



Tarea 1 Unidad 2

"Diseño de las interfaces, base de datos relacional del proyecto asignado y arquitectura de software"

INGENIERIA DE SOFTWARE

11:00 – 12:00 PM

Profesora:

NORMA REBECA GODOY CASTRO

Alumno:

Estrada Valenzuela Jesús Ernesto

Velasco González Diego

Flores Saldaña Martín Alejandro

Meza Quiñonez César Iván

Ibarra Lizárraga Abel Alejandro

7to semestre

ING. En Sistemas Computacionales

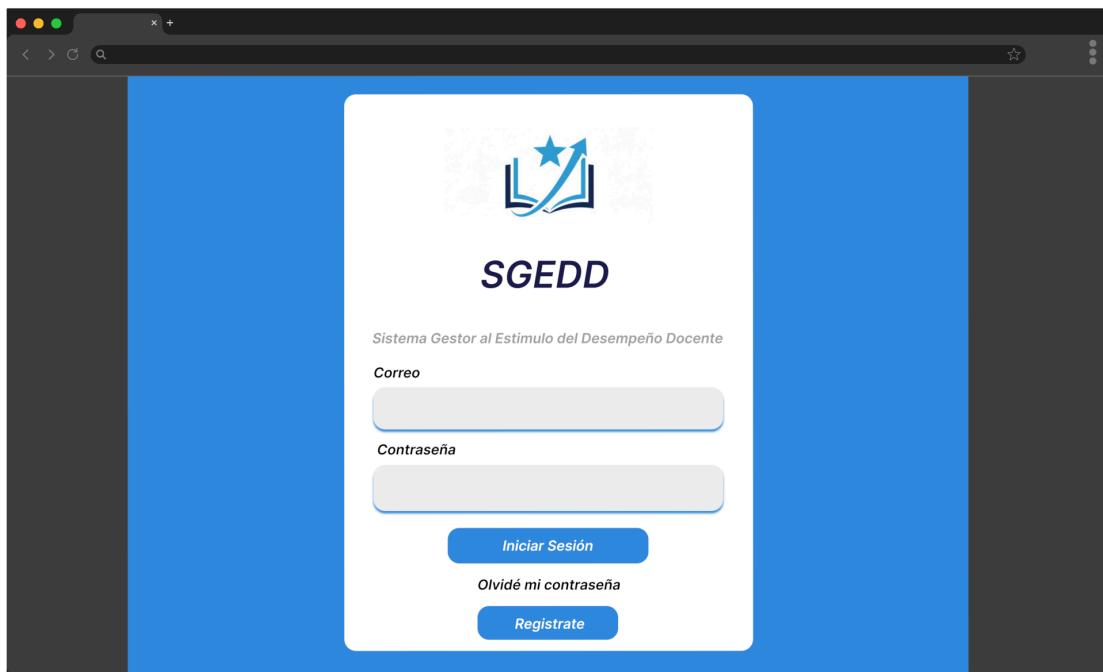
31/10/2025

Índice

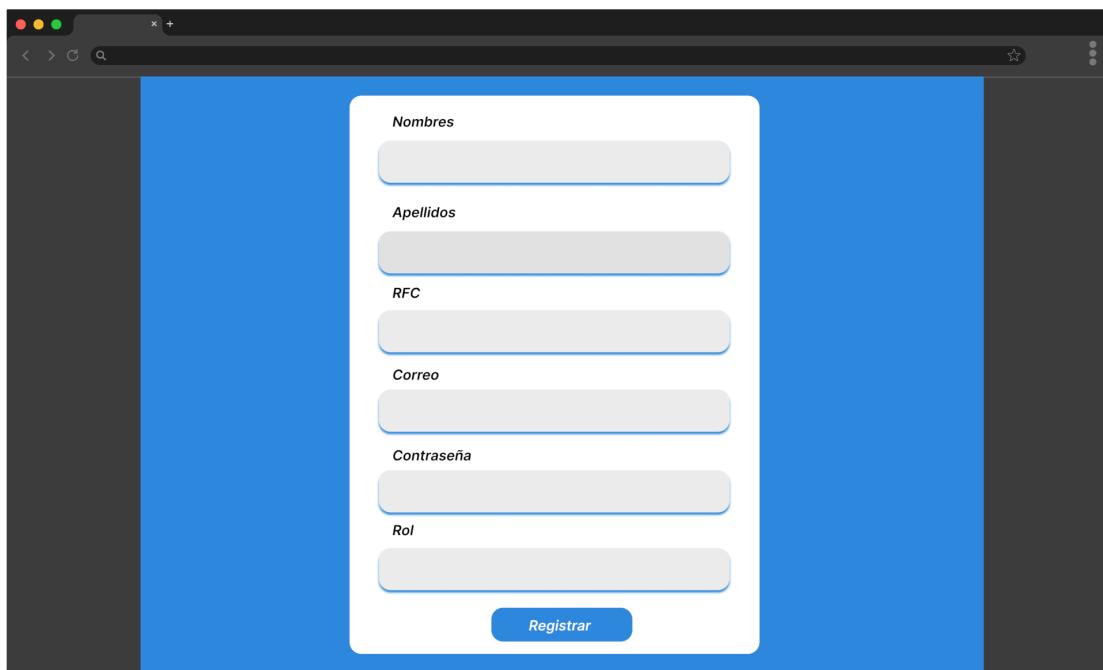
Diseño de interfaces.....	3
1. Login:.....	3
2. Registro.....	3
3. Registro invalido.....	4
4. Recuperación de contraseña.....	4
5. Inicio Docente.....	5
6. Periodo.....	5
7. Expediente - Docente.....	6
8. Expediente - Error en documento.....	6
9. Inicio Subdirección - Revisor.....	7
10. Solicitudes - Revisor.....	7
11. Revisión documento - Revisor.....	8
12. Lista solicitantes - Generador.....	8
13. Visor PDF en documento - Generador.....	9
14. Visor PDF en documento ComboBox - Generador.....	9
15. Notificaciones - Segundo plano.....	10
16. Notificaciones - Completa.....	10
17. Opciones - Segundo plano.....	11
18. Cambio de Contraseña.....	11
19. Soporte.....	12
20. Guia de uso.....	12
Base de Datos Relacional (Modelo Lógico).....	14
Diseño arquitectónico.....	15
¿Qué es la arquitectura monolítica?.....	15
Ventajas.....	15
Desventajas.....	16
¿Qué significa que sea una arquitectura monolítica modular?.....	16
Ventajas.....	16
Desventajas.....	17
¿Qué es la arquitectura cliente-servidor?.....	17
Ventajas.....	17
Desventajas.....	18

