



---

# REPORTE PROYECTO EXAMEN EXTRAORDINARIO

---

programación de interfaces graficas del usuario



6 DE JUNIO DE 2023  
JESSICA ALEJANDRA ZUVIETA TEJEDA  
Edgar Caamal Dzulu

La primera instrucción del primer proyecto nos dice



### **Requisito Funcional #14568-2021.- Preparación del Entorno**

En esta primera versión de su aplicación crea un repositorio en una plataforma GIT (Github, Gitlab).

Inicie creando una aplicación que tenga un método que se llame `author1234567890`, donde los dígitos representan el número de su matrícula, incluya en ese método los datos de nombre, matrícula, correo electrónico, añada de igual forma uno de sus hobbies, ejemplo "leer", "correr", "musica", etc. Guarde todos sus datos e imprima el resultado en pantalla.

En este caso creamos una aplicación cuyo método es `author1234567890`, reemplazamos los datos usando mi nombre y matrícula quedando así `Jessica59640`, en el mismo método añadimos los datos de nombre, matrícula, correo y un hobby.

Entonces queda de la siguiente manera

```
public class Aplicacion {  
    public String Jessica59640() {  
        String nombre = "Jessica";  
        String matricula = "59640";  
        String correo = "al059640@gmail.com";  
        String hobby = "programar";  
    }  
}
```

*(cabe aclarar que me he confundido en el correo y he puesto "@gmail.com" cuando debería ser "@uacam.mx")*

Al imprimir este es el resultado

```
Nombre: Jessica  
Matr cula: 59640  
Correo electr nico: al059640@gmail.com  
Hobby: programar
```

El siguiente proyecto nos dice lo siguiente



**Requisito Funcional #14569-2021.-** Cree una clase que tenga los atributos de su archivo json: id, codigo, nombre, ...

Deberá mostrar en su evidencia la instancia en un frame que tenga los elementos la clase planetas debe utilizar el archivo planetas.json e imprimir los atributos, UA:43.13, id:11, nombre:"Haumea" y así sucesivamente)

Realice su segundo commit o envío por correo electrónico.

10 puntos

Aquí nos pide crear una clase con los atributos del archivo

```
public class Planetas {  
  
    private String UA;  
    private String id;  
    private String nombre;  
  
    public static ArrayList<Planetas> planetas = new ArrayList<>();  
  
    public Planetas(){  
  
    }  
  
    public Planetas(String UA, int id, String nombre) {  
        this.UA = UA;  
        this.id = id;  
        this.nombre = nombre;  
  
    }  
  
    public static void insertarPlanetas(String UA, int id, String nombre){  
        planetas.add(new Planetas(UA, id, nombre));  
    }  
}
```

Lo siguiente que nos pide es



**Requisito Funcional #14875-2021.-** Para darle funcionalidad a su aplicación es necesario segmentar los procesos de captura por tal motivo su aplicación debe contener un menú con dos opciones mínimas (Alta de Información y Mostrar los Datos). Puede utilizar JFrame o Panel, usted decide la forma más funcional para sus procesos.

Realice el commit o envío por correo electrónico.

10 puntos

Ahora nos pide



**Requisito Funcional #14875-2021.-** Para darle funcionalidad a su aplicación es necesario segmentar los procesos de captura por tal motivo su aplicación debe contener un menú con dos opciones mínimas (Alta de Información y Mostrar los Datos). Puede utilizar JFrame o Panel, usted decide la forma más funcional para sus procesos.

Realice el commit o envío por correo electrónico.

10 puntos



**Requisito Funcional #14987-2021.-** Incluya un JTable a su proyecto donde se muestre los datos capturados, incluya las opciones Añadir, Eliminar, Actualizar y Exportar.

Realice el commit o envío por correo electrónico.

20 puntos

Title 1	Title 2	Title 3	Title 4

Actualizar

Eliminar

Exportar

```
Actualizar.setText("Actualizar");
Actualizar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ActualizarActionPerformed(evt);
    }
});

Eliminar.setText("Eliminar");
Eliminar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        EliminarActionPerformed(evt);
    }
});

Exportar.setText("Exportar");
Exportar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ExportarActionPerformed(evt);
    }
});
```



**Requisito Funcional #14991-2022.-** Añada la funcionalidad necesaria para que los registros capturados se permitan guardar en un archivo.

Realice el commit o envío por correo electrónico.

20 puntos

```
private void ArchivoActionPerformed(java.awt.event.ActionEvent evt) {GEN-FIRST:event_ArchivoActionPerformed
    JFileChooser fc = new JFileChooser();
    FileNameExtensionFilter filter = new FileNameExtensionFilter("Archivos de texto","txt");
    fc.setFileFilter(filter);
    int seleccion = fc.showOpenDialog(this);
    if(seleccion == JFileChooser.CANCEL_OPTION){
    }else if(seleccion == JFileChooser.APPROVE_OPTION){
        File archivoSeleccionado = fc.getSelectedFile();
        this.Archivo.setText( archivoSeleccionado.getAbsolutePath());
    }
}
```