Revisión de arquitectura de software

SIGA UC

|  |  |  |
| --- | --- | --- |
| **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** | : | José Antonio Durand |
| **Fecha última modificación** | : |  |
| **Fecha creación** | : | Julio 2.024 |

**Tabla de contenidos**

[1 PROPÓSITO 3](#_Toc174533346)

[2 ÁMBITO 3](#_Toc174533347)

[3 INFORMACIÓN DEL PROYECTO 3](#_Toc174533348)

[4 DESCRIPCIÓN GENERAL DE LA ARQUITECTURA 3](#_Toc174533349)

[4.1 VISIÓN GENERAL DEL SISTEMA 3](#_Toc174533350)

[4.1.1 COMPONENTES PRINCIPALES 3](#_Toc174533351)

[4.1.2 FUNCIONALIDADES PRINCIPALES 5](#_Toc174533352)

[4.2 RELACIÓN ENTRE COMPONENTES 6](#_Toc174533353)

[4.2.1 DIAGRAMA DE CONTEXTO DEL SISTEMA 6](#_Toc174533354)

[4.2.2 DIAGRAMA DE ARQUITECTURA DE INFORMACIÓN (DATOS) 7](#_Toc174533355)

[4.2.3 DIAGRAMA DE ARQUITECTURA DE APLICACIÓN 8](#_Toc174533356)

[4.2.4 DIAGRAMA DE ARQUITECTURA DE TECNOLOGÍA 9](#_Toc174533357)

[4.2.5 DIAGRAMA DE NEGOCIO (PROCESOS) 10](#_Toc174533358)

[4.2.6 TECNOLOGÍAS UTILIZADAS 10](#_Toc174533359)

[5 ESPECIFICACIONES DE HARDWARE 11](#_Toc174533360)

[6 DIMENSIONAMIENTO DE HARDWARE 11](#_Toc174533361)

[7 REQUISITOS DE CALIDAD DEL PROYECTO 11](#_Toc174533362)

[8 HALLAZGOS ENCONTRADOS EN LA ARQUITECTURA DURANTE LA REVISIÓN 12](#_Toc174533363)

[9 OBSERVACIONES 13](#_Toc174533364)

[10 RECOMENDACIONES 13](#_Toc174533365)

[11 DOCUMENTOS RELACIONADOS 14](#_Toc174533366)

[12 DEFINICIONES Y ACRÓNIMOS 14](#_Toc174533367)

[13 HISTORIAL DE REVISIÓN 14](#_Toc174533368)

# 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 Pontificia universidad católica de Chile.

# INFORMACIÓN DEL PROYECTO

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

Responsable del proyecto: Luis Reyes Luis Eduardo Reyes Zelaya

Entidad que entrega: Dirección de Transformación Digital UC (DTDF)

Entidad que recibe: Dirección de Informática UC (DI)

# DESCRIPCIÓN GENERAL DE LA ARQUITECTURA

## 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.

**Servicio de Autenticación de usuarios**: Se ocupará el SSO UC

La integración del sistema propuesto al SSO UC, se realizará usando el protocolo CAS.

De esta integración, el sistema propuesto requiere los siguientes atributos de cuenta de usuario UC:

Para completar

* uid (login)
* mail

**Base de datos de Gestión UC (BDGES) (Ecosistema UC):** La solución propuesta requiere una integración con la base de datos de gestión UC (BDGES), el proyecto propone una conexión directa a la base de datos.

Para completar

¿Como se realizará esta conexión a la BD?

Esta conexión supone accesos a estructuras de datos en BDGES provistas para esta solución como vistas o tablas.

**Ficha Académica Acumulada o equivalente (CAPP) (Ecosistema UC):** Describa su implementación, alcance y método de integración.

Para completar.

Método alternativo de no haber acceso a bdges.

**Cubes (Data Warehouse) (Ecosistema UC):** Describa su implementación, alcance y mecanismo de integración.

Para completar.