Diseño de interfaces

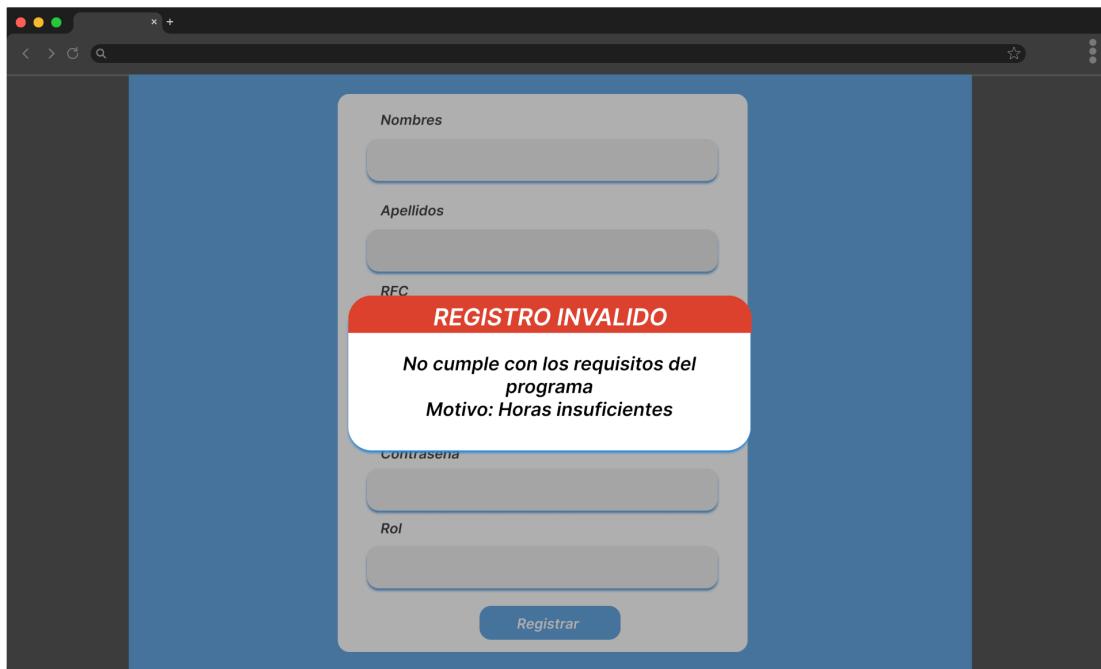
1. Login:



