DOCUMENTACION PROYECTO

BINGO

Ivan Valle Soto

# Índice

1. **Introducción……………………………………………………………Pag 2.**
2. **Desarrollo de la aplicación……………………………………...Pag 3.** 
   1. Fase de diseño…………………………………………………..Pag 3.
   2. Fase de implementación…………………………………Pag 6.
      1. Lógica…………………………………………………Pag 6.
      2. Interfaz………………………………………………Pag 8.
      3. Pruebas……………………………………………..Pag 15.

## -Introducción-

En este proyecto se nos ha “contactado” desde un supuesto centro comercial, el cual quiere implementar un bingo, con 15 tiradas y un cartón de 9 números aleatorios del 1 al 30, que de partida a aquellos clientes que hayan gastado una cantidad determinada 30€, despues se le mostrara al usuario una selección de premios según lo que se haya obtenido en la partida, además se incluye una novedad como el número mágico, el cual si se obtiene en algún premio el usuario registrándose o identificándose sumara 20€ de bono a su saldo para futuras compras.

Con esta descripción del producto se pueden identificar los principales partes de la aplicación que son:

-Validación de tickets: se tiene que comprobar que el código EAN del ticket introducido es válido, eso implica buscar en el archivo donde almacenamos los tickets del centro comercial.

-Implementar el juego del bingo: adaptar el juego físico del bingo a una interfaz gráfica que resulte familiar al usuario, usable y sólida, que además implemente las particularidades que se nos han exigido para nuestro bingo, como el funcionamiento del número mágico, cuando deben ser cantadas las líneas o el bingo, o como se asignan los números al cartón, además de informar de todos los cambios en el sistema al usuario.

-Ofrecer premios al usuario: según los resultados de la partida debemos cargar de la lista de premios del centro comercial, unos determinados, sea bingo o línea, además ver si en el caso de línea puede escoger 2 al haber cantado dos líneas.

-Gestionar el número mágico: mostrarle al usuario la ventana donde pueda registrarse o identificarse para recibir el bono regalo, si optara a el, y también darle la opción de saltarse esta ventana si no quiere dar sus datos informándole además de que si lo hace perderá el bono.

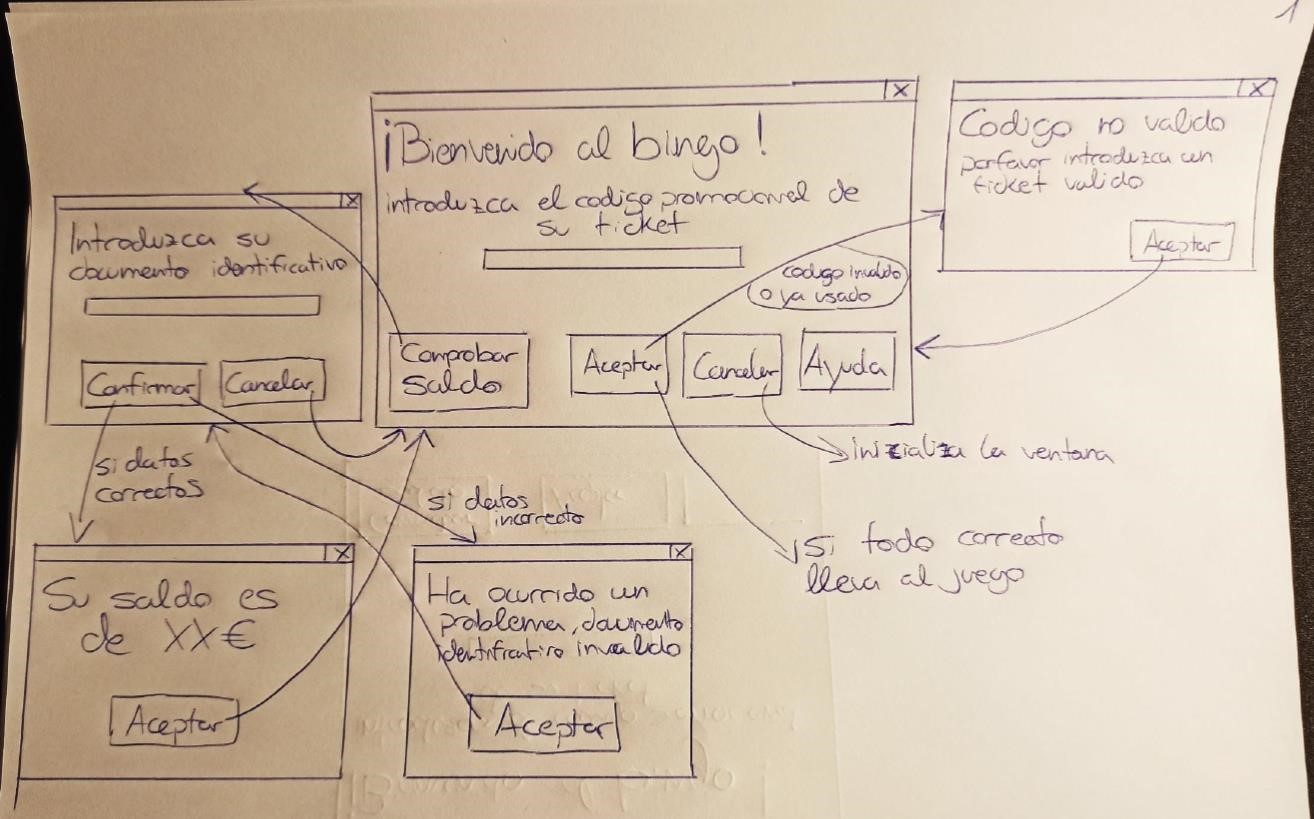
-Ayuda en línea: es imprescindible ofrecerle al usuario un sistema de ayuda donde se le explique, el funcionamiento del sistema como jugar al bingo, normas, como reclamar premios etc.

