

Tarea 4 para Desarrollo de Interfaces

Nueva reserva de pista

Datos del cliente ! No es un entero

Nombre:

Telefono: ES +34

Datos de la reserva

Fecha de la reserva: 20/1/24, 12:12

Hora de reserva: 17:00

Tipo de reserva: Clase Liga Amistoso

Nivel de partido amistoso: Iniciación

Partido abierto: (los jugadores se pueden apuntar por la web)

Plazas libres:

VOLVER ENVIAR



Nueva reserva de pista

Datos del cliente ! no es un entero

Nombre:

Telefono: ES +34

Datos de la reserva

Fecha de la reserva: 20/1/24, 16:18

Hora de reserva: 17:00

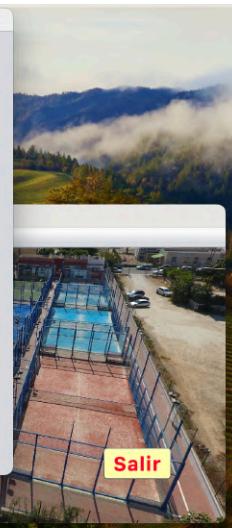
Tipo de reserva: Clase Liga Amistoso

Nivel de partido amistoso: Iniciación

Partido abierto: (los jugadores se pueden apuntar por la web)

Plazas libres:

VOLVER ENVIAR



Nueva reserva de pista

Datos del cliente ! Nombre no puede estar vacío

Nombre:

Telefono: ES +34 Introduce aquí el número de teléfono

Datos de la reserva

Fecha de la reserva: 20/1/24, 16:18

Hora de reserva: 17:00

Tipo de reserva: Clase Liga Amistoso

Nivel de partido amistoso: Iniciación

Partido abierto: (los jugadores se pueden apuntar por la web)

Plazas libres:

VOLVER ENVIAR



Nueva reserva de pista

Datos del cliente

Nombre: ASDFADSf!

Telefono: ES +34

Datos de la reserva

Fecha de la reserva: 20/1/24, 16:18

Hora de reserva: 17:00

Tipo de reserva: Clase Liga Amistoso

Nivel de partido amistoso: Iniciación

Partido abierto: (los jugadores se pueden apuntar por la web)

Plazas libres:

VOLVER ENVIAR



Diego Manuel Carrasco Castañares

Índice:

Enunciado de la tarea.....03

Ejercicio 1:

Qué características básicas y atributos de usabilidad no se cumplen. Indica ejemplos concretos.....05

Qué pautas de diseño usable no se han tenido en cuenta. Indica ejemplos concretos.....06

Qué principios básicos estructurales no se cumplen. Indica ejemplos concretos ..07

Qué elementos y estructura de la interfaz propondrías para aumentar su usabilidad y accesibilidad. Pon ejemplos concretos ..07

Ejercicio 2:

Validación de los campos.....16

Mejora del aspecto de la interface.....20

Detalles de la tarea de esta unidad.

Enunciado.

Esta tarea se va a dividir en dos partes, diferenciando entre una primera más teórica en la que debes basarte en los apuntes, y otra eminentemente práctica:

1. Para la página web de este enlace (<https://arngren.net/>), identifica:
 - o Qué características básicas y atributos de usabilidad no se cumplen. Indica ejemplos concretos.
 - o Qué pautas de diseño usable no se han tenido en cuenta. Indica ejemplos concretos.
 - o Qué principios básicos estructurales no se cumplen. Indica ejemplos concretos.
 - o Qué elementos y estructura de la interfaz propondrías para aumentar su usabilidad y accesibilidad. Pon ejemplos concretos.
2. Retoma la aplicación que desarrollaste en la tarea 1 y, tras visualizar los recursos "Vídeo - Validar formulario utilizando ValidationAPI" y "Vídeo - Cómo hacer una interfaz Java Swing con aspecto web" de esta unidad, aplícalo al formulario de reserva mejorando su aspecto e incluyendo una validación para los principales campos.

Criterios de puntuación. Total 10 puntos.

• Parte 1:

- ▶ Especificación de características y atributos que no se cumplen: 0,5 puntos.
- ▶ Especificación de pautas de diseño usable que no se han tenido en cuenta: 0,5 puntos.
- ▶ Especificación de los principios básicos estructurales que no se han tenido en cuenta: 0,5 puntos.
- ▶ Planificación de estructura y elementos de interfaz propuestos para mejorar la usabilidad: 1 punto.

• Parte 2:

- ▶ Validación de los campos del formulario, especialmente las cajas de texto: 3,5 puntos.
- ▶ Mejora del aspecto de la interfaz: 2 puntos.
- ▶ Documentación explicando con pantallazos los pasos más reseñables de la validación y el cambio de aspecto mediante Material Design: 2 punto.

Recursos necesarios para realizar la Tarea.

Imagen y vídeos de la unidad.

Indicaciones de entrega.

Una vez realizada la tarea tendrás que comprimir los archivos que has generado y subirlos a la plataforma. La estructura a entregar dentro del archivo comprimido es como sigue:

Un directorio llamado leeme donde incluyas un archivo .pdf para la primera parte de la tarea, y otro .pdf para explicar el desarrollo de la segunda parte.

Un directorio llamado proyecto con la aplicación completa.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la cuarta unidad de DI, debería nombrar esta tarea como...

sanchez_manas_begona_DI04_Tarea

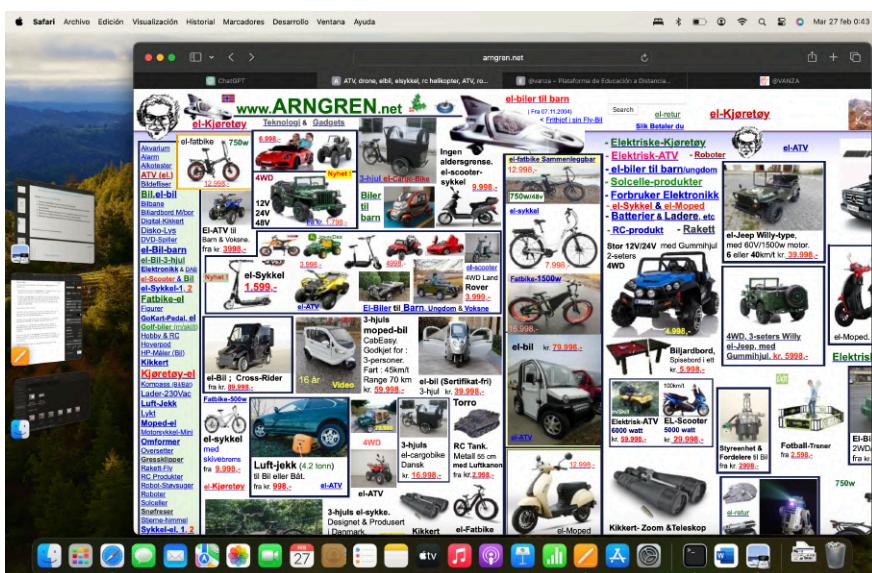
Ejercicio 1

Las características básicas y atributos de usabilidad que no se cumplen son:

- Características básicas de usabilidad: no es fácil de usar, ni de aprender y mucho menos es elegante en su diseño.
- Atributos de usabilidad: El sistema no es fácil de aprender por lo tanto será complicado alcanzar un nivel de eficiencia decente, lo que también conlleva a que será complicado recordar en el tiempo como navegar por la web y a su vez la satisfacción del usuario no será buena.

Unos ejemplos claros serían:

Lo difícil que es encontrar un anuncio en concreto dentro de la web, existen varios buscadores, pero están mezclados con los anuncios y me ha costado encontrarlos.



Aprender algo es complicado por el caos formado por los anuncios mezclados y pisándose, en algunas ocasiones, entre ellos.

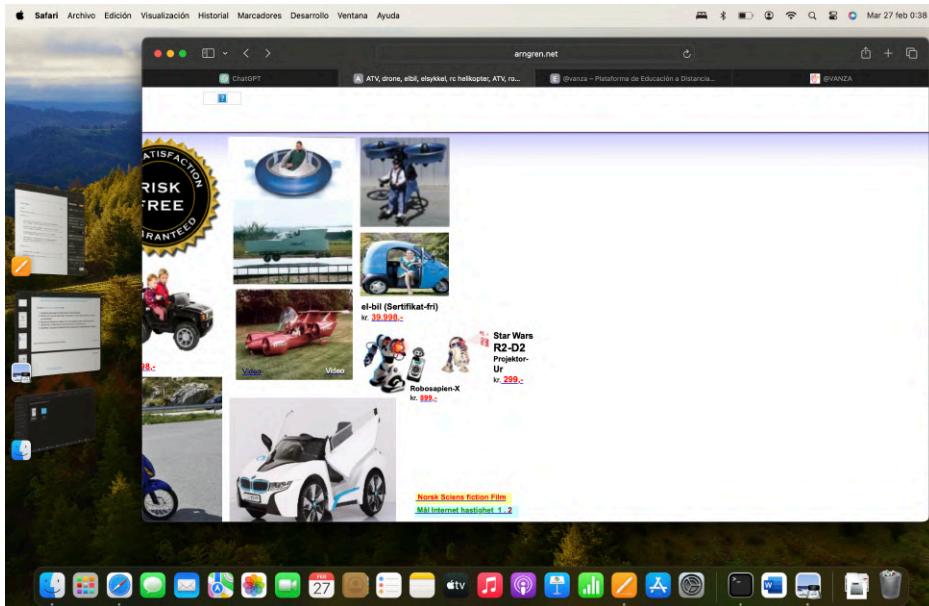
el-biler til barn
(Fra 07.11.2004)
< Frithjof i sin Fly-Bil

Ingen aldersgrense.
el-scooter-sykkelen
9.998,-

el-fatbike Sammenleggbar
12.998,-
750w/48v
el-sykkelen

- E
- E
- e
- S
- F
- e
- B
- R

La elegancia en el diseño esta totalmente ausente, anuncios mezclados entre ellos, precios que no sabes a que anuncio de su alrededor corresponde e incluso quedan zonas en blanco en la pantalla.