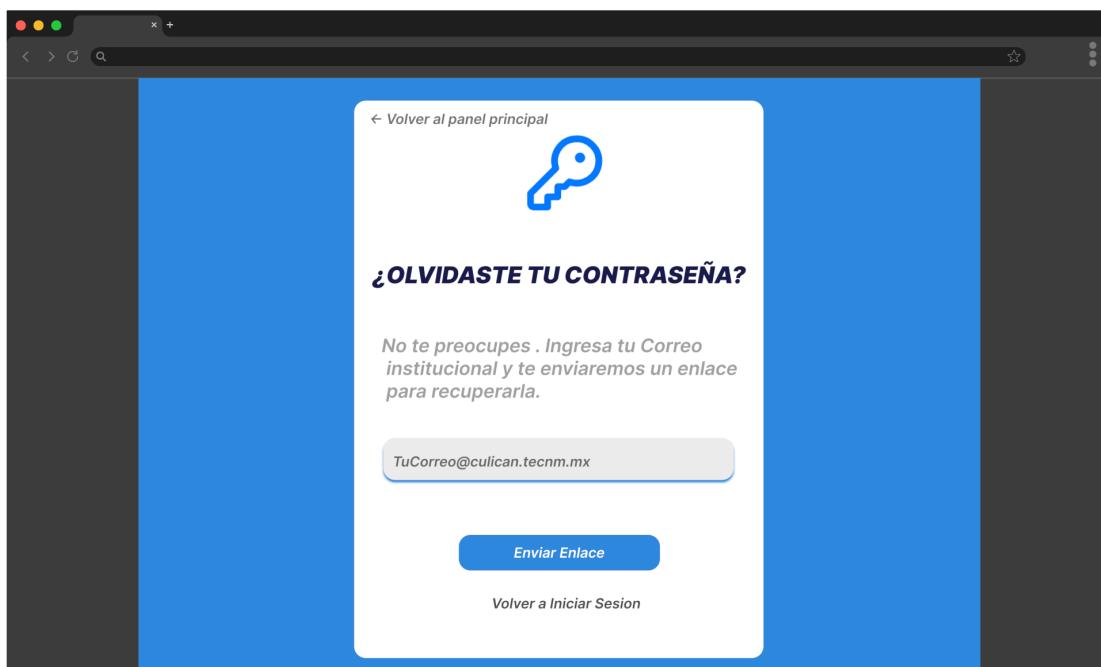
2. Registro



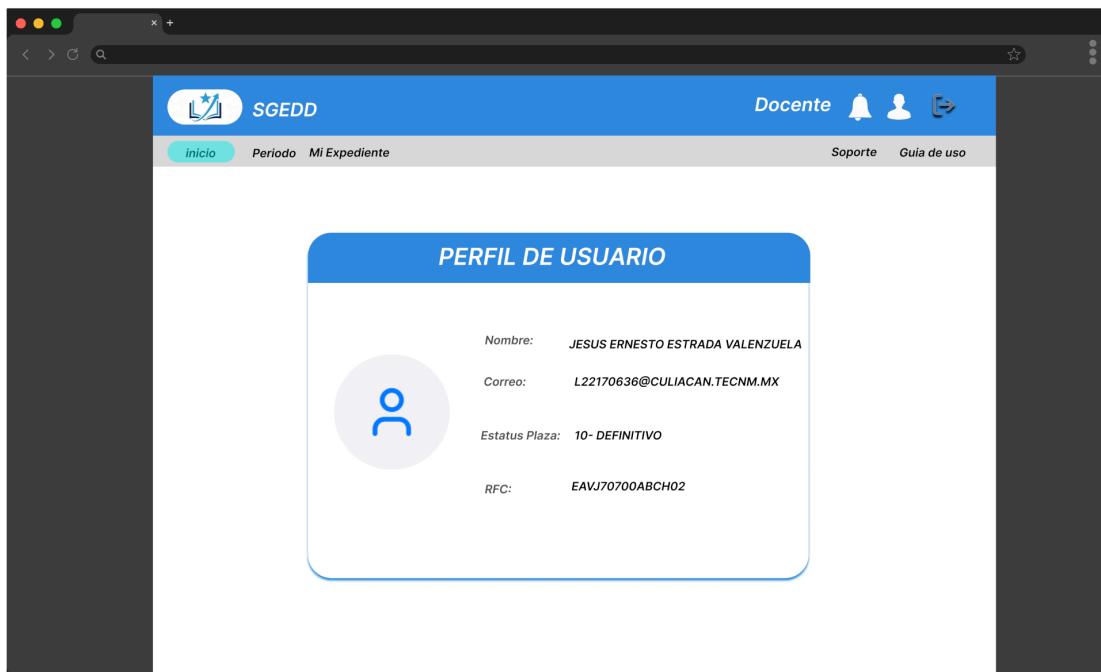
3. Registro invalido



4. Recuperación de contraseña



5. Inicio Docente



6. Periodo



7. Expediente - Docente

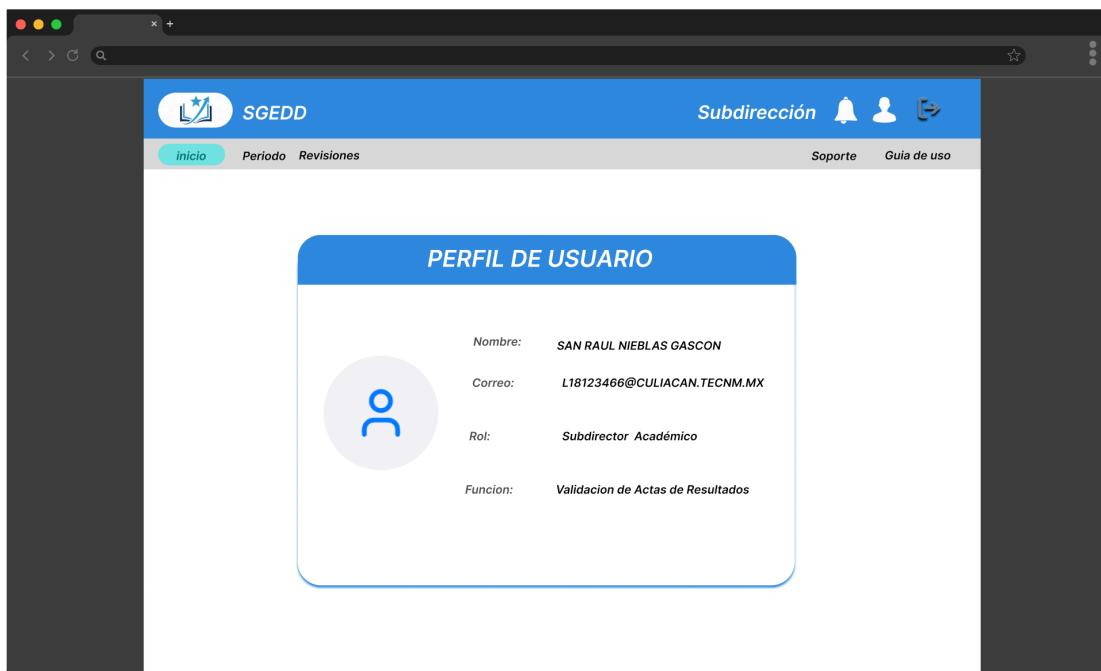
The screenshot shows the SGEDD Docente interface. At the top, there is a navigation bar with links for 'inicio', 'Periodo', and 'Mi Expediente'. On the right side of the header are links for 'Docente', 'Soporte', and 'Guia de uso'. Below the header, the main title 'MI EXPEDIENTE - PERIODO 2025' is displayed. A table lists four documents with their status, last update date, and actions:

DOCUMENTO	SOLICITAR	ESTADO	ULTIMA ACTUALIZACIÓN	ACCIÓN
Constancia de Recursos Humanos	[Icon]	Aprobado	24/10/2025	Ver PDF Reportar
Carta de Exclusividad Laboral	[Icon]	NoGen	23/10/2025	Ver PDF Reportar
Carta de liberacion de Actividades	[Icon]	Pendiente	20/10/2025	Ver PDF Reportar
Constancia de servicios escolares	[Icon]	Rechazado	18/10/2025	Ver motivos Reportar

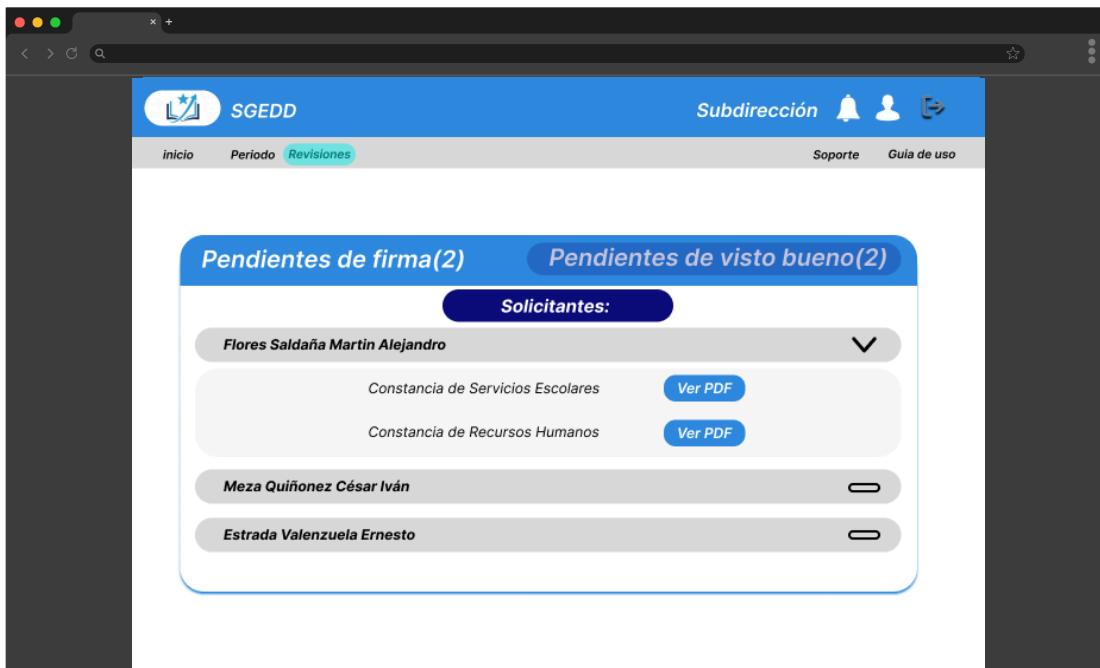
8. Expediente - Error en documento

The screenshot shows the SGEDD interface with the 'Comité Local' role selected. A modal window titled 'Reportar error en documento' is open. It contains a form with fields for 'Documento' (set to 'Constancia de servicios escolares') and 'Mensaje' (a text area with placeholder text 'Describe el problema presente en el documento...'). At the bottom of the modal is a blue button labeled 'Enviar Mensaje'.

