

Programación Orientada a Objetos

Herencia y polimorfismo



Sesión 04

Ing. David Mamani Pari



Misión

“Somos una comunidad universitaria de la Iglesia Adventista del Séptimo Día que **modela** personas a fin de que sean Ingenieros de Sistemas **íntegros**, **misioneros**, e **innovadores** basados en la cosmovisión bíblica-cristiana para servir a Dios y a la humanidad”

Visión



“Ser referentes en el mundo por el **modelamiento** de Ingenieros de Sistemas **íntegros**, **misionero** e **innovadores** con un estilo de vida saludable”

Programa. EP Ingeniería de Sistemas Filial Juliaca Semestre. 2022-2

Leyenda.

76 - 100 % 50 - 75 % 0 - 49 % Sin Asistencia

Mod. resumen Mod. detallado Mod. acta Buscar

Nro.	Código Alumno	Apellidos y Nombres	Sesión. 1	Sesión. 2	Sesión. 3	Sesión. 4	Sesión. 5	Sesión. 6	Sesión. 7	Sesión. 8	Sesión. 9	Sesión. 10
1	201811296	Avalos Humpiri Max Alexander	●	●	●	●	●	●	●	●	●	●
2	202010809	Burgos Pari Diego Armando	●	●	●	●	●	●	●	●	●	●
3	201911332	Condori Cotrado Gabriel Matias	●	●	●	●	●	●	●	●	●	●
4	201911747	Larico Cusi Laydy Maryori	●	●	●	●	●	●	●	●	●	●
5	202012385	Mamani Chura Juan Lenin	●	●	●	●	●	●	●	●	●	●
6	202012387	Morocco Mamani Yohel Aldahyr	●	●	●	●	●	●	●	●	●	●
7	202010800	Paricela Pongo Jack Steve	●	●	●	●	●	●	●	●	●	●
8	202014019	Puma Mestas Brandon Everick	●	●	●	●	●	●	●	●	●	●
9	202012377	Quispe Valero Jhoselyn Noemi	●	●	●	●	●	●	●	●	●	●
10	202013998	Sucasaire Mamani Jhon Erick	●	●	●	●	●	●	●	●	●	●
11	202012384	Ticona Gonzalo Jhobany Elduard	●	●	●	●	●	●	●	●	●	●
12	202012380	Velasquez Rosas Henry Jhonatan	●	●	●	●	●	●	●	●	●	●
13	202012378	Vilca Mamani Jheyson Willy	●	●	●	●	●	●	●	●	●	●

<https://youtu.be/mCE1z0TEQ2A>

Donde almacena
los datos
académico la
UPeU?

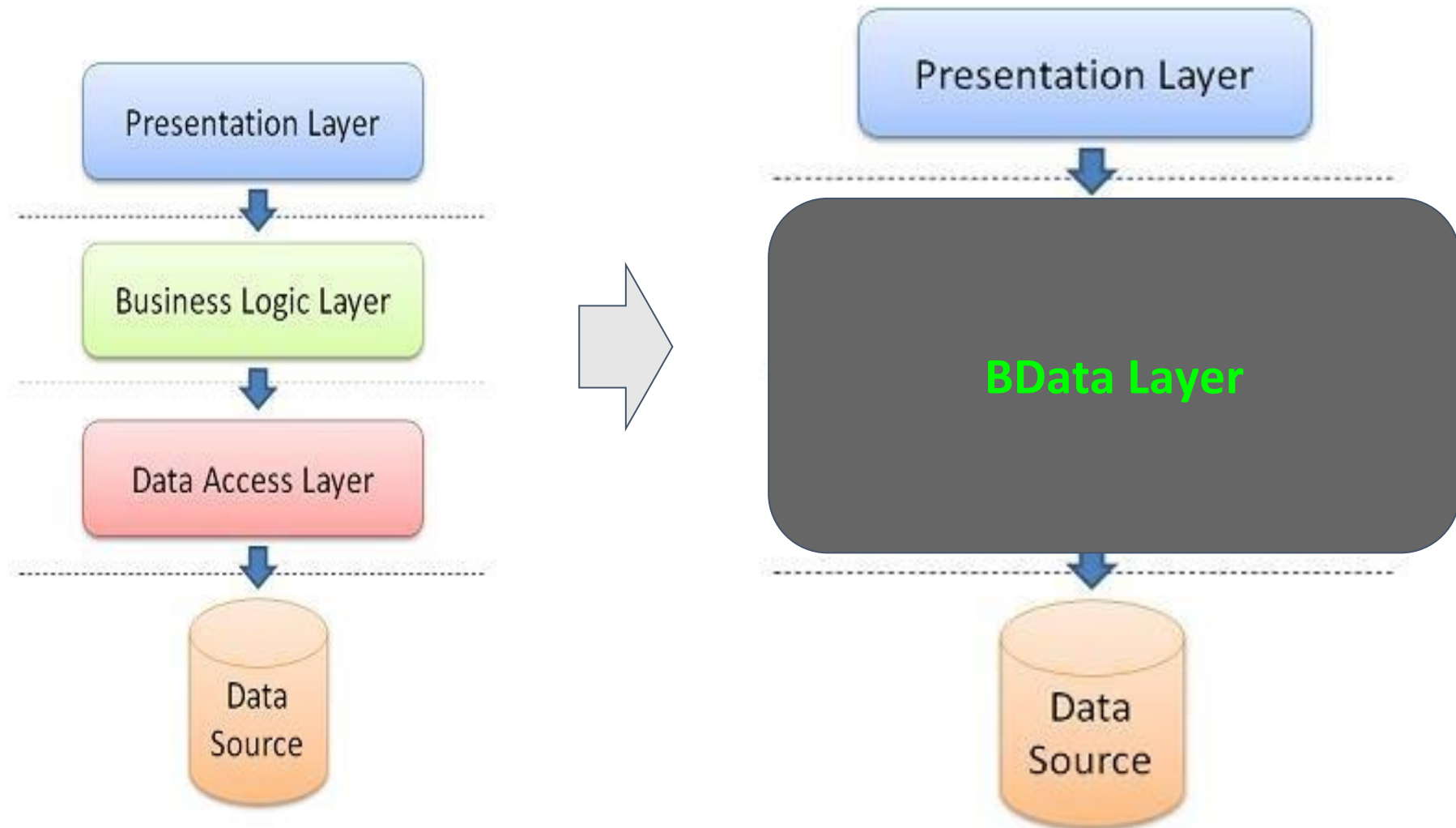
Donde almacena el
Whatsapp mis
conversaciones?