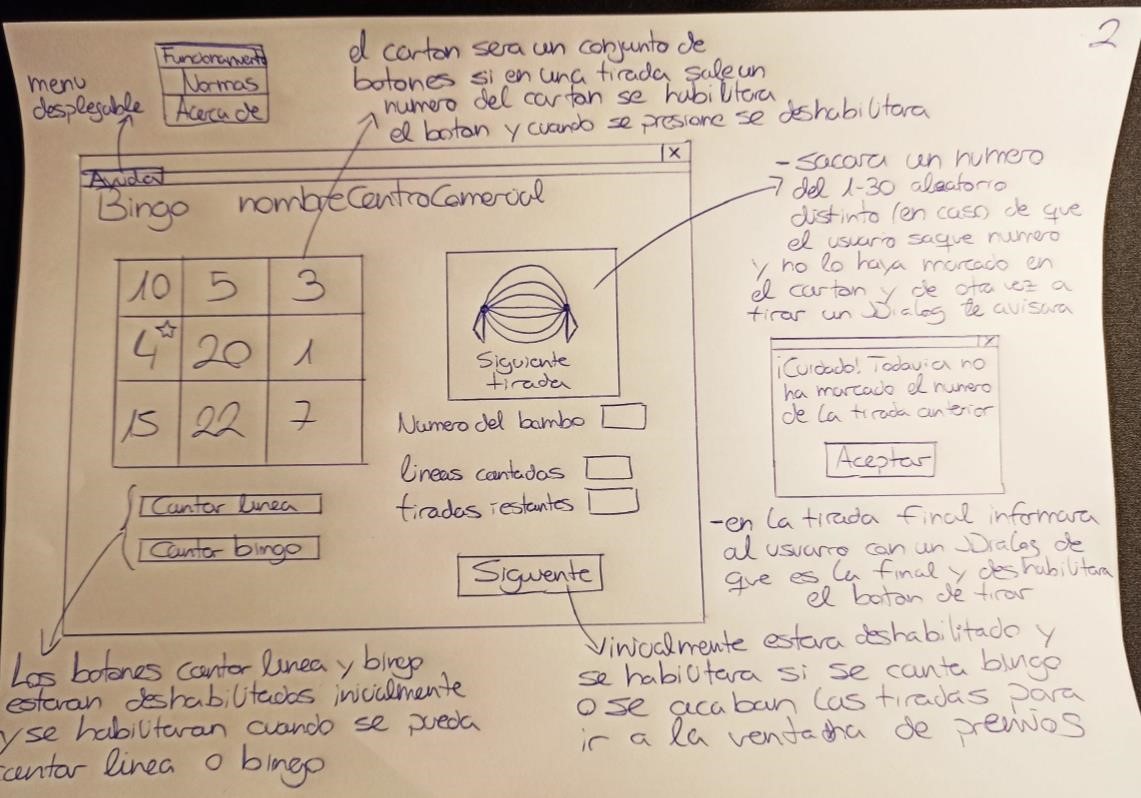
-Internacionalización (opcional): En mi caso he internacionalizado la aplicación, localizándola para el inglés y el español, hay que identificar que elementos necesitan el cambio de idioma y como cambiar los textos según el idioma de los JDialog.

