



**SERVICIO NACIONAL DE APRENDIZAJE SENA**

**INFORME DESARROLLLO DE SOFTWARE  
SAN JOSÉ DEL GUAVIARE, 2023**



## Tabla de contenido

INTRODUCCION.....	3
FASES DEL ANALISIS.....	4
REQUERIMIENTOS.....	4
REQUERIMIENTOS FUNCIONALES.....	4
Requerimientos no funcionales.....	4
DIAGRAMA DE CLASES .....	5
DIAGRAMA MER (MODELO ENTIDAD RELACION) .....	6
DIAGRAMA DE CASOS DE USO .....	7
Documentación diagramas de caso de uso .....	7
CODIGO FUENTE .....	10



## INTRODUCCION

El objetivo de este proyecto es desarrollar un software que permita a un administrador gestionar la información de aprendices de manera eficiente. La aplicación debe permitir almacenar datos básicos como lo son tipo de documento, número de documento, nombres, apellidos, fecha de nacimiento, sexo, teléfono y dirección.



## **FASES DEL ANALISIS.**

- Identificar los requerimientos del cliente.
- Desarrollar los diagramas de clase, diagramas de casos de uso y diagrama MER.
- Desarrollar el modelo vista controlador.
- Desarrollar las validaciones.
- Documentación.

## **REQUERIMIENTOS**

### **REQUERIMIENTOS FUNCIONALES**

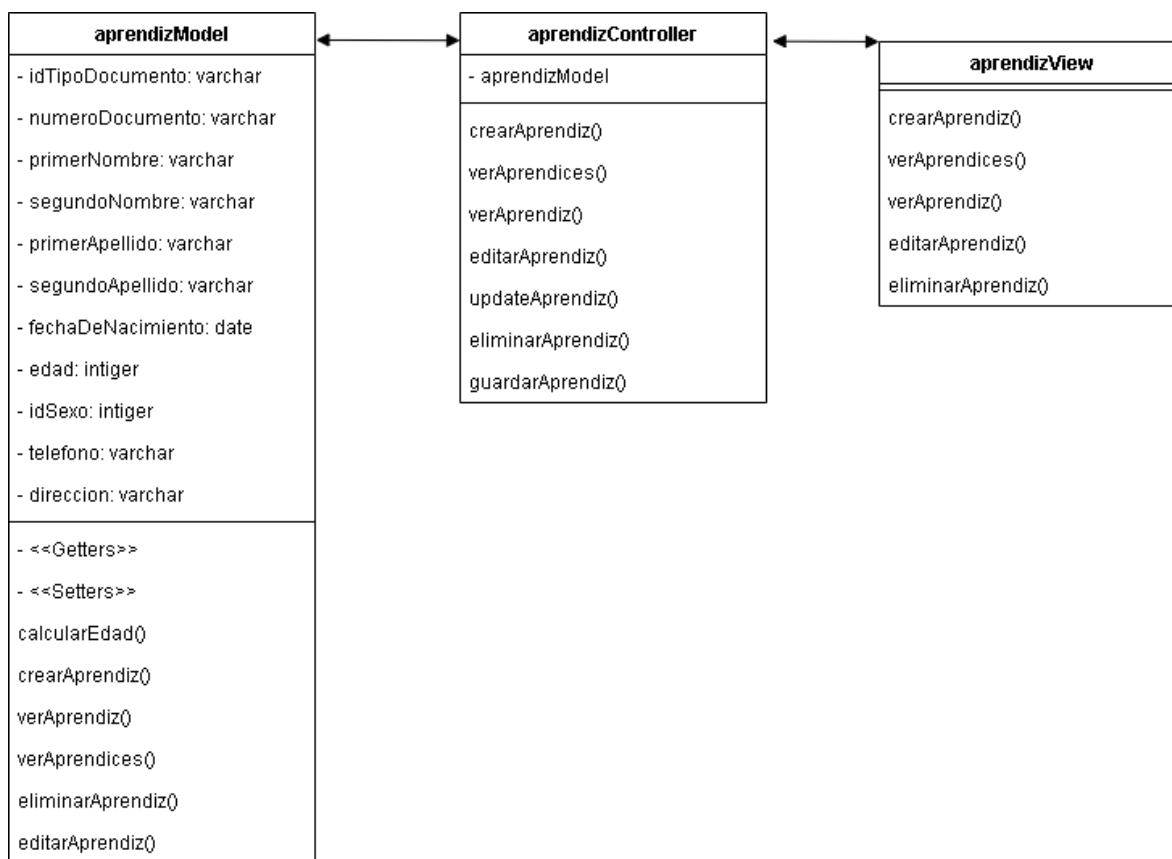
- El software debe permitir gestionar la información de los aprendices Sena de manera eficiente y poder recopilar la información básica (tipo de documento, numero de documento, nombres, apellidos, fecha de nacimiento, sexo, teléfono y dirección.)
- El software debe implementar las funcionalidades CRUD (create-crear, read-leer, update-actualizar, delete-eliminar) en el registro de los aprendices.
- El software debe realizar validaciones para garantizar la integridad de los datos.
- El software debe contar con formularios intuitivos para la creación de aprendices y la actualización de estos.
- El software debe contar con una página para mostrar todos los aprendices registrados.
- El software debe contar con la funcionalidad de actualizar, ver o eliminar cada aprendiz de la lista.

