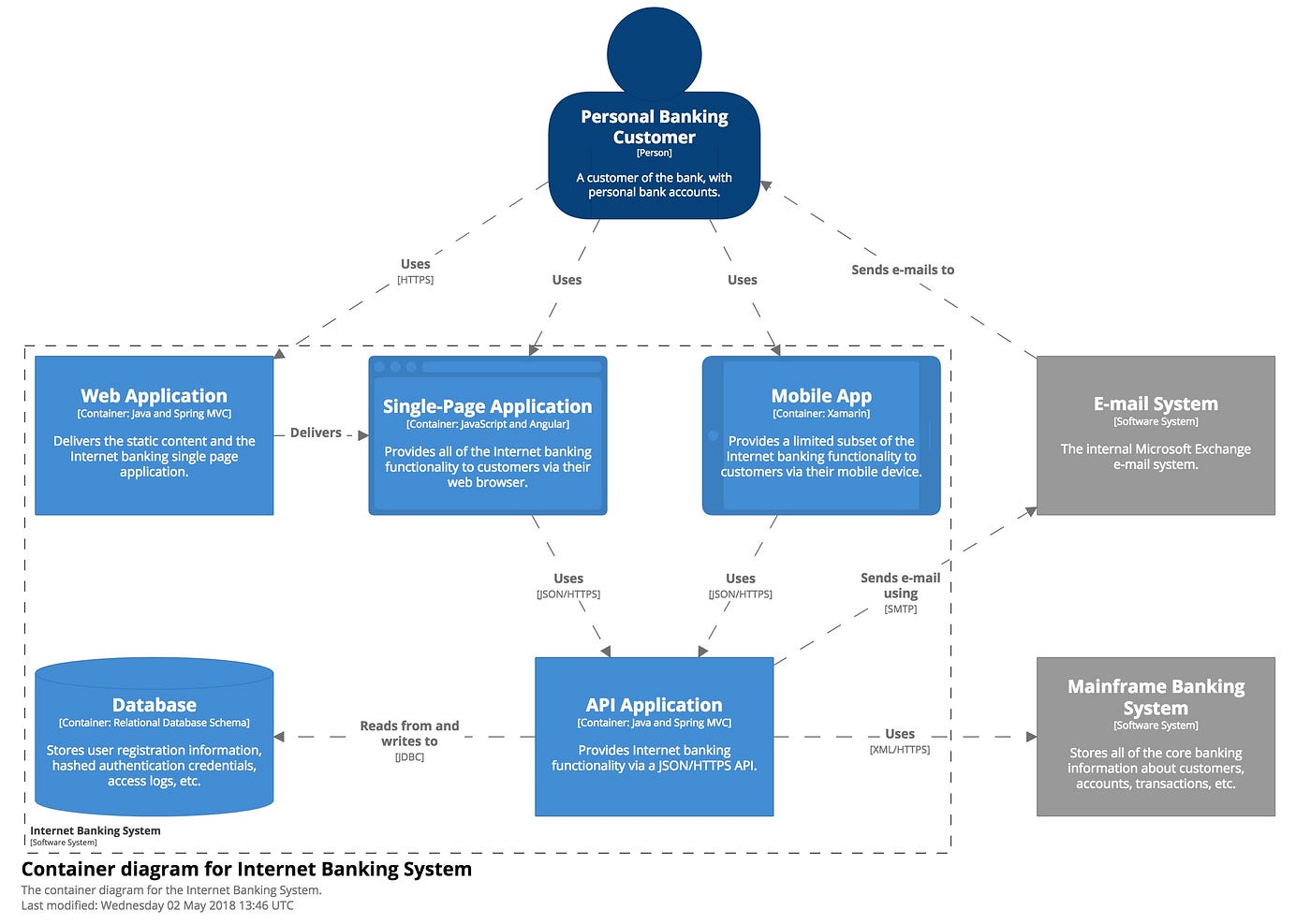
**Gestión de Órdenes de Compra**: Describa la funcionalidad que gestiona la creación, seguimiento y cumplimiento de órdenes de compra.

**Generación de Informes de Inventario**: Describa la funcionalidad que permite generar informes detallados sobre el estado y movimientos del inventario.

**Notificaciones de Stock Bajo**: Describa la funcionalidad que envía alertas y notificaciones cuando el stock de productos alcanza niveles bajos.

## RELACIÓN ENTRE COMPONENTES

### DIAGRAMA DE CONTEXTO DEL SISTEMA



La infraestructura de siga contempla los siguientes componentes:

Explique brevemente los principales componentes de la solución.

Ejemplo:

**Balanceadores de carga.**

**Servicio Web.**

**Servicio de Datos.**

**Servicio de Aplicación y Almacenamiento.**

### DIAGRAMA DE ARQUITECTURA DE INFORMACIÓN (DATOS)

### DIAGRAMA DE ARQUITECTURA DE APLICACIÓN

### DIAGRAMA DE TECNOLOGÍA

### DIAGRAMA DE NEGOCIO (PROCESOS)

### TECNOLOGÍAS UTILIZADAS

En esta sección, enumere brevemente las tecnologías clave utilizadas en el proyecto junto con una breve descripción de su función y razón de elección. Esto proporciona una visión clara de las herramientas y plataformas fundamentales utilizadas en el desarrollo del sistema.