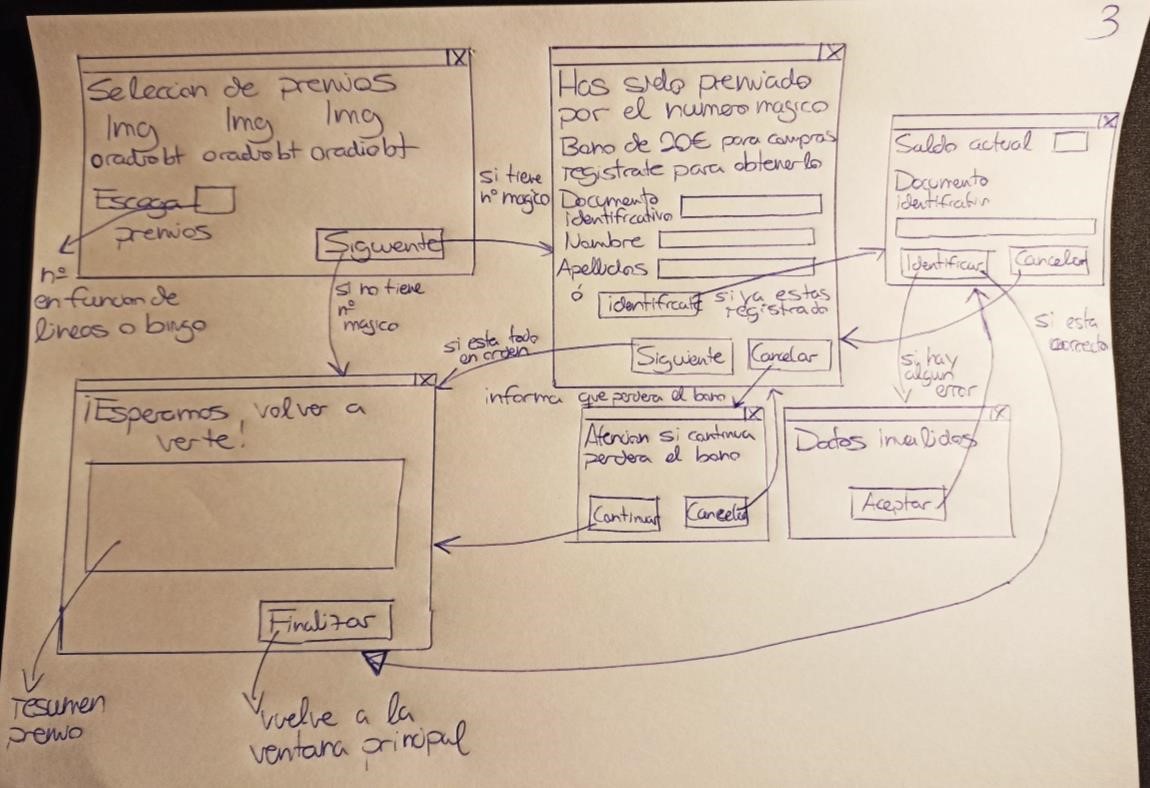
## -Desarrollo de la aplicación-

### -Fase de diseño-

La versión 2 del wireframe es la siguiente:







Los cambios realizados sobre la versión 1 del wireframe recogidos en el informe de corrección son los siguientes:

Página 1 (Imagen 1):

-Añado ventanas y botones para poder comprobar el saldo de una cuenta si se quiere antes de jugar.

-Añado botón que muestra la ayuda desde la primera ventana si así se desea.

Página 2(Imagen 2):

-Esta ventana no tenía ningún problema en el informe de evaluación.

Página 3(Imagen 3):

-Añado los botones y ventanas para permitir un logging con el documento de identificación si ya está registrado el usuario.

