

PROYECTO

COMUNUCACIÓN

PERSONA

MAQUINA

"CHRISTMAS SALES"

INDICE

Introducción	3
Desarrollo de la aplicación	3
Diseño de la interfaz de usuario.....	3
Fase de implementación	5
Lógica:.....	5
Clases receptoras de eventos:.....	5
Interfaz:	6
Pruebas de usuarios:	10
Problemas encontrados	11

Introducción

Una empresa de un centro comercial nos contactó para realizar un juego atractivo para repartir regalos por navidad. Para ello, se dio pautas por parte de la empresa para que este sistema fuese táctil y que contuviese un panel para descubrir casillas en las que nos darán puntos para posteriormente poder canjearlos por algunos regalos que se dispusiesen. Solo podrán acceder a la aplicación aquellos que tuviesen una cuenta registrada para poder identificarse en el panel.

Para la realización de este, no se ha incluido el proceso de internacionalización para clientes que pudiesen venir del extranjero.

Se ha optado por hacer un sistema separado por capas (lógica del sistema y la parte visual y la que organiza los eventos sucedidos durante la interacción).

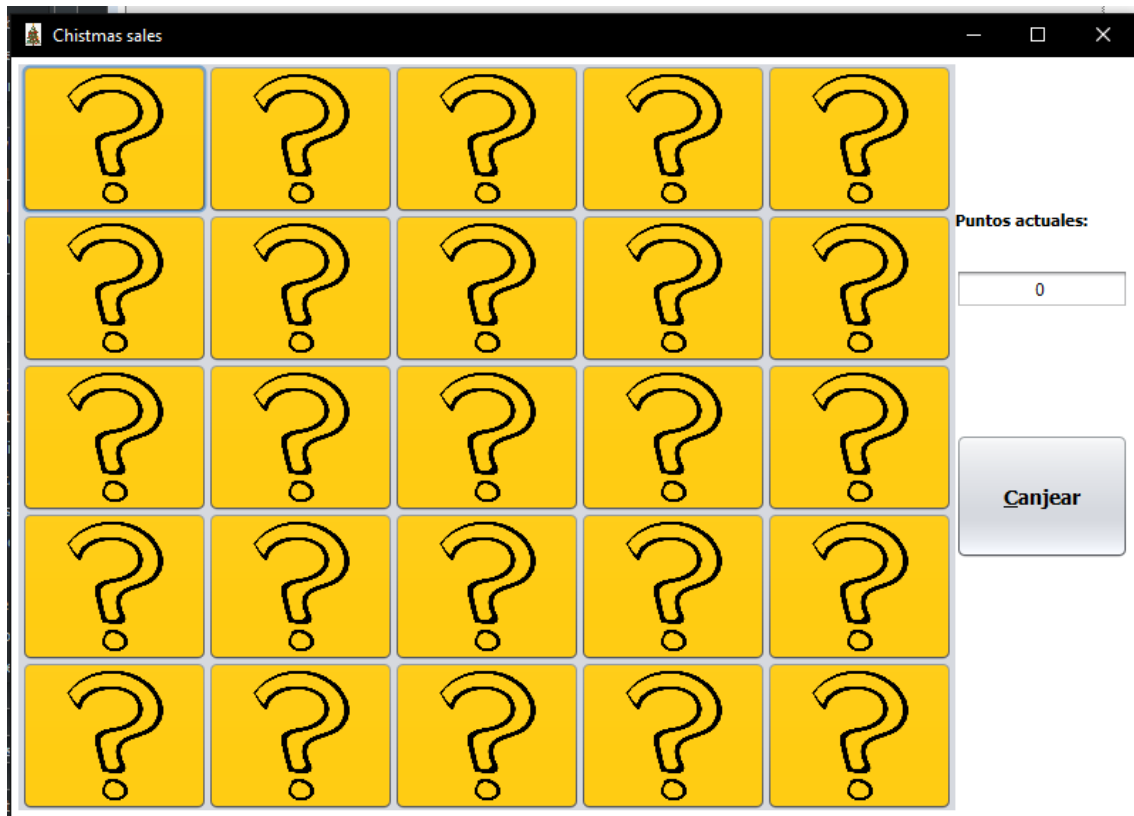
Desarrollo de la aplicación

Diseño de la interfaz de usuario

Para la realización de este proyecto se han realizado 2 wireframes (se adjunta el ultimo de ellos) que eran provisionales. A medida que se desarrollaba el proyecto se optó por realizar una serie de simplificaciones ya que se probó la interfaz con varias personas ajenas al proyecto y se concluyó que las ventanas relacionadas con la elección de los regalos y su canjeo debían cambiarse para ser visualmente mas atractivas y también para añadir los requisitos que faltaban y no se habían tenido en cuenta, estos eran la imagen del regalo que estuviese seleccionado, así como su descripción y coste. También se redujo el numero de pantallas, donde antes eran 4 las relacionadas con los premios ahora son solamente 2 permitiendo así al usuario tener mas información concentrada, bien separada pero sin ser excesiva y sin tener que memorizar que productos había seleccionado previamente.

