

# MANUAL TÉCNICO

## Contenido

|   |   |
|---|---|
| Introducción .....                          | 3 |
| Objetivos .....                             | 3 |
| 1. Requerimientos .....                     | 4 |
| 2. Ejecución .....                          | 4 |
| 3. Almacenamiento de Datos .....            | 4 |
| 4. Diagrama de Clases .....                 | 5 |
| 5. Métodos Constructores .....              | 6 |
| 5.1. Constructor Profesor.....              | 6 |
| 5.2. Constructor Notas de Actividades ..... | 6 |
| 5.3. Constructor Alumno.....                | 7 |
| 5.4. Constructor Curso .....                | 7 |
| 5.5. Constructor Actividad .....            | 8 |
| 6. Variables Globales .....                 | 8 |
| 7. Librerías Externas.....                  | 8 |
| 7.1. json-simple-1.1.jar .....              | 8 |
| 7.2. itextpdf-5.4.0.jar .....               | 9 |
| 7.3. jfreechart-1.5.0.jar .....             | 9 |

## Introducción

El documento da a conocer cada una de las funcionalidades que cumplen los métodos y/o funciones de las clases utilizadas en la construcción del sistema de DTT de la Escuela de Ciencias y Sistemas, que cumple con la función de almacenar información de todo el personal que labora en la educación además de los estudiantes que reciben cursos en pertenecientes al departamento.

## Objetivos

Dar a conocer el uso adecuado del sistema de información, para el acceso y buen funcionamiento del mismo, mostrando la secuencia en que debe ser ejecutado, asimismo la descripción de los archivos relevantes del sistema los cuales resguardan la información del sistema.

Brindar soporte a nuevos técnicos sobre cómo dar un mantenimiento óptimo del software o para poder mejorarlo.

## 1. Requerimientos

Para que el sistema pueda ser utilizado en cualquier sistema operativo debe cumplir con los siguientes requerimientos:

- Tener instalado JDK 14.0.2 o superiores de Oracle.
- Tener instalado Java 8 Update 281.

## 2. Ejecución

Existe más de una forma para ejecutar el programa, sin embargo, a continuación, se describen las formas más sencillas para hacer uso del software en sistemas operativos Windows.

La primera opción está enfocada es mediante el Símbolo del sistema o cmd.

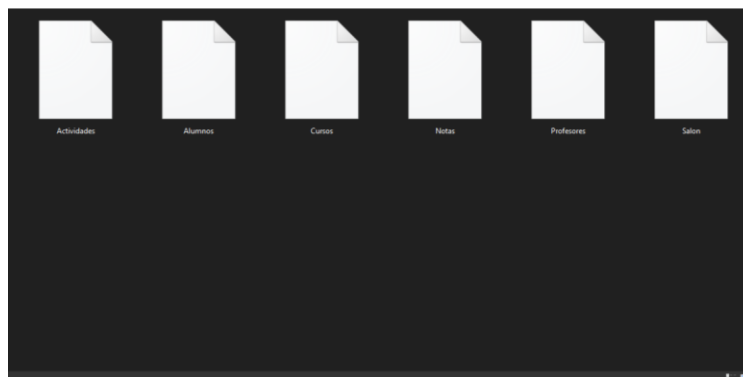
1. Usar la combinación de teclas Win + R, y luego escribir cmd y Aceptar.
2. Escribir "cd" (sin comillas), luego espaciar una vez y escribir la ruta en donde se encuentra el archivo ejecutable (.jar). Ejemplo: cd C:\User\Proyecto
3. Escribir el comando: java -jar; seguido del nombre del archivo con su respectiva extensión. Ejemplo: java -jar Proyecto.jar

Después de realizar estos pasos el programa comenzará su ejecución automáticamente.