Íntegro

Sé Misionero

Sé Innovador



SystemMain@DMP

Archivo Ayuda

prueba1 Prueba 2 Preueba 3 Preueba 4 Preueba 5

GESTIÓN DE CLIENTES

Dato a Buscar

Exportar

DNI/RUC:

Nombres:

Tipo: ▼

#	Dni/Ruc	Nombres	Tipo
1	43631917	David Mamani Pari	Natural
2	43631918	Elias	General



Project

- ☐ Gradle - Groovy ☐ Gradle - Kotlin
☒ Maven

Language

- ☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

- ☐ 3.3.0 (SNAPSHOT) ☐ 3.3.0 (M3) ☐ 3.2.5 (SNAPSHOT) ☒ 3.2.4
☐ 3.1.11 (SNAPSHOT) ☐ 3.1.10

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 22 ☐ 21 ☒ 17

Dependencies

ADD DEPENDENCIES... CTRL + B

Lombok

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Boot DevTools

DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Spring Data JPA

SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

Spring Configuration Processor

DEVELOPER TOOLS

Generate metadata for developers to offer contextual help and "code completion" when working with custom configuration keys (ex.application.properties/.yml files).



GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

<https://mvnrepository.com/>

```
<dependency>
  <groupId>org.xerial</groupId>
  <artifactId>sqlite-jdbc</artifactId>
</dependency>
<dependency>
  <groupId>org.hibernate.orm</groupId>
  <artifactId>hibernate-community-dialects</artifactId>
  <version>6.4.4.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/jakarta.persistence/jakarta.persistence-api -->
<dependency>
  <groupId>jakarta.persistence</groupId>
  <artifactId>jakarta.persistence-api</artifactId>
  <version>3.1.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/jakarta.validation/jakarta.validation-api -->
<dependency>
  <groupId>jakarta.validation</groupId>
  <artifactId>jakarta.validation-api</artifactId>
  <version>3.0.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations -->
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-annotations</artifactId>
  <version>2.17.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.miglayout/miglayout -->
<dependency>
  <groupId>com.miglayout</groupId>
  <artifactId>miglayout</artifactId>
  <version>3.7.4</version>
</dependency>
```

Agregar Plugin

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <version>2.4</version>
  <configuration>
    <archive>
      <manifest>
        <addClasspath>true</addClasspath>
        <classpathPrefix>lib/</classpathPrefix>
      </manifest>
    </archive>
  </configuration>
</plugin>