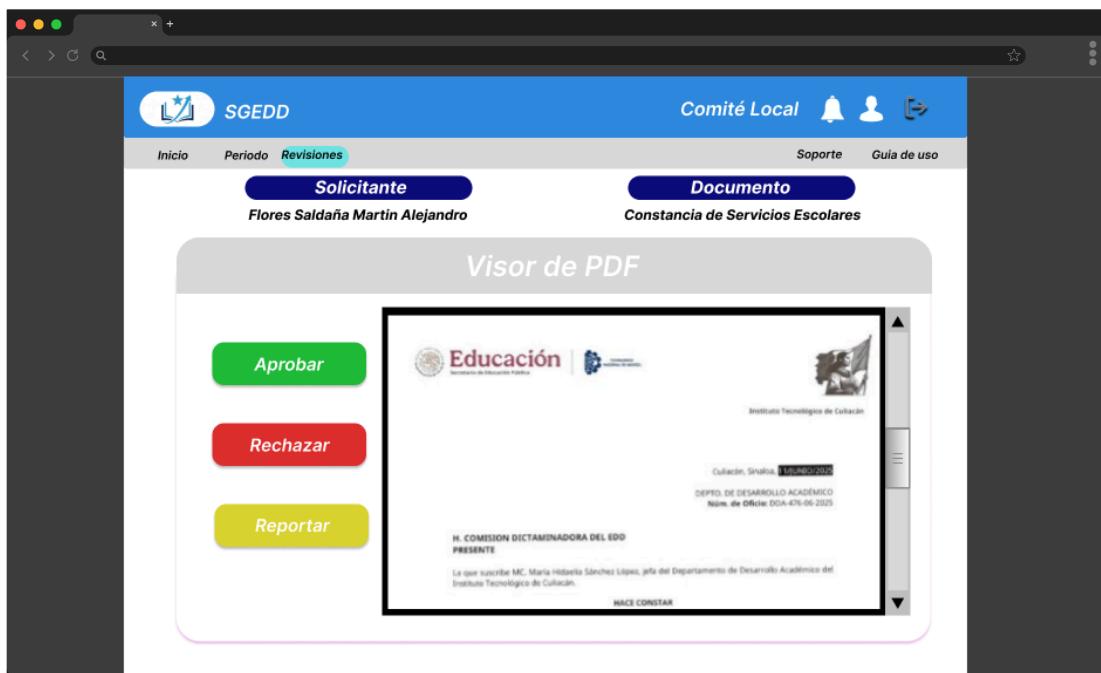
9. Inicio Subdirección - Revisor



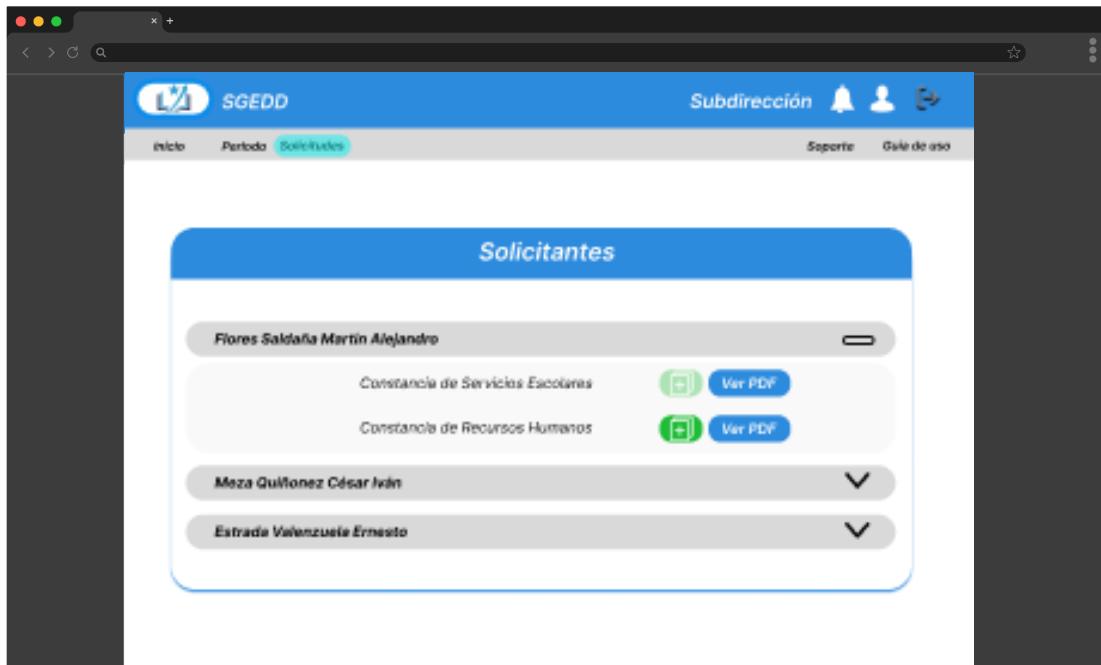
10. Solicitudes - Revisor



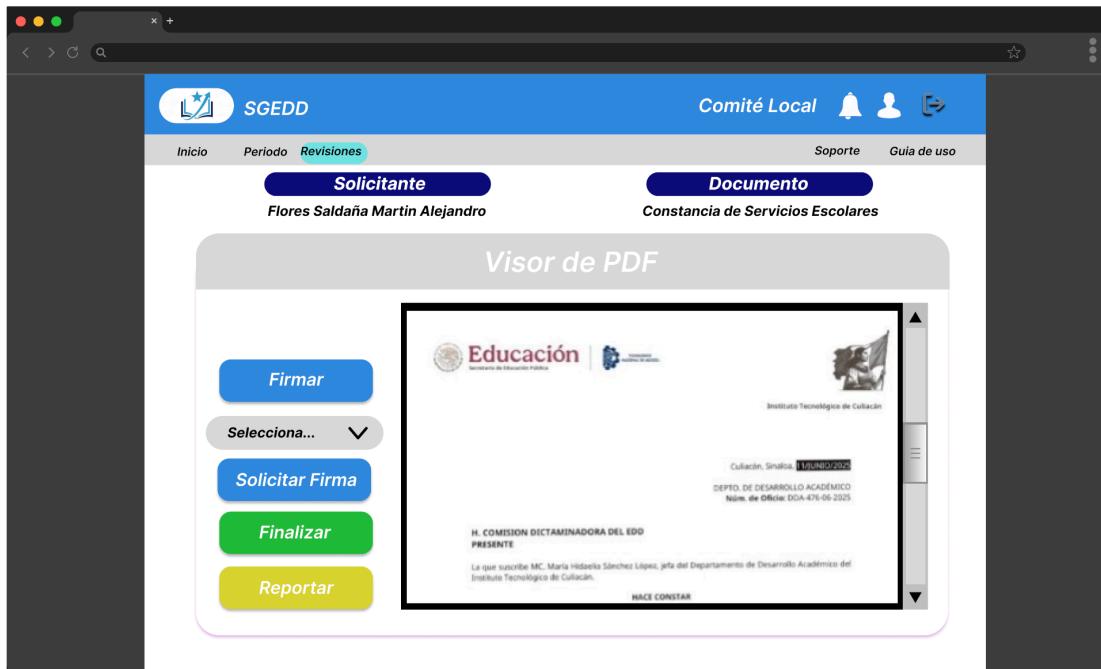
11. Revisión documento - Revisor