A collage of various children's and adult electric vehicles and toys. It includes a yellow dirt bike, a green John Deere tractor, a black electric scooter, a red off-road vehicle, a red Ferrari model, a green Vespa-style scooter, a black electric scooter, a yellow quad bike, a black off-road vehicle, a green off-road vehicle, and a red off-road vehicle. Text overlays include 'El-ATV til Barn & Voksne fra kr. 3998,-', 'Nyhet! el-Sykkel 1.599,-', 'el-ATV', 'El-Biler til Barn, Ungdom & Voksne', 'el-scooter 4WD Land Rover 3.999,-', and 'El-Biler til Barn, Ungdom & Voksne'.

Al ser una web difícil de aprender la eficiencia a la hora de buscar un anuncio, o en su defecto, una categoría no es buena, por ello será difícil recordar como hacerlo, en caso de conseguirlo, en un futuro.

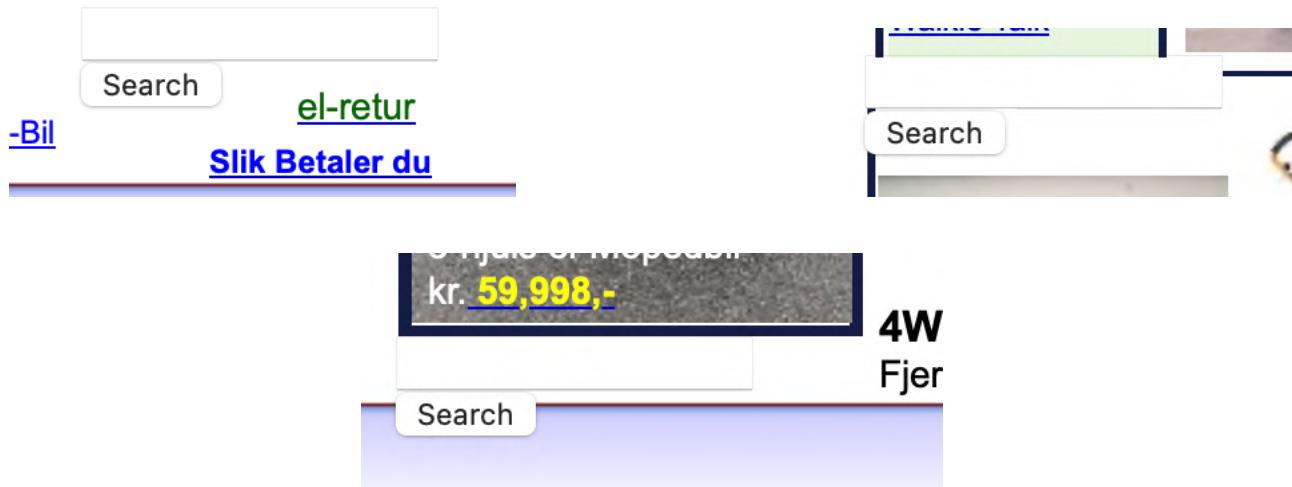
Todo ello conlleva a que la satisfacción del usuario no será buena.

Las pautas de diseño que no se han tenido en cuenta:

No existe ninguna organización de los elementos, que como podemos ver en las imágenes anteriores, todos los anuncios, independientemente de su categoría, están mezclados entre ellos.

En la imagen anterior podemos apreciar, en una zona de la web, varios anuncios, mezclados de coches, patinetes, motor, quad...

No se ha economizado los elementos y contenido, ya que por ejemplo, existen, al menos, tres buscadores repartidos por la web.



El diseño visual es malo ya que los anuncios no guardan un orden o una jerarquía entre otras cosas.

Los principios básicos estructurales que no sigue serían:

La familiaridad de uso es uno de los aspectos que no se siguen, es difícil recordar, entre tanto caos de anuncios, como navegar en la página web.

La consistencia ya que no existen menús que facilite el trabajo del usuario).

Los elementos que añadiríamos, sería primeramente, un menú, donde el usuario pueda recurrir para navegar hacia un punto concreto de la web (por ejemplo, coches, motos, etc...).

Algunos iconos le vendrían muy bien, para identificar el menú por ejemplo y un buscador visible y fácil de usar.

Sobre la estructura, pondría una cabecera, el cuerpo y un título, y en el lateral izquierdo pondría el menú.

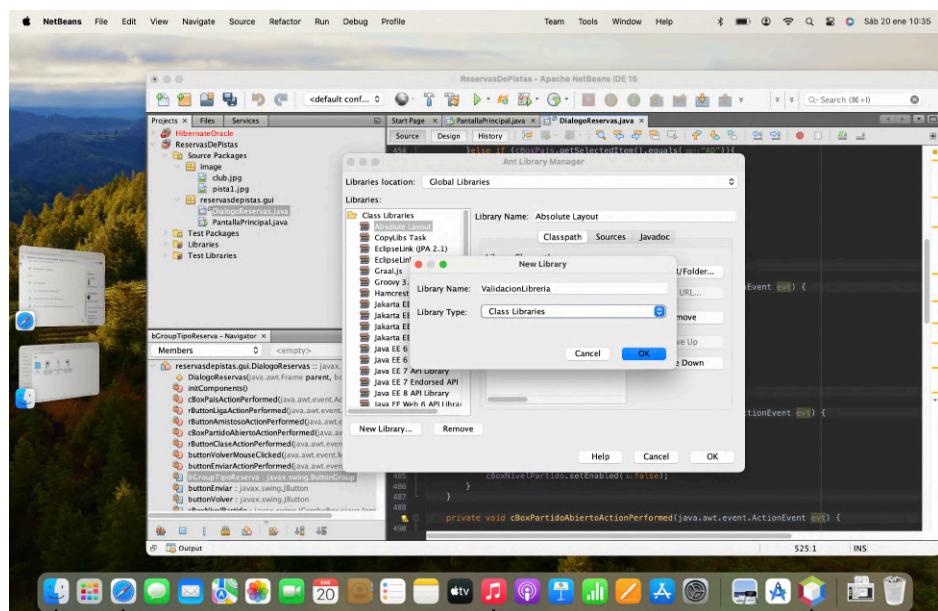
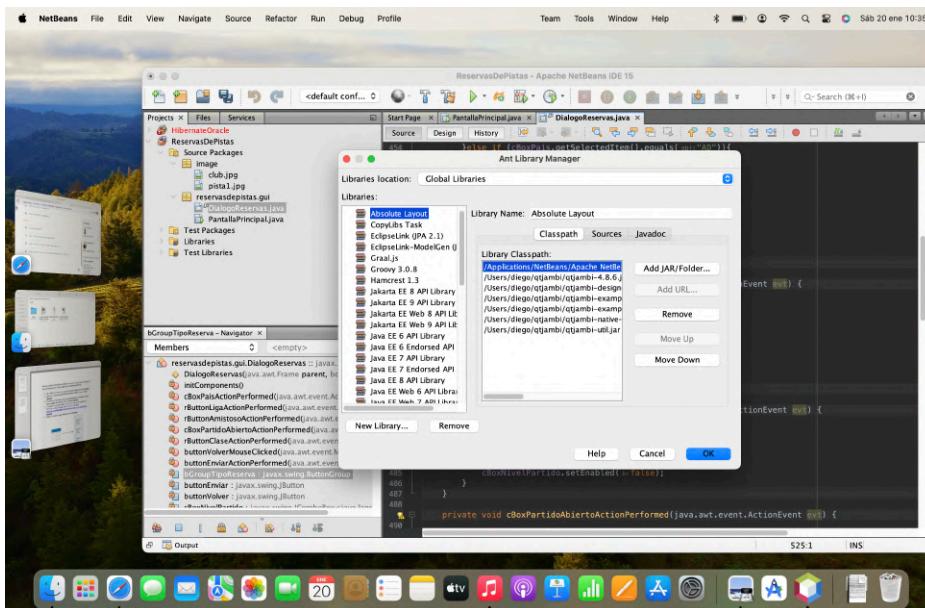
Se usarían colores pasteles para no cansar la vista del usuario.

El tipo de fuente usada sería la misma para toda la web, con un tamaño adecuado, haciendo los títulos un poco mas grande que las descripciones.