<mainClass>pe.edu.upeu.syscenterlife.SysAlmacenVcApplication</mainClass>
```

application.properties

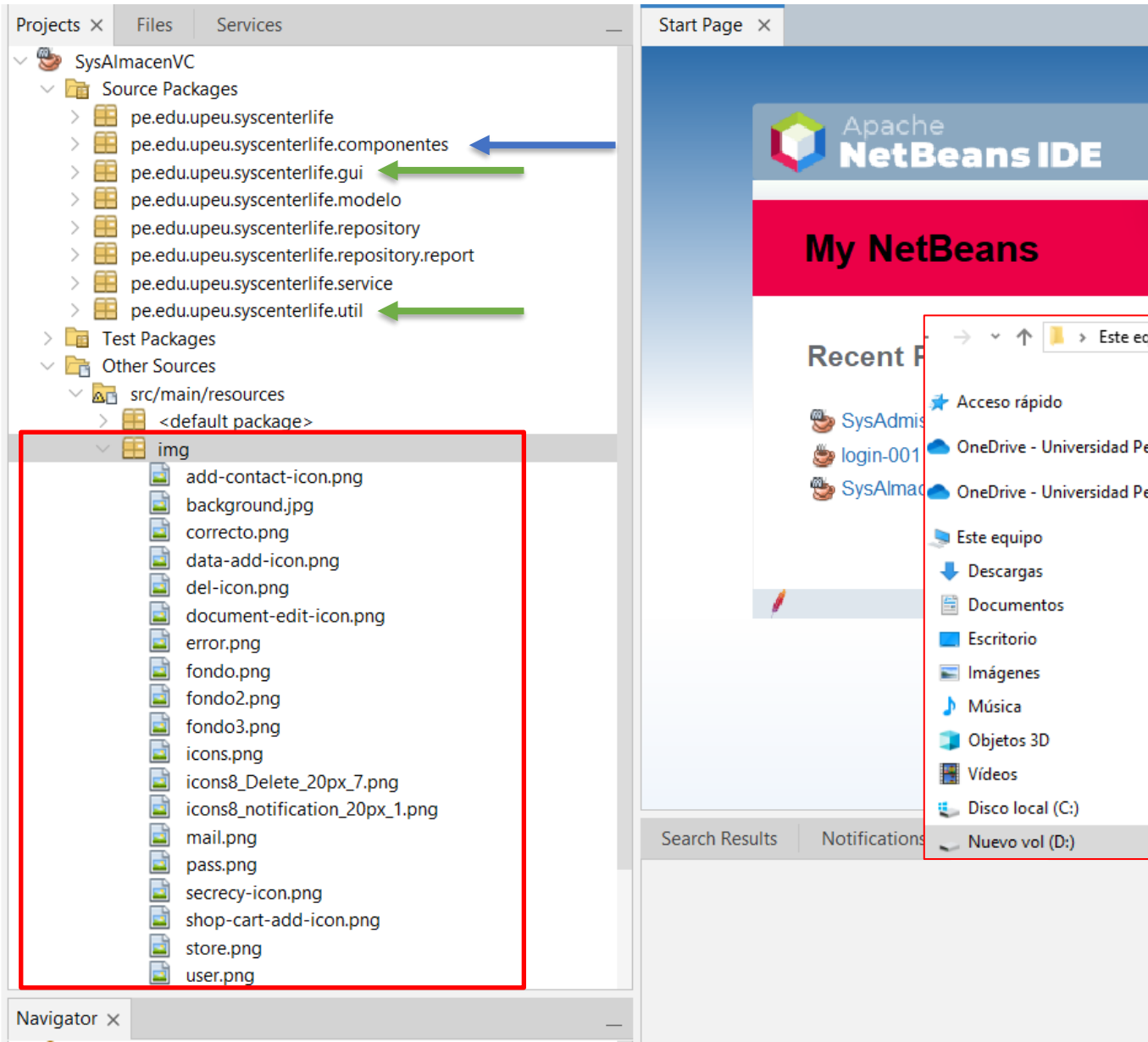
```
#SQLite
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-platform=org.hibernate.community.dialect.SQLiteDialect
spring.datasource.url=jdbc:sqlite:data/db_ventas.db
spring.datasource.driver-class-name=org.sqlite.JDBC
spring.jpa.show-sql=true
```


Repositorio

