

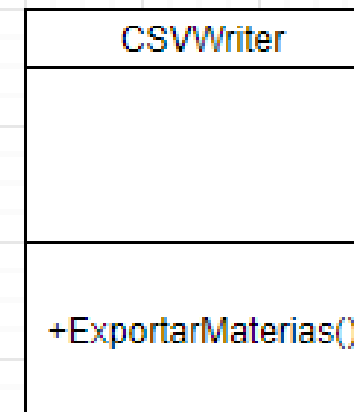
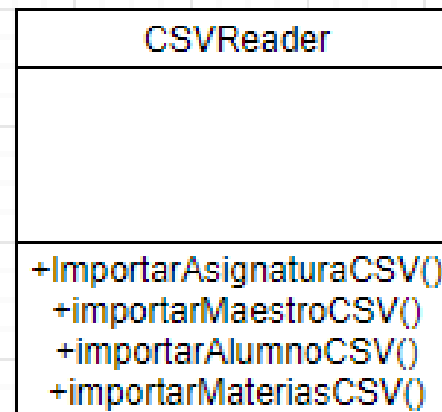
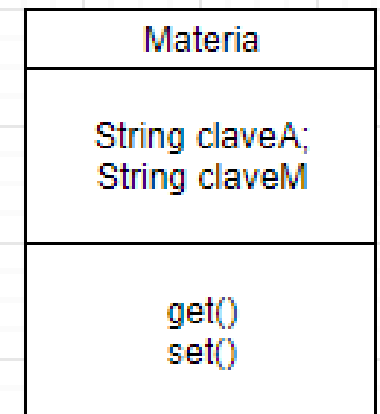
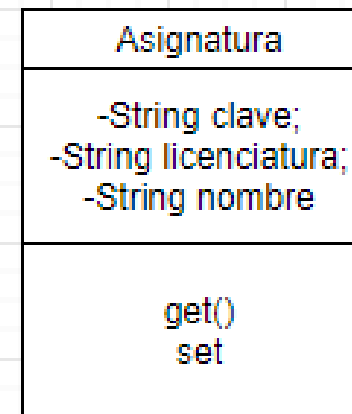
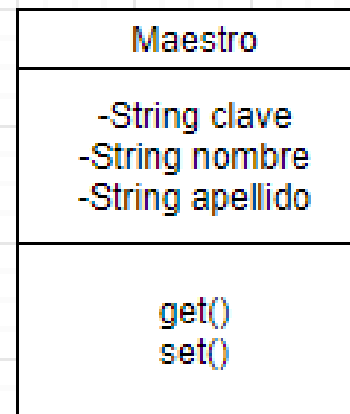
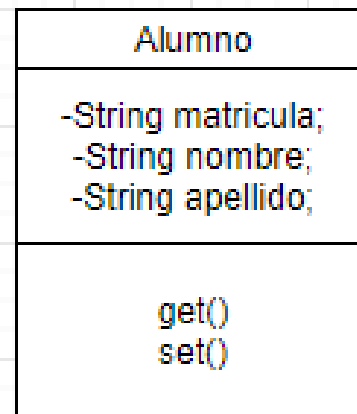
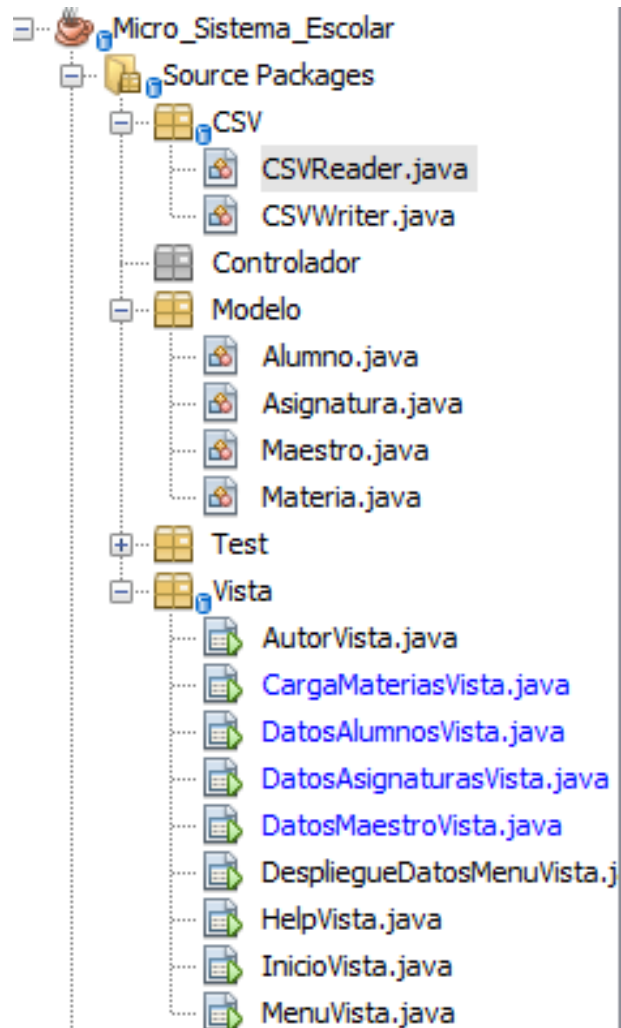
Micro control escolar