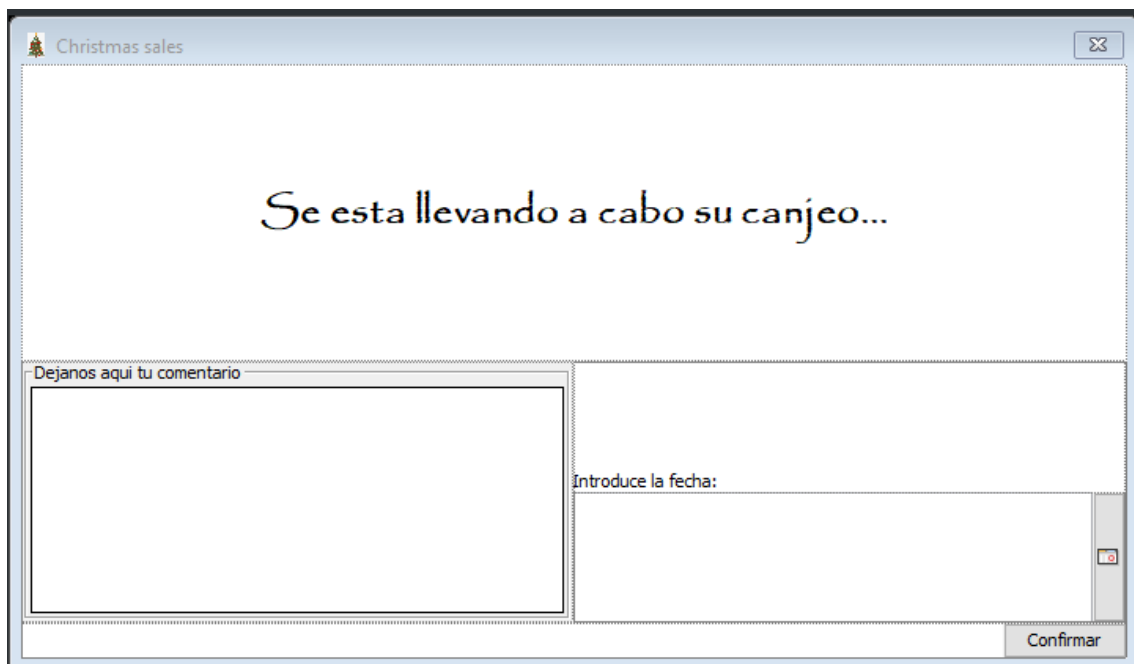


Para la obtención de puntos también se realizó un pequeño cambio, se suprimió un elemento visual que permitía ver que casillas se habían seleccionado, ahora se destapan directamente desvelando que puntos se han obtenido.



The screenshot shows a window titled "Christmas sales". Inside, there is a 5x5 grid of yellow squares, each containing a black question mark. To the right of the grid, the text "Puntos actuales:" is followed by a text input field containing the number "0". Below this is a button labeled "Canjear".

Para canjear los premios se ha modificado ligeramente el diseño original, ahora canjeas directamente tus premios. Solo si tienes en la cesta de productos un artículo de viajes aparecerá un panel que inicialmente está oculto en el que tendrás la opción de añadir comentarios y la fecha.



The screenshot shows a window titled "Christmas sales". The main content area displays the text "Se esta llevando a cabo su canjeo...". Below this, there is a text input area labeled "Dejanos aqui tu comentario". To the right of this is a date input field labeled "Introduce la fecha:". At the bottom right, there is a button labeled "Confirmar".

Fase de implementación

Lógica:

Se procedió a dividir el problema en sencillas clases.

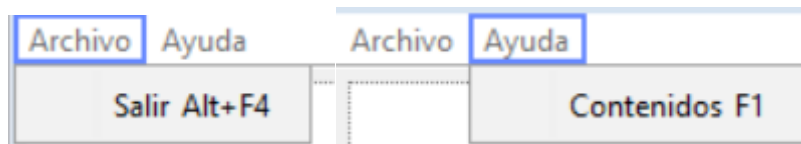
- **Game:** es la clase que contiene toda la estructura principal de la lógica, incluyendo el juego de puntos y los premios y la gestión de estos. También contiene a los clientes que están disponibles y cual está logeado. Los métodos más significativos son cargar los datos de los ficheros de clientes y regalos, también la escritura en fichero en `entregas.dat`. `boolean tieneExperiencias()` nos indica si este tiene algún regalo relacionado con viajes para así poder habilitar en la capa de interfaz un panel. `int identificadorValido()` nos indica a través de un número si el identificador para iniciar sesión es correcto. Dependiendo del tipo de error devolverá un número u otro que será tratado para lanzar un mensaje al cliente (esto último en la capa de interfaz de usuario).
`Void destapar(posicion)` descubre una casilla desencadenando los puntos y actualizaciones de casillas que quedan por ser descubiertas.
`Void añadir()` `void eliminar()` estos métodos reciben un regalo y lo eliminarán o añadirán a la cesta del cliente.
- **Cliente:** tiene los elementos de `identificador; nombre; apellido1; apellido2; puedeJugar`. Se destaca el `toString` para poder guardarse posteriormente el cliente en fichero.
- **Premio:** clase que representa los regalos. Se implementó un comparador para poder ordenarlos por puntos.
- **Tablero:** representa el tablero de casillas de puntos. Destacamos el método `CrearCasillas` que llamara a un método recursivo en el que añadira a posiciones aleatorias la casilla correspondiente de puntos.
- **Casilla:** representa una casilla de puntos en la que se indica los puntos que contiene, si es multiplicador o si es bonus.
- **File útil:** clase abstracta que nos permitirá cargar y guardar todos los ficheros necesarios en la aplicación. `ChangeClientState()` cambia el estado del cliente a no puede jugar para posteriormente ser actualizado en fichero.