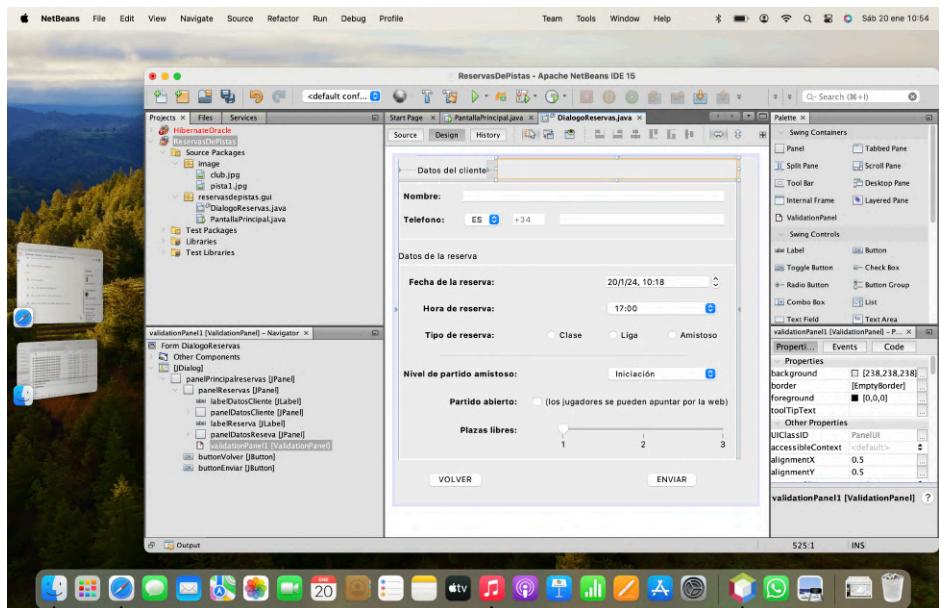
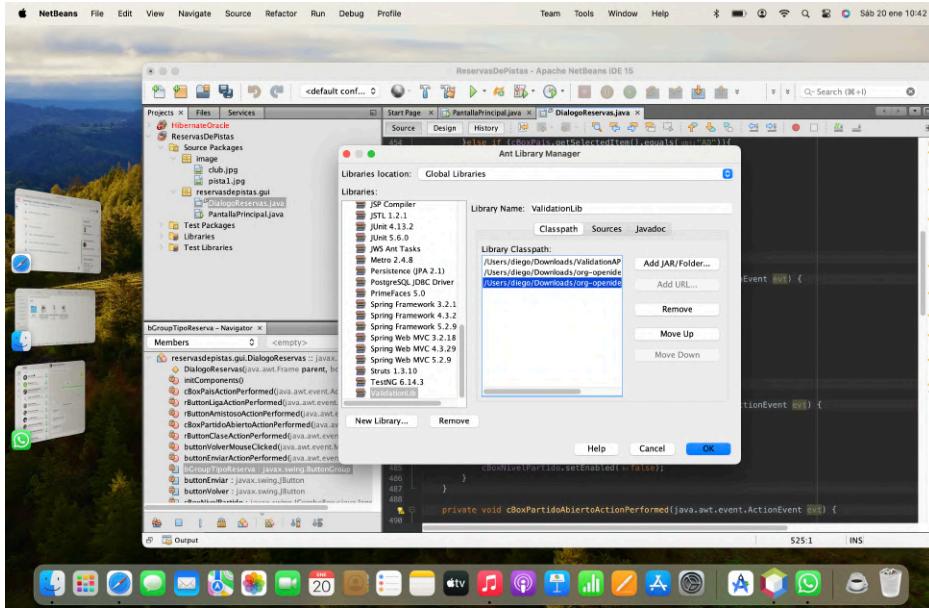
Ejercicio 2

En primer lugar nos dirigiremos a NetBeans, haremos un copia del proyecto de la unidad 1 y la abriremos.

Una vez abierta, añadiremos una nueva librería (botón derecho sobre Libraries -> Add Library -> New Library -> Ok), a la cual vamos a llamar validacionLibreria.

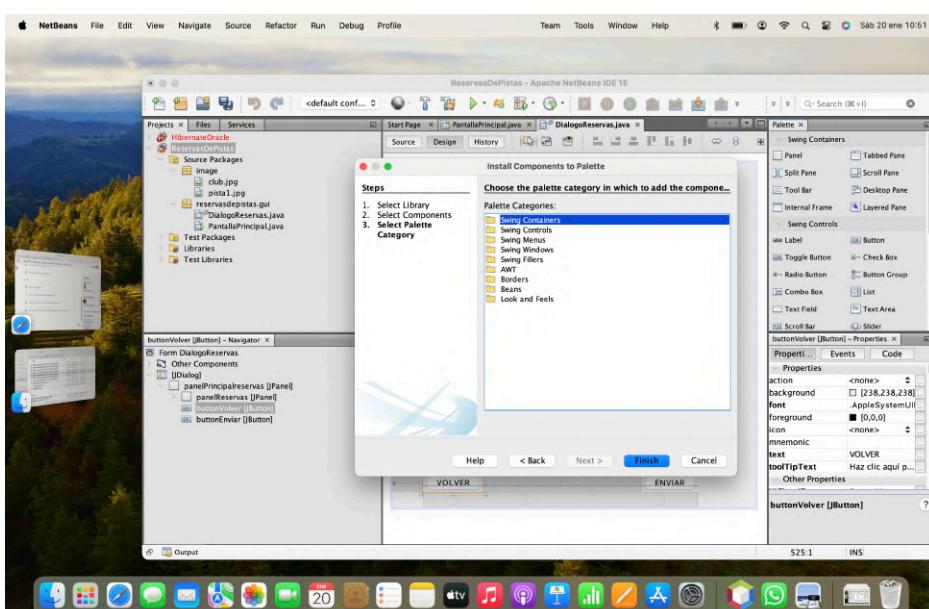
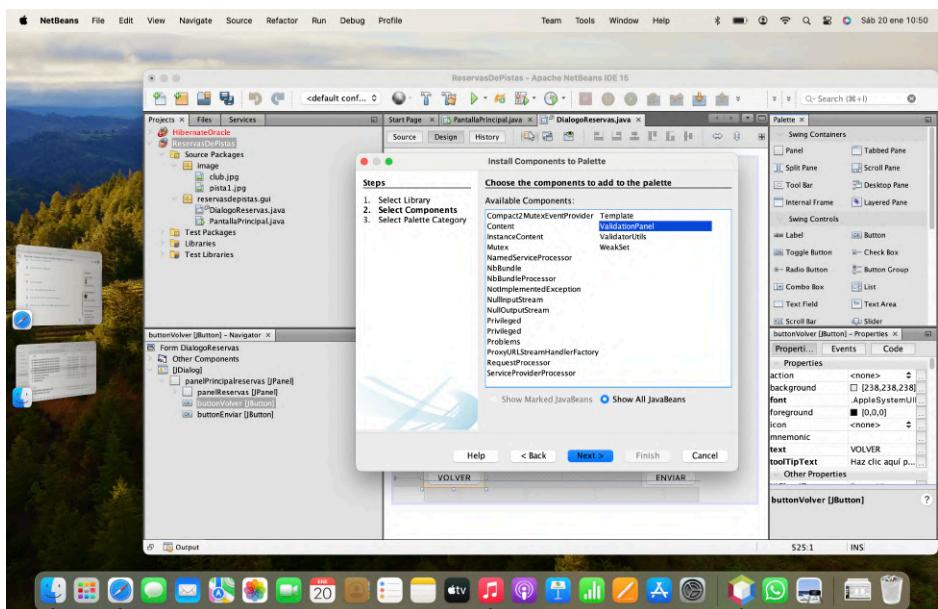
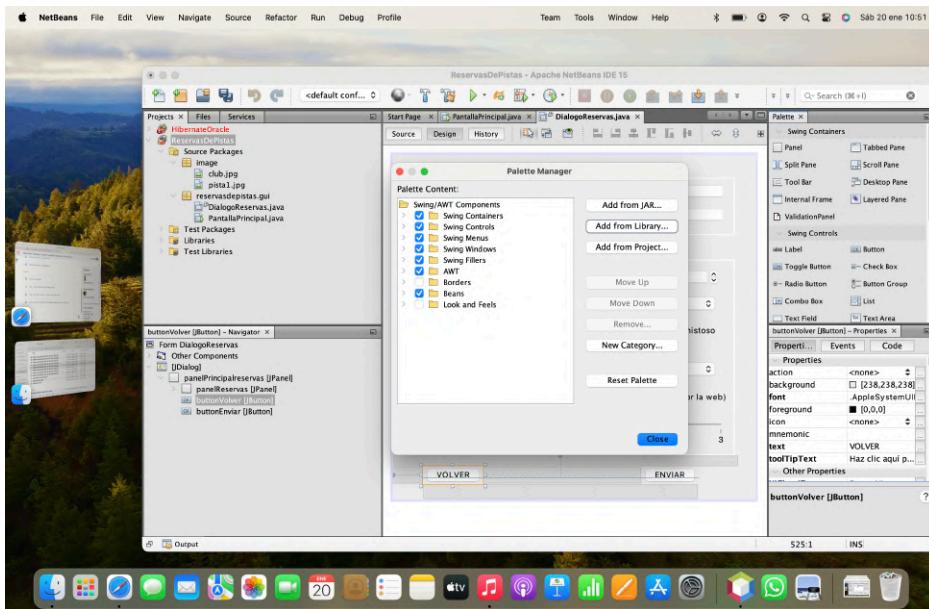


Una vez creada (en mi caso se colocó en último lugar de todas las librerías), nos dirigimos a ella, la seleccionamos y añadimos mediante el botón Add Library.