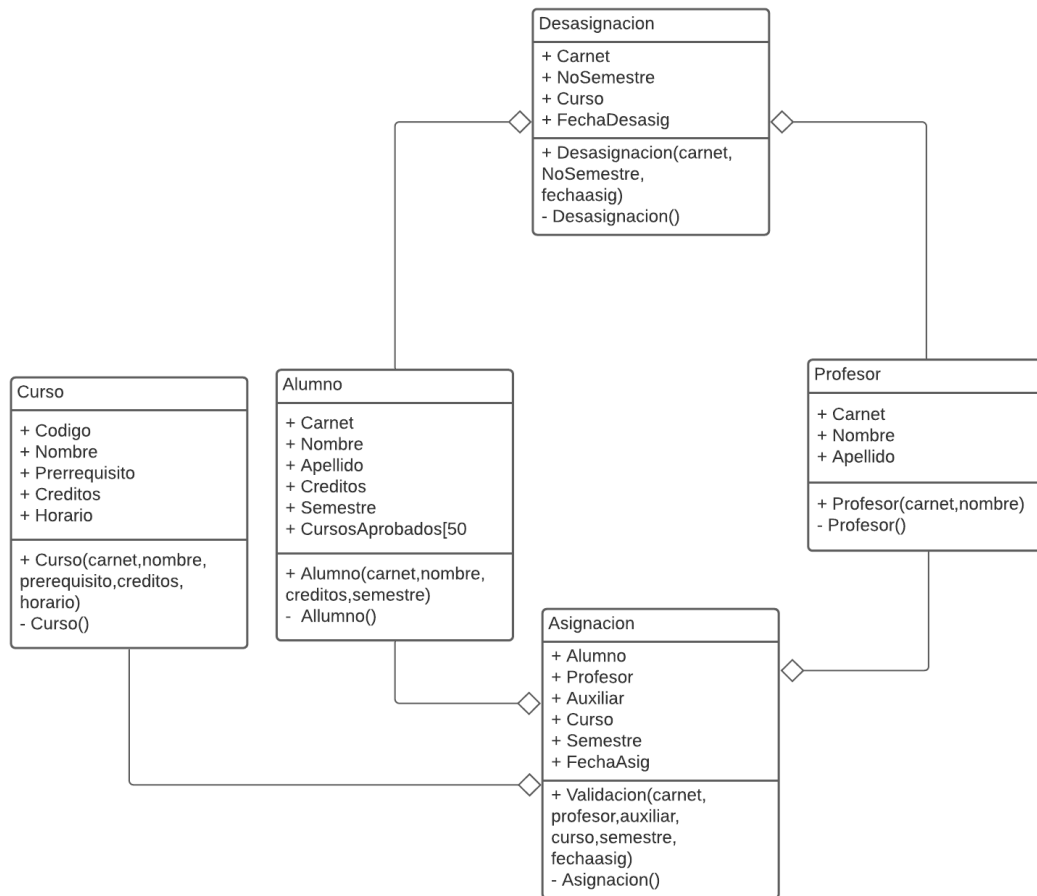
La otra opción, que es más común, es ejecutarla haciendo doble click sobre el ejecutable como con cualquier otro programa.

## 3. Almacenamiento de Datos

Para mantener la consistencia de datos, aunque el programa no esté en ejecución, se implementó Serialización, de tal modo que, al ingresar datos, modificarlos y/o eliminarlos o al terminar la ejecución los datos serán codificados en archivos binarios para que al volver a ejecutar el programa se realice un proceso opuesto al de la codificación para que el programa recupere inmediatamente los datos tal y como fueron codificados en la ejecución anterior.