-Cambio el campo de DNI a documento de identificación por si se diera el caso de que alguien tiene otro documento NIF, Pasaporte, etc

-CAMBIOS REALIZADOS SOBRE EL WIREFRAME 2 EN LA APLICACIÓN FINAL-

Página 1(Imagen 1): En el wireframe, en la ventana inicio, faltaría mostrar los dos botones para cambiar entre la aplicación en español o en inglés que incorpore en la aplicación cuando decidí hacer la parte opcional de internacionalización.

Página 2(Imagen 2): Al solicitar que la ventana se pudiera redimensionar tuve que ajustar la posición de los elementos en la pantalla, además faltarían los JDialog que informar al usuario de que tiene una línea o bingo sin cantar (tanto en el botón de siguiente tirada como en el de siguiente).

Página 3(Imagen 3): La ventana de selección de premios es la que más cambios ha sufrido, inspirado en la practica 11 de laboratorio, me di cuenta de que era más usable si cargaba botones dinámicamente con los premios a elegir que con imágenes y radiobotones, además decidí que esta ventana también pudiera redimensionarse y le introduje un panel donde mostrarle la lista de premios seleccionado hasta el momento, además de un botón para quitar premios seleccionados de la lista y uno para acceder a la ayuda, he quitado la parte donde se informa del número de premios que puede escoger (ya que si tienes una línea o bingo vas a escoger un solo premio pero de una lista distinta), también después de consultar unas dudas con la profesora añadí unos JDialog ligados al botón siguiente informando al usuario que le quedan premios por seleccionar por si le diera al botón por error.

La ventana de registro/identificación para obtener el número mágico consideré oportuno añadirle un botón de ayuda y cambiar el botón de siguiente por uno de registro.

## -Desarrollo de la aplicación-

### -Fase de implementación-

#### -Lógica-

La lógica de la aplicación está compuesta por las siguientes clases:

Ticket->Es la clase para crear los objetos ticket con los atributos del código del ticket (String) y el importe del mismo (double) cuando los cargamos del fichero, ademas tiene un método que lo serializa para guardar en base.

Cliente->Es la clase para crear los objetos cliente con los atributos documento de identificación (String), nombre (String), apellidos (String) y el saldo (int) cuando los cargamos del fichero, además redefine el método equals para comparar clientes.

BaseDatosTicket->Es la clase para gestionar la lista de tickets de la aplicación, tiene como atributo la lista de tickets que cargamos de fichero, además realiza las operaciones de validar el código de un ticket y de actualizar el fichero de tickets cuando por ejemplo eliminamos uno usado.

BaseDatosClientes->Es la clase para gestionar la lista de clientes de la aplicación, tiene como atributo la lista de clientes que cargamos de fichero, además realiza las operaciones de validar un documento de identificación, registrar jugadores en la base y actualizar el fichero con la lista de clientes.

Premio->Es la clase para crear los objeto premio que cargamos del fichero, tiene como atributos el código (String), el nombre (String) y el carácter del tipo de premio (char).

CatalogoPremios->Es la clase para gestionar la lista de premios de la aplicación, tiene como atributo la lista con el catálogo de premios que se carga del fichero, además realiza las operaciones de filtrar aquellos premios que son para línea o para bingo.

FileUtil->Es la clase abstracta que se encarga de las acciones con fichero como: cargar los premios, clientes y ticket de fichero, añadir un cliente al fichero, actualizar la base de clientes y de tickets.

Casilla->Es la clase que crea el objeto casilla, para ser usada posteriormente en el cartón del bingo, tiene como atributos su número (int), si esta seleccionada (boolean) y si contiene el número mágico (boolean).

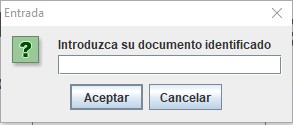
Carton->Es la clase que gestiona el cartón de juego del bingo, tiene como atributo una matriz de dos dimensiones de casillas, además realiza las operaciones de rellenar el cartón, asignar número al cartón, asignar número mágico, marcar un número, saber si se puede cantar una línea, si hay bingo, saber la línea del número mágico.