Clases receptoras de eventos:

- **EventoClickArticulo:** actualiza los valores en la ventana de selección de artículos para que sea mostrado visualmente en el panel central.
- **EventoClickPanel:** Esta relacionada con el panel de puntos en el que al hacer click a un botón, este actualizará los puntos e imagen y otras actualizaciones.
- **EventoClickRbt:** evento asignado a los radiobotones que se encuentran en el panel de premios para poder filtrarlos por tipos.

Interfaz:

VentanaInicial:



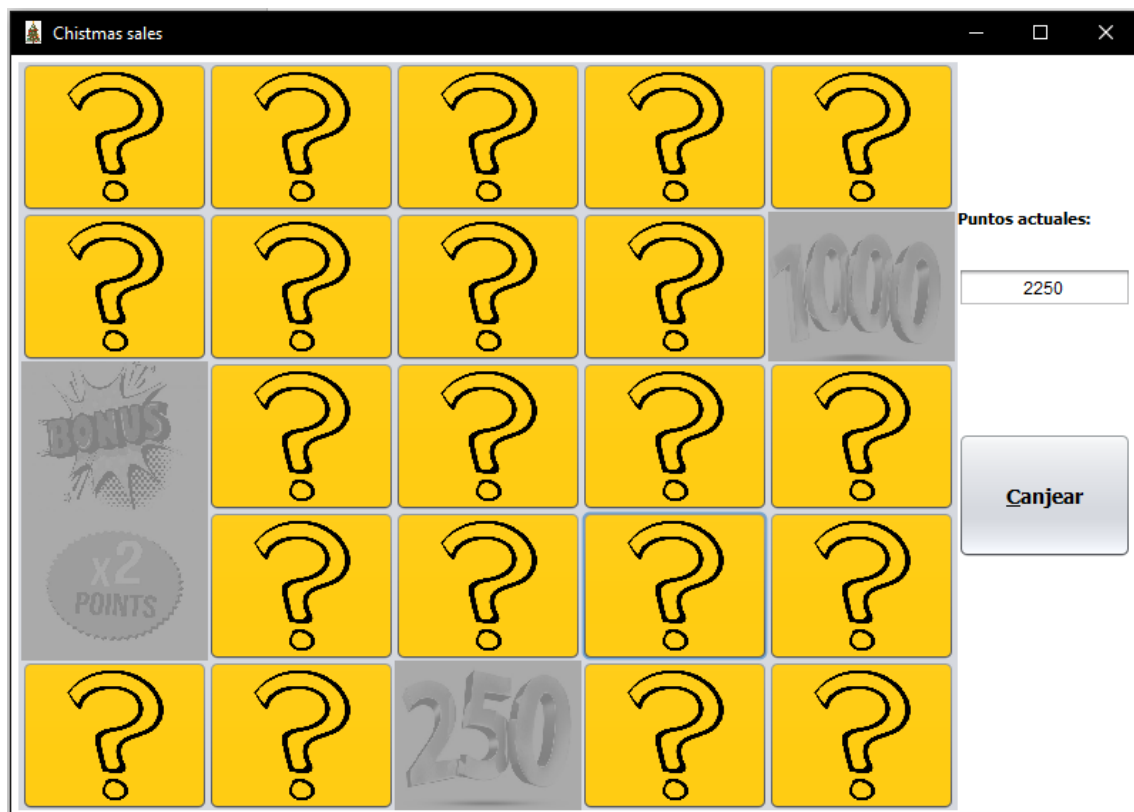
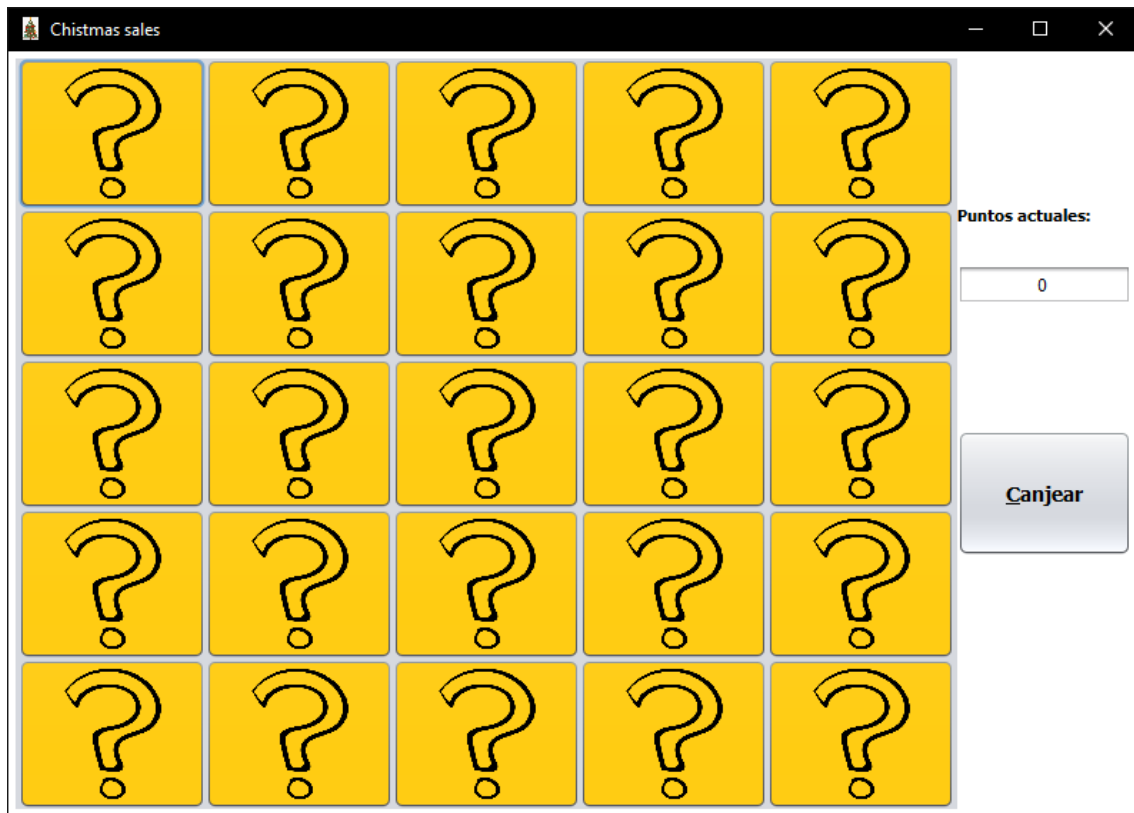
Aquí se encuentra la creación de todas las clases de la lógica y se añadió un menú sencillo con las opciones de salir y ayuda en el que se incorpora todo el sistema de ayuda. No se han añadido otras opciones ya que creo que son irrelevantes para el cliente y no tendrían sentido otras opciones como “Nuevo” o “Ver”.