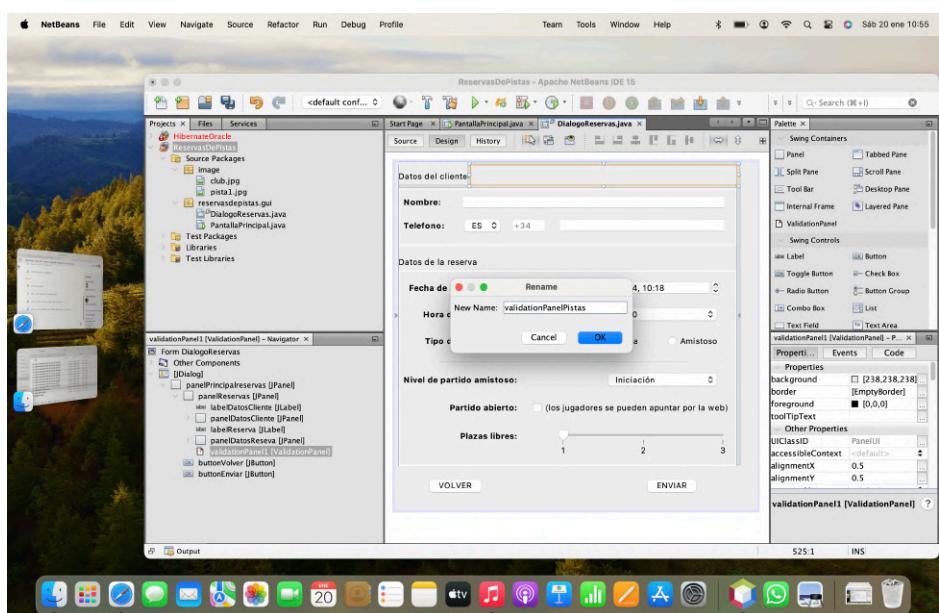
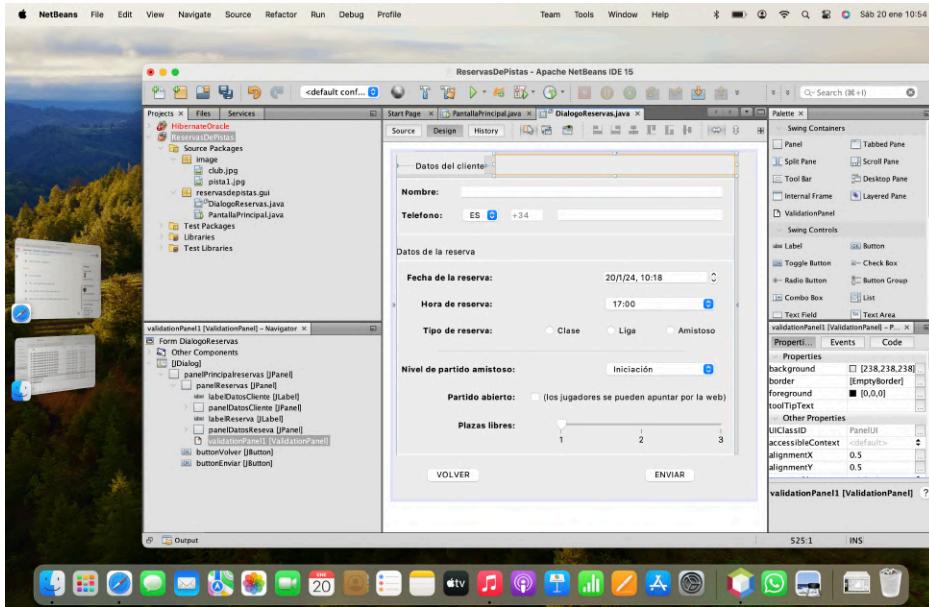
Para añadirlo a la paleta seleccionamos *Palette Manager*, clicamos sobre *add from library* y seleccionamos la librería (ValidationLib en nuestro caso) y clicamos en *next*.

En la siguiente pestaña seleccionamos *ValidationPanel*, clicamos en *next*, elegimos donde queremos almacenarla (para el caso que nos ocupa vamos a seleccionar *Swing Containers*) y clicamos en *finish*.

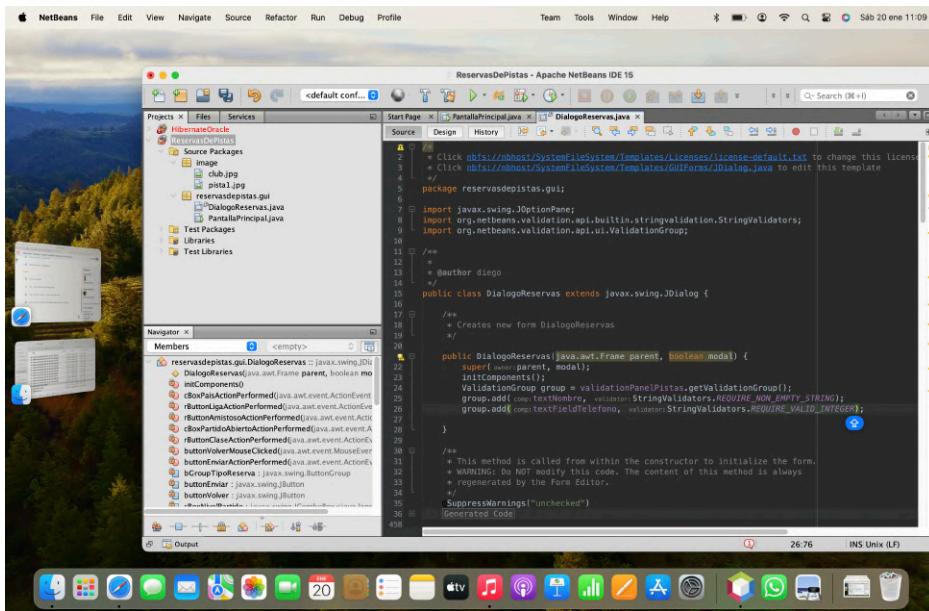


Ahora ya nos aparece en nuestra paleta, en concreto en el apartado de swing containers.

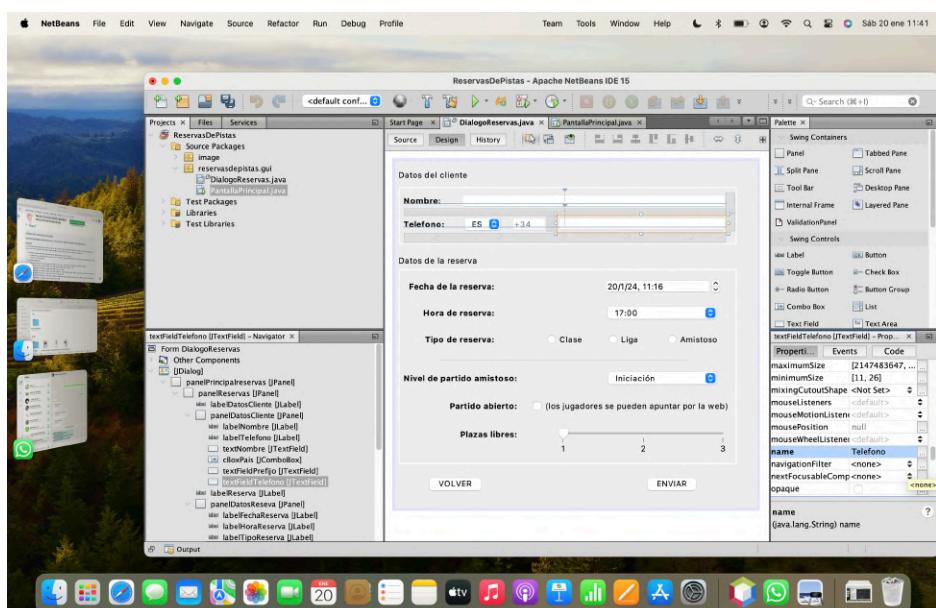
Lo arrastramos a la zona donde queremos que aparezca y le cambiamos del nombre para acceder a él.