Es un sistema de control escolar implementando en el IDE de NetBeans usando java , y una librería de lectura y escritora para archivos CSV.

Diseño: Se opto por implementar una interfaz gráfica y un modelo MVC

Objetivos:

- Interfaz simple y fácil de usar
- Persistencia de datos
- Correcto funcionamiento de relación de datos



Restricciones

El programa esta hecho hasta el momento para relacionar Maestros/Asignaturas, una implementación futura podría agregar más funcionalidades.

El programa sólo puede recibir archivos CSV para leer datos
Maestros no pueden impartir dos veces la misma materia

Descripción de producto

Es un programa hecho en java y diseñado para simular el registro de los profesores que impartirán una asignatura. Lee archivos CSV y escribe en un archivo CSV la carga de los maestros.

Justificación

El diseño esta dirigido a permitir que el usuario seleccione la carga de materias con profesores, también desplegar los datos almacenados con una interfaz gráfica.

Plan de producción

Se realizó un diagrama de entidades y un diagrama UML para modelar el programa, se diseñaron las respectivas clases Alumno, Materia, Asignatura, Maestro, y una sección dedicada a CSV para leer y escribir.

Descripción de uso

Se genera una interfaz de inicio, que con un botón hace transición al menú, ofreciendo dos opciones, una dedicada a ver los datos almacenados en los archivos CSV, y el otro botón despliega el menú para realizar cargas de asignaturas maestros, en dos espacios seleccionables se puede escoger la clave de la materia y la clave del maestro, y por finalizar el botón cargar, que agrega el dato al archivo CSV.

```
package CSV;
import Modelo.*;
import com.csvreader.CsvReader;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author Javier
 */
public class CSVReader {

    public static List<Asignatura> ImportarAsignaturaCSV() {
        List<Asignatura> asignaturas = new ArrayList();
        try{

            CsvReader leerAsignaturas = new CsvReader("Asignaturas.csv");
            leerAsignaturas.readHeaders();

            while(leerAsignaturas.readRecord()) {
                String clave = leerAsignaturas.get(0);
                String licenciatura = leerAsignaturas.get(1);
                String nombreAsignatura = leerAsignaturas.get(2);

                asignaturas.add(new Asignatura(clave, licenciatura, nombreAsignatura));
            }

            leerAsignaturas.close(); // Cierra el archivo
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```



```
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    return asignaturas;  
}
```

```
public static List<Alumno> ImportarAlumnoCSV() {  
    List<Alumno> alumnos = new ArrayList();  
    try{  
  
        CsvReader leerAlumnos = new CsvReader("Alumnos.csv");  
        leerAlumnos.readHeaders();  
  
        while(leerAlumnos.readRecord()) {  
            String matricula = leerAlumnos.get(0);  
            String nombre = leerAlumnos.get(1);  
            String apellido = leerAlumnos.get(2);  
  
            alumnos.add(new Alumno(matricula,nombre,apellido));  
        }  
  
        leerAlumnos.close(); // Cierra el archivo  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
    return alumnos;  
}
```

```
public static List<Maestro> ImportarMaestroCSV() {
```

```
public static List<Maestro> ImportarMaestroCSV() {  
    List<Maestro> maestros = new ArrayList();  
    try{  
  
        CsvReader leerMaestros = new CsvReader("Maestros.csv");  
        leerMaestros.readHeaders();  
  
        while(leerMaestros.readRecord()) {  
            String claveM = leerMaestros.get(0);  
            String nombreM = leerMaestros.get(1);  
            String apellidoM = leerMaestros.get(2);  
  
            maestros.add(new Maestro(claveM,nombreM,apellidoM));  
        }  
  
        leerMaestros.close(); // Cierra el archivo  
    } catch(FileNotFoundException e) {  
        e.printStackTrace();  
    } catch(IOException e) {  
        e.printStackTrace();  
    }  
    return maestros;  
}
```

```
public static List<Materia> ImportarMateriasCSV() {  
    List<Materia> materias = new ArrayList();  
    try{  
  