VentanaInicioSesion: es un Jdialog en que tiene un campo de texto y botones de confirmación.

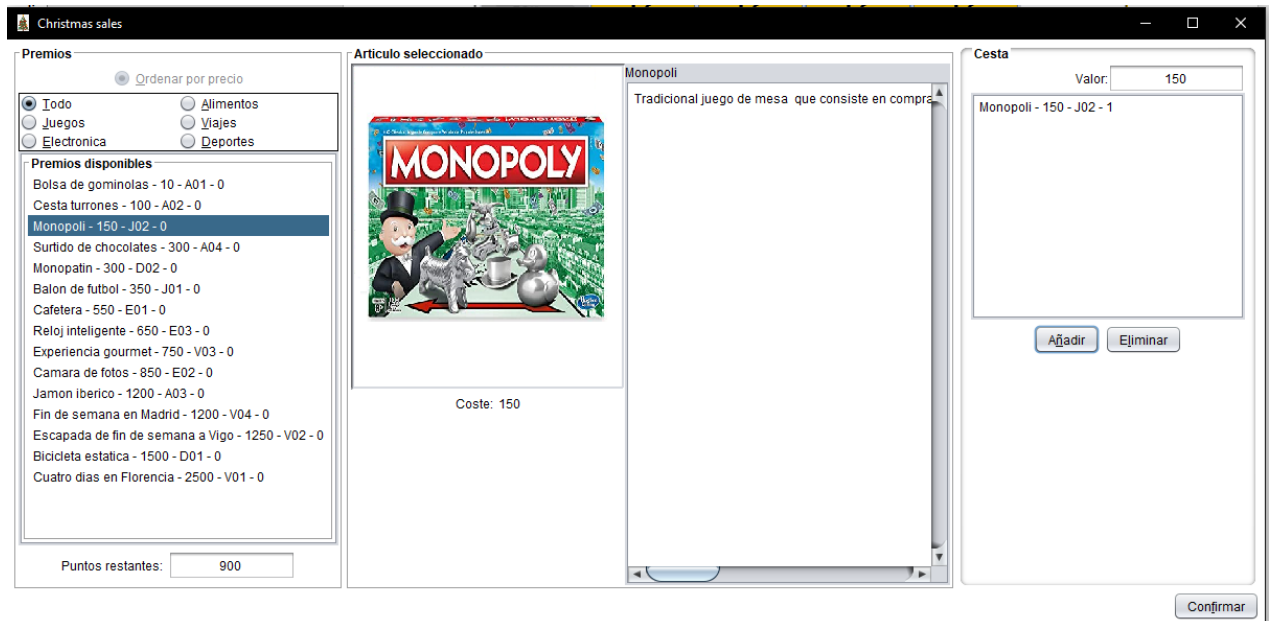


PanelDeJuego:

Se estableció como elemento para las casillas del panel