Método alternativo de no haber acceso a bdges.

**Servicio de Email**: Requiere un servicio de correo para el envío de mensajería.

No se especifica que servicio ocupará el sistema.

Para completar

Se estima una la población objetivo de 4.000 estudiantes a ocupar el sistema, por tanto, se estima un mínimo de 4.000 envío de correos diarios.

**Servicios API UC a implementar**:

Foto Personas UC

**API Gateway**: Para el acceso a servicios API UC se ocupará el API Gateway: api.uc.cl

**Base de Datos Distribuida**: La base de datos a ocupar por el sistema propuesto es MariaDB, versión [10.x], Galera Clúster, con un tamaño inicial de [500] GB. El acceso y gestión de datos será a través de una capa de abstracción vía ORM. Esta base de datos es donde el sistema propuesto almacenará los datos resultantes de sus procesos transaccionales.

**Linux File System**: Para completar, explicar uso/implementación/integración.

Las imágenes y documentos que almacenarán en el file system.

Esta una máquina adicional que procesa datos y los almacena en su file system, los datos almacenados en este file system serán entrada para las funcionalidades del aplicativo propuesto.

### 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.

Para completar

**Administración**: escriba la funcionalidad.

**Ficha Académica**: Describa la funcionalidad.

**Atención de estudiante**: Describa la funcionalidad.

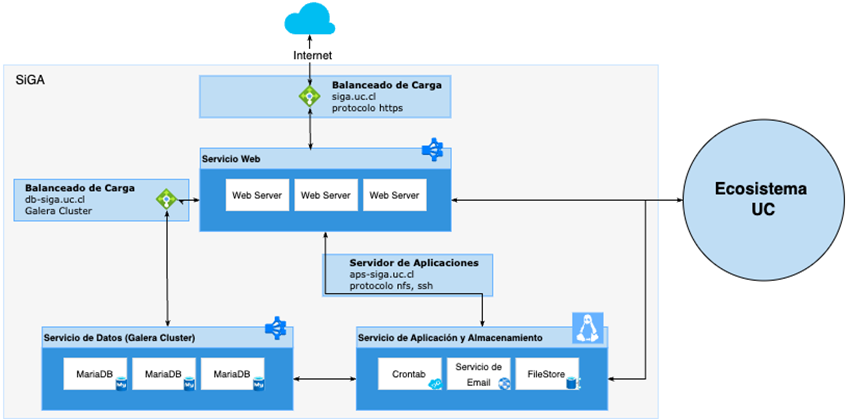
**Integridad académica**: Describa la funcionalidad.

**Egreso y título**: Describa la funcionalidad.

**Ayudantes**: Describa la funcionalidad.

## RELACIÓN ENTRE COMPONENTES

### DIAGRAMA DE CONTEXTO DEL SISTEMA



La infraestructura de siga contempla los siguientes componentes:

**Balanceadores de carga.**

Se han definido dos balanceadores de carga: uno para las aplicaciones web (https con estado para sesiones) y otro para el clúster de bases de datos Galera Cluster para MariaDB. Aunque en el diagrama no se muestran explícitamente los componentes de redundancia, se asume que estos están presentes por defecto.

**Servicio Web.**

El servicio web está compuesto por tres servidores web. La aplicación se carga a través de un disco compartido alojado en el servidor de almacenamiento. Las aplicaciones web se conectan al balanceador de carga de la base de datos para acceder a los datos almacenados.

