



Desarrollo de Aplicaciones con Web Frameworks DWF404

Actividad:	Primera Fase de Proyecto de catedra
Docente:	Daniel Sosa Castillo
Integrantes:	Oswaldo Henríquez HM231279 Jorge Mira Flores MF232232 Jonathan Mendoza MO223025 Carlos Villalobos VR230136 Kevin Ramírez RM230143 Sebastián Torres TG230977 Jennifer Beltrán ZB212784
Fecha de entrega:	22/03/2024

Informe para la Aplicación Web de Centro Escolar

Introducción:

Nuestra aplicación de registro escolar en Java es un sistema diseñado para facilitar la gestión de registros de estudiantes, profesores y otros aspectos relevantes de una escuela. Esta aplicación se desarrolla utilizando tecnologías web basadas en Java para garantizar la eficiencia, seguridad y escalabilidad del sistema.

Objetivos:

- Ofrecer una plataforma centralizada para administrar y mantener registros escolares de manera eficiente.
- Automatizar procesos como la inscripción de estudiantes, asignación de notas y gestión de profesores.
- Proporcionar a los administradores, profesores y estudiantes acceso seguro a la información relevante en tiempo real.
- Mejorar la comunicación entre los diferentes actores dentro de la institución educativa.

Características Generales del Sistema de Registro Escolar:

1. Gestión Integral de Estudiantes: Permite la consulta y actualización de perfiles personales, expedientes académicos, estado de pagos, materias inscritas y otros datos relevantes para un seguimiento completo del progreso educativo de los estudiantes.

2. Administración Avanzada de Profesores: Facilita el registro y gestión de profesores, así como la asignación de cursos y horarios de clases. Permite a los profesores registrar y mantener calificaciones, generando informes de progreso académico para estudiantes y padres. También ofrece herramientas para administrar salones de clases y horarios.

3. Seguridad y Privacidad Mejoradas: Garantiza la seguridad de los datos mediante medidas como la autenticación de usuarios, el control de acceso basado en roles y la encriptación de datos sensibles. Esto asegura que solo usuarios autorizados puedan acceder a la información relevante y que los datos estén protegidos contra accesos no autorizados.

4. **Gestión Eficaz de Pagos:** Ofrece funcionalidades para el seguimiento de pagos de estudiantes, llevando un registro detallado de los pagos realizados y administrando su estado con respecto al curso impartido. Esto facilita la gestión financiera de la institución educativa y garantiza una comunicación clara con los estudiantes y padres sobre los aspectos financieros relacionados con la educación.

Roles y Funcionalidades:

1. Administrador:

Acceso total al sistema, incluida la capacidad de gestionar todas las entidades y sus relaciones.

Puede realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en todas las tablas, lo que incluye la administración de alumnos, profesores, grados, cursos, y grupos.

Generación de informes y estadísticas a través de consultas complejas y agregaciones para evaluar el desempeño académico y financiero.

2. Profesor:

Acceso a funcionalidades específicas relacionadas con la gestión académica.

Puede registrar y actualizar las notas de los estudiantes en las materias asignadas.

Gestión de grupos y horarios de clases, lo que implica la asignación y modificación de grupos de alumnos.

Visualización y seguimiento del historial académico de los estudiantes asignados.

3. Alumno:

Acceso a su perfil personal para visualizar y actualizar su información.

Consulta del estado de sus pagos y registro de nuevos pagos.

Visualización de horarios de clases, detalles de materias inscritas y notas obtenidas en cada materia.

❖ Maven:

Maven es una herramienta de gestión de proyectos de software que se utiliza principalmente para la construcción, el manejo y la gestión de dependencias de proyectos Java. Maven es también una forma estándar de estructurar un proyecto Java, manejar sus dependencias externas y automatizar la construcción, el empaquetado y la distribución del software implementable.



❖ Payara Server

Payara Server es un servidor de aplicaciones Java de código abierto de alto rendimiento basado en la versión de código abierto de GlassFish Server. Está diseñado para ejecutar aplicaciones empresariales Java EE (Enterprise Edition) y Jakarta EE en entornos de producción. Payara Server proporciona muchas funciones y capacidades avanzadas para desarrollar, implementar y administrar aplicaciones en la plataforma Java.