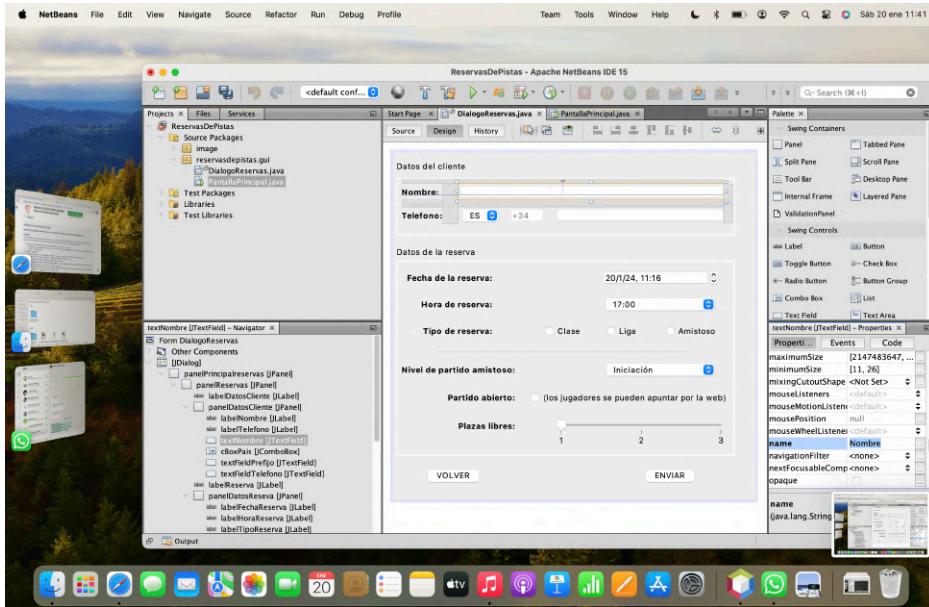


En el constructor de la clase metemos un ValidationGroup y le añadimos, mediante el método add los campos que queremos validar y que queremos validar.

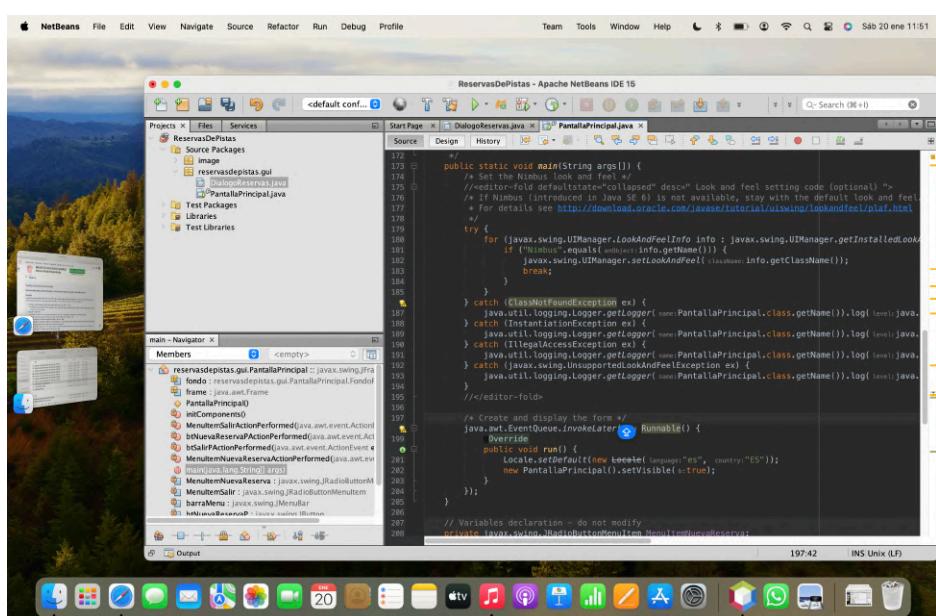


Tras ello accedemos a los atributos de los campos que queremos validar y le ponemos un nombre, para que en la validación nos muestre ese nombre. (Teléfono para jTextField del teléfono y Nombre para el jTextField del nombre).





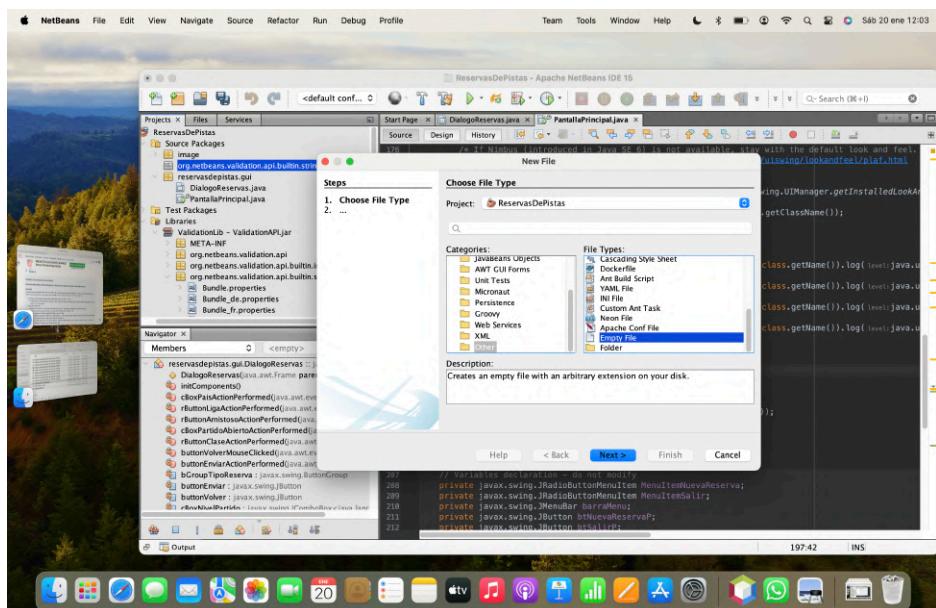
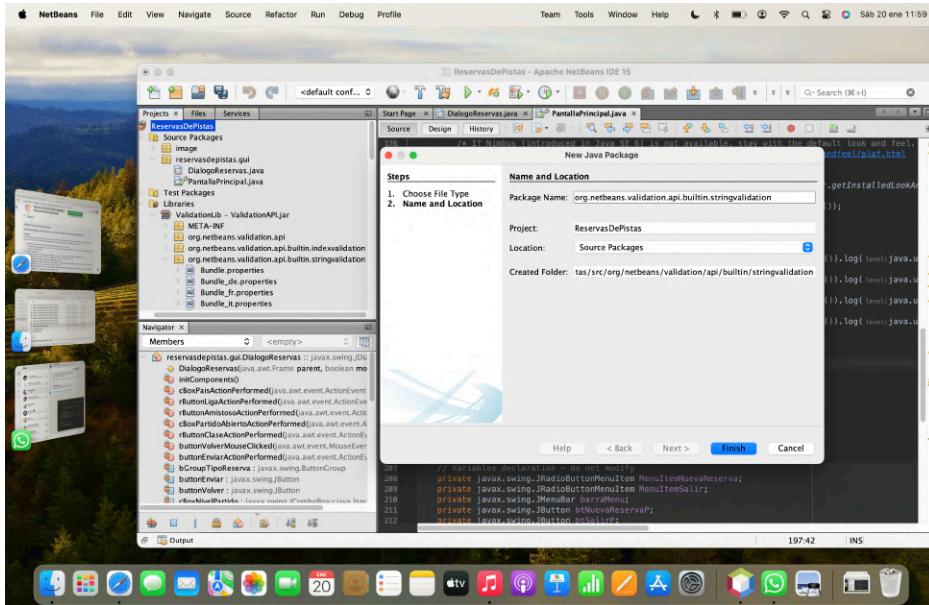
A continuación debemos cambiar el idioma a español, que lo haremos como puede apreciarse en la siguiente imagen.



Tras ello crearemos un nuevo paquete llamado "org.netbeans.validation.api.builtin.stringvalidation".

