EXAMEN FINAL

Diplomado Java ||

20 DE JUNIO DE 2022

ISAEL JIMENEZ SANDDOVAL

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Examen Final

Tabla de contenido

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Problema 1 – Uso del Scanner

```
class Test1 {

	public static void main(String[] args) {

		// TODO: Crea una nueva instancia de la clase Scanner llamada
`scanner`

		System.out.print("Dame tu edad: ")

		int edad = // TODO: Lee un entero desde la instancia `scanner`

		// TODO: Termina de leer la línea en la instancia `scanner`

		System.out.printf("Tu edad es: %d\n", edad);

		// TODO: Cierra la instancia `scanner`

	}

}
```

Pregunta 1

¿Qué error hay en el código?

R-> a) Falta un punto y coma en la primera impresión a consola

b) Falta marcar como ‘public’ la clase `Test`

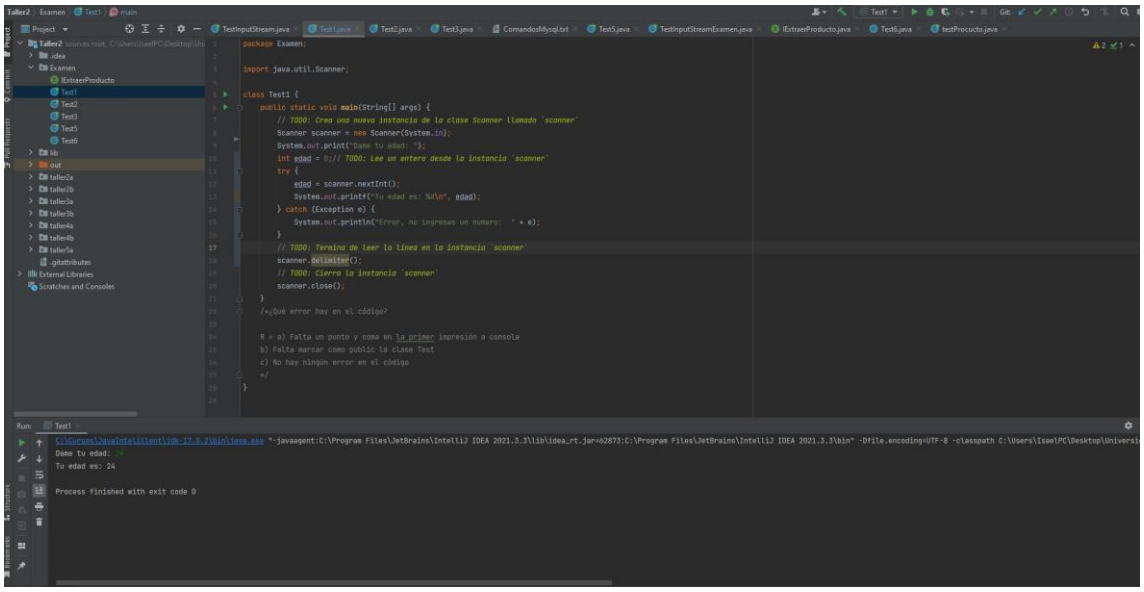
c) No hay ningún error en el código

Solución de los TODOs en el código del problema 1.

```java
package Examen;

import java.util.Scanner;

class Test1 {
    public static void main(String[] args) {
        // TODO: Crea una nueva instancia de la clase Scanner llamada
`scanner`
        Scanner scanner = new Scanner(System.in);
        System.out.print("Dame tu edad: ");
        int edad = 0;// TODO: Lee un entero desde la instancia `scanner`
        try {
            edad = scanner.nextInt();
            System.out.printf("Tu edad es: %d\n", edad);
        } catch (Exception e) {
            System.out.println("Error, no ingresas un numero:  " + e);
        }
        // TODO: Termina de leer la línea en la instancia `scanner`
        scanner.delimiter();
        // TODO: Cierra la instancia `scanner`
        scanner.close();
```

```
    }
    /*¿Qué error hay en el código?

    R = a) Falta un punto y coma en la primer impresión a consola
    b) Falta marcar como public la clase Test
    c) No hay ningún error en el código
    */
}
```

Ejecución del código del problema 1



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Problema 2 - Crear una Ventana con Swing

```
class Test1 {

	public static void main(String[] args) {

		// TODO: Crea una nueva instancia de la clase Scanner llamada
`scanner`

		System.out.print("Dame tu edad: ")

		int edad = // TODO: Lee un entero desde la instancia `scanner`

		// TODO: Termina de leer la línea en la instancia `scanner`

		System.out.printf("Tu edad es: %d\n", edad);

		// TODO: Cierra la instancia `scanner`

	}

}
```