Ejemplo:

Lenguajes de Programación y Frameworks:

* Java (Spring Boot): Backend robusto y escalable.
* JavaScript (React.js): Interfaz de usuario dinámica y interactiva.

Bases de Datos:

* PostgreSQL: Almacenamiento de datos transaccionales por su robustez y capacidad de consultas complejas.
* Redis: Caché en memoria para mejorar la velocidad de acceso a datos.

Herramientas de Desarrollo:

* Git: Control de versiones para gestión colaborativa del código.
* Docker: Contenedorización para un entorno de desarrollo consistente y despliegue eficiente.

Servicios en la Nube:

* Azure: Despliegue escalable y gestión de infraestructura.
* Jenkins: Integración continua para automatización de pruebas y despliegues

# ESPECIFICACIONES DE HARDWARE

Ejemplo:

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente** | **Memoria (GB)** | **CPU (núcleos)** | **Almacenamiento** |
| Servidor de Base de datos | 16 | 8 | 500 (SSD) |
| Servidor de Aplicación | 16 | 8 | 1TB (SSD) |
| Servidor Web | 8 | 4 | 50 (SSD) |

# DIMENSIONAMIENTO DE HARDWARE

Ejemplo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ambiente** | **Servidor de Aplicación** | **Servidor Web** | **Servidor de Base de datos** | **Servidor para GitLab Runner** |
| Desarrollo | 1 | 1 | 1 | 1 |
| Liberación | 1 | 3 | 3 | 1 |
| Producción | 1 | 3 | 3 | 1 |

# REQUISITOS DE CALIDAD DEL PROYECTO

Ejemplo

|  |  |  |
| --- | --- | --- |
| **Tipo de requisito** | **Fecha** | **Descripción** |
| Rendimiento | 01/07/2024 | Tiempo de respuesta de menos de 2 segundos para consultas de inventario.  Capacidad para procesar 500 transacciones simultáneas. |
| Escalabilidad |  | El sistema debe manejar un crecimiento del 200% en el número de transacciones y usuarios en los próximos dos años. |
| Disponibilidad |  | Tiempo de actividad del 99.9%.  Implementación de tolerancia a fallos mediante replicación de bases de datos y balanceo de carga. |
| Seguridad |  | Autenticación de dos factores para todos los usuarios.  Cifrado de datos sensibles tanto en tránsito como en reposo.  Auditoría y registro de todas las operaciones críticas. |

# HALLAZGOS DE ARQUITECTURA ENCONTRADAS EN REVISIÓN

(En adelante, para ser llenado por área de Arquitectura de Integración DI)

|  |  |  |
| --- | --- | --- |
| **Tipo de requisito** | **Fecha** | **Descripción** |
| Front-end está redactado con tecnologías obsoletas | 01/07/2024 | El sistema debe manejar un crecimiento del 200% en el número de transacciones y usuarios en los próximos dos años. |
|  |  | Tiempo de respuesta de menos de 2 segundos para consultas de inventario.  Capacidad para procesar 500 transacciones simultáneas. |
|  |  | Tiempo de actividad del 99.9%.  Implementación de tolerancia a fallos mediante replicación de bases de datos y balanceo de carga. |
|  |  | Autenticación de dos factores para todos los usuarios.  Cifrado de datos sensibles tanto en tránsito como en reposo.  Auditoría y registro de todas las operaciones críticas. |

# OBSERVACIONES

[Para ser llenado por área de ARQ - DI]

# RECOMENDACIONES

[Para ser llenado por área de ARQ - DI]

# DOCUMENTOS RELACIONADOS

[Para ser llenado por área de ARQ – DI o JGPS]

* Matriz de roles y responsabilidades (RACI) integración CAS

# DEFINICIONES Y ACRÓNIMOS

[Para ser llenado por área de ARQ - DI]

**UC:** Pontificia Universidad Católica.

**DI:** Dirección de Informática.

**API:** Application Programming Interface

**SGTR:** Sistema de Gestión de Inventarios en Tiempo Real

**K8s:** Kubernetes

**DB:** Base de datos

**REST:** Representational State Transfer

**TI**: Tecnologías de la Información

**SLA:** Service Level Agreement (Acuerdo de Nivel de Servicio)

**CI/CD:** Continuous Integration/Continuous Deployment

**Microservicio**: Enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de APIs bien definidas.

**Contenedor:** Unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de forma rápida y fiable de un entorno informático a otro.

**Orquestación**: Automatización de la gestión, coordinación y organización de sistemas informáticos complejos, servicios y middleware.

# HISTORIAL DE REVISIÓN

|  |  |  |
| --- | --- | --- |
| **Revisión** | **Fecha** | **Descripción del cambio** |
| R001 | 09/07/2024 | Versión inicial |
|  |  |  |
|  |  |  |
|  |  |  |