Diagrama

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Área de la problemática:

El área de la problemática es una escuela secundaria de Mendoza, donde asisten mas de 150 estudiantes y trabajan mas de 20 profesores, además de directivos y celadores.

Problema detectado:

En esta secundaria están teniendo problemas para verificar la asistencia de los alumnos, dado que algunos vienen para dejar el presente y luego de van y otros ni siquiera asisten a clases. Además de no tener una forma eficaz de guardar un horario para cada clase y materia.

Solución al problema detectado:

Para solucionar este problema, necesitamos un programa que permita comprobar la asistencia de cada alumno y organizar los horarios de cada materia. Además de saber que días no tienen tal materia y que profesor está asignado a cada materia.

Requerimientos funcionales:

**Inicio de Sesión**:

* **Roles**: El programa tendrá diferentes roles de usuario, como profesores, administradores y estudiantes. Cada rol tendrá permisos y accesos específicos.
* **Autenticación**: Los usuarios inician sesión con sus credenciales (nombre de usuario y contraseña).

**Gestión de Clases y Horarios**:

* **Crear Clases**: Los administradores pueden crear y gestionar clases, asignar profesores y estudiantes a cada clase.
* **Horario de Clases**: Los profesores pueden ver el horario de sus clases y la lista de estudiantes asignados a cada clase.

**Tomar Asistencia**:

* **Registro Diario**: Los profesores pueden registrar la asistencia diaria de los estudiantes. Esto puede hacerse manualmente o utilizando métodos automatizados como tarjetas RFID o códigos QR.
* **Estado de Asistencia**: Los estados pueden incluir "Presente", "Ausente", "Tarde" y "Justificado".
* **Notas y Comentarios**: Los profesores pueden añadir notas o comentarios adicionales sobre la asistencia de un estudiante.

**Generar Reportes**:

* **Reportes de Asistencia**: Los administradores pueden generar reportes de asistencia por clase, por estudiante, o por período de tiempo.
* **Alertas y Notificaciones**: El sistema puede enviar alertas a los padres/tutores sobre la asistencia de los estudiantes, especialmente en casos de ausencias repetidas.

**Integración con Otros Sistemas**:

* **Sistema de Gestión Escolar**: Integración con el sistema de gestión escolar para sincronizar datos de estudiantes, clases y horarios.
* **Comunicaciones**: Integración con sistemas de mensajería para enviar notificaciones a padres y estudiantes.

Requerimientos no funcionales:

**Rendimiento**

1. **Tiempo de Respuesta**: El sistema debe responder a las solicitudes de los usuarios en menos de 2 segundos en al menos el 95% de las interacciones.
2. **Escalabilidad**: El sistema debe poder manejar un incremento del 100% en la cantidad de usuarios y datos sin una disminución significativa en el rendimiento.
3. **Capacidad de Concurrencia**: El sistema debe soportar al menos 500 usuarios simultáneos sin pérdida de rendimiento.

**Seguridad**

1. **Autenticación y Autorización**: Implementar mecanismos de autenticación robustos (por ejemplo, OAuth 2.0) y control de acceso basado en roles (RBAC) para asegurar que solo los usuarios autorizados puedan acceder a ciertas funcionalidades.
2. **Protección de Datos**: Todos los datos sensibles, como las credenciales de los usuarios y los registros de asistencia, deben ser almacenados y transmitidos de forma segura usando cifrado (por ejemplo, HTTPS y cifrado en reposo).
3. **Cumplimiento de Normativas**: El sistema debe cumplir con las normativas de protección de datos aplicables, como GDPR o COPPA, según la región.

**Usabilidad**

1. **Interfaz de Usuario Intuitiva**: La interfaz debe ser fácil de usar y entender, con una curva de aprendizaje mínima para los usuarios (profesores, administradores, estudiantes y padres/tutores).
2. **Compatibilidad Multi-dispositivo**: El sistema debe ser accesible y funcionar correctamente en dispositivos móviles, tabletas y computadoras de escritorio.
3. **Accesibilidad**: El sistema debe cumplir con las pautas de accesibilidad web (por ejemplo, WCAG 2.1) para ser accesible a usuarios con discapacidades.

**Disponibilidad y Confiabilidad**

1. **Disponibilidad del Sistema**: El sistema debe estar disponible al menos el 99.9% del tiempo, excluyendo el tiempo programado para mantenimiento.
2. **Tolerancia a Fallos**: El sistema debe ser capaz de recuperarse automáticamente de fallos comunes sin pérdida de datos.
3. **Backup y Recuperación**: Implementar mecanismos de respaldo regular de los datos y proporcionar planes de recuperación ante desastres para asegurar la integridad de los datos en caso de fallo crítico.

**Mantenibilidad**

1. **Modularidad**: El sistema debe estar diseñado de manera modular para facilitar las actualizaciones y el mantenimiento.
2. **Documentación**: El código y las funcionalidades del sistema deben estar bien documentados para facilitar la comprensión y el mantenimiento por parte de los desarrolladores.
3. **Monitorización y Logging**: Implementar herramientas de monitorización y registro para facilitar la identificación y resolución de problemas.

**Escalabilidad**

1. **Escalabilidad Horizontal y Vertical**: El sistema debe ser capaz de escalar tanto horizontalmente (añadiendo más servidores) como verticalmente (mejorando los recursos de un servidor) para manejar incrementos en la carga de trabajo.
2. **Gestión de Sesiones**: Implementar técnicas de gestión de sesiones eficientes para soportar un gran número de usuarios concurrentes.

**Interoperabilidad**

1. **Integración con Sistemas Externos**: El sistema debe ser capaz de integrarse con otros sistemas de gestión escolar existentes mediante API o servicios web.
2. **Formatos de Exportación**: Los reportes y datos deben poder exportarse en formatos comunes como PDF, CSV y Excel.

**Portabilidad**

1. **Independencia de Plataforma**: El sistema debe ser independiente de la plataforma y capaz de ejecutarse en diferentes sistemas operativos y entornos de hardware.

Relevación tecnológica:

Para desarrollar este programa, puedes considerar el uso de las siguientes tecnologías:

* **Frontend**: React, Angular, o Vue.js.
* **Backend**: Node.js con Express, Django, o Spring Boot.
* **Base de Datos**: MySQL, PostgreSQL, o MongoDB.
* **Autenticación**: OAuth, JWT (JSON Web Tokens).
* **Notificaciones**: Twilio, Firebase Cloud Messaging.