Pregunta 2

¿Qué faltó para que se mostrara la ventana?

a) No faltó nada, la ventana se mostró correctamente

b) Faltó llamar al método show()

R-> c) Faltó llamar al método setVisible(true)

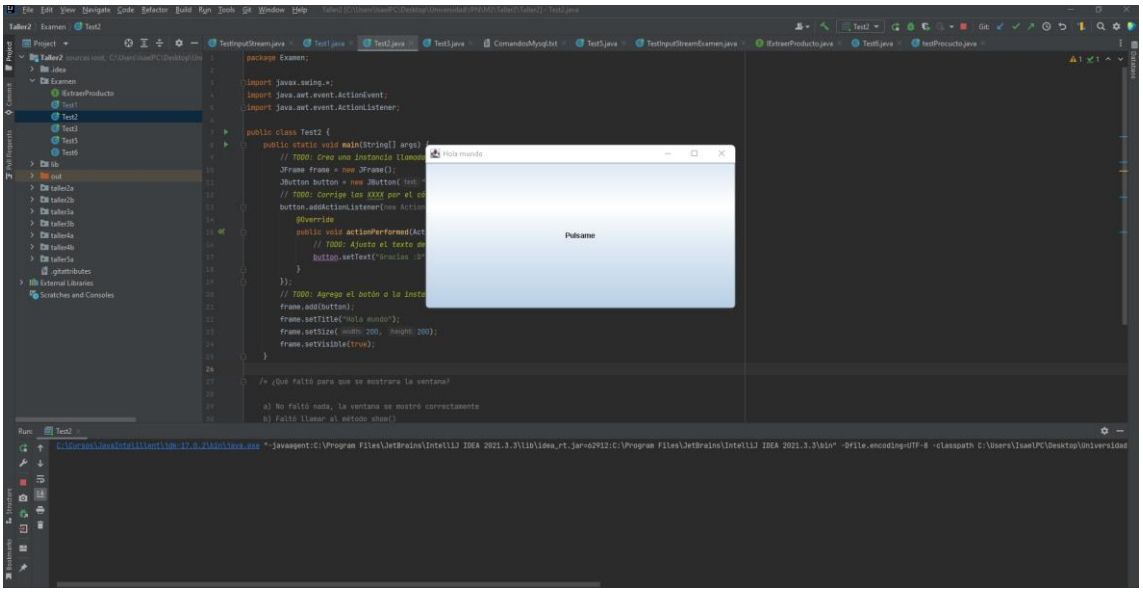
Solución de los TODOs en el código del problema 2.

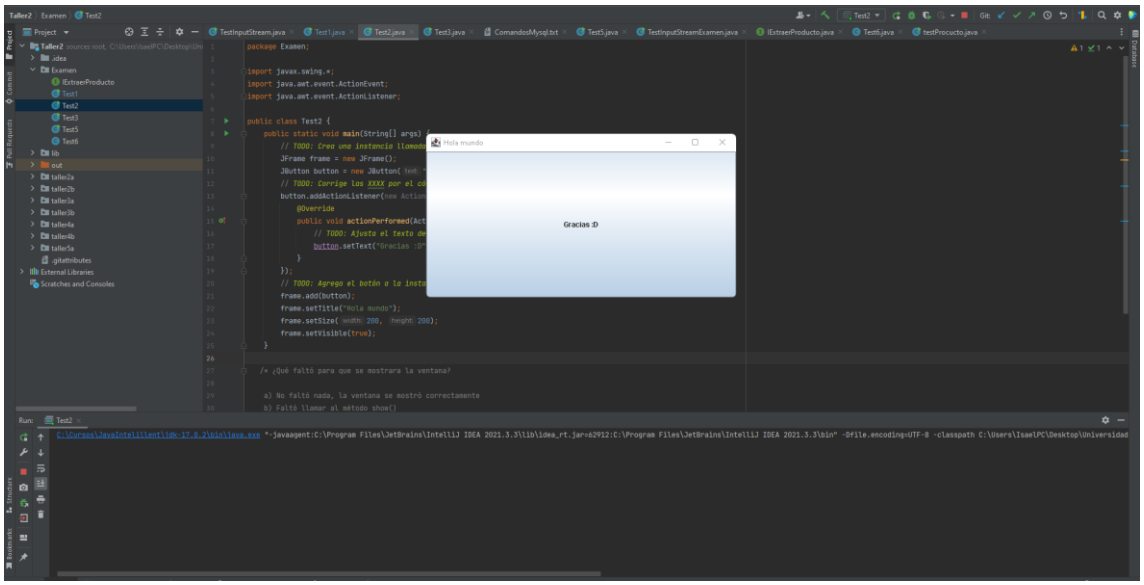
```java
package Examen;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Test2 {
    public static void main(String[] args) {
        // TODO: Crea una instancia llamada `frame` de la clase correcta
        JFrame frame = new JFrame();
        JButton button = new JButton("Pulsame");
        // TODO: Corrige las XXXX por el código correcto
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // TODO: Ajusta el texto del botón a "Gracias :D"
                button.setText("Gracias :D");
            }
        });
        // TODO: Agrega el botón a la instancia `frame`
        frame.add(button);
```

