

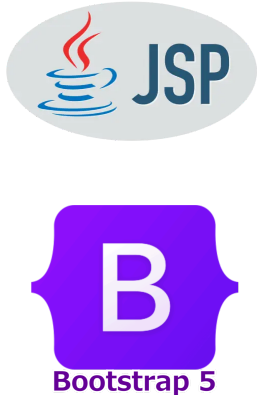




Web Application con JAVA EE + JSP + MAVEN + SQL SERVER UTILIZANDO EL PATRÓN DE DISEÑO MVC

¿Que haremos?

Realizaremos un aplicación web sobre registro de asistencia de un colegio utilizando como back-end JAVA, como front-end páginas .jsp con bootstrap 5 y estaremos utilizando como base de datos SQL Server 2019, con respecto a esto último se establece 2 maestros y una transaccional con el objetivo de cumplir el funcionamiento de nuestra aplicación web.

Imagen:

Front-end	Back-end	Database
		

Patrón de diseño MVC

Estaremos utilizando el patrón de diseño MVC (Modelo Vista Controlador), ya que es el que hemos elegido y según nuestra opinión el patrón de diseño más ordenado y legible a simple vista.

Imagen:



Estructura del aplicativo

1. Configuración del entorno y Base de datos

Estaremos utilizando nuestro [Eclipse IDE for Java EE Developers](#) y [Apache Tomcat](#). Por otro lado necesitaremos nuestra base de datos [SQL Server 2019](#).

Descargamos e instalamos los programas, si necesitas ayuda puedes buscar videos en Youtube.

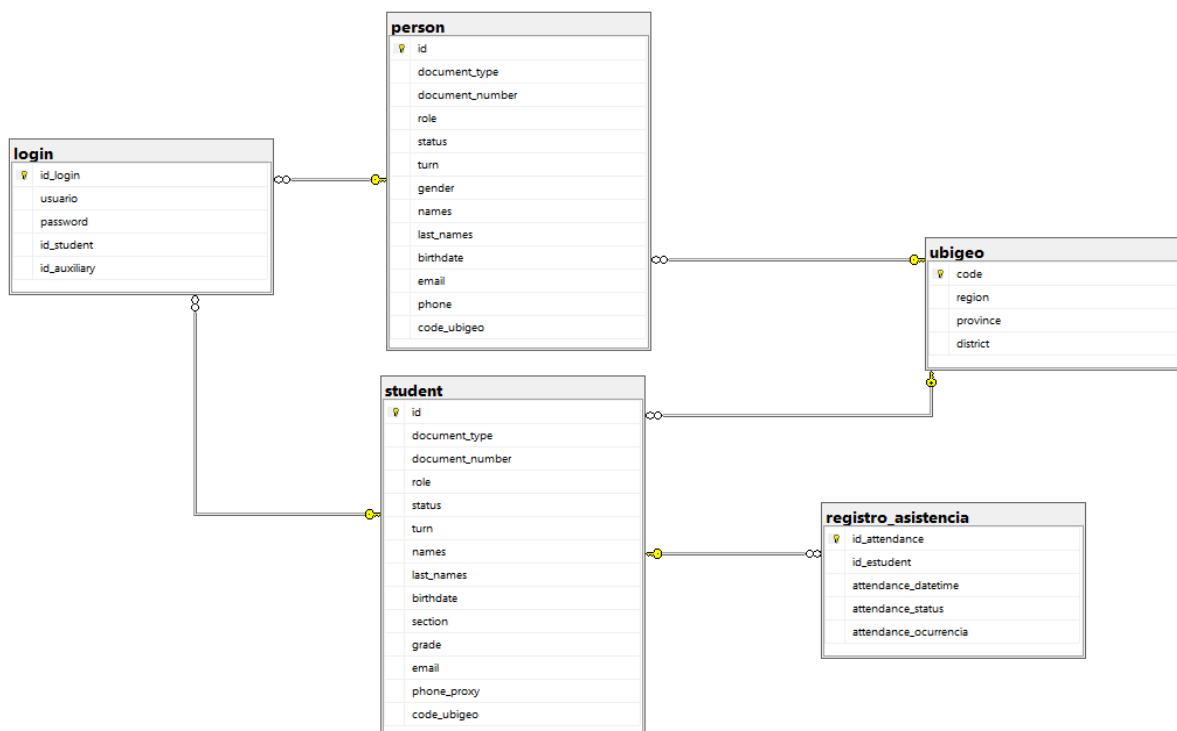


deJava Tip | No olvidar descargar Apache Tomcat, ya que nos proporciona un entorno de servidor web HTTP "Java puro" en el que también se puede ejecutar código Java.

2. Creación de la Base de datos

Creamos nuestra base de datos, según los requerimientos del proyecto y del cliente.