### **Requerimientos no funcionales**

- El backend debe estar desarrollado con PHP.
- La gestión de base de datos debe ser con MySQL.
- El software debe estar diseñado en MVC (Modelo, Vista Controlador)
- La base de datos se llamará sena\_db.
- La tabla aprendices se diseñará con los campos necesarios para almacenar la información requerida.
- El software debe implementar vistas responsivas.

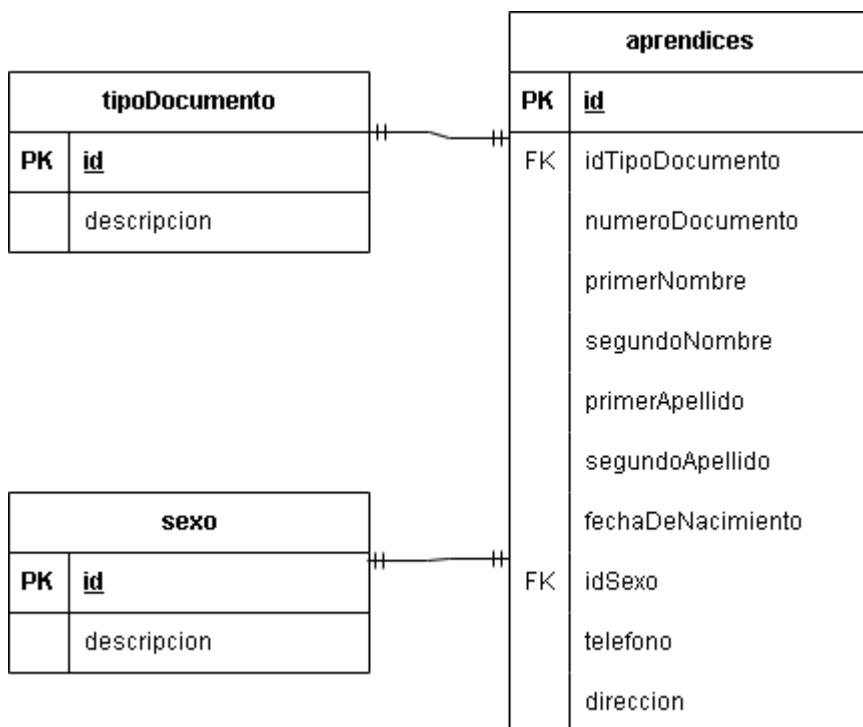


## DIAGRAMA DE CLASES

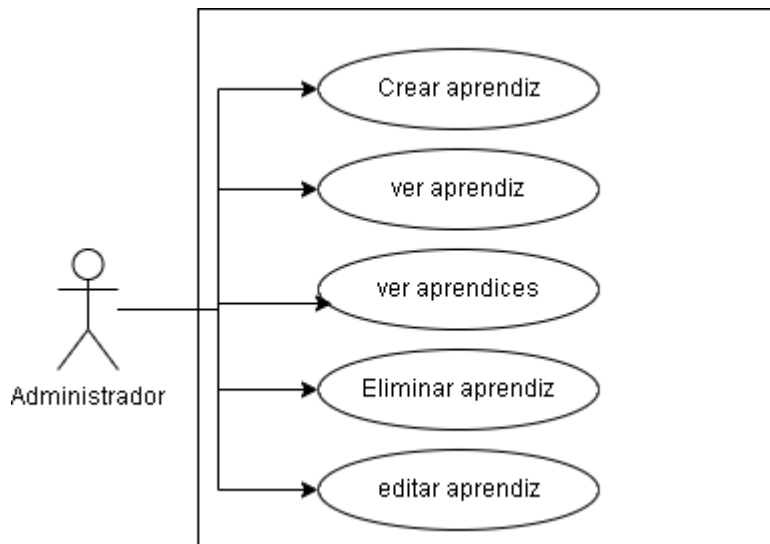




## DIAGRAMA MER (MODELO ENTIDAD RELACION)



## DIAGRAMA DE CASOS DE USO



### Documentación diagramas de caso de uso

#### Crear aprendiz

Caso de uso: crear aprendiz
ID: 1
Breve descripción: Permite al administrador crear a un aprendiz en la base de datos
<b>Actores:</b> Administrador.
<b>Precondiciones:</b> El aprendiz no debe estar registrado en la base de datos
<b>Flujo principal:</b> El administrador ingresa al formulario para crear aprendiz. El administrador suministra los datos para llenar el formulario. El administrador selecciona el botón de guardar aprendiz.
<b>Postcondiciones:</b> El sistema lo redirige a la página principal con la lista de aprendices registrados actualizada. El sistema crea una alerta de éxito.
<b>Flujo alternativo:</b> El sistema arroja una alerta de mala inserción de datos en el formulario. El sistema arroja una alerta por aprendiz ya registrado



## Ver aprendiz

Caso de uso: Ver aprendiz
ID: 2
Breve descripción: Permite al administrador ver toda la información del aprendiz.
<b>Actores:</b> Administrador.
<b>Precondiciones:</b> El aprendiz debe estar registrado en la base de datos
<b>Flujo principal:</b> 1. El administrador da clic al botón de ver, en la lista de los aprendices.
<b>Postcondiciones:</b> El sistema lo dirige a la página donde muestra toda la información registrada del aprendiz
<b>Flujo alternativo:</b> El administrador debe crear al aprendiz

## Ver aprendices

Caso de uso: Ver aprendices
ID: 3