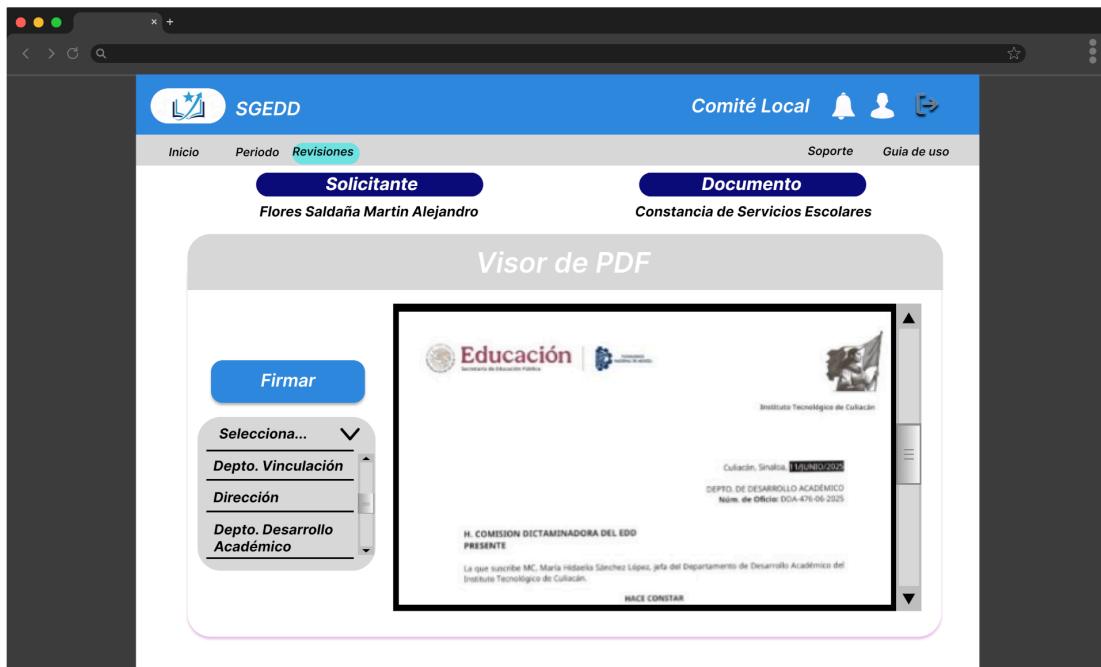
12. Lista solicitantes - Generador



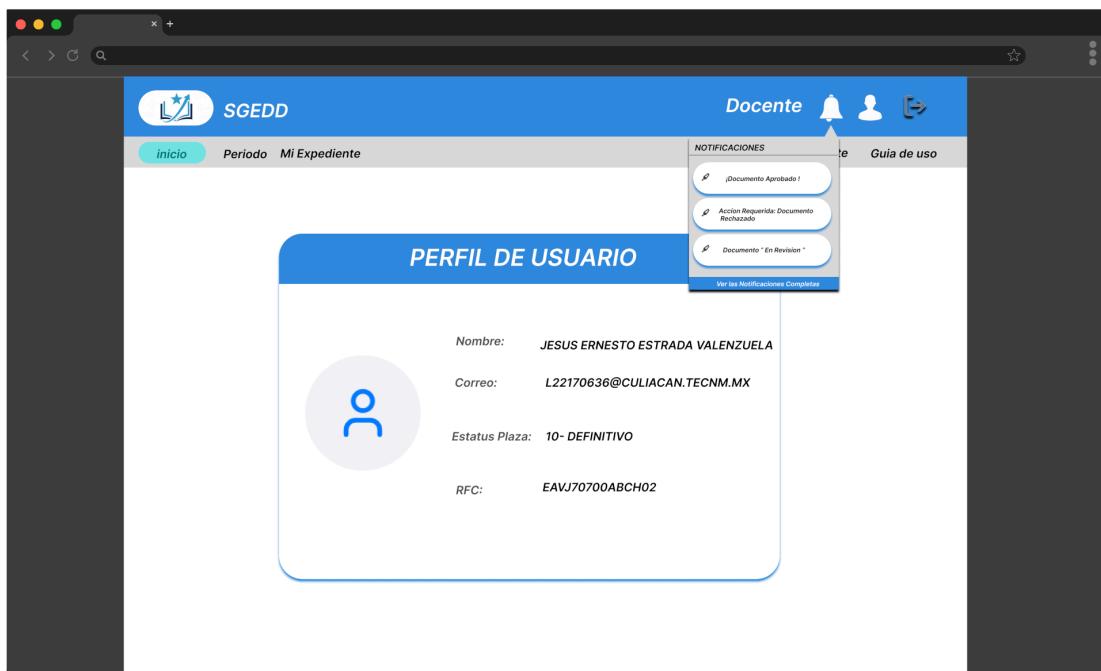
13. Visor PDF en documento - Generador



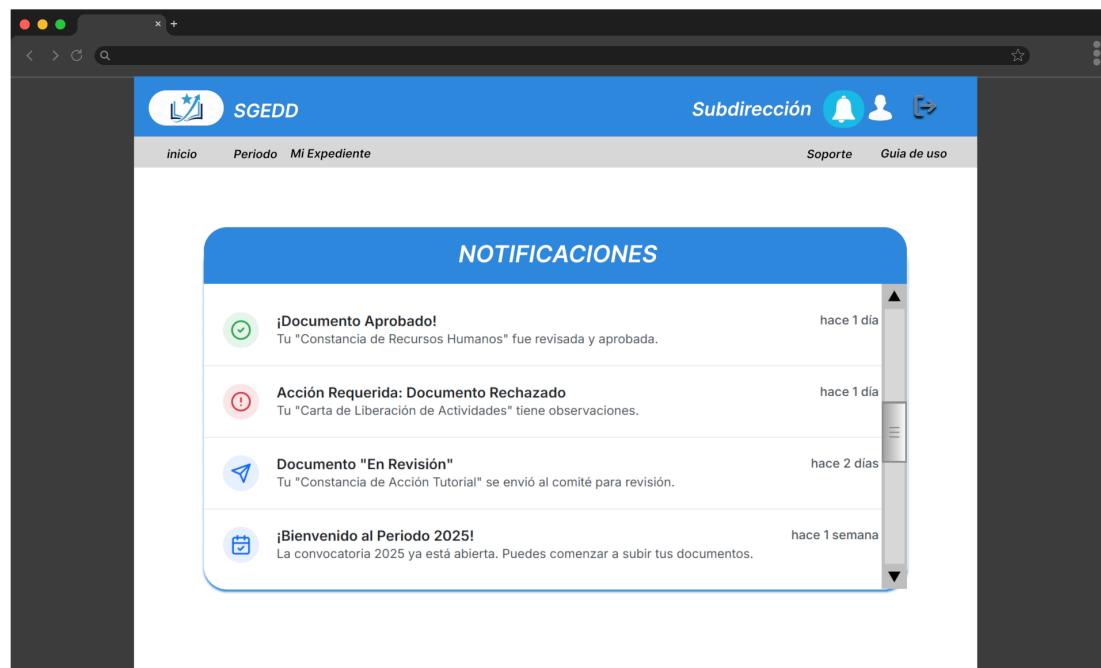
14. Visor PDF en documento ComboBox - Generador