## 4. Diagrama de Clases



## 5. Métodos Constructores

### 5.1. Constructor Profesor

Es el método que recibe y devuelve parámetros enviados masivamente por método lector del archivo json. Se instanció mediante un arreglo de tipo objeto para poder almacenar la información de los profesores.

```
1 package Profesores;
2 import java.io.Serializable;
3 public class Profesor implements Serializable {
4     private static final long serialVersionUID = 1L;
5     private int codigo;
6     private String nombre;
7     private String apellido;
8     private String correo;
9     private String contraseña;
10    private String genero;
11    public Profesor(int codigo,String nombre,String ap
12        this.setCodigo(codigo);
13        this.setNombre(nombre);
14        this.setApellido(apellido);
15        this.setCorreo(correo);
16        this.setContraseña(contraseña);
17        this.setGenero(genero);
18    }
19    public int getCodigo() {
20        return codigo;
21    }
22    public void setCodigo(int codigo) {
23        this.codigo = codigo;
24    }
25    public String getNombre() {
26        return nombre;
27    }
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
31    public String getApellido() {
32        return apellido;
33    }
34    public void setApellido(String apellido) {
35        this.apellido = apellido;
36    }
37    public String getCorreo() {
38        return correo;
39    }
40    public void setCorreo(String correo) {
41        this.correo = correo;
42    }
43    public String getContraseña() {
44        return contraseña;
45    }
46    public void setContraseña(String contraseña) {
47        this.contraseña = contraseña;
48    }
49    public String getGenero() {
50        return genero;
51    }
52    public void setGenero(String genero) {
53        this.genero = genero;
54    }
55 }
```

### 5.2. Constructor Notas de Actividades

Recibe y devuelve los parámetros enviados masivamente por el método que lee el archivo csv que contiene el código del estudiante y su respectiva calificación. Los datos se almacenan en un arreglo de tipo objeto.

```
1 package Actividades;
2 import java.io.Serializable;
3 public class NotaActividad implements Serializable {
4     private static final long serialVersionUID = 1L;
5     private int codAlu;
6     private int codCur;
7     private int codAct;
8     private double nota;
9     public NotaActividad(int codAlu,int codCur,int codAct,double nota) {
10        this.setCodAlu(codAlu);
11        this.setCodCur(codCur);
12        this.setCodAct(codAct);
13        this.setNota(nota);
14    }
15    public int getCodAlu() {
16        return codAlu;
17    }
18    public void setCodAlu(int codAlu) {
19        this.codAlu = codAlu;
20    }
21    public int getCodCur() {
22        return codCur;
23    }
24    public void setCodCur(int codCur) {
25        this.codCur = codCur;
26    }
27    public int getCodAct() {
28        return codAct;
29    }
30    public void setCodAct(int codAct) {
31        this.codAct = codAct;
32    }
33    public double getNota() {
34        return nota;
35    }
36    public void setNota(double nota) {
37        this.nota = nota;
38    }
39 }
```

### 5.3. Constructor Alumno

Es el método que devuelve y recibe parámetros enviados masivamente por método lector del archivo json. Se instanció mediante un arreglo de tipo objeto para poder almacenar la información de los alumnos.

```
1 package Alumnos;
2 import java.io.Serializable;
3 public class Alumno implements Serializable {
4     private static final long serialVersionUID = 1L;
5     private int codigo;
6     private String nombre;
7     private String apellido;
8     private String correo;
9     private String contraseña;
10    private String genero;
11    public Alumno(int codigo,String nombre,String apellido,String correo,String contraseña,String genero) {
12        this.setCodigo(codigo);
13        this.setNombre(nombre);
14        this.setApellido(apellido);
15        this.setCorreo(correo);
16        this.setContraseña(contraseña);
17        this.setGenero(genero);
18    }
19    public int getCodigo() {
20        return codigo;
21    }
22    public void setCodigo(int codigo) {
23        this.codigo = codigo;
24    }
25    public String getNombre() {
26        return nombre;
27    }
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
31    public String getApellido() {
32        return apellido;
33    }
34    public void setApellido(String apellido) {
35        this.apellido = apellido;
36    }
37    public String getCorreo() {
38        return correo;
39    }
40    public void setCorreo(String correo) {
41        this.correo = correo;
42    }
43    public String getContraseña() {
44        return contraseña;
45    }
46    public void setContraseña(String contraseña) {
47        this.contraseña = contraseña;
48    }
49    public String getGenero() {
50        return genero;
51    }
52    public void setGenero(String genero) {
53        this.genero = genero;
54    }
55 }
```