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

```
        frame.setTitle("Hola mundo");
        frame.setSize(200, 200);
        frame.setVisible(true);
    }
}
```

Ejecución del código del problema 2





INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Problema 3 - Conectarse a una Base de Datos MySQL

```
import java.lang.reflect.InvocationTargetException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

// TODO: Corrige las XXXX, puntos y comas, throws e importaciones faltantes

public class Test3 {

    public static void main(String[] args) {

        String driver = // TODO: Coloca la cadena del Driver de MySQL

        Class.forName(driver).getConstructor().newInstance();

        System.out.println("Driver cargado exitosamente");

        String url = "jdbc:mysql://localhost/";

        String dbName = "test";

        String user = "root";

        String password = "password";

        Connection conn = XxxxxxXxxxxxx.xxxXxxxxxxxxx(url + dbName, user,
password);

        System.out.println("Conectado a la BD exitosamente");

        String sql = "SELECT NOW() as Fecha"

        // TODO: Crea una instancia de PreparedStatement llamada `stament`

       // NOTA: Usa la conexión `conn` para crearla y el query `sql`

       XxxxxxXxx resultSet = statement.executeQuery();

       resultSet.next()

       String fecha = resultSet.getString(0);

       System.out.printf("La fecha es: %s\n", fecha)

       conn.close();

        System.out.println("La base de datos ha sido cerrada exitosamente");

    }

}
```

Pregunta 3

¿Cómo corregiste obtener la fecha del resultSet?getString(...)?

a) No tuve que corregirlo, funcionó correctamente

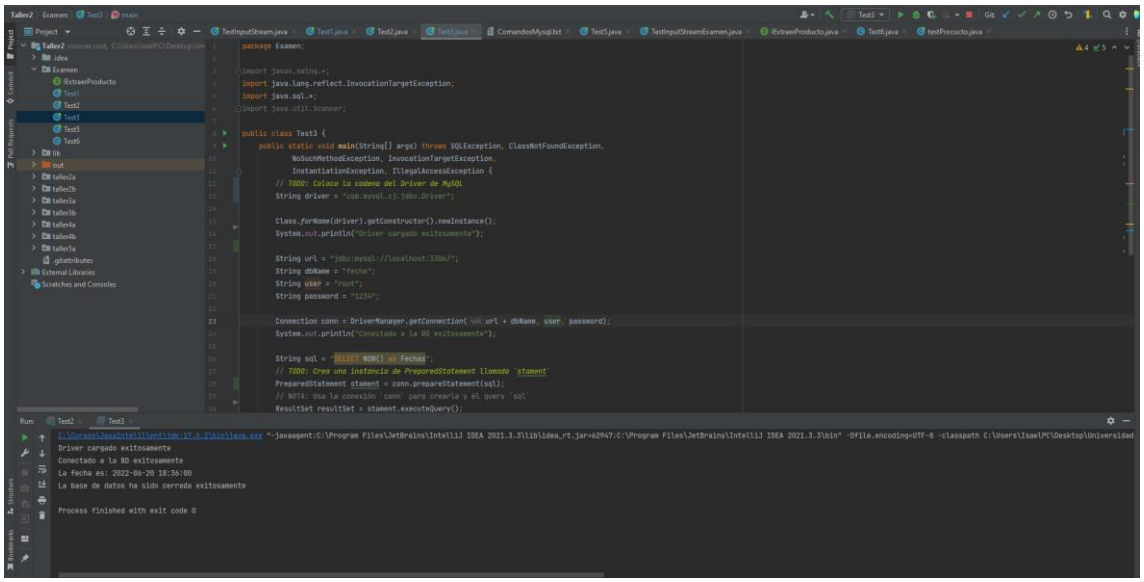
b) Sólo tuve que cambiar el 0 por 1

c) Sólo tuve que cambiar el 0 por "Fecha"