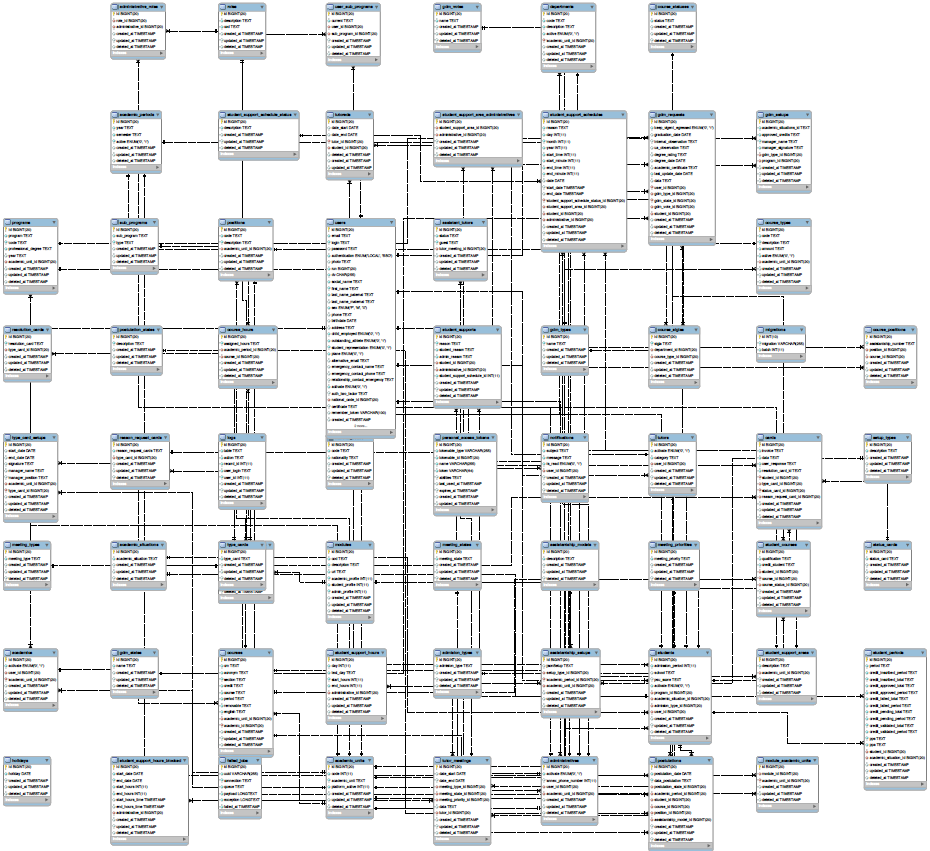
**Servicio de Datos.**

El servicio de datos está compuesto por un clúster Galera para MariaDB con tres nodos operativos. El clúster de Galera funciona en modo multi-master, permitiendo la replicación de datos entre los nodos, lo que garantiza la consistencia y la tolerancia a fallos. El balanceador de carga de la base de datos se conecta a los tres nodos del clúster para distribuir las consultas de manera equilibrada.

