

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS Y SISTEMAS  
LABORATORIO DE IPC 1

PROYECTO 1  
MANUAL TECNICO

DENNIS MAURICIO CORADO MUÑOZ  
CARNET: 202010406  
CUI: 3032329780108  
GUATEMALA, GUATEMALA, 20/03/2021

## **Objetivos y alcances**

### **Objetivos generales**

Poder familiarizarme con el lenguaje de programación Java, aplicar los conocimientos desarrollados a lo largo del curso de Introducción a la Programación y Computación 1, y poder aplicar una lógica correcta para afrontar la problemática planteada.

### **Objetivos específicos**

- Utilizar el lenguaje de programación Java como herramienta de desarrollo de software.
- Aplicación de conceptos de programación orientada a objetos
- Construcción de aplicaciones con interfaz grafica
- Acercamiento al formato JSON y csv.

## Especificación técnica

### Requisitos de Hardware (Mínimos)

- **Microsoft Windows Vista**
  - **Procesador:** Intel Pentium III o equivalente
  - **Memoria:** 512 MB
  - **Espacio en disco:** 750 MB libres
- **Ubuntu 9.10**
  - **Procesador:** Intel Pentium III o equivalente
  - **Memoria:** 512 MB
  - **Espacio en disco:** 650 MB libres
- **Macintosh OS X 10.7 Intel**
  - **Procesador:** Intel Pentium III o equivalente
  - **Memoria:** 512 MB
  - **Espacio en disco:** 750 MB libres

Cualquier hardware superior al establecido es capaz de ejecutar la aplicación y poder ver sus archivos por medio de un editor de código similar a Eclipse o NetBeans.

### Requisitos de Software

- **Sistema Operativo:** Windows, masOS, Linux, Solaris
- **Lenguaje de Programación:** Java SE 12
- **IDE:** NetBeans o Eclipse

## Lógica del Programa

- **Clases Utilizadas (Objetos)**

El programa utiliza un total de 7 clases y una clase principal.

En la clase principal se puede encontrar

- Una variable global utilizada para instanciar el objeto "Login"
- Un método principal dentro del cual se termina de instanciar el objeto login y se llama un método del mismo denominado "displayLogin"

```
package proyectol;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
  
/**  
 * @author dennis  
 */  
public class Proyectol {  
  
    static Login login;  
    public static void main(String[] args) throws IOException, FileNotFoundException, ClassNotFoundException {  
        login = new Login();  
        login.displayLogin();  
    }  
}
```

Clase Login:

- Dentro de la clase Login se puede encontrar una instancia de la clase "ObjetosActividades" así como campos de texto, etiquetas, botones, un panel y un frame, declarados como variables globales.
- Luego se encuentra el método "displayLogin", que sirve para poder desplegar la interfaz grafica de login como también para verificar si el archivo serializado existe.
- Método "login", sirve para poder establecer todos los componentes respectivos de la interfaz grafica en su debido lugar así como añadir ciertas funcionalidades a este.
- Método "loginSave", debido a que el usuario tendrá la opción de cerrar sesión se creo este método que por medio de un objeto de tipo "ObjetosActividades" lo recibe como parámetro para asignarle al objeto instanciado dentro de login los datos ingresados en la clase "Administracion".
- El respectivo método abstracto para poder localizar los botones seleccionados dentro de este se incluye código que es capaz de determinar la entrada del usuario y determinar quien es o que papel tiene la persona dentro de la aplicación, administrador, estudiante o profesor.
- Métodos "leerSerializable" y "serializar", como el nombre de cada una lo indica, el método serializar, crea un archivo binario que será leído por el método leerSerializable para poder iniciar el sistema desde donde se dejó.

