



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

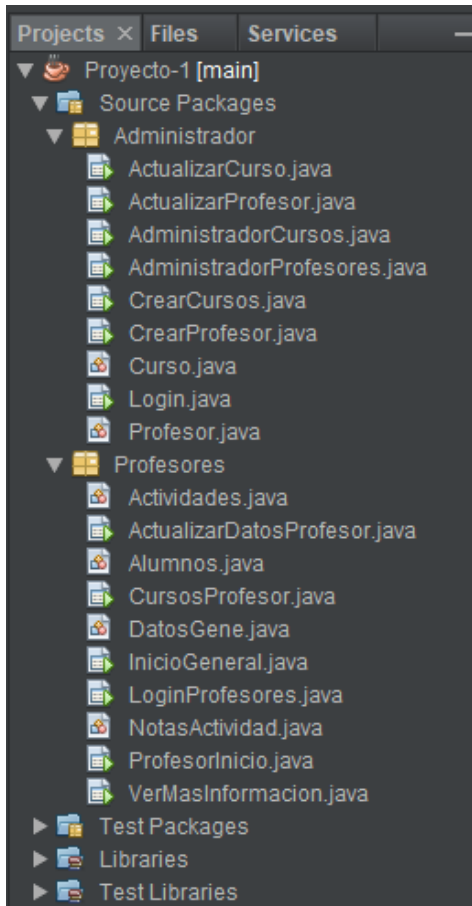
PROYECTO 1

MANUAL TÉCNICO

Introducción a la programación y computación 1

Miguel Adrian Tubac Agustin
202101927

Descripción de la solución



El programa resuelve para la Escuela de Ciencias y Sistemas, la necesidad de gestión de los cursos impartidos en la carrera. A este sistema pueden acceder los siguientes tipos de usuarios:

- Administrador: cargará profesores y cursos al sistema. La plataforma cuenta con un único usuario administrador.

- Profesor: podrá ser asignado a un curso que se encuentre cargado en el sistema, y dentro podrá cargar alumnos, crear actividades, y generar reportes sobre las actividades. La plataforma cuenta con capacidad para almacenar 50 profesores y 300 alumnos.

Requerimientos mínimos del entorno de desarrollo

[Ver información básica acerca del equipo](#)

Edición de Windows

Windows 10 Pro

© Microsoft Corporation. Todos los derechos reservados.



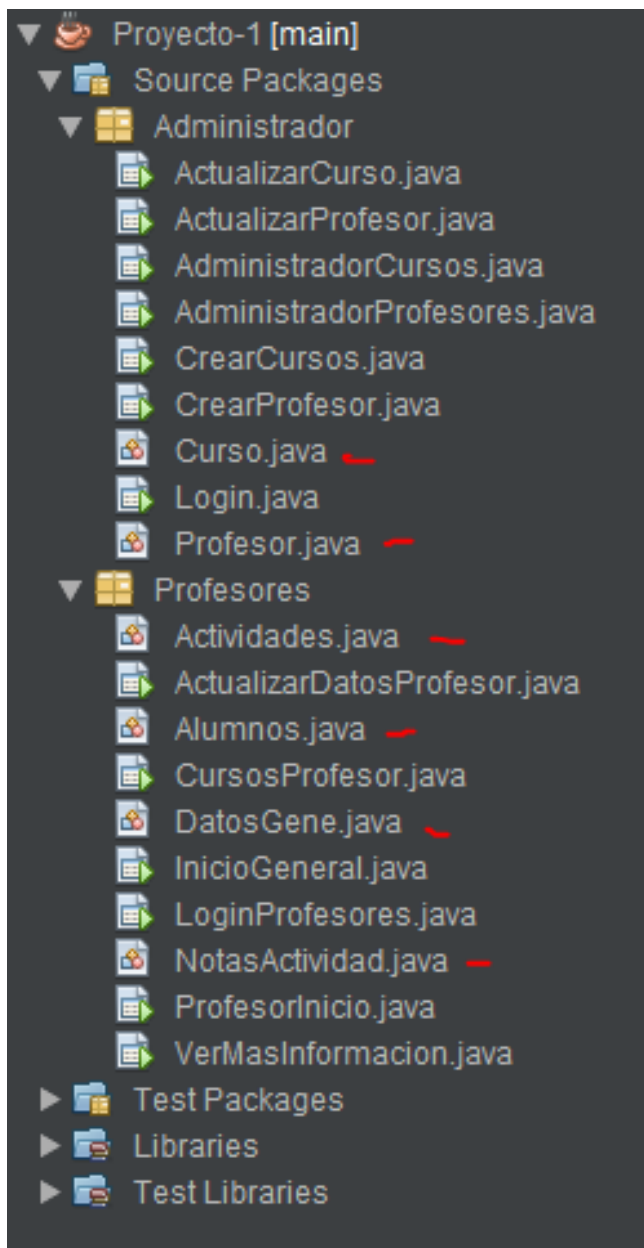
Sistema

Procesador: Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz
Memoria instalada (RAM): 8.00 GB (7.68 GB utilizable)
Tipo de sistema: Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla

Los requerimientos necesarios para la edición y ejecución del programa de gestión de estudiantes deberán ser los siguientes para poder garantizar la correcta depuración del mismo:

- Modelo del procesador i5-6300U
- Procesador Intel® Core™ i5 de la sexta generación
- Capacidad de disco duro 250 GB
- Memoria RAM 8 GB
- Sistema Operativo Windows 10

Diccionario de clases



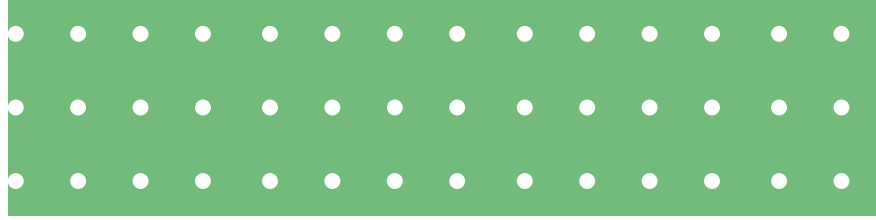
Curso: en esta clase se almacenan los datos correspondientes a los cursos que se ingresen al sistema, este cuentan con los siguientes campos:

- public String codigo;
- public String nombre;
- public String credits;
- public String alumnos;
- public String profesor;

Profesor: en esta clase se almacenan los datos correspondientes a los profesores que se ingresen al

sistema, este cuenta con los siguientes campos:

- public String codigo;
- public String nombre;
- public String apellido;
- public String correo;
- public String contrasenia;
- public String genero;



Actividades: en esta clase se almacenan los datos correspondientes a las actividades que se ingresen al sistema, este cuentan con los siguientes campos:

- public String Nombre;
- public String Descripcion;
- public double Ponderacion;


Alumnos: en esta clase se almacenan los datos correspondientes a los alumnos que se ingresen al sistema, este cuentan con los siguientes campos:

- public String codigo;
- public String nombre;
- public String apellido;
- public String correo;
- public String genero;

DatosGene: en esta clase se almacenan los datos correspondientes a las clases Alumnos y NotaActividad que se ingresen al sistema, este cuentan con los siguientes campos:

- public String codigo;
- public String nombre;
- public String apellido;
- public String correo;
- public double Nota;





NotaActividad: en esta clase se almacenan los datos correspondientes a las notas que obtiene cada alumno en cierta actividad que se ingresen al sistema, este cuentan con los siguientes campos:

- public String codigo;
- public double nota;

Diccionario de métodos

ActualizarCurso:


En esta parte del sistema se realiza la actualización de algún curso que se seleccione.

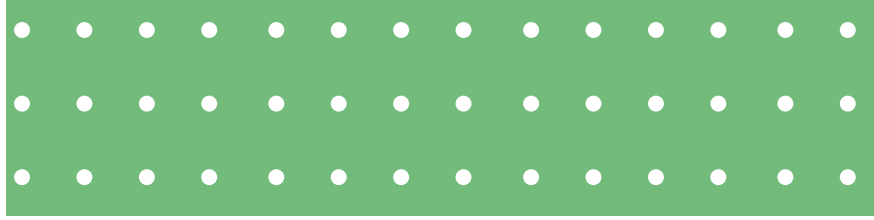
```
public class ActualizarCurso extends javax.swing.JFrame {  
  
    /**  
     * Creates new form CrearProfesor  
     */  
    public ActualizarCurso() {  
        initComponents();  
        meterDatos();  
        CodigoInput.setEnabled(enabled: false);  
    }  
    //clase de degradado  
    class JPanelGradient extends JPanel{  
  
        JPanelGradient() {
```

ActualizarProfesor:

```
public class ActualizarProfesor extends javax.swing.JFrame {  
  
    /**  
     * Creates new form CrearProfesor  
     */  
    public ActualizarProfesor() {  
        initComponents();  
  
        meterDatos();  
        CodigoInput.setEnabled(enabled: false);  
    }  
}
```

En esta parte del sistema se realiza la actualización de algún Profesor que se seleccione.





