ACTIVIDAD 01 PRIMERA ENTREGA: JAVA

GIIN 22 – PROYECTOS DE PROGRAMACIÓN

Ivan Gallego Bravo
gb.ivan@gmail.com

CONTENIDO

1. Introducción al Proyecto

Este proyecto trata sobre la creación de una aplicación en Java sobre la presentación de documentos contables digitales a un organismo fiscal. Para su desarrollo se van a usar Java sobre el IDE Eclipse, una base de datos como MariaDB, un interfaz de usuario implementada en Java y una herramienta para compartir código como GitHub.

1.1 SISTEMAS IMPLEMENTADOS EN ESTA ACTIVIDAD

Al ser esta la primera entrega del proyecto se ha implementado algunas de las funcionalidades del proyecto final. En esa primera entrega me he centrado en la parte de Java y he creado una primera versión que emula el comportamiento del proyecto final, pero sin la parte de la base de datos ni la interfaz gráfica.

Para esta actividad he diseñado las principales clases en java y sus métodos que usaré a lo largo del proyecto.

2. CLASES Y METODOS EN JAVA

El programa en Java por el momento dispone de una clase principal que se encarga de gestionar el menú principal, definir los objetos necesarios y llamar a los métodos para darle la funcionalidad.

2.1 CLASE PRINCIPAL: CONTABLES

Para gestionar el menú se ha implementado con una variable de estado y un switch que gestiona cada una de las opciones del menú.

```
public static void main(String[] args)
{
    int estado = -1, accion = -1;

    Usuario user = new Usuario();
    Municipio mun = new Municipio();
    Convocatoria conv = new Convocatoria();
    Presentacion pres = new Presentacion();

    Scanner leer = new Scanner(System.in);
    while (estado != 0)
    {
        switch(estado)
        {
            case -1:  //Estado Inicial
                System.out.println("Bienvenido al programa de presentaciones digitales contables");
                System.out.println("Introduzca el número del menu al cual quieres acceder:");
                System.out.println("1)Usuarios");
                System.out.println("2)Municipios");
                System.out.println("3)Convocatorias");
                System.out.println("4)Presentaciones");
                System.out.println("5)Información");
                System.out.println("0)Salir");
                System.out.print("Menu: ");
                estado = leer.nextInt();
```

2.2 CLASE: USUARIO

Se ha definido una clase usuario que se encarga de recopilar la información del usuario y poder modificarla, mostrarla o borrarla sin problemas. La información en estas clases se guarda como una cadena de caracteres dentro de un ArrayList que nos permite manejarlo de forma similar a una base de datos.

```
public class Usuario
{
    ArrayList<String> lista_usuarios = new ArrayList<String>();
    Scanner leer = new Scanner(System.in);

    public Usuario()
    {

    }
    public void agregar()
    {
        String usuario_info = "";

        System.out.println("Introduce Nombre: ");
        usuario_info+= leer.next()+' ';
        System.out.println("Introduce clave: ");
        usuario_info+= leer.next()+' '+this.tipo();
        lista_usuarios.add(usuario_info);
    }
    public void eliminar()
    {
        if(lista_usuarios.size() > 0)
        {
            System.out.println("Elije el usuario a eliminar");
            this.mostrar();
            System.out.println("Eliminar: ");
            int eliminar = leer.nextInt();
            lista_usuarios.remove(eliminar);
        }
        else
            System.out.println("No hay usuarios que eliminar");
```

2.2 CLASE: MUNICIPIO

De forma similar al resto se ha creado una clase para controlar los municipios. Cada una de las clases llevan sus propios campos de información.

```
public class Municipio
{

    ArrayList<String> lista_municipios = new ArrayList<String>();
    Scanner leer = new Scanner(System.in);
    public Municipio()
    {

    }

    public void agregar()
    {
        String municipio_info = "";

        System.out.println("Introduce Nombre: ");
        municipio_info+= leer.next()+' ';
        System.out.println("Introduce categoria: ");
        municipio_info+= leer.next()+' ';
        System.out.println("Introduce usuario Cuentadante: ");
        municipio_info+= leer.next();
        lista_municipios.add(municipio_info);
```

2.2 CLASE: CONVOCATORIA

Al igual que resto de clases, se han implementado los métodos específicos para cada caso. Aquí podemos ver un método que emula un menú desplegable con opciones limitadas.

```
}
private String tipo()
{
    System.out.println("Introduzca el tipo de documentacion:");
    System.out.println("1)Libro Diario");
    System.out.println("2)Libro Mayor");
    System.out.println("3)Balance de sumas y saldos");
    System.out.println("4)Registro de ingresos de caja");
    System.out.println("5)Registro de movimientos de bancos");
    System.out.println("Elije una opción: ");
    int tipo= leer.nextInt();
    if (tipo == 1)
        return "Libro Diario";
    else if(tipo == 2)
        return "Libro Mayor";
    else if(tipo == 3)
        return "Balance de sumas y saldos";
    else if(tipo == 4)
        return "Registro de ingresos de caja";
    else
        return "Registro de movimientos de bancos";


}
```

2.2 CLASE: PRESENTACIÓN

Otra clase más, en este caso, para el control de las presentaciones. En esta ocasión muestro el método para modificar un dato ya introducido.