❖ Wamp server

WampServer es una herramienta de entorno de desarrollo local completo que incluye un servidor web, una base de datos y soporte para el desarrollo en PHP. En nuestra aplicación se utilizará en el entorno de base de datos, ya que proporciona un entorno amigable al usuario y de buen rendimiento.



❖ JAVA

Java será nuestro lenguaje de programación para generar el proyecto completo ya que es un lenguaje de alto nivel y orientado a objetos, esto nos permitirá separar nuestro código y aprovechar de la orientación a objetos para que sea más fácil de entender. Por ende, necesitamos tener una versión de java instalado en nuestro equipo, como puede ser JAVA “1.8.0_401” o posteriores.



❖ IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) creado por JetBrains, Será nuestro entorno de trabajo con la finalidad de ayudarnos en el desarrollo por la facilidad que tiene para agregar todas las herramientas para desarrollar un proyecto en JAVA. Se necesita la versión ULTIMATE con la licencia correspondiente para su uso.



Definición de las tecnologías a utilizar (Framework y persistencia)

Para el desarrollo de nuestra aplicación en Java destinada a la gestión académica de una escuela, hemos optado por emplear el framework Hibernate y persistencia JPA (Java Persistence API). A continuación, se presenta una breve descripción de cada una de estas tecnologías:

Hibernate: Hibernate es un framework de mapeo objeto-relacional (ORM) para Java, que facilita la interacción entre una base de datos relacional y una aplicación Java mediante la representación de las tablas de la base de datos como objetos de Java. Utilizando Hibernate, podemos trabajar con objetos Java en lugar de consultas SQL directas, lo que simplifica el desarrollo y el mantenimiento del código, además de ofrecer características avanzadas como el manejo de transacciones, caché de datos y generación de consultas dinámicas.

Persistencia JPA (Java Persistence API): JPA es una especificación de Java para el mapeo objeto-relacional que proporciona una interfaz común para trabajar con bases de datos relacionales en aplicaciones Java. JPA define un conjunto de anotaciones y métodos estándar para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en objetos Java, lo que permite que las aplicaciones sean independientes del proveedor de persistencia subyacente. Al utilizar JPA, podemos escribir código Java que sea portable entre diferentes sistemas de gestión de bases de datos, lo que aumenta la flexibilidad y la interoperabilidad de nuestra aplicación.

La combinación de Hibernate como implementación de JPA nos brinda una potente herramienta para la gestión de la capa de persistencia en nuestra aplicación, permitiendo un desarrollo eficiente y escalable, así como una integración sencilla con otras tecnologías y framework que utilizaremos en el proyecto.

Diagrama Entidad relación de la base de datos

centro_escolar alumno	
id_alumno	int
carnet	varchar(8)
nombre_alumno	varchar(150)
apellido_alumno	varchar(150)
telefono_contacto	varchar(9)
nombre_responsable	varchar(150)
id_estado_alumno	int
id_tipo_alumno	int
fecha_nacimiento	date
id_grado_academico	int

centro_escolar profesor	
id_profesor	int
carnet_profesor	varchar(8)
nombre_profesor	varchar(150)
apellido_profesor	varchar(150)
telefono	varchar(9)
correo	varchar(150)
fecha_nacimiento	date
id_estado_profesor	int

centro_escolar profesores_materia	
id_profesor_materia	int
id_profesor	int
id_materia	int

centro_escolar estado_profesor	
id_estado_profesor	int
estado_profesor	varchar(150)

centro_escolar materia	
id_materia	int
nombre_materia	varchar(150)
id_grado_academico	int

centro_escolar estado_alumno	
id_estado_alumno	int
estado_alumno	varchar(150)

centro_escolar estado_pago	
id_estado_pago	int
estado_pago	varchar(150)

centro_escolar grupos	
id_grupo	int
id_alumno	int
id_profesor	int
id_salón	int

centro_escolar grados_academicos	
id_grado_academico	int
nombre_grado	varchar(100)

centro_escolar salones	
id_salón	int
nombre_salón	varchar(150)

centro_escolar tipo_alumno	
id_tipo_alumno	int
tipo_alumno	varchar(150)

centro_escolar pago_alumno	
id_pago	int
id_alumno	int
monto_pago	decimal(5,2)
fecha_vencimiento	date
fecha_pago	date
mora	decimal(4,2)
concepto	varchar(100)
id_estado_pago	int

centro_escolar notas	
id_notas	int
id_alumno	int
id_profesor_materia	int
nota	decimal(4,2)