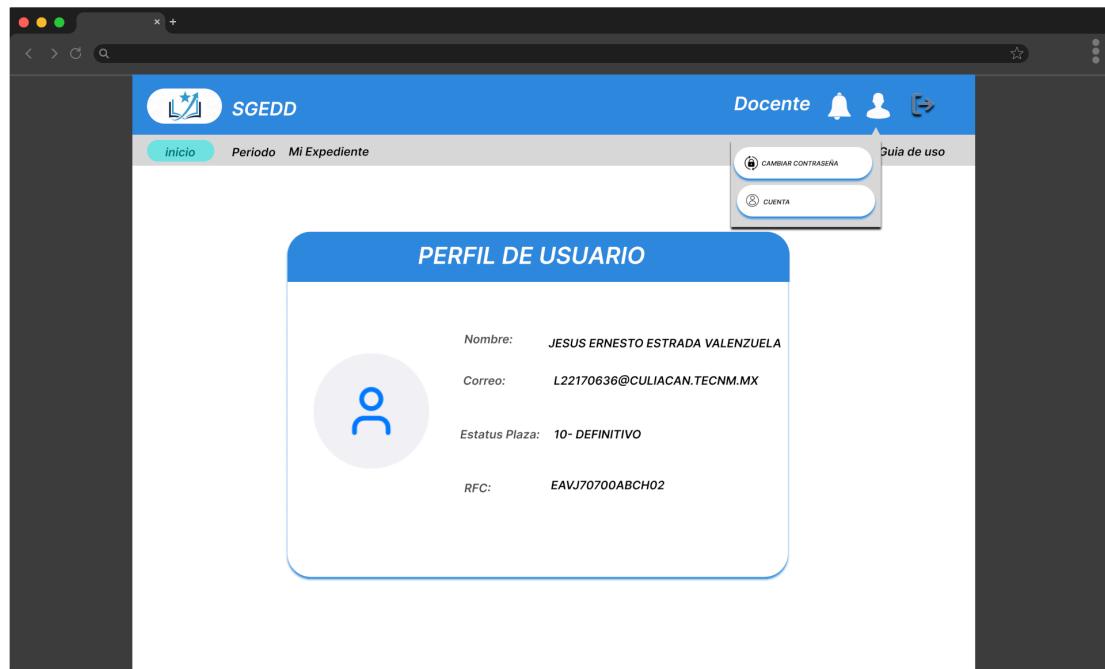
15. Notificaciones - Segundo plano



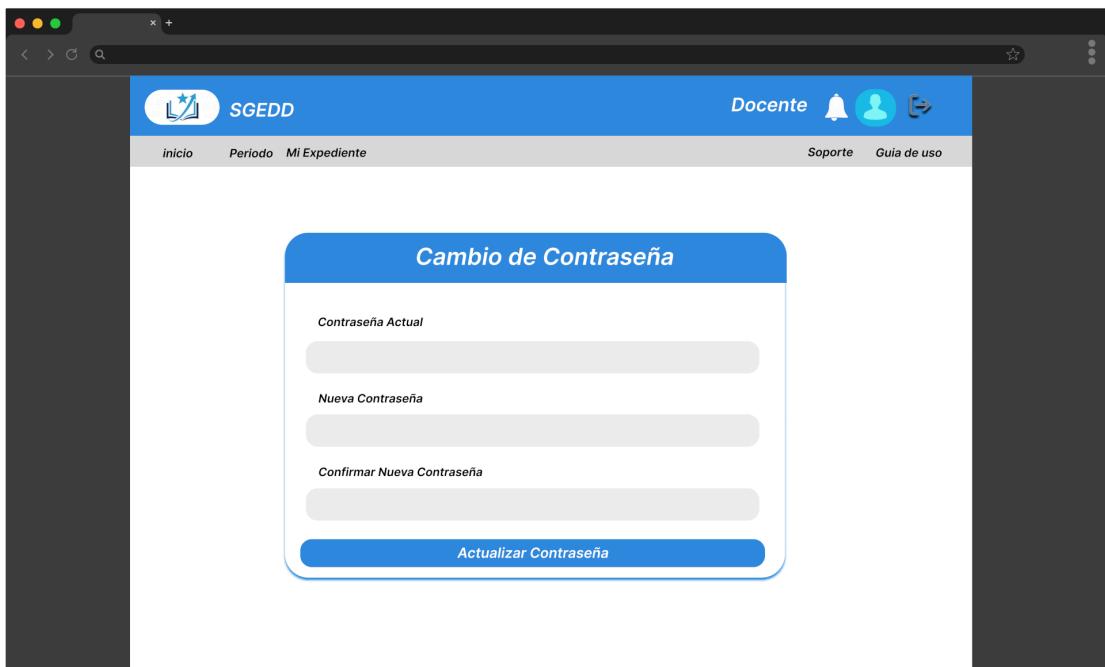
16. Notificaciones - Completa



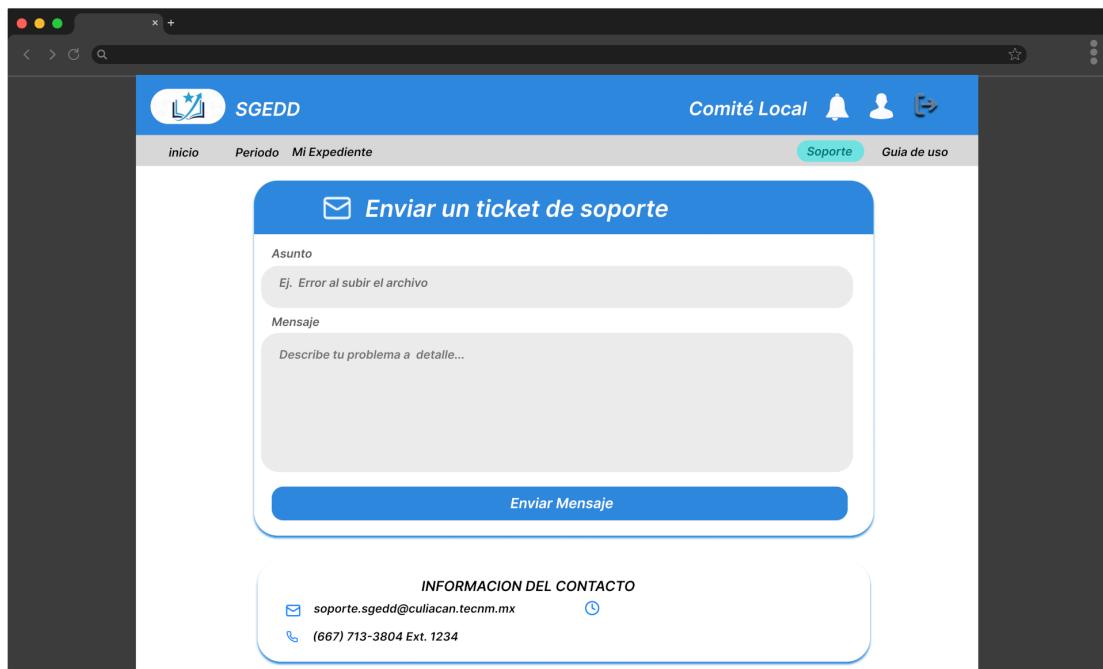
17. Opciones - Segundo plano



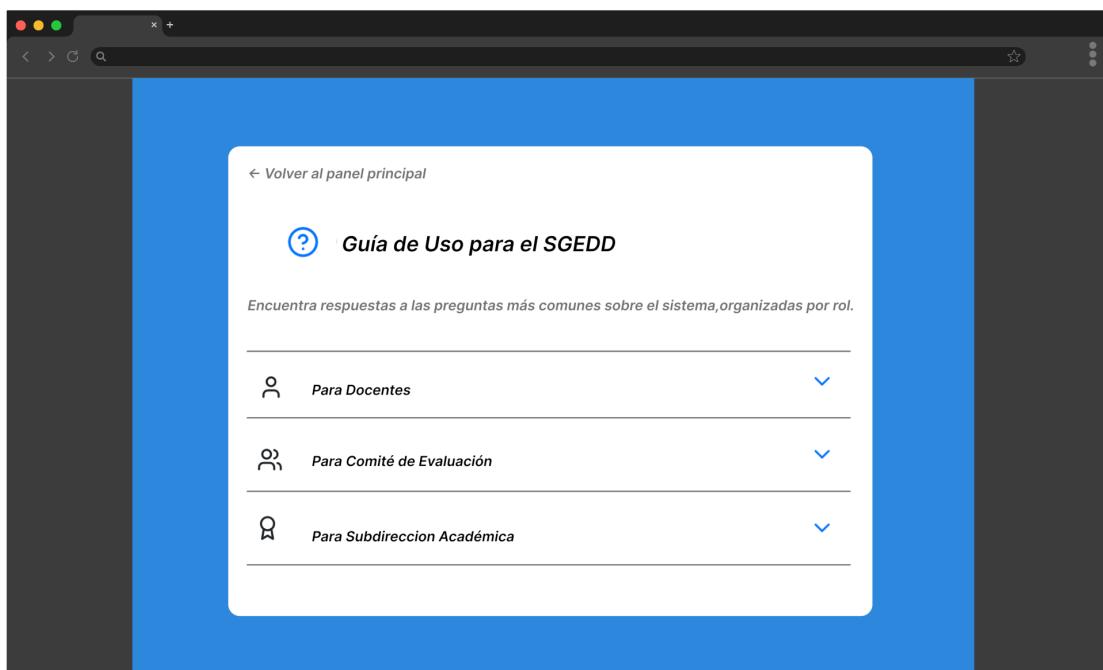
18. Cambio de Contraseña



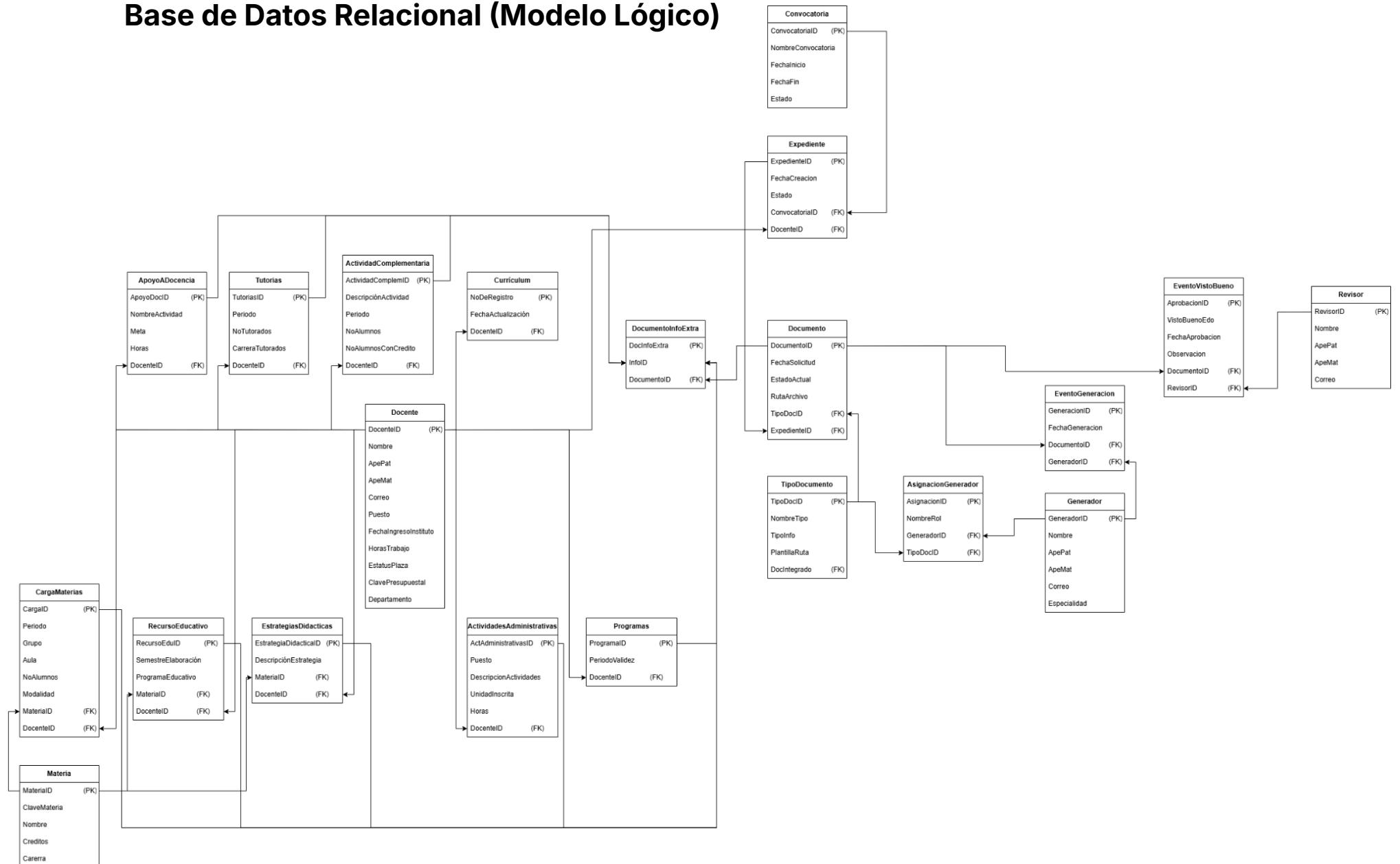
19. Soporte



20. Guia de uso



Base de Datos Relacional (Modelo Lógico)



Diseño arquitectónico

El Sistema Gestor del Estímulo al Desempeño Docente (SGEDD) implementará una arquitectura monolítica modular y cliente-servidor. Se requiere especial atención en el área de seguridad de la información dado que los resultados de cada convocatoria desembocan en estímulos monetarios a nuestros usuarios, además, es necesario contemplar la seguridad de las firmas del documento, esto se logrará mediante la API "Adobe Acrobat Sign", la cual nos facilita la seguridad en el tratamiento de las firmas electrónicas con una constante autenticación y autorización mediante OAuth 2.0 y la integridad del documento con infraestructura de clave pública que utiliza un certificado digital para asegurar que el documento proviene de Adobe Sign y no ha sido alterado. Es crítico que el sistema sea accesible e intuitivo debido al amplio rango de edades y comodidad con el uso de tecnología de nuestros usuarios. El sistema debe poder soportar una alta carga de trabajo en momentos clave de cada convocatoria. El sistema será usado durante un largo periodo de tiempo por lo que es requerido que tenga una alta mantenibilidad, escalabilidad y testabilidad.

¿Qué es la arquitectura monolítica?

Una arquitectura **monolítica** es aquella en la que toda la aplicación se construye como una única unidad o **bloque indivisible**.

Esto significa que **todos los componentes** (la interfaz de usuario, la lógica de negocio y el acceso a datos) **están empaquetados y desplegados juntos**.

Ventajas

1. **Simplicidad inicial:** Es más fácil de desarrollar, probar y desplegar al comienzo.
