
Manual Tecnico



UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA



Materia: Bases de datos

Profesor: José Saul de Lira Miramontes

Por: Carlos Sánchez Fuentes | Daniel Mata Lerma

Matriculas: 368058 | 362040

1. Sistema/Aplicación

Nombre del Sistema: TrackAssist

Descripción del Sistema: TrackAssist es una aplicación web diseñada para escuelas, el cual permite registrar y gestionar la asistencia, calificaciones y reportes académicos de los estudiantes. Está pensada para ser utilizada por docentes, administrativos y directivos, de esta forma facilitando la organización y análisis del rendimiento escolar.

Objetivo General: Desarrollar una aplicación web con una base de datos relacional la cual gestione de forma eficiente los registros académicos y de comportamiento de los distintos estudiantes, brindando acceso rápido y seguro a los usuarios autorizados.

Objetivos Específicos:

Implementar un sistema de registro de asistencias que permita a los docentes capturar la presencia de los estudiantes de forma rápida y confiable, así mismo dando un módulo de evaluaciones académicas para registrar y consultar las calificaciones por estudiante y asignatura. Por nuestra parte queremos facilitar la administración de grupos, asignaturas y usuarios mediante una interfaz intuitiva y segmentada por roles.

Usuarios:

Docente

- Descripción: Personal encargado de impartir clases a los estudiantes.
- Permisos:
 - Consultar listas de estudiantes por grupo y asignatura.
 - Registrar y editar asistencias diarias.
 - Registrar y modificar las calificaciones de sus asignaturas.
 - Acceder a reportes individuales de estudiantes.

Administrativo

- Descripción: Usuario encargado de gestionar la estructura académica del sistema.
- Permisos:

- Registrar y administrar estudiantes, grupos, docentes y asignaturas.
- Asignar docentes a asignaturas.
- Acceder y generar reportes generales.

Entorno Operativo:

Entorno de Desarrollo

- Frontend: React.js o Vue.js
- Estilos: Tailwind CSS o Bootstrap
- Backend: Node.js con Express.js
- Base de Datos: Oracle SQL
- Conector Oracle: oracledb oficial para Node.js

2. Entorno de Ejecución

- Servidor Web: Localhost (durante desarrollo) o servidor Node.js en la nube
- Servidor de Base de Datos: Oracle DB (local o remoto)
- Sistema Operativo Compatible: Windows, Linux o macOS (para el servidor)
- Navegadores compatibles: Chrome, Firefox, Edge, Safari (últimas versiones)

3. Entorno de Producción (Sugerido)

- Hosting en un servidor con soporte para Node.js
- Oracle Cloud o servicio externo para base de datos Oracle
- Certificado SSL para asegurar tráfico HTTPS
- Monitoreo de logs y métricas del servidor

2. Especificación de requerimientos

Funcionales:

- Desarrolladores, administradores de bases de datos.
- Registro de asistencias y evaluaciones.
- Generación de reportes académicos.

Diagrama de Clases:

Clases principales:

- **Estudiante**
 - id_estudiante, nombre, apellido, fecha_nacimiento
 - Relaciones: pertenece a Grupo, tiene Asistencias y Evaluaciones
- **Docente**
 - id_docente, nombre, apellido, especialidad
 - Relaciones: imparte Asignaturas
- **Grupo**
 - id_grupo, nombre_grupo, grado, ciclo_escolar
 - Relaciones: tiene muchos Estudiantes y Asignaturas
- **Asignatura**
 - id_asignatura, nombre
 - Relaciones: asignada a un Grupo, impartida por un Docente, relacionada con Evaluaciones
- **Asistencia**
 - id_asistencia, fecha, estado
 - Relaciones: pertenece a un Estudiante
- **Evaluación**
 - id_evaluacion, calificación, fecha
 - Relaciones: conecta Estudiante y Asignatura

Diagrama de Casos de Uso:

Actores:

- Docente
- Administrativo
- Directivo

Casos de uso por actor:

- **Docente**
 - Registrar asistencia
 - Registrar calificaciones
 - Consultar lista de estudiantes
 - Ver reportes individuales

- **Administrativo**
 - Gestionar estudiantes
 - Gestionar grupos y asignaturas
 - Asignar docentes
 - Consultar estadísticas globales

Diagrama de Actividades:

Docente inicia sesión

Selecciona grupo y fecha

Visualiza lista de estudiantes

Marca estado: Presente / Ausente / Justificado

Confirma y guarda

Sistema registra asistencia en la base de datos

Notificación de éxito

No funcionales:

- Interfaz responsiva.
- Seguridad de acceso por roles.
- Conexión estable con Oracle DB.

3. Arquitectura

Frontend: React.js / Vue.js + Tailwind o Bootstrap

Backend: Node.js + Express.js

Base de Datos: Oracle SQL

Conexión: oracledb oficial para Node.js

4. Base de Datos

Modelo Entidad-Relación (E/R):

Incluye entidades como Estudiantes, Docentes, Asignaturas, Evaluaciones.

Tablas principales:



| Tabla | Atributos Clave | Relaciones Principales |
|---------------------|--|--|
| Estudiantes | id_estudiante, nombre, apellido, fecha_nacimiento, grupo_id | Pertenece a un grupo |
| Docentes | id_docente, nombre, apellido, especialidad | Imparte varias asignaturas |
| Grupos | id_grupo, nombre_grupo, grado, ciclo_escolar | Tiene muchos estudiantes y varias asignaturas |
| Asignaturas | id_asignatura, nombre, docente_id | Relacionada con un grupo y un docente |
| Asistencias | id_asistencia, estudiante_id, fecha, estado | Relacionada con un estudiante y una fecha específica |
| Evaluaciones | id_evaluacion, estudiante_id, asignatura_id, calificación, fecha | Evalúa a un estudiante en una asignatura |

Representación básica de relaciones:

Docentes --< Asignaturas >-- Grupos

Grupos --< Estudiantes

Estudiantes --< Asistencias

Estudiantes --< Evaluaciones >-- Asignaturas

Especificación Tablas:

Estudiantes

| Campo | Tipo de dato | Restricciones |
|------------------|---------------------|-----------------------|
| id_estudiante | INT | PRIMARY KEY, NOT NULL |
| nombre | VARCHAR | NOT NULL |
| apellido | VARCHAR | NOT NULL |
| fecha_nacimiento | DATE | NOT NULL |
| grupo_id | INT | FOREIGN KEY → Grupos |

Docentes

| Campo | Tipo de dato | Restricciones |
|--------------|---------------------|-----------------------|
| id_docente | INT | PRIMARY KEY, NOT NULL |
| nombre | VARCHAR | NOT NULL |
| apellido | VARCHAR | NOT NULL |
| especialidad | VARCHAR | |

Grupos

| Campo | Tipo de dato | Restricciones |
|--------------|---------------------|-----------------------|
| id_grupo | INT | PRIMARY KEY, NOT NULL |
| nombre_grupo | VARCHAR | NOT NULL |

| | | |
|---------------|---------|----------|
| grado | VARCHAR | NOT NULL |
| ciclo_escolar | VARCHAR | NOT NULL |

Asignaturas

| Campo | Tipo de dato | Restricciones |
|---------------|--------------|------------------------|
| id_asignatura | INT | PRIMARY KEY, NOT NULL |
| nombre | VARCHAR | NOT NULL |
| docente_id | INT | FOREIGN KEY → Docentes |
| grupo_id | INT | FOREIGN KEY → Grupos |

Asistencias

| Campo | Tipo de dato | Restricciones |
|---------------|--------------|--------------------------------------|
| id_asistencia | INT | PRIMARY KEY, NOT NULL |
| estudiante_id | INT | FOREIGN KEY → Estudiantes |
| fecha | DATE | NOT NULL |
| estado | VARCHAR | NOT NULL (Ej: 'Presente', 'Ausente') |

