

Interacción Persona Ordenador

Práctica 5

INTERNACIONALIZACIÓN

A continuación describiré y aportaré capturas de pantalla del proceso llevado a cabo para tener una aplicación de escritorio java swing con internacionalización.

Para la implementación he utilizado el método explicado en clase de la matriz. Este método consiste en tener una matriz en memoria en la que cada una de sus filas pertenezca a un idioma distinto y por tanto se aplique de manera dinámica el texto o imágenes que se deben poner en la aplicación.

En mi caso he decidido crear una clase que maneje estos datos "Datos.java"

```
public class Datos {  
    //array marcas moto  
    private ArrayList<String> marcas = new ArrayList<String>(); //Array de marcas de moto  
    private ArrayList<String> modelos = new ArrayList<String>(); //Array de modelos de moto  
    private ArrayList<String> fechas = new ArrayList<String>(); //Array de fechas de reserva  
    private ArrayList<ArrayList<String>> reservas = new ArrayList<ArrayList<String>>(); //Matriz de reservas (Modelo x Fechas)  
    private ArrayList<ArrayList<String>> idiomas = new ArrayList<ArrayList<String>>(); //Matriz de internacionalización (Idioma x Frases)
```

Para cargar los datos se puede realizar de distintas maneras, en mi caso para evitar el manejo de archivos externos he inicializado todo en el constructor de la clase de datos, de tal manera que cuando se crea la ventana principal de la aplicación se inicia una instancia de esta clase para poder aportar todos los datos necesarios a nuestros botones, labels, etc...

A continuación voy a poner un pequeño trozo de la matriz de internacionalización donde se crea una fila para cada uno de los idiomas y se ve como en esa fila se insertan todas las líneas de texto necesarias para la aplicación. A veces es necesario no solo introducir en esta matriz el texto si no también iconos o imágenes ya que como sabemos dependiendo de las distintas culturas o países puede ser necesario cambiarlo, aunque en mi caso no he

introducido ni iconos ni imágenes puesto que son neutros, es decir no importa en qué parte del mundo estés que van a ser reconocidos y entendidos.

```
idiomas.add(new ArrayList<String>());
idiomas.get(0).add("Español");

idiomas.get(0).add("RESERVAR");
idiomas.get(0).add("Complete el formulario para reservar: ");
idiomas.get(0).add("Nombre Completo:");
idiomas.get(0).add("Correo Electronico:");
idiomas.get(0).add("Telefono:");
idiomas.get(0).add("Fecha:");
idiomas.get(0).add("Contratos:");
idiomas.get(0).add("Acepto recibir información adicional via email");
idiomas.get(0).add("Acepto recibir la confirmación de mi reserva por mensaje en el movil");

//ingles
idiomas.add(new ArrayList<String>());
idiomas.get(1).add("Ingles");

idiomas.get(1).add("RESERVATION");
idiomas.get(1).add("Complete the form to make a reservation: ");
idiomas.get(1).add("Full name:");
idiomas.get(1).add("Email:");
idiomas.get(1).add("Telephone:");
idiomas.get(1).add("Date:");
idiomas.get(1).add("Contracts:");
idiomas.get(1).add("I agree to receive additional information via email");
idiomas.get(1).add("I agree to receive confirmation of my reservation by mobile phone messa
```

Son filas gemelas con la traducción en cada idioma, es decir, en la fila 0(español) posición 1 nos encontramos la palabra “RESERVAR” y en la fila 1(inglés) posición 1 nos encontramos la misma palabra pero en inglés “RESERVATION”, de esta forma veremos lo sencillo que es cambiar de un idioma a otro en ejecución.

```
public PaginaPrincipal() {
    initComponents();
    cargarDatos();
}

private void cargarDatos(){
    this.datos = new Datos();
}
```

Como vemos en la creación de la página principal se llama la función cargarDatos(), esta será la encargada de asignar a cada componente (jcombobox, jlabel, jbutton, etc...) el texto correspondiente a cada uno.

```

this.titulo1.setText(datos.getIdiomas().get(idiomaActual).get(10));
this.descripcion1.setText(datos.getIdiomas().get(idiomaActual).get(11));

this.titulo2.setText(datos.getIdiomas().get(idiomaActual).get(13));
this.descripcion2.setText(datos.getIdiomas().get(idiomaActual).get(14));

this.titulo3.setText(datos.getIdiomas().get(idiomaActual).get(16));
this.descripcion3.setText(datos.getIdiomas().get(idiomaActual).get(17));

this.titulo4.setText(datos.getIdiomas().get(idiomaActual).get(19));
this.descripcion4.setText(datos.getIdiomas().get(idiomaActual).get(20));

```

Como podemos observar en esta imagen a cada componente se le asigna el texto según la matriz de idiomas que se encuentra en la clase de datos explicada anteriormente.

Si queremos modificar el idioma mientras estamos usando la aplicación deberemos modificar el siguiente jcombobox



De tal manera que al seleccionar otro idioma se llama a una función listener que estará a la espera de esto para desencadenar una nueva asignación de texto a cada componente.

```

private void comboIdiomaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    // TODO add your handling code here:
    String aux = (String) this.comboIdioma.getSelectedItem();
    if (aux.equals("Español")) {
        this.idiomaActual = 0;
    } else if (aux.equals("Inglés")) {
        this.idiomaActual = 1;
    }
}

```

Esta primera parte de la función se encargará de seleccionar un nuevo idioma según el seleccionado y la siguiente parte se encargará de modificar todos los componentes asignando su nuevo texto al igual que antes.

```
this.descripcion1.setText(datos.getIdiomas().get(idiomaActual).get(11));  
this.descripcion2.setText(datos.getIdiomas().get(idiomaActual).get(14));  
this.descripcion3.setText(datos.getIdiomas().get(idiomaActual).get(17));  
this.descripcion4.setText(datos.getIdiomas().get(idiomaActual).get(20));
```

Como podemos ver idiomaActual se encarga de seleccionar la fila según el idioma y la sentencia(el segundo get) siempre es la misma para su componente.