

## Manual Técnico

### Descripción general de la solución.

Fue realizado en el lenguaje de java con el IDE NetBens, utilizando las distintas herramientas de interfaz gráfica. El uso de diferentes arreglos que contendrán la cantidad de profesores, cursos, alumnos permitidos por cada arreglo. Se necesitó el uso de constructores para los arreglos y los “setters and getters”. La incorporación de librerías para abrir archivos CSV y descomponerlo, así como para tablas.

### Requerimientos mínimos del entorno de desarrollo.

- IDE para el desarrollo, en este caso es Apache NetBeans IDE 17.
- JDK acorde a la versión del IDE, JDK 20.
- Memoria RAM de 8.
- Gráficos integrados para la fluidez.
- Procesador Intel Core i5.
- Sistema Operativo Windows.
- Almacenamiento de 1 GB.

### Diccionario de Clases y Métodos.

```
public String codigo;
public String nombre;
public String apellido;
public String correo;
public String contraseña;
public String genero;

public Profesor(String codigo, String nombre, String apellido, String correo, String contraseña, String genero) {
    this.codigo = codigo;
    this.nombre = nombre;
    this.apellido = apellido;
    this.correo = correo;
    this.contraseña = contraseña;
    this.genero = genero;
}
```

Clase Profesor, utilizado para los arreglos de los atributos del objeto Profesor, así como la creación del constructor.

```
public String codigo;
public String nombre;
public String creditos;
public String alumnos;
public String profesor;

public Curso(String codigo, String nombre, String creditos, String alumnos, String profesor) {
    this.codigo = codigo;
    this.nombre = nombre;
    this.creditos = creditos;
    this.alumnos = alumnos;
    this.profesor = profesor;
}
```

Clase Curso, utilizado para los arreglos de los atributos del objeto Curso, así como la creación del constructor.

```
public void actualizarListadoProfesores() {
```

Método utilizado para que la tabla de Listado de Profesores se mantenga actualizada luego de cualquier cambio realizado a las mismas.

```
public void actualizarListadoCursos() {
```

Método utilizado para que la tabla de Listado de Cursos se mantenga actualizada luego de cualquier cambio realizado a las mismas.

```
private void CargaMBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Método creado para seleccionar un archivo en formato CSV del ordenador, descomponerlo de acuerdo a como están separados los campos del archivo y posteriormente mostrarlos en la tabla de Listado de Profesores.

```
public class CrearProfesor extends javax.swing.JFrame {
```

Utilizada para mostrar un JFrame en donde se ingresan los campos para la creación de un nuevo Profesor y de igual manera acciones para regresar al cerrar la ventana.

```
private void ActualizarBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Se usará el getSelectRow que hará que cuando se seleccione una fila de la tabla, mandara los datos de la fila seleccionada a la ventana en donde se ingresaran los campos del Profesor.

```
private void EliminarCursosBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Se usará para cuando se tenga seleccionada la fila que se desea eliminar. Haciendo uso del ciclo for para ocupar el espacio de la fila eliminada.

```
private void ExportarBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Utilizada para exportar la tabla de Listado de Profesores a un formato de documento de HTML, haciendo uso del ciclo for y cadenas de texto.

```
private void ActualizarDatosBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Utilizado para actualizar los datos del Profesor, haciendo uso del constructor creado anteriormente y llamando el arreglo de la Tabla para ser mostradas.

```
private void CursoBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Utilizado para entrar al Administrador de Curso, en donde se podrán realizar diversas acciones como lo son nuevas actividades, cargar notas, etc.

```
public String codigo;  
public String nombre;  
public String apellido;  
public String correo;  
public String genero;  
  
public Alumno(String codigo, String nombre, String apellido, String correo, String genero) {  
    this.codigo = codigo;  
    this.nombre = nombre;  
    this.apellido = apellido;  
    this.correo = correo;  
    this.genero = genero;  
}
```

Clase Alumno, utilizado para los arreglos de los atributos del objeto Alumno, así como la creación del constructor Alumno.

```
public String codigo;  
public String nota;  
  
public Nota(String codigo, String nota) {  
    this.codigo = codigo;  
    this.nota = nota;  
}
```

Clase Nota, utilizado para los arreglos de los atributos del objeto Nota, así como la creación del constructor Alumno.

```
private void CargaMAlumnosActionPerformed(java.awt.event.ActionEvent evt) {
```

Método creado para seleccionar un archivo en formato CSV del ordenador, descomponerlo de acuerdo a como están separados los campos del archivo y posteriormente mostrarlos en la tabla de Listado de Alumnos.

```
private void EliminarEstuBtnActionPerformed(java.awt.event.ActionEvent evt)
```

Se usará para cuando se tenga seleccionada la fila que se desea eliminar. Haciendo uso del ciclo for para ocupar el espacio de la fila eliminada.

```
private void HTMLAlumnosBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Utilizada para exportar la tabla del Listado de Alumnos a un formato de documento de HTML, haciendo uso del ciclo for y cadenas de texto.

```
public void sumar() {
```

Utilizada para hacer el acumulado, empleando un ciclo for y que cada actividad creada se le sea aumentada la ponderación que se estipula al cargarla.

```
public void actualizarListadoNotas() {
```

Método utilizado para que la tabla de Listado de Alumnos se mantenga actualizada luego de cualquier cambio realizado a las mismas.

```
private void CSVActividadBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Método creado para seleccionar un archivo en formato CSV del ordenador, descomponerlo de acuerdo a como están separados los campos del archivo y posteriormente mostrarlos en la tabla de Listado de Alumnos.

```
private void CrearActividadBtnActionPerformed(java.awt.event.ActionEvent evt) {
```

Utilizado para crear las diferentes actividades, haciendo uso de arreglos y insertando los campos ingresados a la tabla.