los botones en las que tendrán iconos. Un contador para los puntos conseguidos y un botón para continuar.



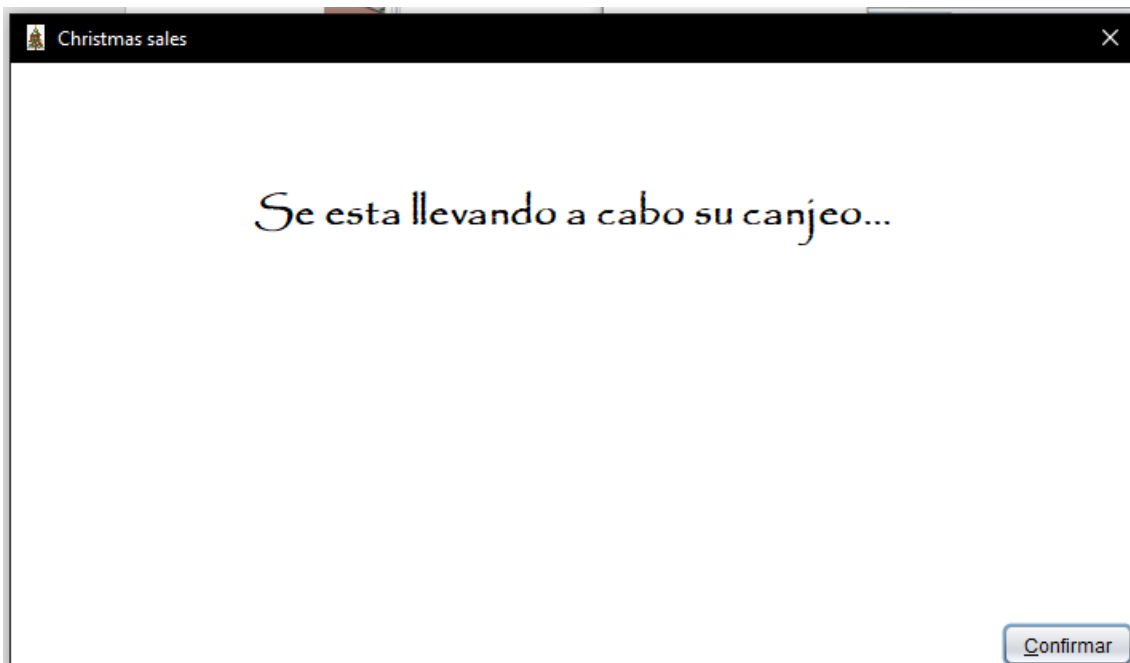
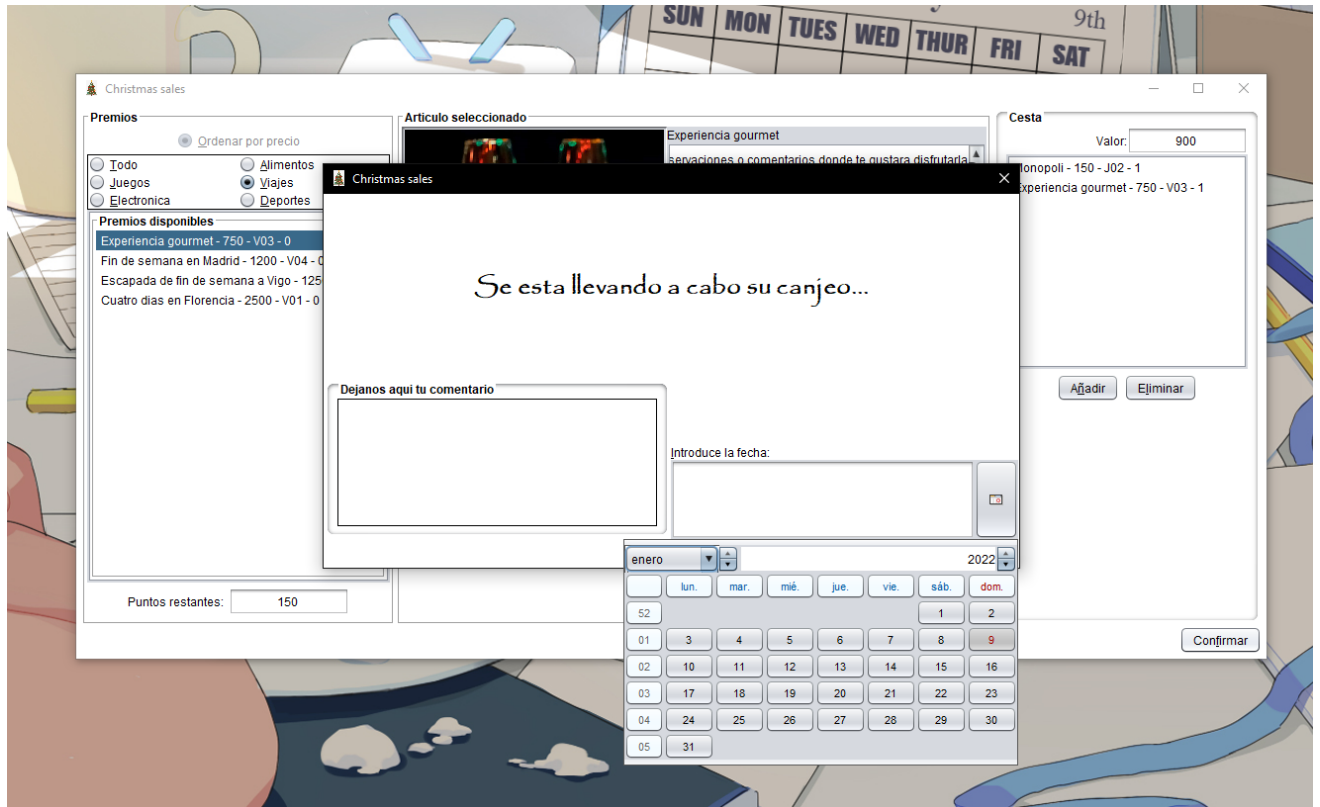
VentanaPremios:



Esta ventana dividida en 3 secciones permitiendo así al usuario tener todo a simple vista (Premios, Artículo seleccionado y la cesta del cliente). En los regalos se distinguen radiobotones agrupados para que solo pueda seleccionarse un elemento para poder filtrarlos y otro radio botón que permite ordenarlos por precio (este se actualiza al seleccionar otro radioboton de filtrado de la Jlist), también en la parte inferior los puntos restantes. En el centro destacamos los datos del artículo en una label tanto su imagen como coste, y en un JText con scrollpane para la descripción de este.

En la cesta de la compra se pueden observar los artículos seleccionados en una Jlist para indicar como se compone el artículo y función de esta pantalla se encuentra en el sistema de ayuda incluido indicando el formato que se siguió para los artículos (Nombre, puntos, identificador y cantidad que se tiene sobre ellos). Dos botones para añadir o eliminar (añadimos desde el elemento seleccionado en los premios disponibles y eliminamos desde los premios de la cesta). Otro botón para confirmar la selección obtenida.

VentanaDeConfirmacion: JDialog



En el caso de canjear experiencias se habilitaran los campos de los comentarios y la fecha(JDateChooser) para la reserva. Siempre se puede editar la cesta hasta que se de al botón de confirmar, que enviara el pedido a entregas.dat y preparara la aplicación para volver a ser usada.

No se han incluido aquí en la bitácora los numerosos mensajes emergentes o advertencias debido a errores del usuario.

Pruebas de usuarios:

Prueba Ana:

Obtiene los 2500 puntos, al destapar la tercera casilla da al botón de canjear, añade correctamente los premios, confirma y se le avisa de que le sobran puntos. Escoge la fecha y observación y confirma. Solo puede escoger una fecha ya que en las especificaciones adjunto imagen del documento:

3.11. En el caso de que uno de los regalos elegidos pertenezca a la categoría de **Viajes/Experiencias**, el cliente deberá indicar la **fecha de inicio** en la que desea disfrutarlo así como indicar alguna **observación o comentario** que considere oportuno. La única comprobación que se debe realizar respecto a la fecha indicada es que ésta sea **posterior a la fecha actual**.

CPM - Módulo de prácticas 21-22

4

De ahí deduje que debido a que contiene al menos una experiencia solo debe indicar una fecha.

