**Documentación Técnica del Proyecto**

**“Diseño de Arquitectura y Lineamientos del Desarrollo del Software”**

**Fecha 24/02/2025**

Autor: INAP

Fecha de creación: 08 de agosto, 2022

Última modificación: 08 de agosto, 2022

Documento de Referencia: N/A

Versión: V.1

**Contenido**

Contenido

[1.0 Arquitectura de sistemas 3](#_Toc193207596)

[2.0 Estructura de la base de datos. 5](#_Toc193207597)

[3.0 Estructura de Microservicios. 6](#_Toc193207598)

[4.0 Diagrama entidad relación de la Base de datos. 7](#_Toc193207599)

[5.0 Instancias de Desarrollo, QA y Producción. 8](#_Toc193207600)

[5.1 Instancia de Desarrollo 8](#_Toc193207601)

[5.2 Instancia QA 8](#_Toc193207602)

[5.3 Instancia de Producción 9](#_Toc193207603)

[6.0 Lineamientos para el diseño de la base de datos. 10](#_Toc193207604)

[7.0 Lineamientos para el desarrollo del Software. 12](#_Toc193207605)

[8.0 Tecnologías sugeridas por TI de la Secretaría de Finanzas. 13](#_Toc193207606)

[8.1 Base de datos: 14](#_Toc193207607)

[8.2 Sugerencias Framework, lenguaje de programación, SO. 14](#_Toc193207608)

[8.3 Versiones de Software definidos para el desarrollo: 15](#_Toc193207609)

[8.4 Sistema de control de versiones: 15](#_Toc193207610)

[8.5 Configuración de ambientes de desarrollo. 16](#_Toc193207611)

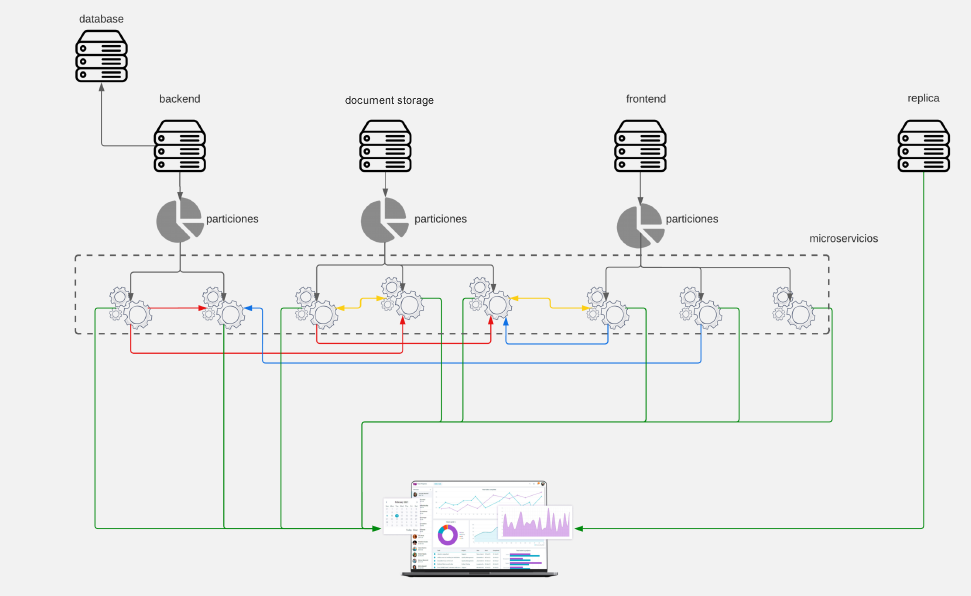
[8.6 Micro servicios: 16](#_Toc193207612)

[8.7 Administración del seguimiento del proyecto: 16](#_Toc193207613)

# 1.0 Arquitectura de sistemas

El **Portal PPAFE** estará basado en una arquitectura web modular con un enfoque en seguridad, escalabilidad y disponibilidad.