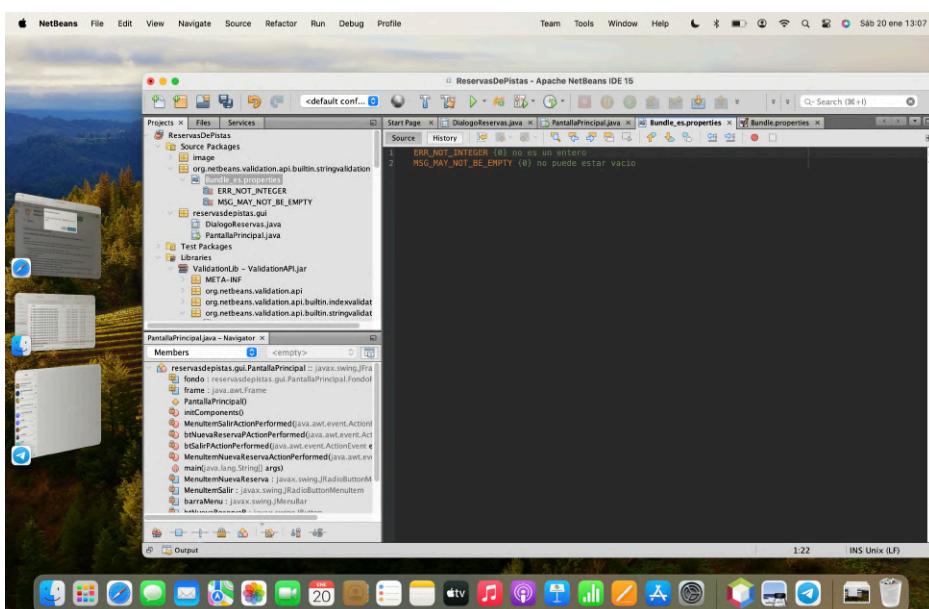
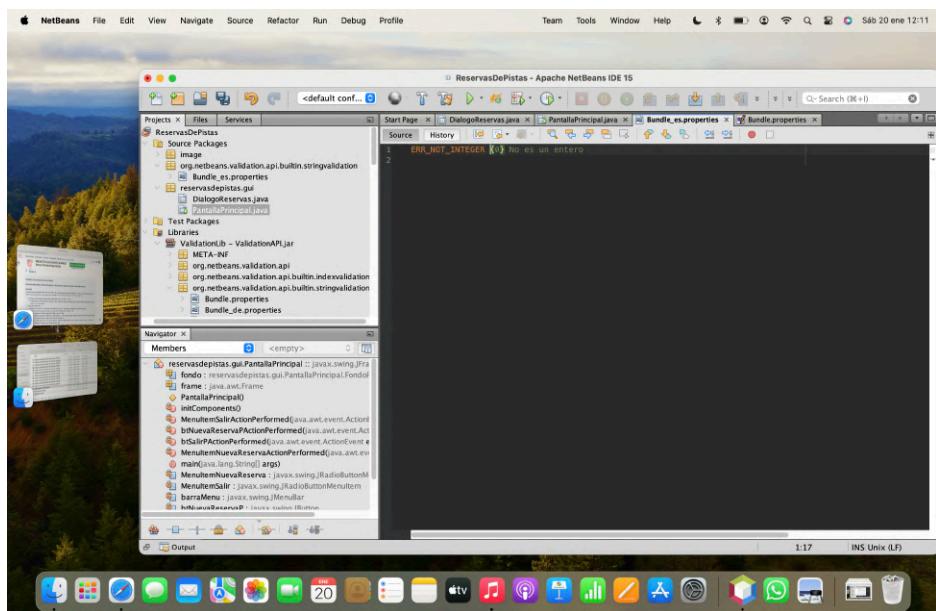
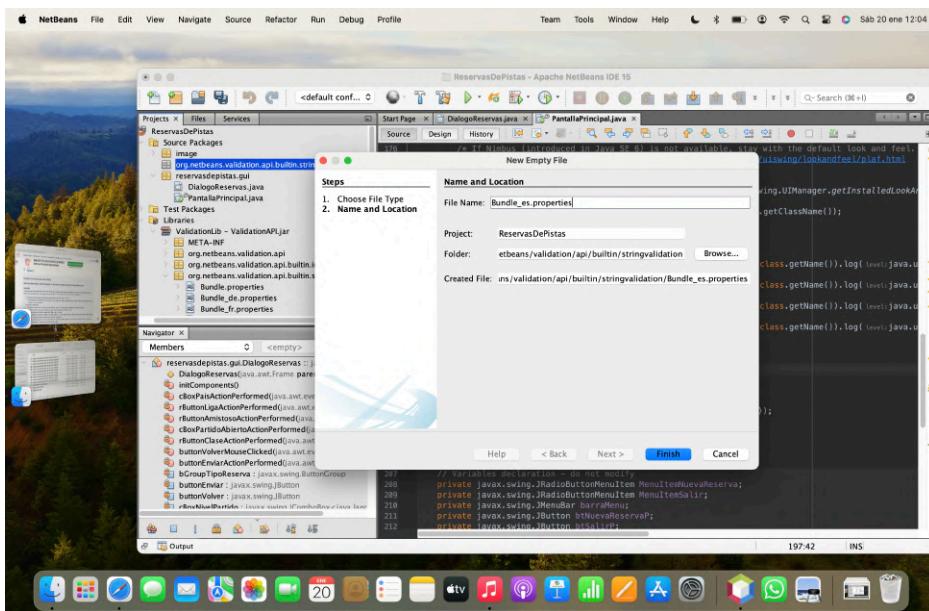
Diego Manuel Carrasco Castañares

Página 13 de 20

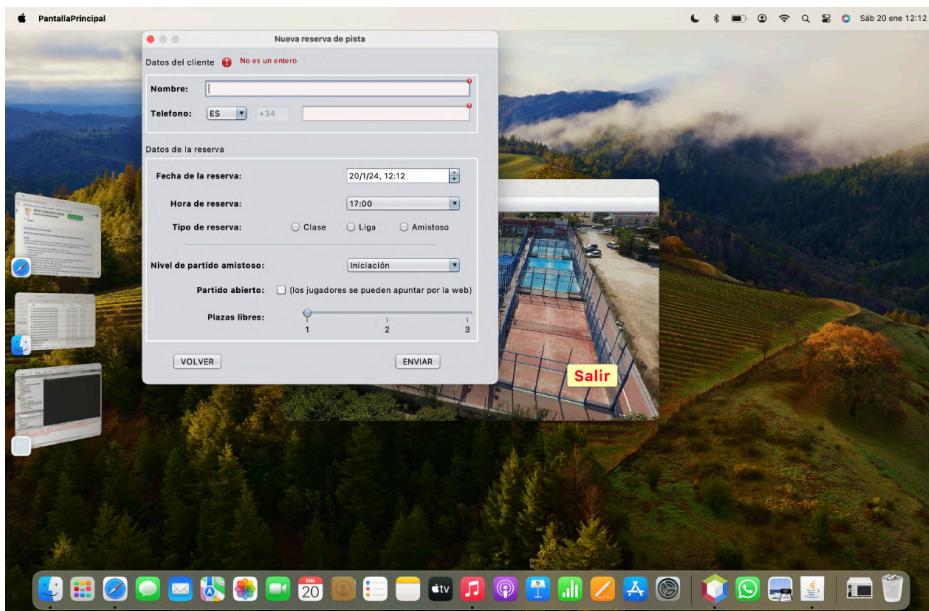


Dentro de este paquete añadiremos un nuevo archivo vacío al cual llamaremos "Bundle_es.properties".

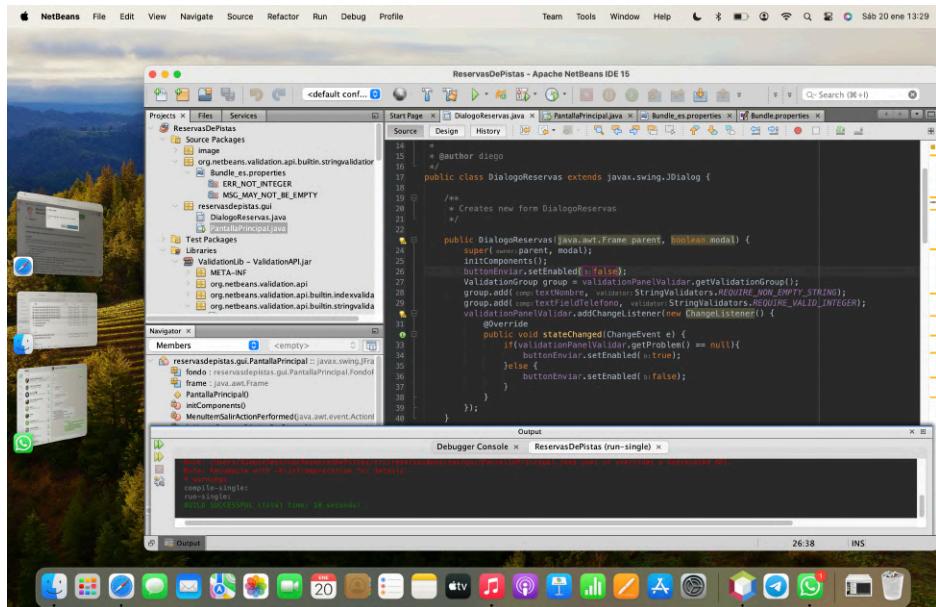
Dentro de dicho archivo escribiremos el texto que queremos que nos muestre a la hora de validar el campo según puede apreciarse en siguiente imagen.