Juego->Es la clase que gestiona todo el juego del bingo, tiene como atributos el catálogo de premios, la base de clientes, la base de tickets, el cartón, un booleano para saber si hay bingo, el código del ticket de juego (String), la línea sin cantar marcada (int), número de líneas cantadas (int), numero de tiradas (int) y un booleano para saber si se ha obtenido el número mágico, además realiza todas las acciones del juego del bingo como inicializar el cartón, devolver el saldo del cliente, comprobar si la partida ha finalizado, marcar una casilla, cantar línea y bingo, comprobar si se ha obtenido el número mágico en algún premio, obtener la lista de premios según el resultado de la partida, registrar jugadores en base, eliminar un tickets y actualizar las base de datos de tickets y clientes.

-Interfaz- Ventana inicio

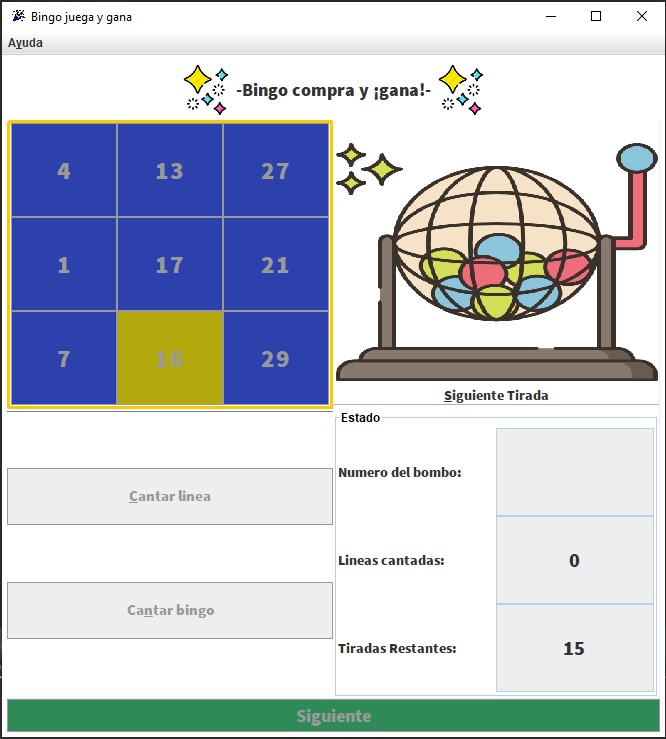


Esta es la ventana de inicio de la aplicación como características generales no se puede redimensionar e implementa un JPanel con AbsoluteLayout para poder colocar todos los elementos tal y como quería. En cuanto a los componentes podemos observar en la parte superior los dos botones de cambio de idioma a español e inglés, después debajo de las JLabel que nos indican que hay que introducir el código promocional del ticket vemos un campo de texto TextField donde introduciremos el código EAN de nuestro ticket, después en la parte inferior a la izquierda se puede ver el botón de consultar saldo, este nos abrirá un InputDialog para introducir el documento identificativo y después nos mostrara un MessageDialog con nuestro saldo si el documento es correcto o un mensaje de error si algo ha fallado.



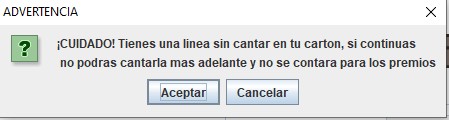
En la parte derecha encontraremos los botones de siguiente que si el código es correcto nos llevara a la siguiente ventana de juego (Ventana Principal) o si no nos informara con un MessageDialog del error, el de cancelar que inicializa la ventana y el de ayuda que nos abre la ayuda en línea.

Ventana Principal



Esta es la ventana de juego, al requerirse de que se pueda redimensionar, escogí para el panel principal un BorderLayout, en el cual dividí la ventana en 3 secciones: la norte para la cabecera un FlowLayout con las JLabel de las imágenes de los brillos y del *-Bingo compra y ¡gana!-*. la sur donde se aloja el botón de siguiente, y la central, a la hora de elegir como distribuir los componentes del bingo, decidí que se podían localizar 4 zonas diferenciadas y para ello lo mejor era un GridLayout de 2 columnas y 2 filas, y en cada área meter un panel de tal manera que siguiendo un orden de izquierda a derecha y de arriba abajo seguiría así: El primero el panel del cartón del bingo un GridLayout de 3 columnas y 3 filas tal y como se nos indica, (los botones de las casillas se cargan dinámicamente). El segundo un GridLayout de 1 columna y 1 fila ya que solo va a alojar el botón del bombo de siguiente tirada. El tercero un GridLayout de 1 columna y 5 filas, de tal manera que los botones de cantar línea y cantar bingo queden centrados y separados decidí poner dos JSeparator en los huecos de las fila 1 y 3 y dejar la fila 5 vacía. Y el cuarto y último el panel, de estado, otro GridLayout de 2 columna y 3 filas para alojar la JLabel que indica que dato es el que se muestra en el JTextField de su derecha, ademas le puse un TitleBorder al panel.