        CsvReader leerMaterias = new CsvReader("CargaMaterias.csv");  
        leerMaterias.readHeaders();  
  
        while(leerMaterias.readRecord()) {  
            String claveA = leerMaterias.get(0);  
            String claveM = leerMaterias.get(1);  
  
            materias.add(new Materia(claveA, claveM));  
        }  
  
        leerMaterias.close(); // Cierra el archivo  
    } catch(FileNotFoundException e) {  
        e.printStackTrace();  
    } catch(IOException e) {  
        e.printStackTrace();  
    }  
    return materias;  
}
```

```
package CSV;
import Modelo.*;
import com.csvreader.CsvWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.List;

/**
 *
 * @author Javier
 */
public class CSVWriter {
    public static void ExportarCSV(List<Materia> materias) {
        String salidaArchivo = "CargaMaterias.csv";
        boolean existe = new File(salidaArchivo).exists();

        if(existe){
            File archivoMaterias = new File(salidaArchivo);
            archivoMaterias.delete();
        }

        try{
            CsvWriter salidaCSV = new CsvWriter(new FileWriter(salidaArchivo, true), ',');

            salidaCSV.write("claveA");
            salidaCSV.write("claveM");

            salidaCSV.endRecord();

            for (Materia materia: materias) {
                salidaCSV.write(materia.getClaveA());
            }
        }
    }
}
```

```
        salidaCSV.write(materia.getClaveM());

        salidaCSV.endRecord();
    }

    salidaCSV.close();

} catch (IOException e) {
    e.printStackTrace();
}

}

}
```

```
public class Alumno {
    String matricula;
    String nombreAlumno;
    String apellidoAlumno;

    public Alumno(String matricula, String nombreAlumno, String apellidoAlumno) {
        this.matricula = matricula;
        this.nombreAlumno = nombreAlumno;
        this.apellidoAlumno = apellidoAlumno;
    }

    public String getMatricula() { ...3 lines }

    public String getNombreAlumno() { ...3 lines }

    public String getApellidoAlumno() { ...3 lines }

    public void setMatricula(String matricula) { ...3 lines }

    public void setNombreAlumno(String nombreAlumno) { ...3 lines }

    public void setApellidoAlumno(String apellidoAlumno) { ...3 lines }
}
```

```
public class Asignatura {
    String clave;
    String licenciatura;
    String nombreAsignatura;

    public Asignatura(String clave, String licenciatura, String nombreAsignatura) {
        this.clave = clave;
        this.licenciatura = licenciatura;
        this.nombreAsignatura = nombreAsignatura;
    }

    public String getClave() { ...3 lines }

    public String getLicenciatura() { ...3 lines }

    public String getNombreAsignatura() { ...3 lines }

    public void setClave(String clave) { ...3 lines }

    public void setLicenciatura(String licenciatura) { ...3 lines }

    public void setNombreAsignatura(String nombreAsignatura) { ...3 lines }

}
```

```
package Modelo;

/**
 *
 * @author Javier
 */
public class Maestro {
    String clave;
    String nombreMaestro;
    String apellidoMaestro;

    public Maestro(String clave, String nombreMaestro, String apellidoMaestro) {
        this.clave = clave;
        this.nombreMaestro = nombreMaestro;
        this.apellidoMaestro = apellidoMaestro;
    }

    public String getClave() { ...3 lines }

    public String getNombreMaestro() { ...3 lines }

    public String getApellidoMaestro() { ...3 lines }

    public void setClave(String clave) { ...3 lines }

    public void setNombreMaestro(String nombreMaestro) { ...3 lines }

    public void setApellidoMaestro(String apellidoMaestro) { ...3 lines }
```



```
public class Materia {  
    String claveA;  
    String claveM;  
  
    public Materia(String claveA, String claveM) {  
        this.claveA = claveA;  
        this.claveM = claveM;  
    }  
  
    public String getClaveA() { ...3 lines }  
  
    public String getClaveM() { ...3 lines }  
  
    public void setClaveA(String claveA) { ...3 lines }  
  
    public void setClaveM(String claveM) { ...3 lines }  
  
}
```

VISTAS





Datos existentes Asignar materias/maestros

Revisar

Cargar



Datos almacenados

Maestros

Cargar

Asignaturas

Cargar

Alumnos

Cargar

Regresar



Alumnos

Matricula	Nombre	Apellido
A13003975	Javier	Chi
A14002842	Fernando	Ortiz

Regresar



ASIGNATURAS

Clave	Licenciatura	Asignatura
DISOFT	LIS	DISEÑO DE SOFTWARE
ARQSOFT	LIS	ARQUITECTURA DE SOFT...
ALGOR	LCC	ALGORITMIA

[Regresar](#)



CARGA DE MATERIAS

ARQSOFT



6978



Regresar

Cargar