Prueba Antonio:

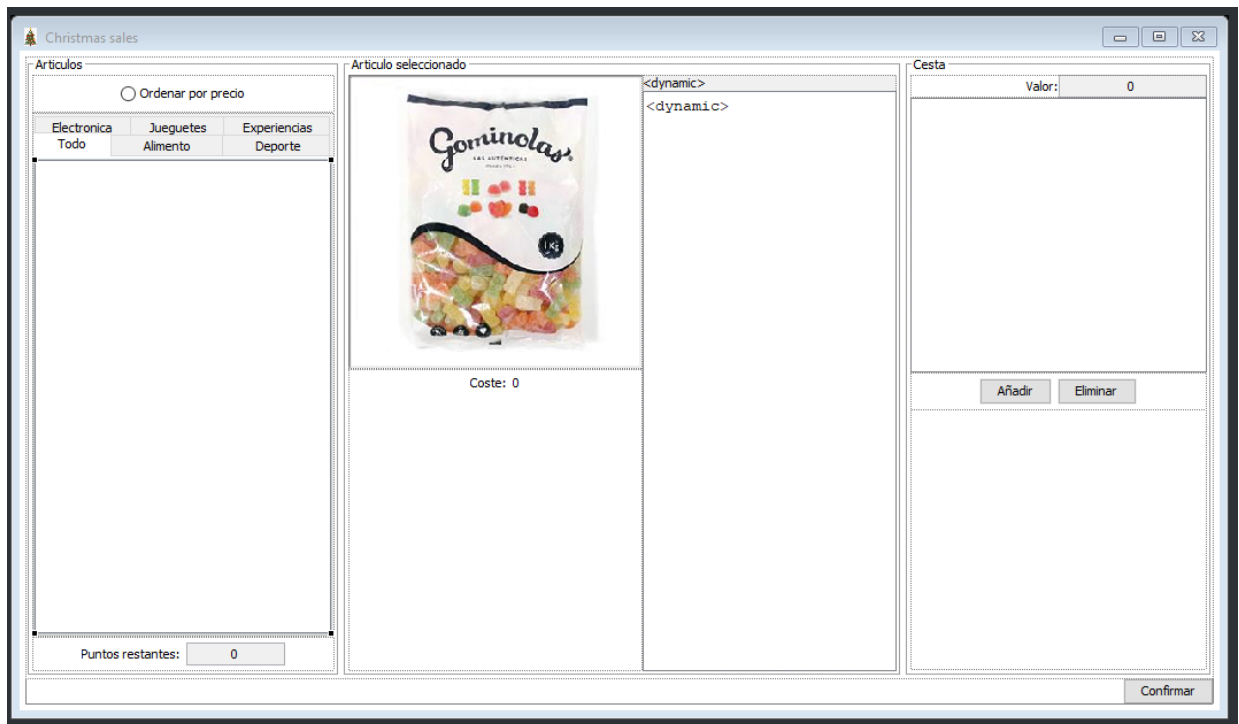
Realiza sin problema todos los pasos visuales ya que esta todo indicado correctamente, al llegar al panel puede distinguir la opción de ordenar por puntos y así le es más fácil distinguir los artículos que quiere conseguir. No se desechan los artículos que tienen más puntos, siguen apareciendo pero ordenados por puntos.

Prueba George:

No se encuentra la opción de idioma ya que no está internacionalizada la aplicación. Al introducir el identificador se muestra un aviso por pantalla en el que indica que debe registrarse antes en el centro comercial para poder hacer la tarjeta de socio. Destapa las casillas sin problema puede observar los puntos u el bonus de forma visual. Selecciona de forma correcta la sección de alimentación y añade ambos artículos (la aplicación le deja), intenta canjear y la aplicación le avisa de que sobrepasa los puntos obtenidos y por tanto debe eliminar algún artículo o cambiar su selección. Finalmente elimina el artículo que no puede llevarse y confirma los premios.

Problemas encontrados

Se quiso rediseñar la ventana de los premios del wireframe v2 ya que este no convencía su aspecto. Y se optó por utilizar el siguiente.



El problema que surgió aquí fue la implementación de los listeners para llevar a cabo la acción de según el elemento(de la JList) seleccionado del tabedpane que apareciesen los datos del premio en la zona central. Se intento hacer una clase externa para gestionar este evento pero al añadirlo y depurar para comprobar su correcto funcionamiento, se llego a la conclusión que no se activaba el listener correspondiente y por tanto no hacia nada.

Adjunto código que se tenía previamente:

```
private JScrollPane getScTodos() {
    if (scTodos == null) {
        scTodos = new JScrollPane();
        scTodos.setViewportView(getListTodo());
    }
    return scTodos;
}

private JList<Premio> getListTodo() {
    if (listTodo == null) {
        listTodo = new JList<Premio>();
        modelTodo = new DefaultListModel<>();
        listTodo.setModel(modelTodo);
        listTodo.addListSelectionListener(eventoClickArticulo);
    }
    return listTodo;
}
```