Menor sobrecarga técnica: No hay necesidad de gestionar comunicación entre servicios ni infraestructura compleja.
2. **Desarrollo rápido en etapas tempranas:** Ideal para prototipos o startups.
3. **Rendimiento eficiente:** Las llamadas internas dentro del mismo proceso son más rápidas que las de red.

Desventajas

1. **Difícil de escalar por partes:** No puedes escalar solo un módulo (por ejemplo, "usuarios"), tienes que escalar toda la aplicación.
2. **Despliegue lento y riesgoso:** Cualquier cambio (aunque sea pequeño) obliga a reconstruir y desplegar toda la aplicación.
3. **Fuerte acoplamiento:** Los módulos dependen mucho entre sí, lo que dificulta el mantenimiento y la evolución.
4. **Complejidad creciente con el tiempo:** A medida que la aplicación crece, el código se vuelve más difícil de entender y modificar.
5. **Baja flexibilidad tecnológica:** No puedes usar diferentes lenguajes o frameworks en distintos módulos.

¿Qué significa que sea una arquitectura monolítica modular?

La aplicación sigue siendo **una sola unidad desplegable** (un solo código base, un solo servidor o contenedor), pero su **interior está dividido en módulos bien definidos e independientes entre sí**.

La aplicación se divide en **módulos funcionales**.

Cada módulo (como users, products, orders) tiene todo lo necesario para funcionar:

- Sus modelos (estructura de datos)
- Sus controladores (lógica y endpoints)
- Sus servicios (reglas de negocio)
- Sus rutas (puntos de entrada HTTP o funciones)

Y todos se integran en una misma aplicación, pero con fronteras bien definidas.

Ventajas

1. **Simplicidad de despliegue:** Un solo proyecto, un solo servidor. Ideal para equipos pequeños.
2. **Organización clara:** Cada módulo tiene su espacio y responsabilidades.
3. **Facilita la colaboración:** Varios desarrolladores pueden trabajar en módulos distintos sin interferirse.
4. **Mantenible a largo plazo:** Puedes refactorizar o reemplazar un módulo sin afectar a todo el sistema.