```
<repositories>
  <repository>
    <id>jitpack.io</id>
    <url>https://jitpack.io</url>
  </repository>
</repositories>
```

Dependencia

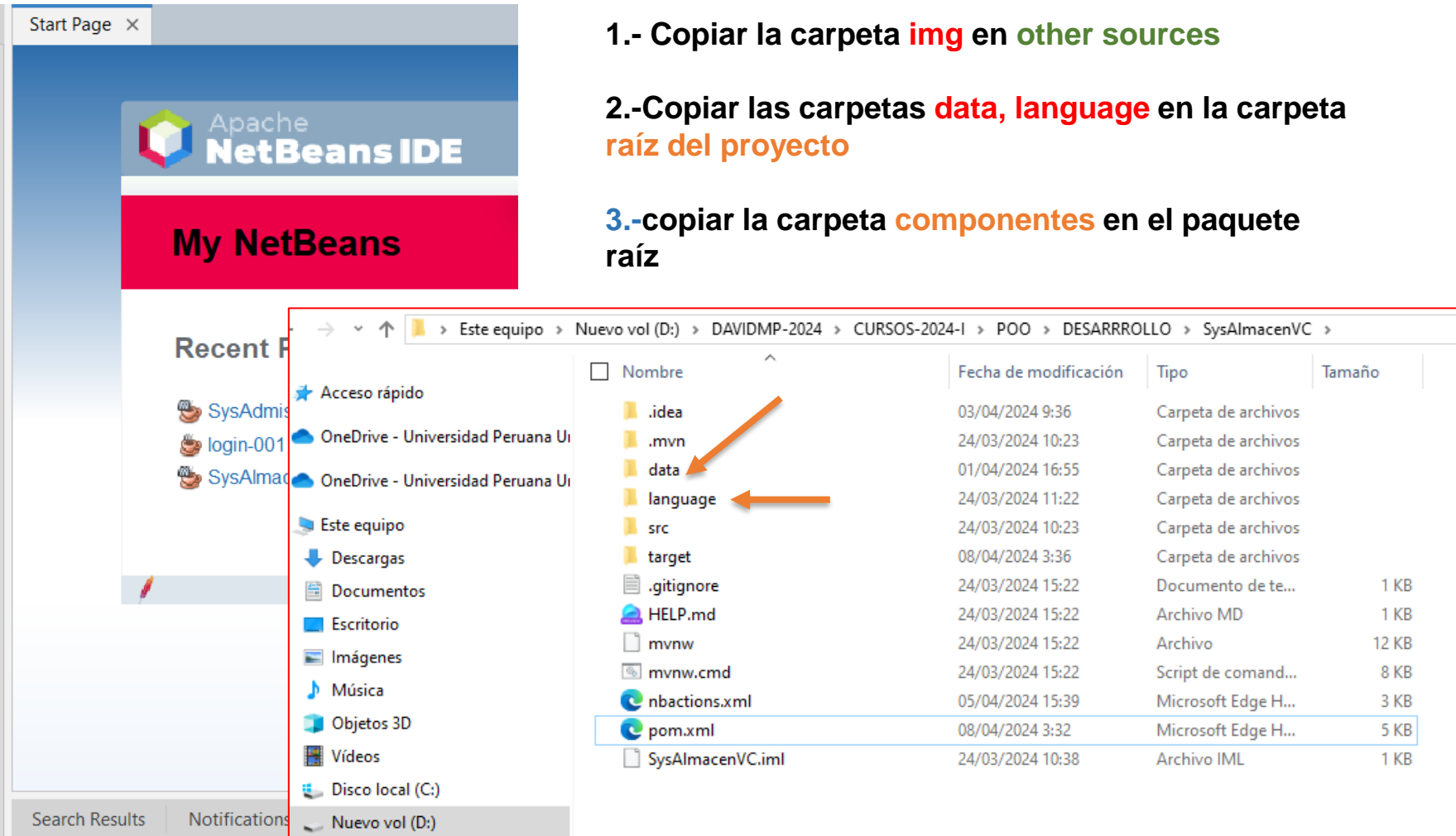
```
<dependency>
  <groupId>com.github.davidmp</groupId>
  <artifactId>SysSwingDmp</artifactId>
  <version>v1.9</version>
</dependency>
```

1.- Copiar la carpeta **img** en **other sources**

2.- Copiar las carpetas **data**, **language** en la carpeta **raíz del proyecto**

3.- copiar la carpeta **componentes** en el paquete **raíz**



Crear Clase **UtilsX**

```
public class UtilsX {  
    public URL getFile(String ruta) {  
        return this.getClass().getResource("/") + ruta);  
    }  
    public File getFileExterno(String carpeta, String filex) {  
        File newFolder = new File(carpeta);  
        String ruta = newFolder.getAbsolutePath();  
        Path CAMINO = Paths.get(ruta + "/" + filex);  
        return CAMINO.toFile();  
    }  
  
    public File getFolderExterno(String carpeta) {  
        File newFolder = new File(carpeta);  
        String ruta = newFolder.getAbsolutePath();  
        Path CAMINO = Paths.get(ruta + "/");  
        return CAMINO.toFile();  
    }  
}
```

Agregar **métodos** en la Clase **UtilsX**

```
public Properties detectLanguage(String idioma) {  
    Properties myresourcesx = new Properties();  
    try {  
        FileInputStream in = new FileInputStream(  
            getFileExterno("language", "idiomas-" + idioma + ".properties").getAbsolutePath());  
        try {  
            myresourcesx.load(in);  
        } catch (IOException e) {e.printStackTrace();}  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
    return myresourcesx;  
}  
  
public String readLanguageFile() {  
    Properties myresourcesx = new Properties();  
    String idioma = "";  
    try {  
        FileInputStream in = new FileInputStream(  
            getFileExterno("language", "SysCenterLife.properties").getAbsolutePath());  
        try {  
            myresourcesx.load(in);  
            idioma = myresourcesx.getProperty("syscenterlife.idioma");  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    } catch (FileNotFoundException e) {  
        e.printStackTrace();  
    }  
    return idioma;  
}
```

```
import javax.swing.*;

/**
 * Clase Principal para manejo de alertas
 *
 * @see <br>
 * El constructor resive 3 parametros (mensaje, tipo_de_alerta,
 * icono_personalizado)
 * <h1>El parámetro tipoDialog tiene las siguientes opciones: JOptionPane</h1>
 *
 * <ol><li>WARNING_MESSAGE</li><li>ERROR_MESSAGE</li><li>PLAIN_MESSAGE</li><li>INFORMATION_MESSAGE</li>
 * <li>QUESTION_MESSAGE</li></ol>
 *
 * @see <a href = "https://github.com/davidmp" />Aqui Github</a>
 * @return No hay retorno de metodos
 */
public class MsgBox {
    UtilsX obj = new UtilsX();
    JPanel p;
    ImageIcon icon = null;
    public MsgBox() {}
    public MsgBox(String msg, int tipoDialog, String iconPropio) {
        if (!iconPropio.trim().equals("")) {
            icon = new ImageIcon(obj.getFile(iconPropio));
            JOptionPane.showMessageDialog(p, msg, "Holas", tipoDialog, icon);
        } else {
            JOptionPane.showMessageDialog(p, msg, "Holas", tipoDialog);
        }
    }