Este es el listener que añadí a cada uno de ellos para que se actualice el panel del Artículo seleccionado pero no funciona:
¿Quizás es otro evento o quizás se debe añadir el listener en el tabbedPane...?

```
public class EventoClickArticulo implements ListSelectionListener{

    private VentanaPremios vPremios;
    private Game game;

    public EventoClickArticulo(VentanaPremios vPremios) {
        this.vPremios = vPremios;
        this.game = vPremios.getPanelDeJuego().getInicioSesion().getVentanaInicial().getGame();
    }

    /**
     * Metodo para recalcular la imagen
     * @param label
     * @param rutaImagen
     */
    private void setImagenAdaptada(JLabel label, String rutaImagen){
        Image imgOriginal = new ImageIcon(getClass().getResource(rutaImagen)).getImage();
        Image imgEscalada = imgOriginal.getScaledInstance(label.getWidth(), label.getHeight(), Image.SCALE_FAST);
        ImageIcon icon = new ImageIcon(imgEscalada);
        label.setIcon(icon);
    }

    @Override
    public void valueChanged(ListSelectionEvent e) {
        Premio seleccionado = ((JList<Premio>)e.getSource()).getSelectedValue();
        setImagenAdaptada(vPremios.getLbImagen(), "img/"+seleccionado.getCodigo()+".png");
        vPremios.getLbNombreArticulo().setText(seleccionado.getDenominacion());
        vPremios.getTxDescripcion().setText(seleccionado.getDescripcion());
        vPremios.getLbCoste().setText(String.valueOf(seleccionado.getPuntos()));
        vPremios.validate();
        vPremios.repaint();
    }
}
```

Otro problema surgido en esta ventana era:

Otra cuestión a parte es, en el botón añadir ¿Como obtengo la JList del scrollpane en el que me encuentro? Se que para obtener el scrollpane en el que se esta es getTabbedPane().getSelectedComponent() pero no me deja indagar mas...

```
private void añadirACesta(ActionEvent e) {
    JList<Premio> tienda = (JScrollPane)(getTabbedPane().getSelectedComponent()).getComponentAt(0, 0);
    //POR HACER
}

//getComponentAt(Point p): Component - Component
//getComponentAt(int x, int y): Component - Component
//getComponentListeners(): ComponentListener[] - Component
//getComponentOrientation(): ComponentOrientation - Component
```

Finalmente el problema que iba a dar la solución a esta nueva interfaz fue un fracaso y opte por otro tipo de interfaz, esta vez con radio botones y una única lista con su modelo correspondiente para evitar estos problemas. Pero aun así no fue suficiente, seguía habiendo problemas en los listener al llevarlos a clases externas para que el código estuviese mas limpio pero fue imposible, ¡al hacer click en los radiobotones seguía sin entrar en la clase receptora del evento a pesar de estar inicializada en el frame y añadida como listener en los radiobotones! (Para que este funcionase puse actioncomands a los radiobotones para poder

distinguir a cual de ellos se hacia referencia, pero finalmente no fue desechado al no funcionar)

```
public class EventoClickRbt implements ActionListener{

    private VentanaPremios vpremios;


    public EventoClickRbt(VentanaPremios vpremios) {
        this.vpremios=vpremios;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        //Obtenemos que rbtn fue llamado
        JRadioButton radioButton = (JRadioButton)e.getSource();

        //Eliminamos el anterior contenido
        vpremios.getModelArticulos().removeAllElements();

        List<Premio> premios = vpremios.getPanelDeJuego().getInicioSesion().getVentanaInicial().getGame().getPremios();
        for (Premio p : premios) {
            if(radioButton.getActionCommand().equals("T") && p.getSeccion().equals("T")) {
                vpremios.getModelArticulos().addElement(p);
            }else if(radioButton.getActionCommand().equals("A") && p.getSeccion().equals("A")) {
                vpremios.getModelArticulos().addElement(p);
            }else if(radioButton.getActionCommand().equals("D") && p.getSeccion().equals("D")) {
                vpremios.getModelArticulos().addElement(p);
            }else if(radioButton.getActionCommand().equals("J") && p.getSeccion().equals("J")) {
                vpremios.getModelArticulos().addElement(p);
            }else if(radioButton.getActionCommand().equals("V") && p.getSeccion().equals("V")) {
                vpremios.getModelArticulos().addElement(p);
            }else if(radioButton.getActionCommand().equals("E") && p.getSeccion().equals("E")) {
                vpremios.getModelArticulos().addElement(p);
            }
        }
    }
}
```

Christmas sales

Articulos	Articulo seleccionado	Cesta
<div><input type="radio"/> Ordenar por precio</div> <div><input checked="" type="radio"/> Todo <input type="radio"/> Alimentos</div> <div><input type="radio"/> Juegos <input type="radio"/> Viajes</div> <div><input type="radio"/> Electronica <input type="radio"/> Deportes</div>	<div></div> <div>Coste: 0</div>	<div><div><dynamic></div><div><dynamic></div></div> <div>Valor: 0</div> <div><input type="button" value="Añadir"/> <input type="button" value="Eliminar"/></div>

Puntos restantes: 0