5. **Camino natural hacia microservicios:** Si un módulo crece mucho, puede separarse fácilmente en un servicio independiente.

Desventajas

1. **Despliegue unificado:** Un pequeño cambio requiere volver a desplegar todo el sistema.
2. **Escalabilidad parcial limitada:** No puedes escalar un solo módulo de forma independiente.
3. **Possible acoplamiento accidental:** Si no se cuidan las dependencias, los módulos pueden terminar entrelazados.
4. **Más disciplina de equipo:** Requiere seguir convenciones y buenas prácticas para mantener la modularidad real.

¿Qué es la arquitectura cliente-servidor?

La **arquitectura cliente-servidor** es un **modelo de diseño de software** que divide una aplicación en dos partes principales:

1. Cliente:
 - Es quien solicita servicios o recursos.
 - Normalmente es el programa con el que interactúa el usuario (por ejemplo, un navegador web, una app móvil o un programa de escritorio).
2. Servidor:
 - Es quien provee los servicios o recursos solicitados.
 - Generalmente se ejecuta en una máquina distinta y atiende múltiples clientes al mismo tiempo.

Ventajas

1. **Centralización de recursos:** los datos y servicios se gestionan desde el servidor.
2. **Facilidad de mantenimiento:** el servidor puede actualizarse sin necesidad de cambiar a todos los clientes.
3. **Seguridad mejor controlada:** el servidor puede gestionar permisos y autenticación.
4. **Escalabilidad:** se pueden conectar muchos clientes a un mismo servidor.
5. **Interoperabilidad:** permite que distintos tipos de clientes (web, móvil, escritorio) usen los mismos servicios.

Desventajas

1. **Dependencia del servidor:** si el servidor falla, todos los clientes se ven afectados.
2. **Carga del servidor:** con muchos clientes, puede saturarse.
3. **Requiere conexión constante:** no funciona bien sin red o con baja conectividad.
4. **Costo del hardware del servidor:** se necesitan equipos o infraestructura potente para soportar muchos usuarios.