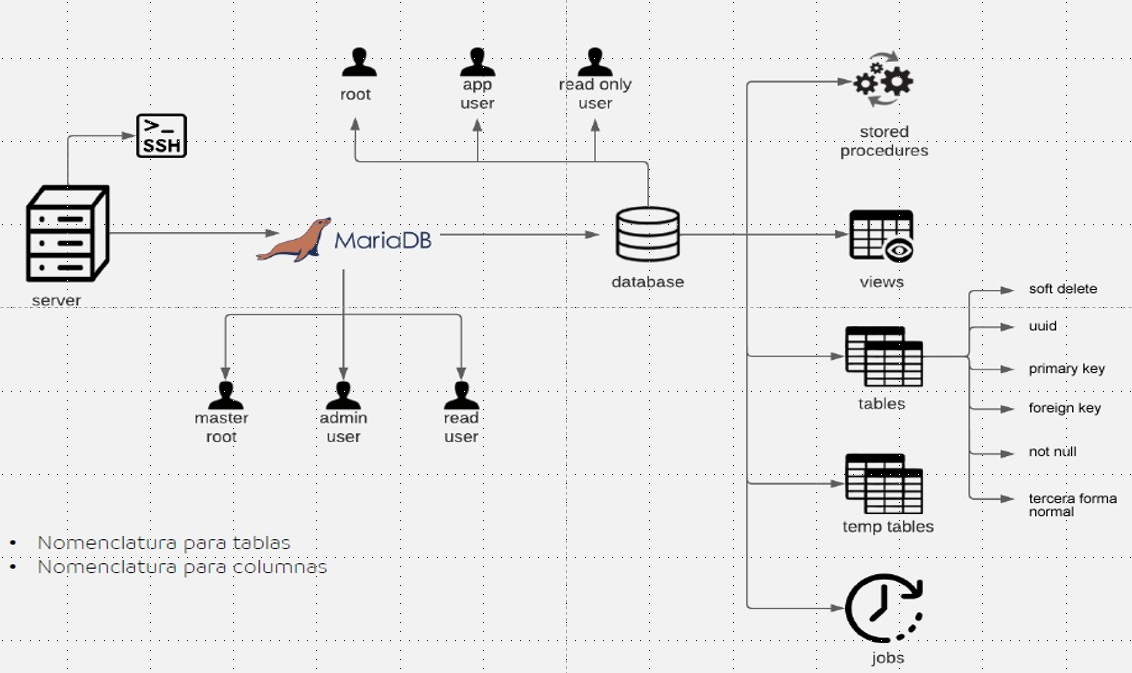
* **Frontend:** Aplicación web responsiva basada en tecnologías modernas para visualización de datos financieros.
* **Backend:** API RESTful para la gestión de datos y autenticación.
* **Base de Datos:** Motor de base de datos optimizado para almacenamiento y consultas de alta demanda.
* **Seguridad:** Implementación de autenticación, permisos y cifrado de datos sensibles.
* **Despliegue en servidores Linux**, optimizados para carga de información financiera.



# 2.0 Estructura de la base de datos.

La base de datos será diseñada para garantizar la **integridad de los datos** y la **eficiencia en las consultas**, contemplando:

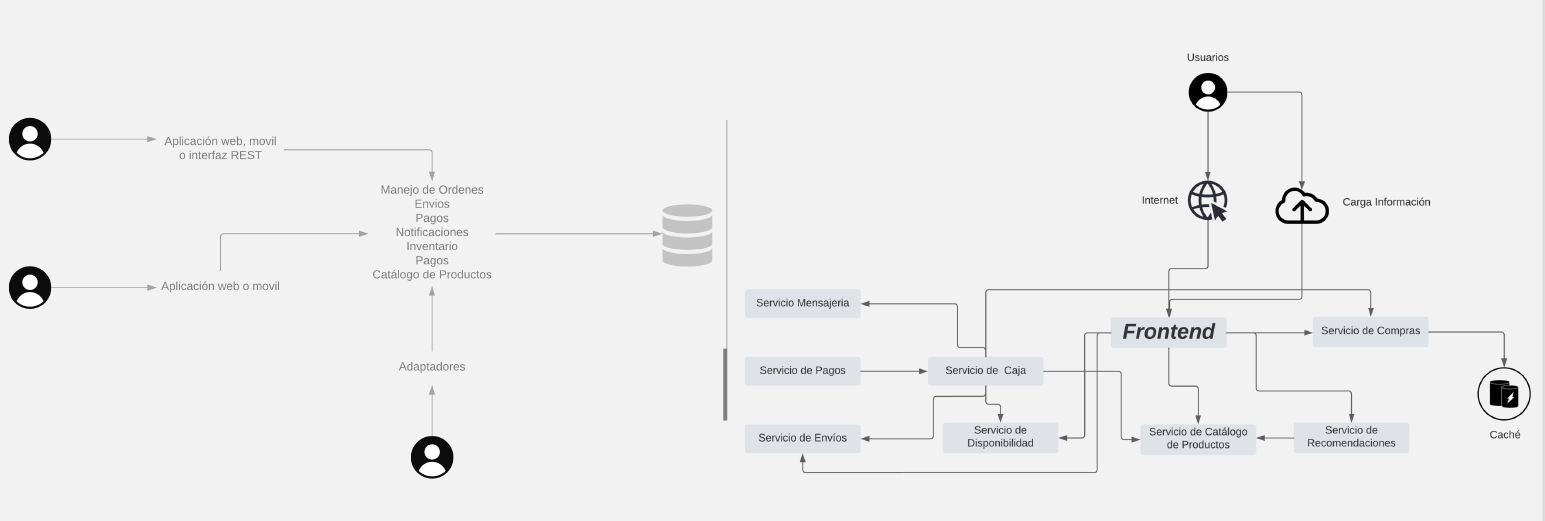
* Tablas organizadas por módulos funcionales: **Montos, Acuerdos, Cálculos, Usuarios, Roles, Catálogos**.
* Uso de **índices optimizados** para consultas de montos por municipio, año y fondo.
* Implementación de **relaciones entre tablas** mediante claves foráneas.
* Manejo de trazabilidad y control de cambios en los registros.



# 3.0 Estructura de Microservicios.

El sistema utilizará una **arquitectura basada en microservicios** para garantizar escalabilidad y modularidad:

* **Servicios de Datos:** Manejo de información sobre montos, acuerdos y fórmulas de cálculo.
* **Servicios de Seguridad:** Autenticación, gestión de permisos y auditoría.
* **Servicios de Reportes:** Generación de documentos descargables en PDF y Excel.
* **Servicios de Administración:** Carga y validación de datos desde el panel administrativo.



# 4.0 Diagrama entidad relación de la Base de datos.

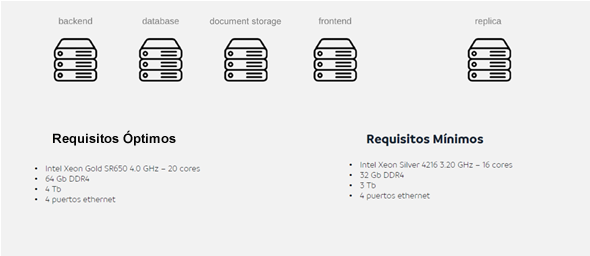
Se implementará un **modelo de datos relacional**, con integridad referencial para asegurar la consistencia de la información.

**Entidades clave:** Fondos, Municipios, Montos, Acuerdos, Cálculos.

**Relaciones:** Definición de claves foráneas para asegurar la correcta vinculación entre datos.

# 5.0 Instancias de Desarrollo, QA y Producción.

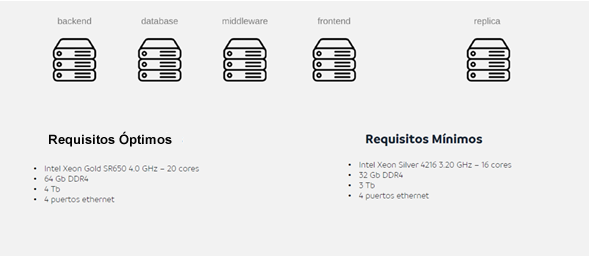
## 5.1 Instancia de Desarrollo



## 5.2 Instancia QA



## 5.3 Instancia de Producción



# 6.0 Lineamientos para el diseño de la base de datos.

**Usuarios**

* Supe usuario – root
* Usuario App – admin
* Usuario Consulta – consulta

**Nombre de Base de Datos**

* Siglas o nombre del proyecto sin exceder 20 caracteres, 1 BD por aplicación, mayúsculas.