    public static void main(String[] args) {
        MsgBox msgBox = new MsgBox("Mensaje", JOptionPane.ERROR_MESSAGE, "img/secrecy-icon.png");
    }
}
```

Crear la clase **MsgBox**

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.*;

public class ErrorLogger {

    protected static final Logger logger = Logger.getLogger("MYLOG");
    private static String classs = "";

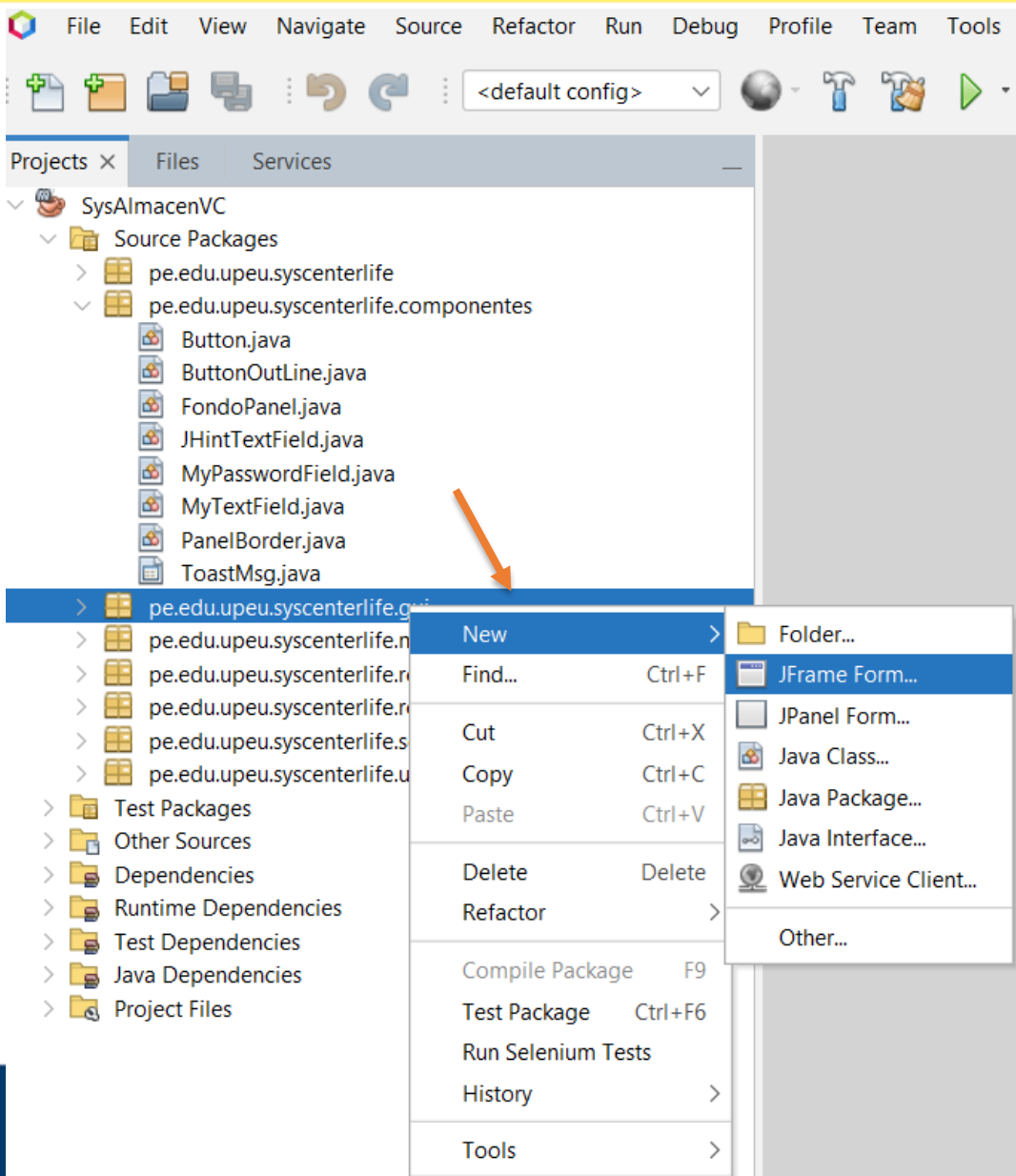
    public ErrorLogger(String classs) {
        this.classs = classs;
        configure();
    }

    private static void configure() {
        try {
            String logsDirectoryFolder = "logs";
            Files.createDirectories(Paths.get(logsDirectoryFolder));
            FileHandler fileHandler = new FileHandler(logsDirectoryFolder + File.separator + getCurrentTimeString() + ".log", true);
            logger.addHandler(fileHandler);
            SimpleFormatter formatter = new SimpleFormatter();
            fileHandler.setFormatter(formatter);
        } catch (IOException exception) {
            // exception.printStackTrace();
            System.err.println("configure:" + exception.toString());
        }
        addCloseHandlersShutdownHook();
    }

    private static void addCloseHandlersShutdownHook() {
        Runtime.getRuntime().addShutdownHook(new Thread(()
            -> {
                for (Handler handler : logger.getHandlers()) {
                    handler.close();
                }
            }
        ));
    }

    private static String getCurrentTimeString() {
        DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
        return dateFormat.format(new Date());
    }

    public static void log(Level level, String me, Exception exception) {
        if (classs.equals("")) {
            configure();
            logger.log(level, me, exception);
        } else {
            logger.log(level, classs + "." + me, exception);
        }
    }
}
```