AdministradorCursos:

En esta parte del sistema se muestra la información correspondiente a los cursos que se ingresen en el sistema.

```
public class AdministradorCursos extends javax.swing.JFrame {  
  
    public static Curso[] cursos = new Curso[100];  
    public static int contadorCursos;  
  
    // Para actualizar curso  
    public static int fila;  
  
    public static String codigo;  
    public static String nombre;  
    public static String credits;  
    public static String alumnos;  
    public static String profesor;  
  
    /**  
    */  
}
```

AdministradorProfesores:

```
public class AdministradorProfesores extends javax.swing.JFrame {  
  
    public static Profesor[] profesores = new Profesor[50];  
  
    public static int contadorProfesores;  
    public static String[] datos;  
    public static String[] codi;  
    public static String[] contra;  
  
}
```

En esta parte del sistema se muestra la información correspondiente a los profesores que se ingresen en el sistema.

CrearCursos:

En esta parte del sistema se crean nuevos cursos y se guardan en el mismo, para posteriormente mostrarlo en Administrador de cursos.

```
public class CrearCursos extends javax.swing.JFrame {  
  
    public CrearCursos() {  
        initComponents();  
  
        meterDatos();  
    }  
  
}
```





CrearProfesor:

```
public class CrearProfesor extends javax.swing.JFrame {  
    public CrearProfesor() {  
        initComponents();  
    }  
    class jPanelGradient extends JPanel{
```

En esta parte del sistema se crean nuevos profesores y se guardan en el mismo, para posteriormente mostrarlo.

Login:

Esta parte se muestra y se comprueba que el usuario que está ingresando corresponde al administrador.

```
public class Login extends javax.swing.JFrame {  
    /**  
     * Creates new form Login  
     */  
    public Login() {  
        initComponents();  
    }
```

ActualizarDatosProfesor:

```
public class ActualizarDatosProfesor extends javax.swing.JFrame {  
    public ActualizarDatosProfesor() {  
        initComponents();  
        meterDatos();  
    }
```

En esta parte del sistema se actualiza la información del Profesor que se encuentre ingresado al sistema.





CursosProfesor:

En esta parte del sistema se muestra la información correspondiente a los alumnos y actividades que se ingresen en el sistema y estén asociadas a un profesor.

```
public class CursosProfesor extends javax.swing.JFrame implements MouseListener {  
  
    public static Alumnos[] alumnos = new Alumnos[300];  
    public static Actividades[] actividad = new Actividades[100];  
    public static NotasActividad[] notas = new NotasActividad[100];  
    public static DatosGene[] datosGene = new DatosGene[100];  
    public static DatosGene[] arregloCopia;  
  
    public static int contadorAlumnos;  
    public static int contadorNotas;  
    public static int contadorActividades=0;  
    public static int filaS;  
    public static double acumulado;  
    public static double ponderacionActividad;  
  
    public static double[] resguardo;  
    JLabel etiqueta = new JLabel();  
}
```

InicioGeneral:

```
public class InicioGeneral extends javax.swing.JFrame {  
  
    /**  
     * Creates new form InicioGeneral  
     */  
    public InicioGeneral() {  
        initComponents();  
    }  
}
```

En esta parte del sistema se muestra al usuario los diferentes roles que puede acceder al mismo.

LoginProfesores:

Esta parte se muestra y se comprueba que el usuario que está ingresando corresponde a un profesor que se encuentre registrado al mismo.

```
public class LoginProfesores extends javax.swing.JFrame {  
    public static String codigo;  
  
    public LoginProfesores() {  
        initComponents();  
    }  
}
```





ProfesorInicio:

```
public class ProfesorInicio extends javax.swing.JFrame {  
  
    /**  
     * Creates new form ProfesorInicio  
     */  
    public ProfesorInicio() {  
        initComponents();  
        tablaActual();  
    }  
}
```

En esta parte del sistema se muestra el curso que se le asigne al profesor que ingrese en este apartado.

VerMasInformacion:

Esta parte muestre la información a detalle de algún alumno y además puede eliminar al propio alumno.

```
public class VerMasInformacion extends javax.swing.JFrame {  
  
    public VerMasInformacion() {  
        initComponents();  
        meterDatos();  
    }  
}
```