Solución de los TODOs en el código del problema 3.

```
package Examen;

import javax.swing.*;
import java.lang.reflect.InvocationTargetException;
import java.sql.*;
import java.util.Scanner;

public class Test3 {
    public static void main(String[] args) throws SQLException,
ClassNotFoundException,
            NoSuchMethodException, InvocationTargetException,
            InstantiationException, IllegalAccessException {
        // TODO: Coloca la cadena del Driver de MySQL
        String driver = "com.mysql.cj.jdbc.Driver";

        Class.forName(driver).getConstructor().newInstance();
        System.out.println("Driver cargado exitosamente");

        String url = "jdbc:mysql://localhost:3306/";
        String dbName = "fecha";
        String user = "root";
        String password = "1234";

        Connection conn = DriverManager.getConnection(url + dbName, user,
password);
        System.out.println("Conectado a la BD exitosamente");

        String sql = "SELECT NOW() as Fechas";
        // TODO: Crea una instancia de PreparedStatement llamada `stament`
        PreparedStatement stament = conn.prepareStatement(sql);
        // NOTA: Usa la conexión `conn` para crearla y el query `sql`
        ResultSet resultSet = stament.executeQuery();
        resultSet.next();

        String fecha = resultSet.getString(1);
        System.out.printf("La fecha es: %s\n", fecha);

        conn.close();
        System.out.println("La base de datos ha sido cerrada exitosamente");
    }
}
```

Ejecución del código del problema 3



Problema 4 Diseña la Interfaz del Modelo IExtraerProducto

```
// TODO: Define una interfaz llamada IExtraerProducto con los siguientes métodos

{

        int getId()

        String getNombre()

        double getPrecio()

        int getExistencias()

}
```

Pregunta 4

¿Cuál sería la mejor explicación de lo que es una interface?

a) Se utilizan para modelar estereotipos que abstraen la funcionalidad, y esta puede ser implementada posteriormente en clases

b) Se utilizan para definir los métodos que tendrá que implementar una clase para que no haya error en el código

c) Se utilizan para agrupar clases por funcionalidad y poder pasarlas como parámetros o usarlas como objetos similares

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Solución de los TODOs en el código del problema 4.

```java
public interface IExtraerProducto {
    int getId();

    String getNombre();

    double getPrecio();

    int getExistencias();
}
```

Problema 5 - Uso de InputStream y OutputStream

```java
import java.io.*;

public class Test5 {

    public static void main(String[] args) {

        InputStream inputStream = new FileInputStream("<ruta archivo>");

        OutputStream outputStream = new FileOutputStream("<ruta archivo>");

        byte[] bytes = inputStream.readAllBytes();

 // TODO: Guarda los bytes leídos del `inputStream` y alamacenados

// en el arreglo `bytes` en el `outputStream`, pero, invierte los bytes

// IMPORTANTE: Los bytes guardados en el `outputStream`

//tienen que quedar invertidos

// PISTA: Lee al revés los bytes y usa outputStream.write(byte);

        inputStream.close();

        outputStream.close();

    }

}
```

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Pregunta 5

¿Para qué sirven InputStream y OutputStream?

a) Son dos clases que nos permiten leer y escribir bytes en la fuente conectada, la cuál puede ser un archivo, un socket o algún otro.

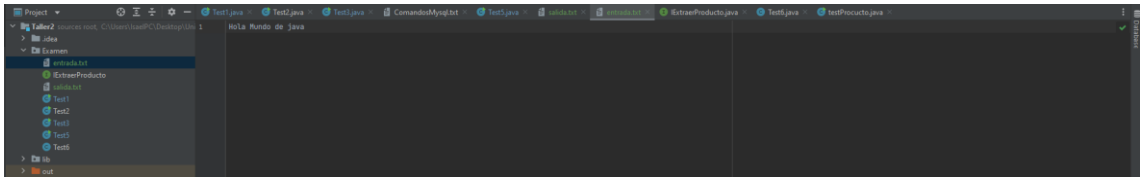
b) Son dos clases que nos permiten leer y escribir bytes de forma temporal sobre archivos, sockets o algunos otros, sin modificar los flujos originales.

c) Son dos clases que nos permiten leer y escribir bytes directamente en la memoria RAM para almacenar variables en forma de bytes