Una vez explicada la estructura de paneles pasare a identificar los elementos más relevantes. El cartón del bingo es un conjunto de botones que se cargan dinámicamente de tal manera que son una representación gráfica de la matriz bidimensional que aloja los objetos casilla del cartón, estos botones se cargan y se les asigna el numero a cada uno de columna en columna de tal manera que la columna uno aloja números del 1-10, la dos del 11-20 y la tres del 21-30, además uno de esos botones será asignado como el de número mágico y se le cambiara el fondo a un amarillo/dorado para diferenciarlo del resto, estos botones estarán deshabilitados y se habilitara un botón si sale su número del bombo. El botón de siguiente tirada simula como saldrían números de un bombo, se deshabilitará al agotar las tiradas o cantar bingo, además si tenemos un numero sin marcar o línea o bingo sin marcar nos avisa con un ConfirmDialog de que si aceptamos perderemos la opción de cantar/marcar según el caso.



(Ejemplo de ConfirmDialog de línea sin cantar, tanto el de bingo como el de marcar número son análogos a este)

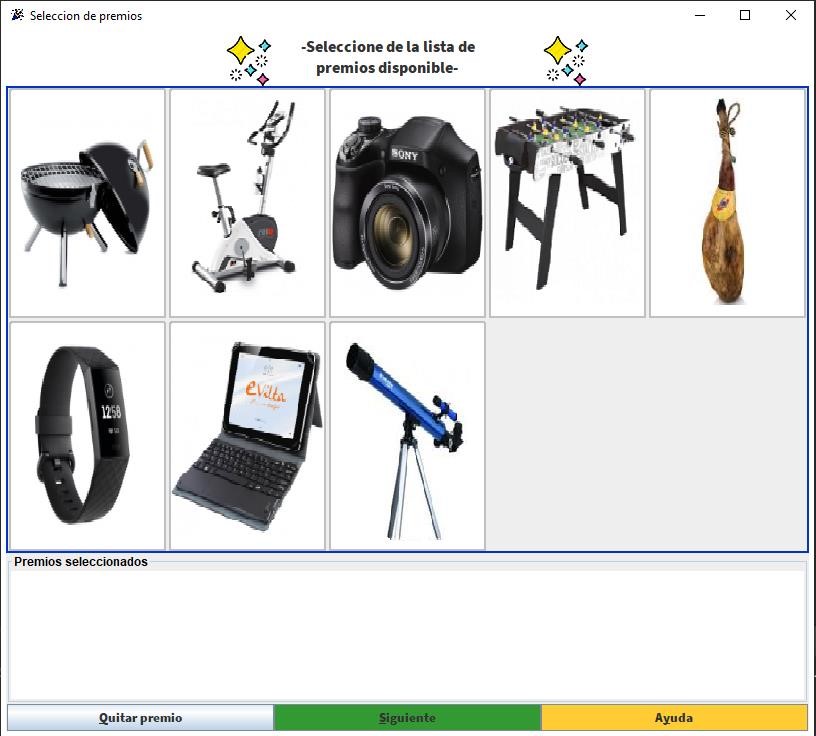
Los botones de cantar línea y cantar bingo estarán desactivados y se activarán cuando se pueda cantar alguna línea o bingo. Los elementos del panel de estado simplemente muestran al usuario información de cómo está la partida: el número que ha salido del bombo, las líneas cantadas y las tiradas restantes.

Además, a la ventana le he incorporado una JMenuBar para el menú de ayuda que consta de dos JMenuItem, uno que nos lleva a la ayuda y otro de Acerca de con información de la aplicación



El botón de siguiente esta inicialmente deshabilitado y se habilitara si se canta bingo, si es la última tirada y quedan números sin marcar o línea/bingo sin cantar avisara al usuario de la misma manera que con el botón de siguiente tirada. El botón siguiente nos llevara a la ventana de premios, si se han obtenido alguno y en caso contrario a la ventana final.