**Servicio de Aplicación y Almacenamiento.**

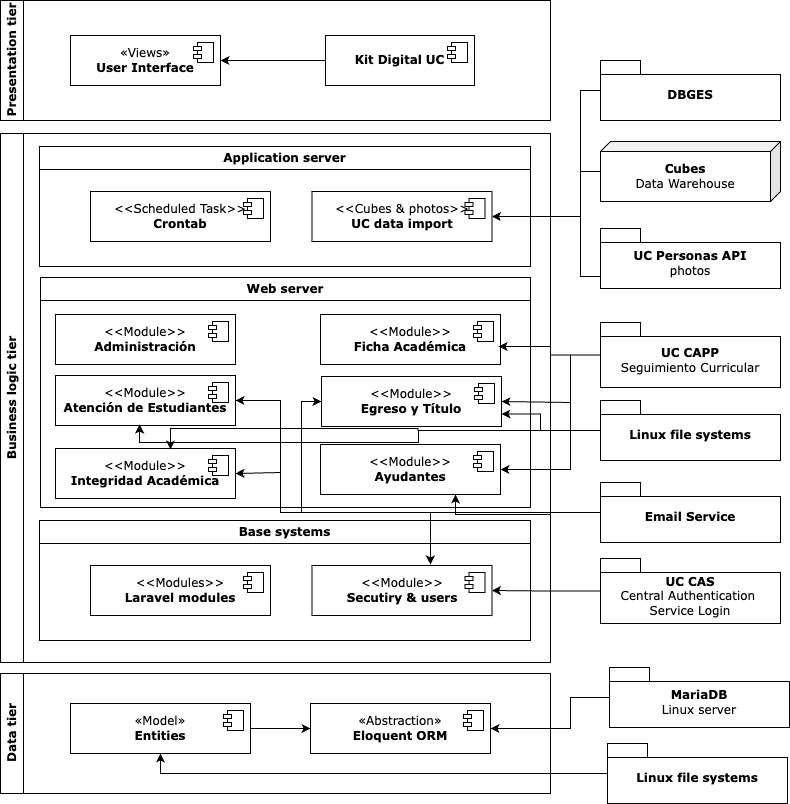
El servidor de aplicaciones y almacenamiento realiza tareas programadas para extraer datos desde el ecosistema UC, lo que asegura el correcto funcionamiento de la aplicación SiGA. Además, el servicio NFS comparte el código de la aplicación con el servicio web y proporciona almacenamiento para archivos que no forman parte de la base de datos, como imágenes o documentos. También se utiliza para ejecutar otras aplicaciones auxiliares requeridas por la aplicación principal. El servidor NFS se conecta tanto al servicio web como al servicio de datos.

### DIAGRAMA DE ARQUITECTURA DE INFORMACIÓN (DATOS)

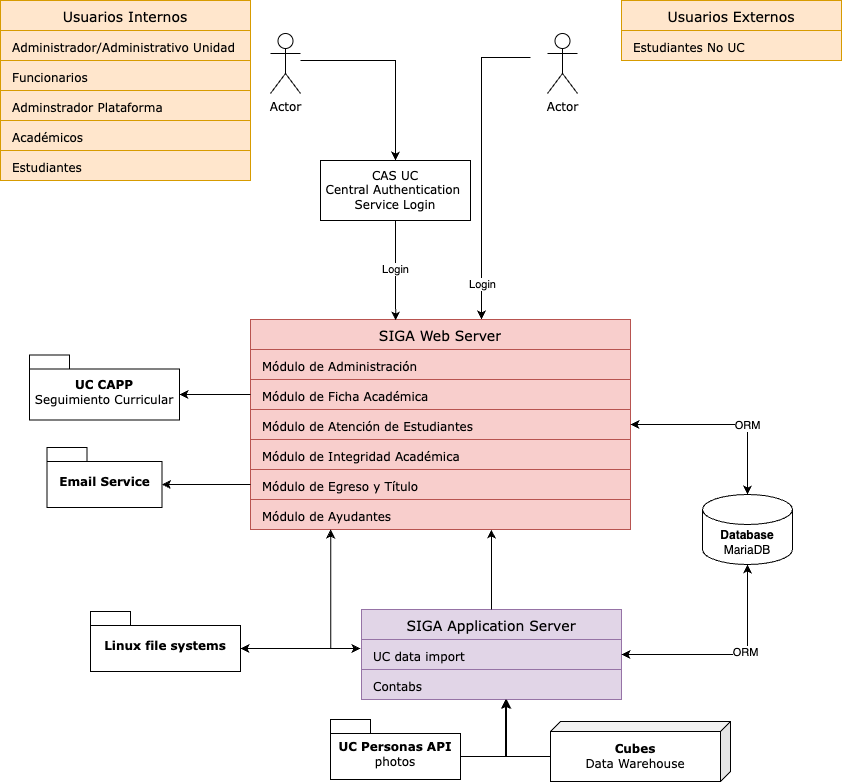




### DIAGRAMA DE ARQUITECTURA DE APLICACIÓN



### DIAGRAMA DE ARQUITECTURA DE TECNOLOGÍA



### DIAGRAMA DE NEGOCIO (PROCESOS)

Se debe incorporar (macro proceso)

### 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.

* **Sistema Operativo:**

Ubuntu Server 24.04 LTS

* **Lenguajes de Programación y Frameworks:**

Lenguaje de programación: php 8.2

Framework: Laravel, versión 10, módulos:

Eloquent: acceso y gestión de datos en base de datos del sistema propuesto

Apache web server: 2.4

* **Bases de Datos:**

MariaDB: Galera Clúster versión 10.x

* **Herramientas de Desarrollo:**

Git: Control de versiones para gestión colaborativa del código.