Breve descripción: Permite al administrador ver una lista de los aprendices registrados.
<b>Actores:</b> Administrador.
<b>Precondiciones:</b> Deben haber aprendices registrados
<b>Flujo principal:</b> 1. El administrador da clic al botón de inicio
<b>Postcondiciones:</b> El sistema lo dirige a la página donde muestra la lista de aprendices registrados.
<b>Flujo alternativo:</b> El administrador debe crear aprendices.





## Eliminar aprendiz

Caso de uso: Eliminar aprendiz
ID: 4
Breve descripción: Permite al administrador eliminar a un aprendiz de la base de datos
<b>Actores:</b> Administrador.
<b>Precondiciones:</b> El aprendiz debe estar registrado en la base de datos
<b>Flujo principal:</b> 1. El administrador da clic al botón eliminar, en la lista de los aprendices.
<b>Postcondiciones:</b> El sistema elimina al aprendiz de la base de datos. El sistema actualiza la lista de aprendices. El sistema lanza una alerta de éxito.
<b>Flujo alternativo:</b> El administrador debe crear al aprendiz

## Editar aprendiz

Caso de uso: Editar aprendiz
ID: 4
Breve descripción: Permite al administrador editar la información del aprendiz.
<b>Actores:</b> Administrador.
<b>Precondiciones:</b> El aprendiz debe estar registrado en la base de datos
<b>Flujo principal:</b> 1. El administrador da clic al botón editar, en la lista de los aprendices. 2. El sistema lo dirige a la página donde se encuentra el formulario con los datos del aprendiz editable. 3. El administrador actualiza la información del aprendiz. 4. El administrador da clic en el botón editar aprendiz.
<b>Postcondiciones:</b> El sistema actualiza la información del aprendiz en la base de datos. El sistema lo redirige a la página principal con la lista de aprendices registrados actualizada. El sistema crea una alerta de éxito.
<b>Flujo alternativo:</b> El sistema arroja una alerta de mala inserción de datos en el formulario.



## **CODIGO FUENTE**

[https://github.com/Williamlo21/sena\\_db.git](https://github.com/Williamlo21/sena_db.git)