1.- Crear GUI Login

2.- Agregar el siguiente código en la clase de Login

```
JPanel panelGeneral;
FondoPanel fondoPanel = new FondoPanel();
PanelBorder panelBorder = new PanelBorder();
MyPasswordField txtPassword;
MyTextField txtUsername = new MyTextField();
Button loginButton;

UtilsX obj = new UtilsX();
```

3.- Agregar anotación antes del nombre de la clase de Login

```
@Component
```

Agregar Código en Login

```
public void addEventListeners() {
    loginButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (txtUsername.getText().equals("admin")
                && String.valueOf(txtPassword.getPassword()).equals("admin")) {
                new GUIMain().setVisible(true);
                dispose();
            } else {
                new MsgBox("Error al ingresar!", NORMAL, "");
            }
        }
    });
    txtUsername.addFocusListener(new FocusListener() {
        @Override
        public void focusLost(FocusEvent e) {
            if (txtUsername.getText().equals("")) {
                txtUsername.setText("Coloca tu usuario");
                txtUsername.setForeground(Color.gray);
            }
        }
        @Override
        public void focusGained(FocusEvent e) {
            if (txtUsername.getText().equals("Coloca tu usuario")) {
                txtUsername.setText("");
                txtUsername.setForeground(Color.black);
            }
        }
    });
    txtPassword.addFocusListener(new FocusListener() {
        @Override
        public void focusLost(FocusEvent e) {
            if (txtPassword.getText().equals("")) {
                txtPassword.setText("Coloca tu clave");
                txtPassword.setForeground(Color.gray);
                txtPassword.setEchoChar((char) 0);
            }
        }
        public void focusGained(FocusEvent e) {
            if (txtPassword.getText().equals("Coloca tu clave")) {
                txtPassword.setText("");
                txtPassword.setEchoChar('*');
                txtPassword.setForeground(Color.black);
            }
        }
    });
}
```

```
public void initCustom() {
    panelBorder.setPreferredSize(new Dimension(250, 360));
    txtUsername.setPrefixIcon(new ImageIcon(obj.getFile("img/user.png")));
    txtUsername.setHint("User");
    txtPassword.setPrefixIcon(new ImageIcon(obj.getFile("img/pass.png")));
    txtPassword.setHint("Pass");
    loginButton.setBackground(new Color(66, 245, 114));
    loginButton.setFocusPainted(false);
    txtUsername.setText("Coloca tu usuario");
    txtUsername.setForeground(Color.gray);
    txtPassword.setText("Coloca tu clave");
    txtPassword.setForeground(Color.gray);
    txtPassword.setEchoChar((char) 0);

    panelBorder.setLayout(new MigLayout("wrap", "push[center]push", "push[]25[]10[]10[]25[]push"));
    JLabel label = new JLabel("SysCenterlife");
    label.setFont(new Font("sansserif", 1, 30));
    label.setForeground(new Color(7, 164, 121));

    panelBorder.add(label);
    panelBorder.add(txtUsername, "w 80%");
    panelBorder.add(txtPassword, "w 80%");
    panelBorder.add(loginButton, "w 60%");

    fondoPanel.setPreferredSize(new Dimension(250, 360));
    panelGeneral.add(fondoPanel);
    panelGeneral.add(panelBorder);
    addEventListeners();
}
```