* **Servicios en la Nube:**

N/A

# ESPECIFICACIONES DE HARDWARE

|  |  |  |  |
| --- | --- | --- | --- |
| **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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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

Completar

|  |  |  |
| --- | --- | --- |
| **Tipo de requisito** | **Fecha** | **Descripción** |
| Rendimiento | 07/08/2024 | Tiempo de respuesta de menos de X segundos para consultas de inventario.  Capacidad para procesar YYY transacciones simultáneas. |
| Escalabilidad | 07/08/2024 | El sistema debe manejar un crecimiento del X% en el número de transacciones y usuarios en los próximos dos años. |
| Disponibilidad | 07/08/2024 | Tiempo de actividad del 99.9%.  Implementación de tolerancia a fallos mediante replicación de bases de datos y balanceo de carga. |
| Seguridad | 07/08/2024 | Autenticación de usuarios a través del Servicio Central de Autenticación UC (CAS).  Auditoría y registro de todas las operaciones críticas en el sistema.  Los accesos de red solicitados vía ssh, tendrán alcance para el equipo desarrollador y proveedor hasta ambiente de desarrollo UC, los despliegues de las versiones de software a ambientes de liberación y producción serán gestionados por el equipo de la Dirección de Informática UC. |

# HALLAZGOS ENCONTRADOS EN LA ARQUITECTURA DURANTE LA REVISIÓN

|  |  |  |
| --- | --- | --- |
| **Tipo de requisito** | **Fecha** | **Descripción** |
| Protección de datos | 07/08/2024 | En la integración con SSO UC, de requerir algún otro atributo de cuenta UC diferente a uid, estos deben ser visados por protección de datos UC. |
| Integración a Base de datos de gestión UC | 07/08/2024 | No es factible establecer una conexión a la base de datos BDGES. Esto independiente del tipo de conexión: directa, vía api o algún otro mecanismo de conexión.  Debe incorporar componente(s) alternativo(s) a BDGES para la ingesta de datos que requiere la aplicación. |
| Arquitectura de aplicación | 07/08/2024 | No hay correspondencia entre los componentes informados en el diagrama de arquitectura de la aplicación y Diagrama de arquitectura de tecnología, en el primero se muestra la componente DBGES que no se muestra en el segundo diagrama como tecnología a ocupar. |
| Arquitectura de tecnología | 07/08/2024 | Se debe enriquecer el diagrama, incorporando las rutas url/uri y protocolos de comunicación entre componentes.  Se debe regularizar el diagrama para que represente los componentes reales a ocupar.  Dado que no es factible la conexión a la Base de BDGES debe actualizar la arquitectura para desincorporar el componente BDGES e incorporar los componentes alternativos para la ingesta de datos.  Para las componentes: UC CAPP (seguimiento curricular) y Cubes (data warehouse) debe describir su implementación, alcance y mecanismo de integración. |
| Servicio de Email | 07/08/2024 | Dada la estimación de envío de correos diarios, se sugiere la implementación de un servicio de correo masivo en donde el transporte de correo sea una capacidad externa al aplicativo propuesto. |
| Infraestructura solicitada **Onpremise** | 07/08/2024 | Esta solución es candidata a ser alojarla en plataforma **Azure** dado sus características:  Virtualización de máquinas a través de vmware y uso de base de datos MySQL. |

# OBSERVACIONES

* Conexión directa a base de datos de Gestión UC BDGES, **no es factible**, debe darse solución a este hallazgo en la arquitectura para poder otorgar recursos solicitados y dar avance al desarrollo del proyecto.