Imagen:



deJava Tip | Realizar pruebas CRUD (Create, Read, Update, Delete). En este proyecto realizaremos el eliminado físico y lógico de acuerdo a los requerimientos del cliente

3. Creación de una web application con maven

En este punto creamos nuestra Aplicación Web con Maven en nuestro Eclipse IDE.



deJava Tip | Si tienes dudas al respecto, puedes buscar tutoriales en YouTube

4. Conexión a la Base de datos

Creamos la conexión entre nuestra aplicación web y la base de datos.

Imagen:

```
1 package pe.edu.vallegrande.app.db;
2
3 import java.sql.Connection;
4
5
6 public class Conexion {
7
8     // Declaración de la variable de conexión
9     Connection con;
10
11     // URL de conexión a la base de datos
12     String url = "jdbc:sqlserver://localhost:1433;databaseName=CASSIATEC;encrypt=true;TrustServerCertificate=True;";
13
14     // Nombre de usuario y contraseña para acceder a la base de datos
15     String user = "sa";
16     String pass = "francisDE";
17
18     // Método para establecer una conexión a la base de datos
19     public Connection conectar() {
20         try {
21             // Cargar el controlador del driver de la base de datos
22             Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
23
24             // Establecer la conexión utilizando la URL, usuario y contraseña
25             con = DriverManager.getConnection(url, user, pass);
26         } catch (Exception e) {
27             // Capturar cualquier excepción ocurrida durante la conexión
28             e.printStackTrace();
29         }
30         return con;
31     }
32 }
```

Crearemos nuestra clase de prueba para verificar la conexión.

Imagen:

```
1 package pe.edu.vallegrande.app.pruebasPerson;
2
3 import java.sql.Connection;
4
5
6 public class ProbarConexion {
7
8     public static void main(String[] args) {
9         Conexion conexion = new Conexion(); // Crear una instancia de la clase Conexion
10        Connection con = conexion.conectar(); // Establecer la conexión a la base de datos
11
12        if (con != null) {
13            System.out.println("Conexión exitosa a la base de datos CASSIATEC");
14        } else {
15            System.out.println("No se pudo establecer la conexión");
16        }
17    }
18 }
19
20 }
21
```

Servers Console X

<terminated> ProbarConexion (1) [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (29 feb. 2024 19:11:51 – 19:11:52) [pid: 19016]

Conexión exitosa a la base de datos CASSIATEC

5. Pruebas de consultas por Back-end

Crearemos la lógica para realizar consultas por Back-end



deJava Tip | Realizar pruebas CRUD (Create, Read, Update, Delete). En este proyecto realizaremos el eliminado físico y lógico de acuerdo a los requerimientos del cliente

6. Creación de las interfaces del aplicativo

Las interfaces es uno de los apartados más importantes, ya que es donde el usuario estará interactuando con la aplicación web



deJava Tip | Recuerda que las interfaces deben de ser intuitivas y las validaciones deben de ayudar a que la información ingresada es soportada por la Base de datos

6. Pruebas de consultas por Front-end

Realizaremos pruebas del funcionamiento desde las interfaces creadas.



deJava Tip | Realizar pruebas CRUD (Create, Read, Update, Delete). En este proyecto realizaremos el eliminado físico y lógico de acuerdo a los requerimientos del cliente

The screenshot shows a web application interface for 'Reporte Individual'. The page has a dark theme with a red sidebar on the left containing navigation links: HOME, ACTUALIZAR MIS DATOS, REGISTRAR ASISTENCIA (highlighted in yellow), AGREGAR ESTUDIANTE, AGREGAR PERSON, and ACTUALIZAR ESTUDIANTE. The main content area has a header with the time '19:58:58' and date 'Jueves 29 de febrero de 2024', along with a 'Cerrar sesión' link. Below the header, there's a 'Reporte Individual' section with a 'Buscar Otra Aula' button and an 'Exportar' button. To the right, there's a 'Filtrar por fecha' section with input fields for 'Fecha de inicio' and 'Fecha de termino', and a 'Generar Reporte Filtro Fecha' button. Below this, there's a red box containing 'Información del Reporte:' with 'N° Doc: 55569048' and 'Alumna: Mendieta Cardena, Luzia'. The main part of the page is a table with columns: Fecha, Estado de Asistencia, Ocurrencia, and Opciones. The table contains five rows of attendance data. The Opciones column has 'Editar' and 'Eliminar' buttons for each row.

Fecha	Estado de Asistencia	Ocurrencia	Opciones
15/12/2023 - 12:17:40 a. m.	Faltó	falto a clases por que estaba enfermo de la cabeza	Editar Eliminar
16/12/2023 - 1:07:30 a. m.	Faltó	Falto por que estaba enfermo	Editar Eliminar
16/12/2023 - 12:30:13 a. m.	Asistió	hola	Editar Eliminar
18/12/2023 - 12:30:16 a. m.	Faltó	hola	Editar Eliminar
19/12/2023 - 12:30:18 a. m.	Justificó	oooooooooooo	Editar Eliminar