**Nomenclatura Tablas**

* Nombre descriptivo no mayor a 20 caracteres y no menor a 6 caracteres, si son palabras compuestas usar PascalCase, plural, sin carácter especial.

**Nomenclatura Columnas**

* Nombre descriptivo no mayor a 15 caracteres y no menor a 6 caracteres, si son palabras compuestas usar PascalCase, singular, sin carácter especial.

**Llaves Primarias**

* Obligatorias

**Llaves Foráneas**

* Obligatorias

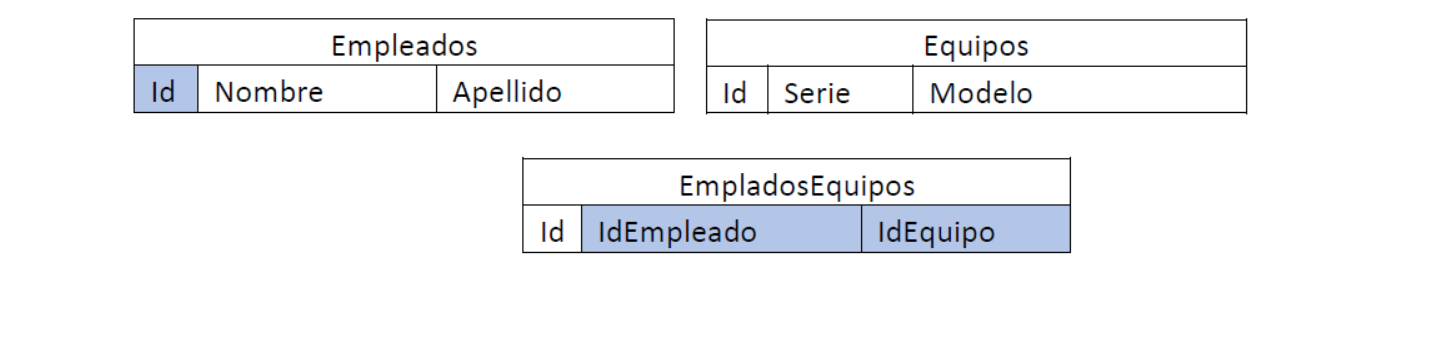
**Llaves Primarias Compuestas**

* Opcionales

**Tablas de Relación**

* Obligatorias

**Ejemplo.**



**Respaldo Base de Datos**

* 02:00 am

**Duplicar Datos**

* TBD

**Estructura de Tablas**

**Eliminado Lógico**

* Obligatorio, nombre de campo: activo, deleted, st. Tipo BOOLEAN, BOOL, BIN

**Última Actualización**

* Obligatorio, DATETIME.

**Fecha Creación**

* Obligatorio, DATETIME.

**Modificado Por**

* Obligatorio, IdUsuario

**Creado Por**

* Obligatorio, IdUsuario

**Store Procedures**

* Obligatorio para transacciones, nomenclatura sp\_{descripción}, no mayor a 16 caracteres.
* Parámetros
* Input: abreviación tipo de variable + nombre
* Output: abreviación tipo de variable + nombre

**Jobs**

* Opcional, nomenclatura jb\_{descripción}, no mayor a 16 caracteres.

**Triggers**

* Opcional, nomenclatura tr\_{descripción}, no mayor a 16 caracteres.

**Funciones**

* TBD

**Idioma**

* Español

**Passwords**

* Encriptados
* MD5, Hash. TBD

**Id**

* UUID CHAR(36)

**Ficheros (IMAGEN, PDF)**

* BLOB, VARBINARY. Base64, repo tradicional. Tablas Independientes.

**Select \***

* No aplica

**Not Null**

* Obligatorio para todas las columnas

**Comodín Like**

* TBD

**Evitar Operador desigualdad <>,!, not**

* TBD

Evitar uso de queries, creación de sp’s directo de back o front.

Realizar manipulación de cadenas, concatenaciones, etc. En back o front.