1.- Agregar este fragmento de código en el constructor de la clase **Login**

```
this.setTitle("Formulario de Ingreso-SysCenterlife");
this.setIconImage(new ImageIcon(obj.getFile("img/store.png")).getImage());
try {
    panelGeneral = new JPanel() {
        BufferedImage bufferedImage = ImageIO.read(obj.getFile("img/background.jpg"));
        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            g.drawImage(bufferedImage, 0, 0, this);
        }
    };
} catch (Exception e) {}

txtPassword = new MyPasswordField();
loginButton = new Button();
loginButton.setFont(new Font("sansserif", 1, 20));
loginButton.setForeground(new Color(7, 164, 121));
loginButton.setEffectColor(Color.yellow);
loginButton.setText("Ingresar");
initCustom();
this.setContentPane(panelGeneral);
this.setResizable(false);
this.setDefaultCloseOperation(this.EXIT_ON_CLOSE);
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
this.setSize(new Dimension(screenSize.width / 2, (screenSize.height - 36) / 2));
this.setLocationRelativeTo(null);
```



```
import lombok.Data;
@Data
public class MenuMenuItenTO {
    public String menunombre, menuitemnombre, nombreObj;

    public MenuMenuItenTO(String menunombre, String menuitemnombre, String nombreObj) {
        this.menunombre = menunombre;
        this.menuitemnombre = menuitemnombre;
        this.nombreObj = nombreObj;
    }
    public MenuMenuItenTO() {
    }
}
```

```
import pe.edu.upeu.syscenterlife.modelo.MenuMenuitemTO;
import java.util.List;
import java.util.Properties;

public interface MenuMenuitemDao {

    public List<MenuMenuitemTO> listaAccesos(String perfil, Properties
idioma);
}
```

```
@Service
public class MenuMenuitemDao implements MenuMenuitemDao {
    @Override
    public List<MenuMenuitemTO> listaAccesos(String perfil, Properties idioma) {
        List<MenuMenuitemTO> lista = new ArrayList<>();
        lista.add(new MenuMenuitemTO(idioma.getProperty("menu.nombre.archivo"), "", "mifile"));
        lista.add(new MenuMenuitemTO("Edit", "cut", "micut"));
        lista.add(new MenuMenuitemTO("Edit", "copy", "micopy"));
        lista.add(new MenuMenuitemTO("Edit", "paste", "mipaste"));
        lista.add(new MenuMenuitemTO("Edit",
            idioma.getProperty("menuitem.nombre.postulante"), "miselectall"));
        lista.add(new MenuMenuitemTO("Help", "Ver1", "miver1"));
        lista.add(new MenuMenuitemTO("Help", "Ver2", "miver2"));
        lista.add(new MenuMenuitemTO("Principal", "A.Periodo Result", "aperidoresult"));
        lista.add(new MenuMenuitemTO("Principal", "Area Periodo", "areaperiodo"));

        List<MenuMenuitemTO> accesoReal = new ArrayList<>();
        switch (perfil) {
            case "Administrador":
                accesoReal.add(lista.get(0));
                accesoReal.add(lista.get(1));
                accesoReal.add(lista.get(2));
                accesoReal.add(lista.get(3));
                accesoReal.add(lista.get(4));
                accesoReal.add(lista.get(5));
                accesoReal.add(lista.get(6));
                break;
            case "Root":
                accesoReal = lista;
                break;
            case "Reporte":
                accesoReal.add(lista.get(0));
                accesoReal.add(lista.get(5));
                accesoReal.add(lista.get(6));
                break;
            default:
                throw new AssertionError();
        }
        return accesoReal;
    }
}
```

```
public class GUIMain extends JFrame {
    Preferences userPrefs = Preferences.userRoot();
    JMenuBar menuBar;
    JTabbedPane jtpane;
    JScrollPane scrollPane;
    JMenu[] menu;
    JMenuItem[] jMenuItem;
    UtilsX util = new UtilsX();
    Properties myresources = new Properties();
    List<MenuMenuItemTO> lista;
    MenuMenuItemDao mmiDao;

    public GUIMain() {
    }
}
```

Agregar dentro de la Clase **GUIMain**

```
class SampleMenuListener implements MenuListener {
    @Override
    public void menuSelected(MenuEvent e) {
        System.out.println("menuSelected");
        if (((JMenu) e.getSource()).getName().equals("mmiver1")) {
            System.out.println("llego help");
        }
    }
    @Override
    public void menuDeselected(MenuEvent e) {
    }
    @Override
    public void menuCanceled(MenuEvent e) {
    }
}
```

Agregar dentro de la Clase **GUIMain**

```
class MenuItemListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        System.out.println("pasa por aqui");
        Container contai = GUIMain.this.getContentPane();
        if(((JMenuItem)e.getSource()).getName()
            .equals("miaperidoresult")){
            System.out.println("Holas si llega");
            /*jtpane.removeAll();
            MainAreaPeriodoResult map=new MainAreaPeriodoResult();
            map.setPreferredSize(new Dimension(1024, 600));
            scrollPane=new JScrollPane(map);
            scrollPane.setHorizontalScrollBarPolicy(
                JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
            scrollPane.setVerticalScrollBarPolicy(
                JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
            jtpane.add(scrollPane, "A.P.Resultado");
            contai.add(BorderLayout.CENTER, jtpane);
            contai.validate();
            contai.repaint();*/
        }
        if(((JMenuItem)e.getSource()).getName().equals("miareaperiodo")){
            System.out.println("Si llega!");
        }
        if (((JMenuItem) e.getSource()).getName().equals("mimiselectall")) {
            jtpane.removeAll(); //remueve todo el contenido
        }
    }
}
```

Agregar dentro de la clase **GUIMain**

```
public int[] contarMenuMunultem(List<MenuMunultemTO> data) {
    int menui = 0, menuitem = 0;
    String menuN = "";
    for (MenuMunultemTO mmi : data) {
        if (!mmi.menunombre.equals(menuN)) {
            menuN = mmi.menunombre;
            menui++;
        }
        if (!mmi.menuitemnombre.equals("")) {
            menuitem++;
        }
    }
    return new int[]{menui, menuitem};
}
```