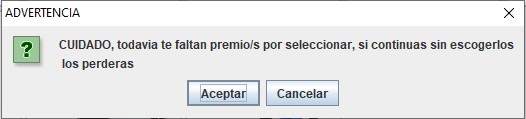
Ventana Premios



La ventana de selección de premios también se puede redimensionar, por decisión propia, el panel principal de la ventana es un BorderLayout que está dividido en: sección norte la cabecera un GridLayout con tres columnas con las JLabel de las imágenes y del texto -*Seleccione de la lista de premios disponible-*, la zona central que aloja el panel de premios un GridLayout de 5 columnas donde los botones para los distintos premios se van a cargar dinámicamente, y la zona sur con un BorderLayout que aloja en la zona norte un ScrollPane con una JList con los premios seleccionados, y en la zona sur un GridLayout de 3 columnas para los botones quitar premios, siguiente y ayuda.

Los componentes principales son los botones de los premios, se cargan dinamicamente y cada botón es principalmente una imagen del premio en cuestión, cuando se hayan escogido todos los premios posibles se deshabilitan todos. La JList que incluye los premios seleccionados, el botón de quitar premio que quita el que este seleccionado en la lista y reactiva los botones de los premios si estuvieran deshabilitados.

El botón de siguiente que si se han escogido todos los premios nos llevara dependiendo de si se ha obtenido el número mágico a la ventana de registro/identificación o si no se ha obtenido número a la ventana final, en caso de que no se hayan escogido todos los premios se informara al usuario que si continua perderá los premios no seleccionados.



Y por último el botón de ayuda que nos abre la ayuda en línea.

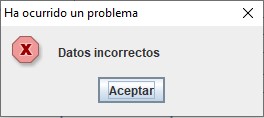
Ventana Número Mágico

Interfaz de usuario gráfica, Aplicación

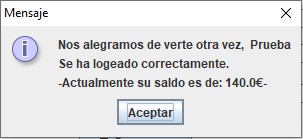
Descripción generada automáticamente

La ventana del número mágico no se puede redimensionar y implementa un

AbsoluteLayout para que queden todos los componentes a mi gusto, para diferenciar la sección de registro he metido un panel con AbsoluteLayout donde se ubican todos los elementos del registro, la JLabel que indican que dato se debe introducir en el campo de texto (textField) de su derecha y al final un botón de registro que al darle, nos comprueba los campos de texto, si no son válidos nos informa con un messageDialog y en caso de que sean correctos nos da la bienvenida con un messageDialog tambien.



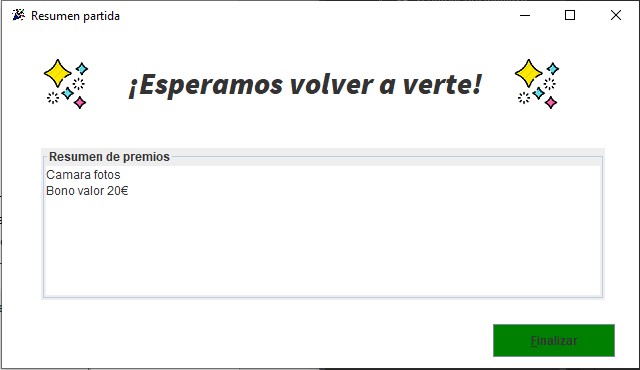
En la ventana tenemos el botón de identificación que nos abre un InputDialog donde introducimos el documento de identificación y si es correcto nos da la bienvenida y nos muestra también el saldo actualizado, en caso de que no sea correcto nos informa con un MessageDialog que el dato es incorrecto.



Además también tenemos un botón cancelar por si no queremos ni registrarnos, ni identificarnos, el cual antes nos avisara con un ConfirmDialog de que si continuamos perderemos el bono del 20€. Al lado del botón de cancelar tenemos el botón ayuda.



Ventana Final



La ventana final no se puede redimensionar e implementa un AbsoluteLayout para poder poner los componentes a mi gusto, los componentes principales son un scrollPane con un textArea que o bien nos pone una frase si no hemos obtenido premios, o nos pone un resumen de los premios obtenidos, y el botón finalizar que nos lleva a la ventana de inicio dejando lista la aplicación para una nueva partida.