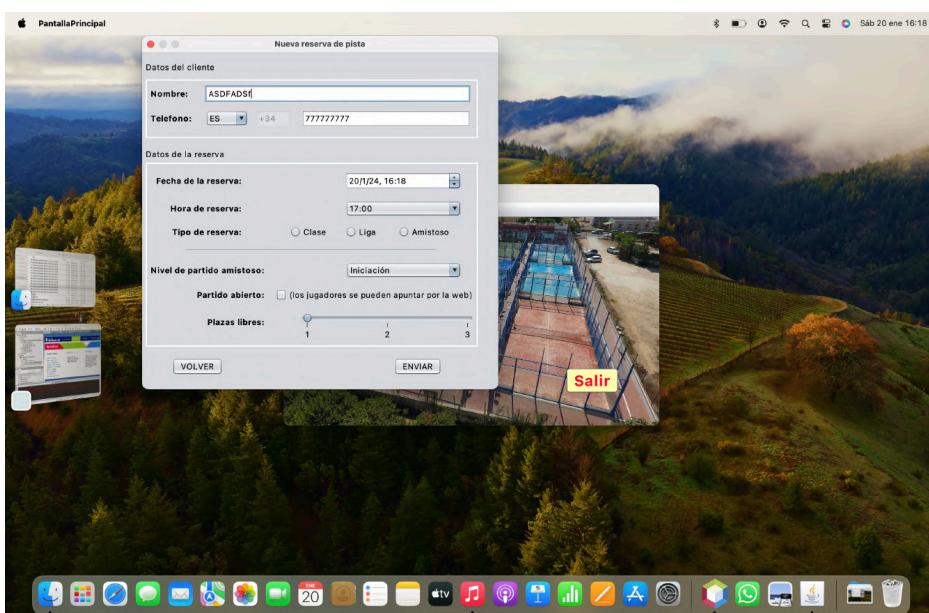
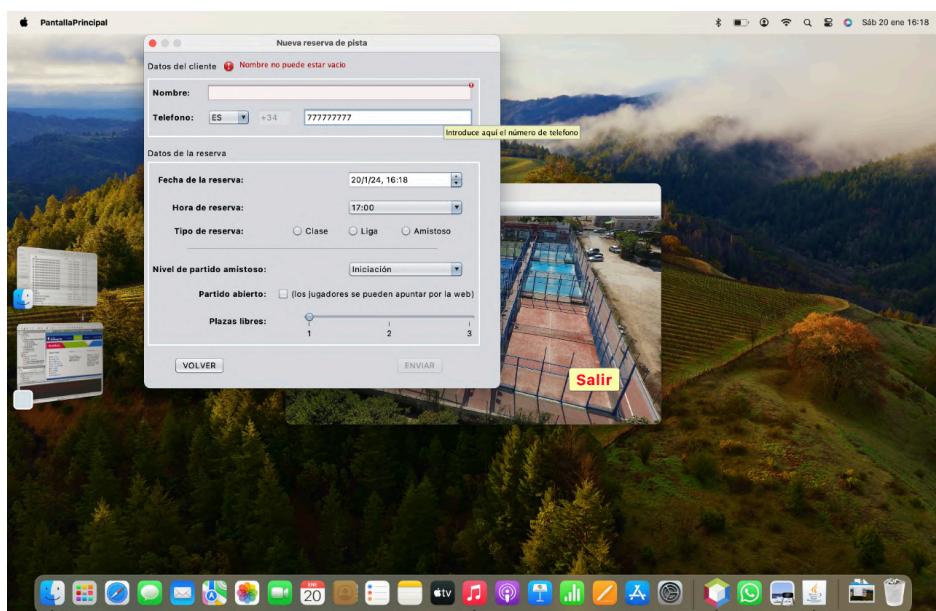
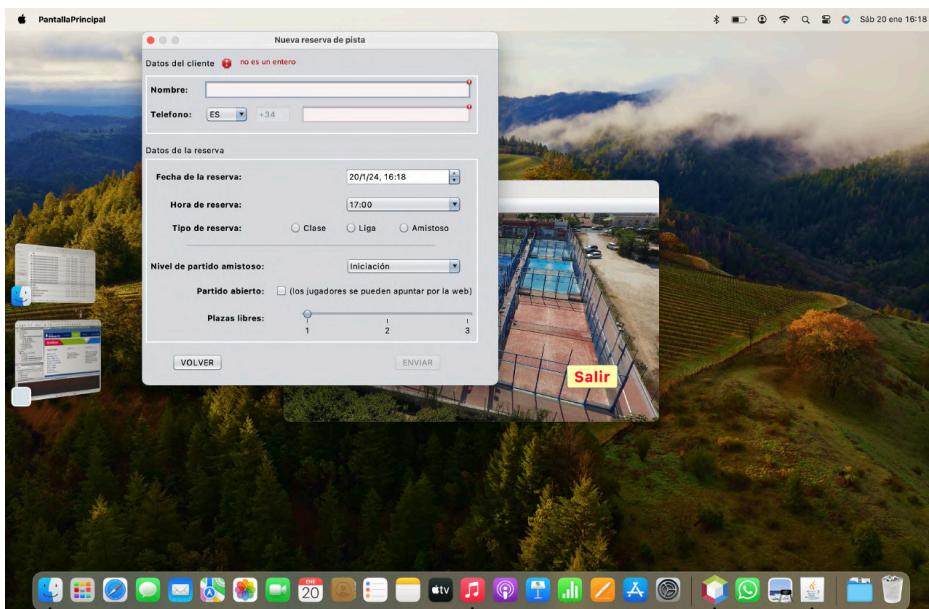
En las siguientes imágenes podemos ver como ya nos muestra el texto que hemos puesto a la hora de validar.



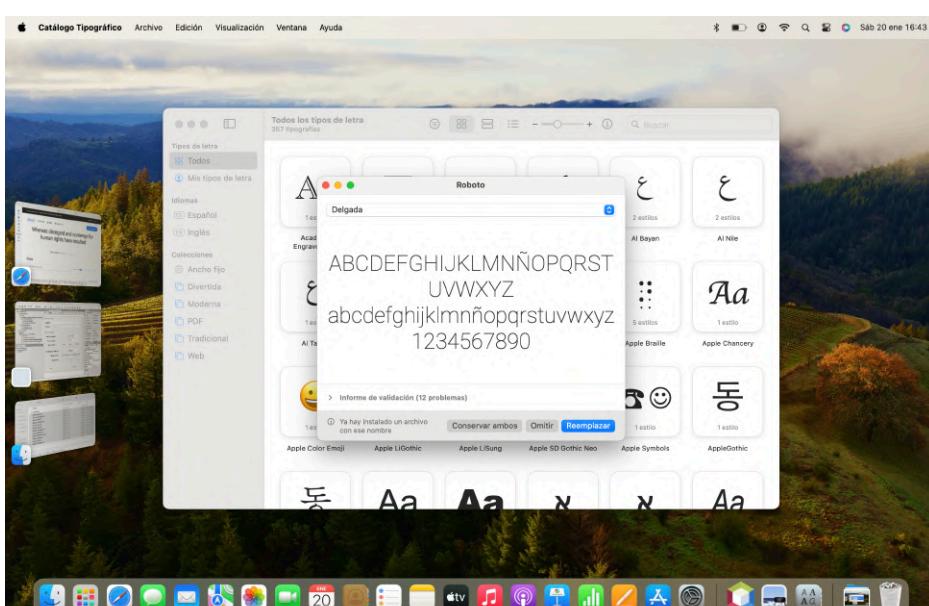
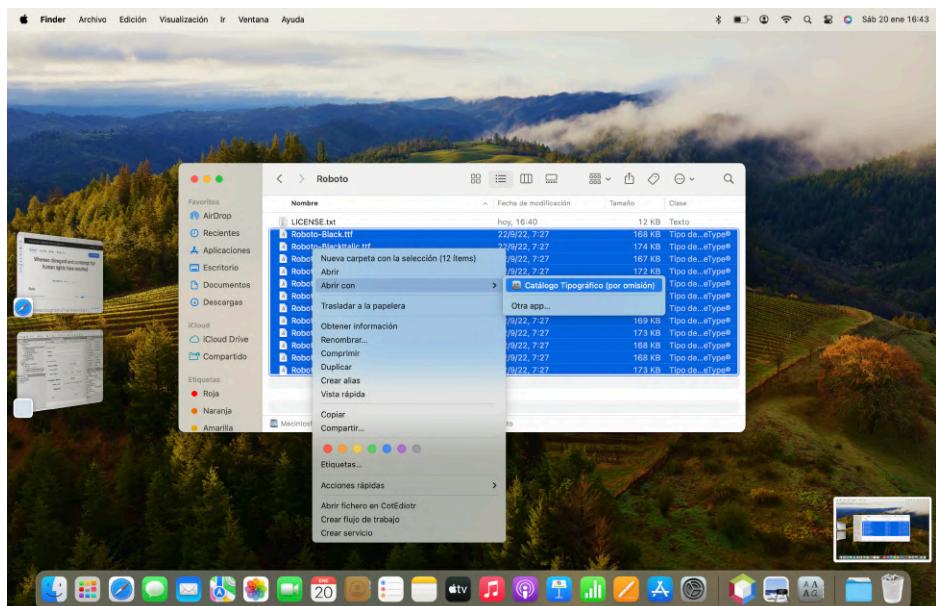
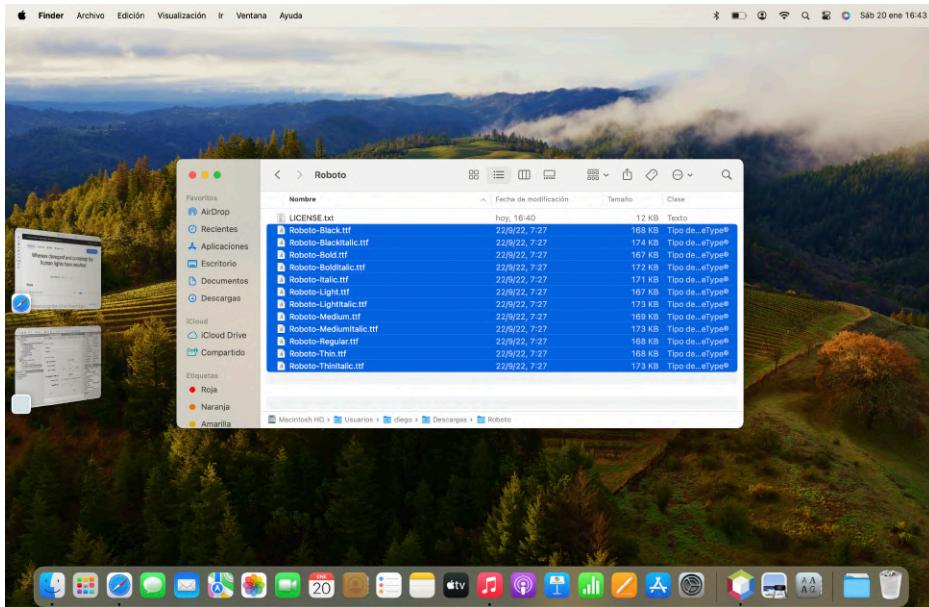
Tras ellos vamos a deshabilitar el botón enviar en el constructor por defecto, para que solo este habilitado cuando los dos campos que vamos a validar estén rellenos, cosa que conseguiremos mediante un if y el método `getProblem` aplicado al panel que hemos creado.