Se debe evaluar mecanismo alternativo que permita la ingesta de datos que requiere la aplicación.

De no tener mecanismo alternativo y requerir una excepción a la observación encontrada, esta debe ser solicitada personalmente por el director de Transformación Dirección Digital UC a su homologo Diego Biscar director de la Dirección de informática UC.

De lo acordado en la sesión antes mencionada, debe generarse un documento con el visado de la excepción y un documento de compromiso de adecuación de la arquitectura del aplicativo que dé solución a la observación encontrada. Ambos documentos deben estar firmados por el director de Transformación Digital UC, como entidad que da visado a la excepción y su homólogo el director de la Dirección de Informática UC, entidad que recibe el software con la observación encontrada. El documento de compromiso debe indicar fecha en corto plazo de la adecuación a implementar, así como la priorización de esta modificación en el backlog del aplicativo.

Las gestiones para el visado de la excepción y generación de documento de compromiso están a cargo del jefe de proyecto de la Dirección de Transformación Digital UC.

* Dado las características de la aplicación; virtualización de máquinas de aplicación a través de vmware y uso de base de datos MySQL, es factible técnicamente alojar esta aplicación en plataforma Azure, pudiéndose otorgar los recursos solicitados en esta plataforma bajo tecnología de virtualización.

Los recursos en plataforma Azure suponen un costo, a través del responsable del proyecto por la Dirección de Informática UC, podrá tener detalles e información asociados a los costos.

* Se debe regularizar el diagrama como los componentes reales a ocupar.

# RECOMENDACIONES

* Implementar un sistema de monitoreo y alertas para detectar y responder a problemas de rendimiento y disponibilidad.
* Realizar pruebas de carga y estrés para asegurar que el sistema puede manejar el crecimiento proyectado, considerar para estas pruebas definiciones de carga real para el servicio API UC Foto Personas UC y aplicativo propuesto.
* Evaluar la posibilidad de migrar a una arquitectura de microservicios para mejorar la escalabilidad del aplicativo.

# DOCUMENTOS RELACIONADOS

* Matriz de roles y responsabilidades (RACI) integración Plataforma CAS
* Matriz de roles y responsabilidades (RACI) integración Plataforma API UC

# DEFINICIONES Y ACRÓNIMOS

**UC:** Pontificia Universidad Católica de Chile.

**DI:** Dirección de Informática.

**DTDF:** Dirección de Transformación Digital UC.

**SSO:** Single Sing-On (inicio de sesión único).

**CAS:** Servicio Central de Autenticación UC

**API:** Application Programming Interface (Interfaz de programación de aplicaciones)

**DB:** Base de datos

**MER:** Modelo entidad-relación

**ORM:** Mapeo Relacional de objetos

**TI**: Tecnologías de la Información

**RACI**: Responsible, Accountable, Consulted, Informed

(Matriz de asignación de responsabilidades)

**Protocolo**: un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.

**Data Warehouse**: un almacén de datos o repositorio de datos es una colección de datos orientada a un determinado ámbito, integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

**API Gateway:** gestiona todas las tareas relacionadas con la aceptación y el procesamiento de centenares de miles de llamadas simultáneas a la API. Estas tareas incluyen la administración del tráfico, el control de la autorización y el acceso, el monitoreo y la administración de versiones de la API.

**File System**: Un sistema de archivos o sistema de ficheros, en informática, es un elemento que controla cómo se almacenan y recuperan los datos.

# HISTORIAL DE REVISIÓN

|  |  |  |
| --- | --- | --- |
| **Revisión** | **Fecha** | **Descripción del cambio** |
| R001 | 0/08/2024 | Versión inicial |
|  |  |  |
|  |  |  |
|  |  |  |