Evaluaciones

| Campo | Tipo de dato | Restricciones |
|---------------|--------------|---------------------------|
| id_evaluacion | INT | PRIMARY KEY, NOT NULL |
| estudiante_id | INT | FOREIGN KEY → Estudiantes |

| | | |
|---------------|---------|---------------------------|
| asignatura_id | INT | FOREIGN KEY → Asignaturas |
| calificacion | DECIMAL | NOT NULL (Ej: 0.0–10.0) |
| fecha | DATE | NOT NULL |

Integridad:

Para garantizar la calidad y fiabilidad de los datos, se implementan las siguientes **restricciones (constraints)**:

PRIMARY KEY: en cada tabla para identificar de forma única los registros.

FOREIGN KEY: para mantener las relaciones lógicas entre entidades.

- grupo_id en Estudiantes → Grupos(id_grupo)
- docente_id en Asignaturas → Docentes(id_docente)
- grupo_id en Asignaturas → Grupos(id_grupo)
- estudiante_id en Asistencias y Evaluaciones → Estudiantes(id_estudiante)
- asignatura_id en Evaluaciones → Asignaturas(id_asignatura)

NOT NULL: en campos críticos como nombres, fechas y estados.

CHECK: opcionalmente en estado de Asistencias para restringir valores a un conjunto válido.

CHECK (estado IN ('Presente', 'Ausente', 'Justificado'))

5. Lógica de Negocio

Las reglas de negocio definen el comportamiento operativo de TrackAssist con base en el modelo académico:

- Un estudiante solo puede pertenecer a un grupo a la vez.
- Un docente puede impartir múltiples asignaturas, pero cada asignatura es impartida sólo por un docente.
- Una evaluación está siempre ligada a un estudiante y a una asignatura específica.

- Cada asistencia está asociada a un único estudiante en una fecha específica, sin duplicidad.
- Los reportes académicos sólo pueden ser consultados por usuarios autorizados según su rol.
- Solo el docente asignado a una asignatura puede registrar evaluaciones para ella.

6. Interfaz de la aplicación

La interfaz de TrackAssist está diseñada para ser intuitiva y responsiva. Sus componentes principales son:

- **Inicio de sesión:** Acceso según rol (docente, administrativo, directivo).
- **Panel de usuario personalizado:**
 - Docente: Lista de grupos, registro de asistencias y calificaciones.
 - Administrativo: Módulo de gestión de usuarios, asignaturas y grupos.
 - Directivo: Panel de visualización de estadísticas y reportes.
- **Menú lateral de navegación:** Acceso rápido a módulos clave.
- **Formularios dinámicos:** Para registro y edición de datos.
- **Validaciones en tiempo real:** Para asegurar la integridad de los datos ingresados.

7. Seguridad

TrackAssist implementa medidas básicas de seguridad para controlar el acceso y proteger los datos:

1. **Control de acceso por roles:** Cada usuario solo puede acceder a las secciones autorizadas según su perfil.
2. **Autenticación:** Inicio de sesión obligatorio con usuario y contraseña.
3. **Validación de formularios:** Para evitar inyecciones de datos erróneos.
4. **Separación de responsabilidades:** Los docentes no acceden a configuración, los administrativos no editan calificaciones.
5. **Conexión segura sugerida:** Uso de HTTPS y cifrado de contraseñas en producción.
6. **Logs de acceso (opcional):** Para auditoría y rastreo de actividades críticas.

8. Conclusión

El sistema TrackAssist representa una solución integral para la gestión académica de instituciones educativas. Su diseño modular, su base de datos relacional en Oracle y su enfoque web responsivo permiten una experiencia eficiente, segura y adaptable a distintas necesidades escolares. Su implementación promueve una mejora significativa en la organización, el análisis del rendimiento estudiantil y la transparencia en los procesos educativos.

9. Anexos

- Script de creación de la base de datos en Oracle SQL.
- Manual de Usuario