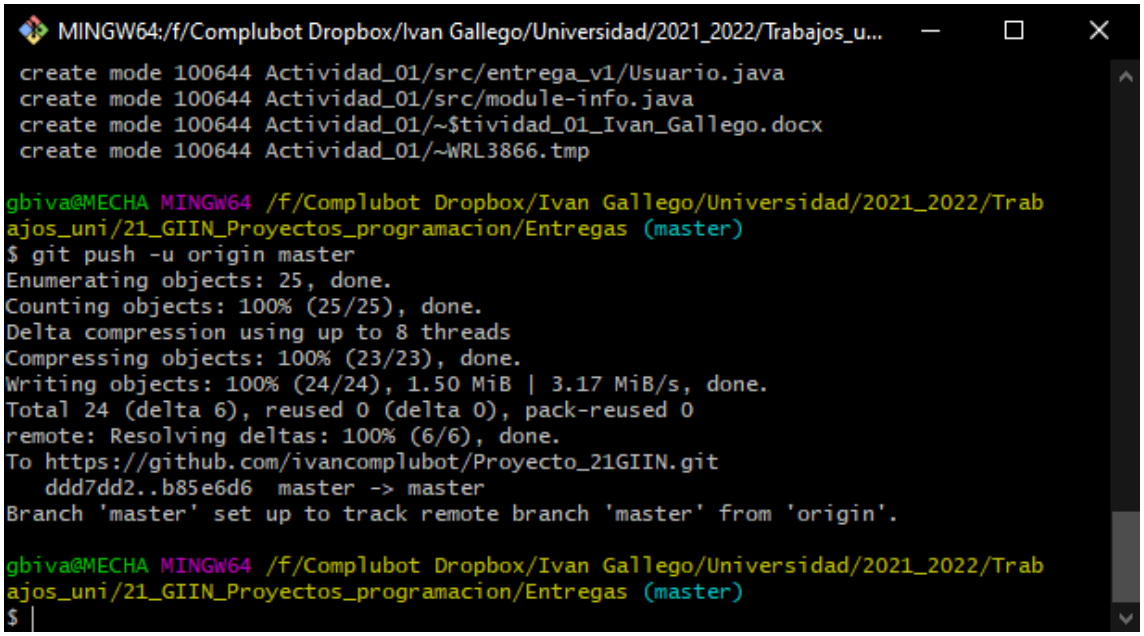
```
}
public void modificar()
{
    if(lista_presentacion.size() > 0)
    {
        String presentacion_info = "";
        System.out.println("Elije la presentacion a modificar");
        this.mostrar();
        System.out.println("Modificar: ");
        int modificar = leer.nextInt();

        System.out.println("Fecha de presentacion: ");
        presentacion_info+= leer.next()+' ';
        System.out.println("Estado: ");
        presentacion_info+= leer.next()+' '+this.tipo();
        presentacion_info+= leer.next()+' '+this.tipo();
        lista_presentacion.set(modificar,presentacion_info);
    }
    else
        System.out.println("No hay presentaciones que modificar");
```

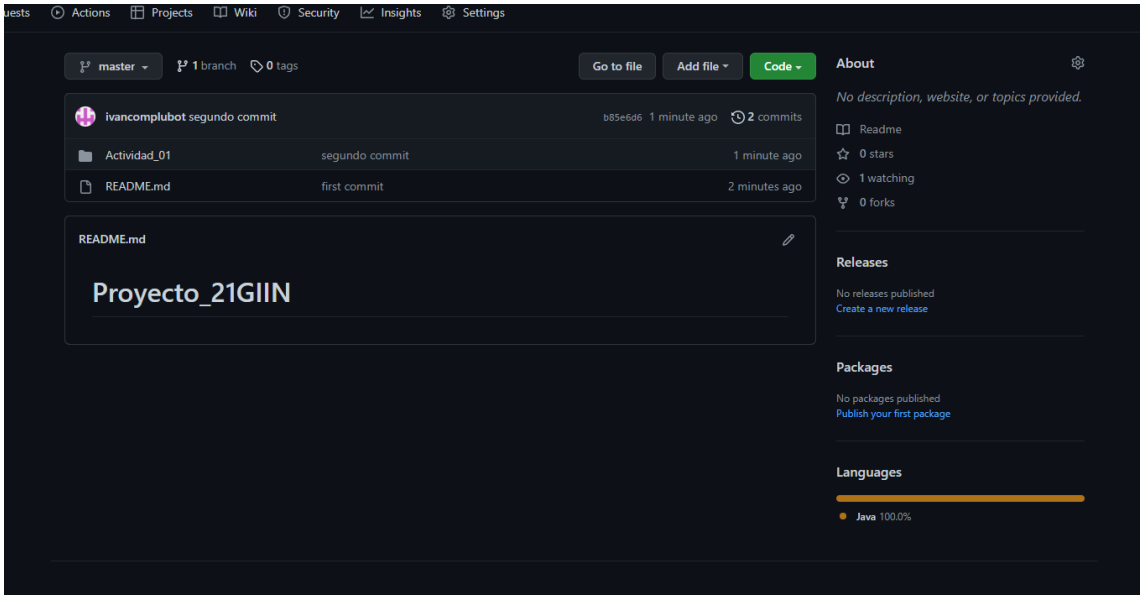
3. HERRAMIENTA PARA COMPARTIR CÓDIGO: GITHUB

Para compartir el código generado en las distintas actividades voy a usar mi repositorio de GitHub y voy a subir los archivos utilizando GIT.

Este es mi repositorio: https://github.com/ivancomplubot/Proyecto_21GIIN

Usando GIT para subir el código:





4. CONCLUSIONES

Creo que esta actividad ha sido una buena primera aproximación, en la siguiente entrega me voy a centrar más en darle la estructura final al proyecto intentando implementar las funciones que faltan desde el punto de vista de Java.