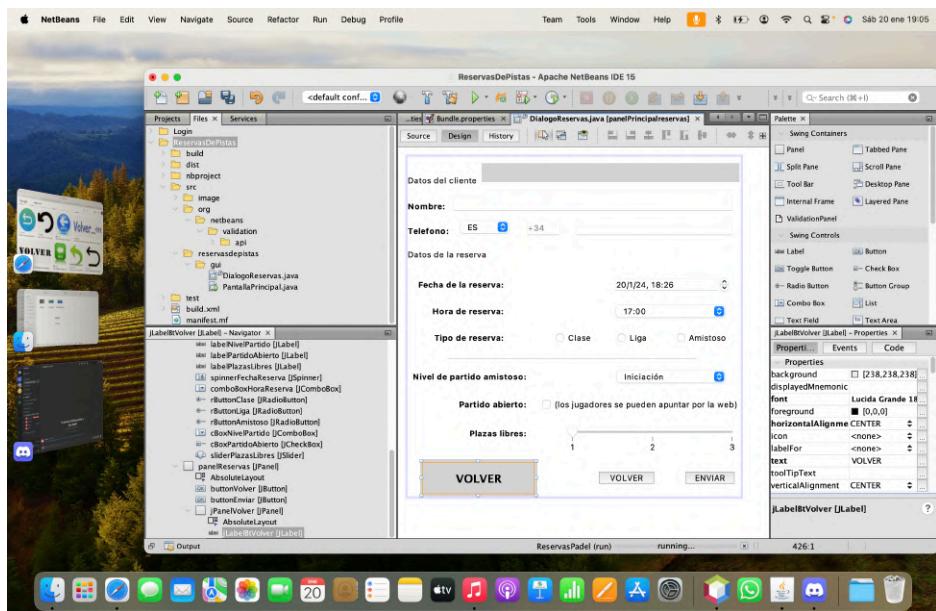
En las siguientes imágenes podemos ver como funciona.



En las siguientes imágenes podemos ver el proceso de instalación de la fuente de texto roboto.



Y en la siguiente imagen podemos ver como hemos cambiado el botón volver por un label, lo mismo haremos con el botón enviar y con el botón de cerrar la ventana.



Al igual que los botones tenían sus eventos, tenemos que pasar estos eventos a los label. El código empleado es el mismo que tenían los botones.

Además, vamos a añadir un evento para controlar cuando entra y cuando sale el ratón de dichos label, para que, una vez que entre el ratón, cambie de color y cuando salga vuelva a su color original.

Podemos apreciar dicho código en la siguientes imágenes.

```

private void jLabelEnviaMouseClicked(java.awt.event.MouseEvent evt) {
    int option = JOptionPane.showConfirmDialog(parentComponent.rootPane, "Seguro que no te importa que se genere una reserva?", titleConfirmation, JOptionPane.YES_NO_OPTION);
    if (option == JOptionPane.YES_NO_OPTION) {
        this.dispose();
    }
}

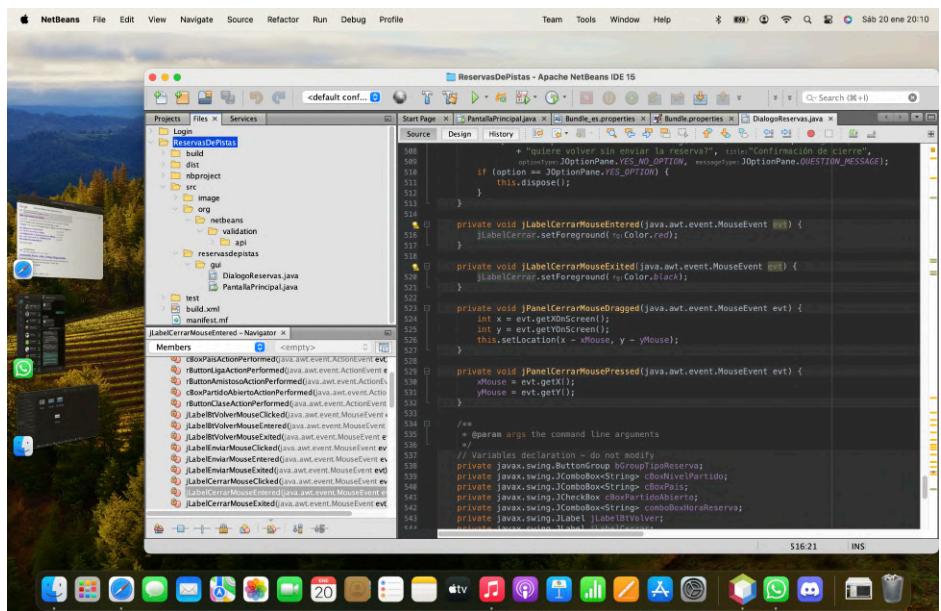
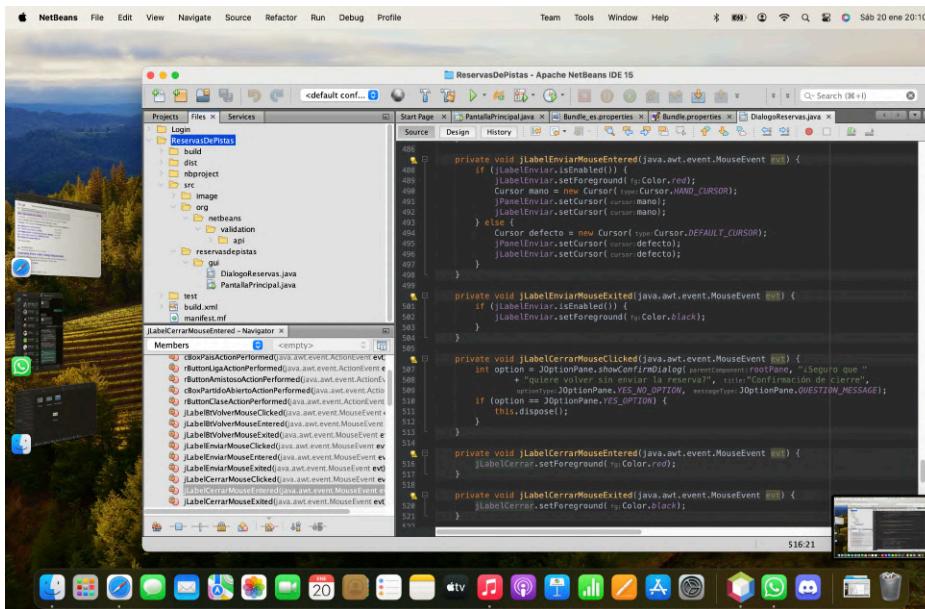
private void jLabelEnviaMouseEntered(java.awt.event.MouseEvent evt) {
    jLabelEnvia.setForeground(Color.red);
}

private void jLabelEnviaMouseExited(java.awt.event.MouseEvent evt) {
    jLabelEnvia.setForeground(Color.black);
}

private void jLabelEnviaMouseMoved(java.awt.event.MouseEvent evt) {
    if (!jLabelEnvia.isEnabled()) {
        JOptionPane.showMessageDialog(parentComponent.rootPane, message);
        this.dispose();
    }
}

private void jLabelEnviaMouseClicked(java.awt.event.MouseEvent evt) {
    if (!jLabelEnvia.isEnabled()) {
        jLabelEnvia.setForeground(Color.red);
        Cursor mano = new Cursor(Cursor.HAND_CURSOR);
        jLabelEnvia.setCursor(mano);
        jLabelEnvia.setCursor(mano);
        jLabelEnvia.setCursor(mano);
    } else {
        Cursor defecto = new Cursor(Cursor.DEFAULT_CURSOR);
        jLabelEnvia.setCursor(defecto);
        jLabelEnvia.setCursor(defecto);
        jLabelEnvia.setCursor(defecto);
    }
}

```



En la siguiente imagen podemos ver las mejoras de la interfaz.