Agregar dentro del constructor de **GUIMain**

```
myresources = util.detectLanguage(userPrefs.get("IDIOMAX", "es"));
mmdao = new MenuMunultemDao();
lista = mmdao.listaAccesos("Root", myresources);

int[] mmi = contarMenuMunultem(lista);
menu = new JMenu[mmi[0]];
jmltem = new JMenuItem[mmi[1]];
menuBar = new JMenuBar();
MenuMunultemListener d = new MenuMunultemListener();
SampleMenuListener m = new SampleMenuListener();
this.setTitle("SystemMain@DMP");
this.setIconImage(new ImageIcon(util.getFile("img/store.png")).getImage());
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
this.setSize(new Dimension(screenSize.width, (screenSize.height
- 36)));

String menuN = "";
int menui = 0, menuitem = 0;
char conti = 'N';
for (MenuMunultemTO mmix : lista) {
    if (!mmix.menunombre.equals(menuN)) {
        menu[menui] = new JMenu(mmix.menunombre);
        menu[menui].setName("m" + mmix.nombreObj);
        menu[menui].addMenuListener(m);
        if (!mmix.menuitemnombre.equals("")) {
            jmltem[menuitem] = new JMenuItem(mmix.menuitemnombre);
            jmltem[menuitem].setName("mi" + mmix.nombreObj);
            jmltem[menuitem].addActionListener(d);
            menu[menui].add(jmltem[menuitem]);
            menuitem++;
        }
        menuBar.add(menu[menui]);
        menuN = mmix.menunombre;
        conti = 'N';
        menui++;
    } else {
        conti = 'S';
    }
}
System.out.println("MI:" + menuitem);
if (!mmix.menuitemnombre.equals(""))
    && mmix.menunombre.equals(menuN) && conti == 'S' {
    jmltem[menuitem] = new JMenuItem(mmix.menuitemnombre);
    jmltem[menuitem].setName("mi" + mmix.nombreObj);
    jmltem[menuitem].addActionListener(d);
    menu[menui - 1].add(jmltem[menuitem]);
    menuitem++;
}
}

jtpane = new JTabbedPane();
this.getContentPane().add(BorderLayout.NORTH, menuBar);
this.add(BorderLayout.CENTER, jtpane);
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;
import pe.edu.upeu.syscenterlife.util.UtilsX;
import java.awt.*;
import java.util.prefs.Preferences;
import pe.edu.upeu.syscenterlife.gui.Login;

@Component
public class SpringBootSwingCommandLineRunner implements CommandLineRunner {
    @Autowired
    Login mainGUI;
    static Preferences userPrefs = Preferences.userRoot();
    @Override
    public void run(String... args) throws Exception {
        userPrefs.put("IDIOMAX", new UtilsX().readLanguageFile());
        userPrefs.put("PERFIL", "Admin");
        EventQueue.invokeLater(()->{
            mainGUI.setVisible(true);
        });
    }
}
```

```
import org.springframework.boot.WebApplicationType;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.builder.SpringApplicationBuilder;

@SpringBootApplication
public class SysAlmacenVcApplication {

    public static void main(String[] args) {
        //SpringApplication.run(SysAlmacenVcApplication.class, args);
        new SpringApplicationBuilder(SysAlmacenVcApplication.class)
            .headless(false)
            .web(WebApplicationType.NONE)
            .run(args);
    }
}
```

Clona:

git clone URL

Ver estado:

git status

Subir cambios al Repositorio Local:

git add -A .

git commit -a -m "mensaje de cambio"

Subir al Repositorio Remoto:

git pull

git push

Actualizar cambios del repositorio remoto a repositorio local:

git pull

#	COMANDOS	DESCRIPCIÓN
Comandos Básicos		
1	git checkout -b "name branch"	// -b permite crear una nueva rama y ubicarse en esa rama
2	git checkout "name branch"	//moverse a otra rama
3	git status	//ver informacion
4	git add -A .	//adicionar los cambios
5	git branch "nombre rama"	//crear nueva rama
6	git branch	//ver en que ramas estamos
7	git push --set-upstream origin “nombre rama”	// Subir a repositorio remoto
8	git branch -a	//Ver todas las ramas
9	git fetch origin <nombre-branch>	//Traer una determinada rama
Comandos para integrar a una rama		
1	git checkout "name branch"	//moverse a otra rama
2	git merge "nombre de la rama a integrar"	//permite integrar la rama especificada con la rama donde se está actualmente
Comandos para eliminar ramas		
1	git branch -d "nombre rama a eliminar"	//Eliminar rama localmente
2	git push origin --delete "nombre rama a eliminar"	//Eliminar rama remotamente
3	git fetch -p	//Es el comando que hace que tu repositorio Git local se actualice con la última información que hay en el repositorio remoto
Clonar una rama específica		
1	git clone --branch <branchname> <remote-repo-url>	//Permite clonar una rama específica
2	git clone -b <branchname> <remote-repo-url>	
Solución de Problemas al integrar ramas		
1	git log --oneline --decorate --all --graph	Para ver solución de conflictos de forma manual una vez unido con rama master es decir rama por rama se debe solucionar

1.- Subir su trabajo de investigación a su github



MUCHAS GRACIAS

Sé Íntegro, Sé Misionero, Sé Innovador

www.upeu.edu.pe