# 7.0 Lineamientos para el desarrollo del Software.

Esta guía técnica entrega los lineamientos generales y recomendaciones específicas que debe seguir todo equipo que desarrolle software al interior de la Administración del Estado, contribuyendo a la construcción de sistemas de alta calidad en todas las instituciones.

1. Hacer uso de librerías y frameworks de seguridad
2. Consultas seguras a las bases de datos
3. Validar todos los datos de entrada desde el front, back y en la base de datos.
4. Codificar y escapar los datos
5. Validación sintáctica de datos
6. Implementar mecanismos de autenticación seguros
7. Uso de local storage para manejo de sesión
8. Uso de tokens de sesión
9. Uso de TLS y SSL
10. Archivo de variables de entorno excluido de las versiones.
11. Log en el código
12. Calidad de código (Codeclimate, Sonarqube ).
13. Manejo seguro de errores y excepciones
14. Paradigma de la programación orientada a objetos y utilizando un patrón de arquitectura modelo-vista-controlador (MVC).
15. Tareas fuera de línea o programadas: Para las tareas asíncronas o que tengan la característica de una tarea programada diaria o de tiempo definido, éstas deben usar tecnologías de encolamiento, como las que se mencionan en Tecnologías de preferencia, y el consumo de las colas debe ser con procesos que no tengan relación con el sitio y que sea posible desarrollarlos e implementarlos como un servicio separado.

# 8.0 Tecnologías sugeridas por TI de la Secretaría de Finanzas.

Para cada proyecto se recomienda el uso de ciertas tecnologías de desarrollo y arquitectura del software, basado en las recomendaciones del equipo de infraestructura y del equipo de desarrollo de la División de Gobierno Digital.

## 8.1 Base de datos:

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Versiones** | **Comentarios** |
| Mysql/MariaDB | 10.3.21 |  |
| Mongo | La más reciente estable  5.0.8 | Compatible con Centos 7 ,8 u 8.2 |

## 8.2 Sugerencias Framework, lenguaje de programación, SO.

|  |  |  |
| --- | --- | --- |
| **Tipo** | **Versiones** | **Comentarios** |
| Laravel | 9.x | Para trabajar con Laravel es necesario trabajar con una versión mayor a la 7 de PHP |
| NEXT.JS | 12.x | Para trabajar con Next JS y/o VUE JS es recomendable utilizar alguna versión mayor a 12.x de Node JS. Es recomendable utilizar la última versión de Node, NPM y para el manejo de librerías es mejor utilizar yarn en lugar de npm. La versión de react recomendable es la versión 12.x |
| VUE | >=2.x | Si se quiere trabajar en integración con Laravel es recomendable utilizar la versión 2.x ya que la versión 3.x tiene algunos |
| PHP | >=7.x |  |
| Rocky Linux | >9.0 |  |

## 8.3 Versiones de Software definidos para el desarrollo:

|  |  |  |
| --- | --- | --- |
| **Tipo** |  | **Versión** |
| Sistema Operativo | Rocky Linux | 9 |
| Base de datos | MariaDB | Mariadb Ver 15.1 Distrib 10.8.4-MariaDB, for Linux (x86\_64) using readline 5.1 |
| Servidor Web | Apache | apache Server versión: Apache/2.4.6 (CentOS) |
| Lenguaje del servidor | PHP | PHP 8.1.11 (cli) (built: Sep 28 2022 09:08:05) (NTS gcc x86\_64) |
|  | Node | Node v16.18.0 |
| Framework de desarrollo | Laravel | Laravel 9.35.1 |

## 8.4 Sistema de control de versiones:

Git y GitHub.

## 8.5 Configuración de ambientes de desarrollo.

Docker

## 8.6 Micro servicios:

Desarrollo y consumo de API’s.

## 8.7 Administración del seguimiento del proyecto:

Jira Software

## 

|  |  |
| --- | --- |
| **ELABORA / REVISA** | **AUTORIZA** |
| Ing. Alberto Sobrado Garnica  Gerente General de Proyecto INAP | Ing. Néstor Ibarra Palomares  Coordinación de Gestión de Información de la SFyTGE |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FIN DEL DOCUMENTO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***