### 5.4. Constructor Curso

Devuelve y recibe parámetros enviados masivamente por método lector del archivo json. Se instanció mediante un arreglo de tipo objeto para poder almacenar la información de los cursos.

```
1 package Cursos;
2 import java.io.Serializable;
3 public class Curso implements Serializable {
4     private static final long serialVersionUID = 1L;
5     private int codigo;
6     private String nombre;
7     private int creditos;
8     private int alumnos;
9     private int profesor;
10    public Curso(int codigo,String nombre,int creditos,int alumnos,int profesor) {
11        this.setCodigo(codigo);
12        this.setNombre(nombre);
13        this.setCreditos(creditos);
14        this.setAlumnos(alumnos);
15        this.setProfesor(profesor);
16    }
17    public int getCodigo() {
18        return codigo;
19    }
20    public void setCodigo(int codigo) {
21        this.codigo = codigo;
22    }
23    public String getNombre() {
24        return nombre;
25    }
26    public void setNombre(String nombre) {
27        this.nombre = nombre;
28    }
29    public int getCreditos() {
30        return creditos;
31    }
32    public void setCreditos(int creditos) {
33        this.creditos = creditos;
34    }
35    public int getAlumnos() {
36        return alumnos;
37    }
38    public void setAlumnos(int alumnos) {
39        this.alumnos = alumnos;
40    }
41    public int getProfesor() {
42        return profesor;
43    }
44    public void setProfesor(int profesor) {
45        this.profesor = profesor;
46    }
47 }
```

## 5.5. Constructor Actividad

El método devuelve y recibe parámetros enviados masivamente por método lector del archivo json. Se instanció mediante un arreglo de tipo objeto para poder almacenar la información de las actividades.

```
1 package Actividades;
2 import java.io.Serializable;
3 public class Actividad implements Serializable {
4     private static final long serialVersionUID = 1L;
5     private int curso;
6     private int codigo;
7     private String nombre;
8     private String descripcion;
9     private double ponderacion;
10    public Actividad(int curso,int codigo,String nombre,String descripcion,double ponderacion) {
11        this.setCurso(curso);
12        this.setCodigo(codigo);
13        this.setNombre(nombre);
14        this.setDescripcion(descripcion);
15        this.setPonderacion(ponderacion);
16    }
17    public int getCurso() {
18        return curso;
19    }
20    public void setCurso(int curso) {
21        this.curso = curso;
22    }
23    public int getCodigo() {
24        return codigo;
25    }
26    public void setCodigo(int codigo) {
27        this.codigo = codigo;
28    }
29    public String getNombre() {
30        return nombre;
31    }
32    public void setNombre(String nombre) {
33        this.nombre = nombre;
34    }
35    public String getDescripcion() {
36        return descripcion;
37    }
38    public void setDescripcion(String descripcion) {
39        this.descripcion = descripcion;
40    }
41    public double getPonderacion() {
42        return ponderacion;
43    }
44    public void setPonderacion(double ponderacion) {
45        this.ponderacion = ponderacion;
46    }
47 }
```

## 6. Variables Globales

Se usaron variables globales para el registro de alumnos, profesores, asignación de cursos y asignación de alumnos a cursos, y para mostrar datos en tablas y las representaciones numéricas en las tablas.

También fueron utilizados componentes establecidos como globales, como labels, botones campos de texto etc., con el fin de extraer datos directamente de la fuente de origen.

## 7. Librerías Externas

Ya que Java no cuenta con librerías propias para realizar algunas acciones se importaron librerías externas.

### 7.1. json-simple-1.1.jar



Permite la lectura de archivos en formato .json se utilizó ésta, para poder importar los campos que contiene la información requerida del archivo.

```
▼ JSON
  ▼ json-simple-1.1.jar - C:\Users
    > org.json.simple
    > org.json.simple.parser
    > META-INF
```





## 7.2. itextpdf-5.4.0.jar

Permite exportar archivos en formato pdf.

▼  iText  
    >  itextpdf-5.4.0.jar - C:\Users'

## 7.3. jfreechart-1.5.0.jar

Permite realizar gráficas en el entorno gráfica mediante páneles.

▼  JFreeChart  
    >  jfreechart-1.5.0 (1).jar - C:\Users'