## Clase Administracion

- Dentro de la clase administración se logra evidenciar que se utilizaron bastante variables globales con el objetivo de poder leerlas y cambiar su valor en cualquier parte del código todos los componentes declarados como globales cambian su valor por los métodos dentro de esta clase, así como variables enteras y arreglos de tipo String para poder controlar ciertos cambios más adelante.
- Método administración, este método sirve para poder ingresar todo lo relacionado a la interfaz grafica así como actionlisteners a botones para después crear su funcionalidad.
- Método print, este método funciona por medio de una librería instalada en el programa que permite crear y escribir archivos de extensión pdf, recibe como parámetro la tabla a imprimir y el título que tendrá el archivo creado.
- Método crear, este método sirve en conjunto junto con los botones crear del módulo administración, este método recibe como parámetro un título, un arreglo de etiquetas, un arreglo de objetos, y un arreglo de strings que contendrán las opciones que tendrá dicho frame.
- Método cargaMasiva, dicho método su funcionalidad es por medio de una librería leer archivos tipo JSON y poder almacenarlos en los objetos deseados, únicamente recibe como parámetro un arreglo de objetos.
- Método actualizar, este método sirve para poder actualizar la información de profesores o cursos dentro de la aplicación, funciona seleccionando un dato respectivo de la table, este método reconocerá la fila selecciona y hará el cambio respectivo en la tabla como en el objeto.
- Método cambio, funciona en conjunto con el método actualizar para poder realizar los cambios en el objeto.
- Método eliminar, recibe como parámetro un arreglo de objetos, este método tiene una deficiencia y es que no se puede eliminar objetos de forma aleatoria por lo cual es un método que puede ocasionar problemas
- Método gráfica, utiliza una librería para poder crear gráficos de los distintos objetos es un método que debe ser llamado para poder funcionar, en este caso es llamado en la clase carga masiva.

## Clase ObjetosActividades

- Utiliza como variables globales 3 contadores para los 3 diferentes objetos que contendrá la aplicación así como crea un arreglo de cada tipo de objeto.
- Esta clase contiene métodos bastante similares ya que para poder guardar objetos se utilizan métodos que tienen como parámetros las distintas variables declaradas en cada objeto respectivo y en estos métodos se instancia a un objeto de cada tipo y se almacena dentro de los arreglos establecidos como globales
- Método asignarAlumno y asignarNotas funcionan de forma similar, se apoyan en dos funciones que retornan un objeto, básicamente estas funcionan checkean si el objeto existe y coincide con el ID proporcionado para así retornar el objeto en su posición deseada y asignan un valor a una variable no instanciada dentro de cada objeto.

```

public void guardarAlumno(int codigo, String nombre, String apellido, String correo, String genero){
    Alumno nuevoAlumno = new Alumno(codigo, nombre, apellido, correo, genero);
    alumnos[contadorAlumnos] = nuevoAlumno;
    contadorAlumnos++;
}

public void asignarAlumno(int codigo, int idCurso, int posicion){
    Alumno alumno_asignable = getAlumnoAs(codigo);
    Curso curso_asignable = getCursoAs(idCurso);
    curso_asignable.setAlumnos(alumno_asignable, posicion);
}

```

Las clases Alumno, Curso y Profesor son las clases donde se crea el objeto y sus atributos por medio del constructor y sus propios getters y setters.

### Clase ModuloProfesor

- Esta clase tiene gran cantidad de variables y componentes globales, cada uno de estos cambia su valor en diferentes métodos algunos sirven solo para poder proporcionar información a otros métodos
- Constructor ModuloProfesor, este constructor sirve para crear toda la interfaz grafica y crear los botones que corresponden al respectivo profesor, si un profesor no tiene cursos asignados este no tendrá botones para manejarlos.
- Método actualizarDatos, funciona al ser presionado el botón del mismo nombre y permite tal y como dice, actualizar datos, así como también hace el cambio en el objeto respectivo
- Método adminCurso, este método se encarga de la interfaz grafica que se utilizara al momento de que se hace clic sobre el botón que representan los distintos cursos.
- Método cargaMasiva, al igual que en administración este método lee un archivo de tipo JSON y muestra información en una tabla del método adminCurso
- Método cargarCsv, este método como su nombre lo indica sirve para leer un archivo csv, delimitado por comas y almacena su información en los objetos respectivos
- Método crearActividad, funciona al ser presionado el botón del mismo nombre y lo que realiza es que envía la información respectiva a una tabla de actividades así como asigna actividades a los distintos alumnos.
- Función bubblesort, esta función es un método de ordenamiento que ordena los datos proporcionados de tipo double de mayor a menor para poder hacer más fácil el reconocimiento de los alumnos con mejores y peores rendimientos.
- Método print, funciona igual que en administración y su función es imprimir las tablas respectivas a la información de los alumnos con mejor y peor rendimiento. Por alguna razón no imprime la tabla respectiva y crea un documento en blanco, método no terminado en su totalidad
- Los métodos top5 mejor y peor funcionan de forma similar y se apoyan en el método de ordenamiento contenido en la función bubblesort, para obtener el mejor se toman los últimos 5 datos del arreglo de notas y para poder obtener el peor se toman los primeros 5 datos del arreglo de notas.