Solución de los TODOs en el código del problema 5.

```
package Examen;

import java.io.*;

public class Test5 {
    public static void main(String[] args) throws IOException {

        InputStream inputStream = new
FileInputStream("C:\\Users\\IsaelPC\\Desktop\\Universidad\\IPN\\M2\\Taller2\\T
aller2\\Examen\\entrada.txt");
        OutputStream outputStream = new
FileOutputStream("C:\\Users\\IsaelPC\\Desktop\\Universidad\\IPN\\M2\\Taller2\\
Taller2\\Examen\\salida.txt");

        byte[] bytes = inputStream.readAllBytes();

        // TODO: Guarda los bytes leídos del `inputStream` y alamacenados
        //       en el arreglo `bytes` en el `outputStream`, pero, invierte
los bytes

        for (int i = bytes.length -1; i >= 0; i--){
            outputStream.write(bytes[i]);
        }
        System.out.println( outputStream.toString());

        // IMPORTANTE: Los bytes guardados en el `outputStream`
        //             tienen que quedar invertidos

        // PISTA: Lee al revés los bytes y usa outputStream.write(byte);

        inputStream.close();
        outputStream.close();

    }
}
```

Ejecución del código del problema 5



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Problema 6 - Diseña la siguiente UI en FXML para JavaFX

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>

<?import java.util.*?>

<?import javafx.scene.*?>

<?import javafx.scene.control.*?>

<?import javafx.scene.layout.*?>


<VBox xmlns="http://javafx.com/javafx"

            xmlns:fx="http://javafx.com/fxml"

            fx:controller="com.example.hello.MyController"

            prefHeight="400.0" prefWidth="600.0" alignment="CENTER">


    <HBox alignment="CENTER">

        <Button text="Pulsame" />

    </HBox>


</VBox>
```

Pregunta 6

¿Qué mejoras tiene *JavaFx* respecto a *Swing*?

a) *JavaFx* separa la *Vista* mediante archivos *FXML* de forma nativa, a diferencia de *Swing* que tiene que definir sus componentes directamente en el código

b) *JavaFx* separa el *Controlador* mediante una clase tradicional de *Java* y utiliza los *JavaBeans* para conectar eventos que surgen desde la interfaz, a diferencia de *Swing*

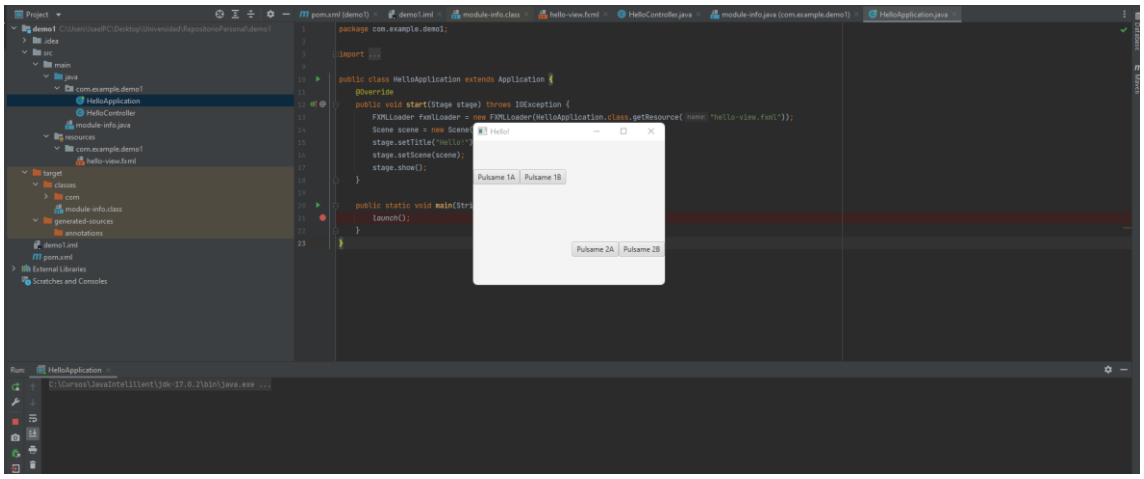
c) No hay diferencias, ambos se pueden interoperar

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CÓMPUTO
DEPARTAMENTO DE DIPLOMADOS Y EXTENSIÓN PROFESIONAL
PROFESOR: Alan Badillo Salas
ALUMNO: Isael Jimenez Sandoval

Solución de los TODOs en el código del problema 6.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.VBox?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.layout.HBox?>
<VBox alignment="CENTER" spacing="20.0" xmlns:fx="http://javafx.com/fxml"
      fx:controller="com.example.demo1.HelloController">

    <HBox alignment="TOP_LEFT" prefHeight="100.0" prefWidth="300.0">
        <Button text="Pulsame 1A"/>
        <Button text="Pulsame 1B"/>
    </HBox>

    <HBox alignment="BOTTOM_RIGHT">
        <Button text="Pulsame 2A"/>
        <Button text="Pulsame 2B"/>
    </HBox>
</VBox>
```

Ejecución del código del problema 6



Enlace de talleres y examen de modulo II en github.com

https://github.com/Jimenez94/Talleres.git