



MÓDULO PRÁCTICAS CPM

PANEL DE REGALOS

2021/2022

Documentación del trabajo



23 DE DICIEMBRE DE 2021
MIGUEL GONZÁLEZ NAVARRO
GRUPO PL3 (ESPAÑOL)
UO282337

ÍNDICE

1. INTRODUCCIÓN	2
2. DESARROLLO DE LA APLICACIÓN	3
2.1. FASE DE DISEÑO	3
2.2. FASE DE IMPLEMENTACIÓN	7
2.2.1. LÓGICA.....	7
2.2.2. INTERFAZ.....	17
2.2.3. PRUEBAS	32

1. INTRODUCCIÓN

Todos los años por Navidad, unos conocidos grandes almacenes regalan diversos artículos a sus clientes. Estos han de probar suerte girando las casillas de varios paneles ubicados en las instalaciones de estos grandes almacenes. Este año hemos decidido instalar nuestras propias terminales para sustituir a dichos paneles físicos e implementar una aplicación visual que permita prescindir de los paneles giratorios que se utilizaba hasta ahora. De esta forma, le resultará más cómodo al usuario poder jugar una partida y conseguir algún regalo por Navidad.

En esta campaña, cada vez que un cliente realiza una compra y pasa por la caja la tarjeta de fidelización, entra en un proceso aleatorio que le ofrece la posibilidad de destapar 3 casillas de un panel. Si el cliente resulta agraciado, se podrá acercar a uno de los paneles a probar suerte.

El aspecto inicial de la aplicación es el mostrado a continuación; en esta ventana podemos resumir que hay un campo de texto donde tenemos que introducir nuestro identificador, un botón verde para validarlo en la página y un botón “Siguiente” para avanzar a la siguiente ventana.



Cabe destacar que no se ha internacionalizado (ni localizado) la aplicación (ni se ha incluido cualquier otro elemento que tenga que ser tenido en cuenta a la hora de corregir), por lo que jugar resultará especialmente cómodo para un usuario que hable o entienda el español, pero también consta de elementos visuales claves que pueden ayudar a un usuario con idioma extranjero a desenvolverse con la web.

También se ha habilitado un sistema completo de ayuda (en español) con capturas de pantalla y explicaciones acerca de cada elemento, que pueden ser de gran ayuda para todo tipo de usuarios.

2. DESARROLLO DE LA APLICACIÓN

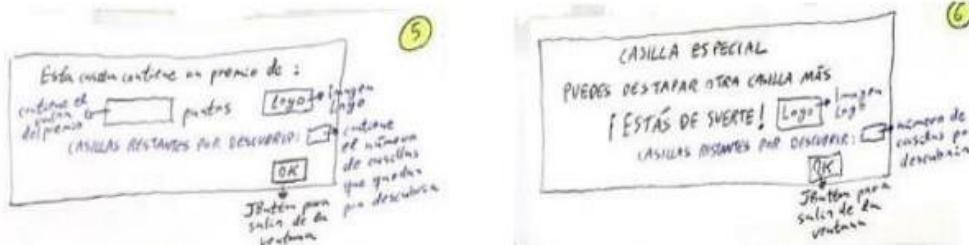
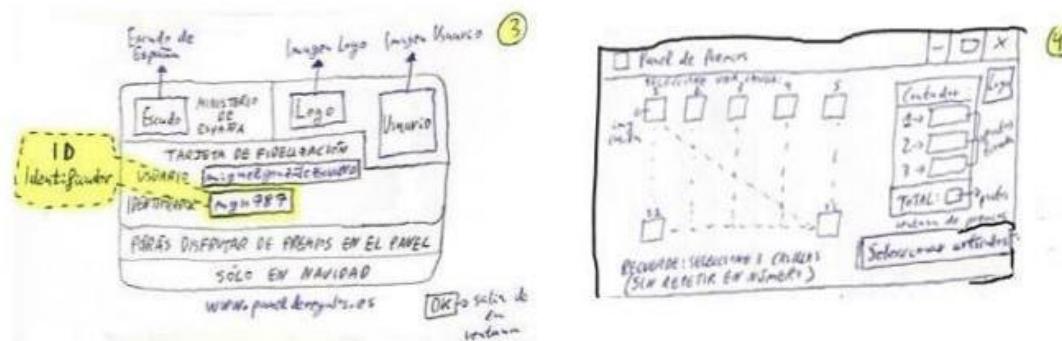
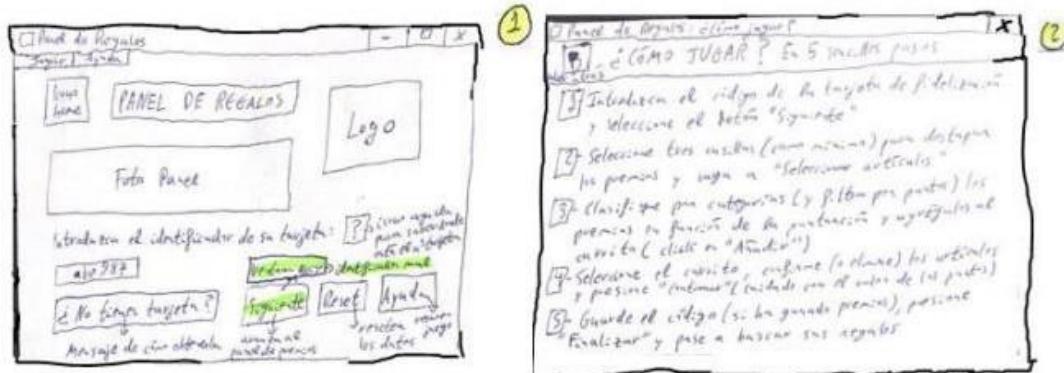
2.1. FASE DE DISEÑO

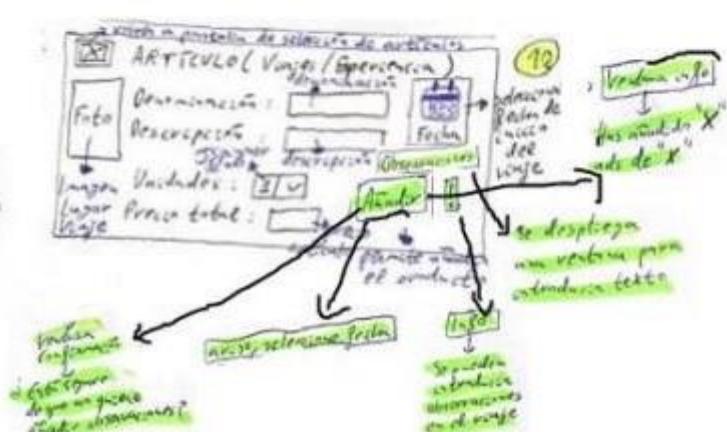
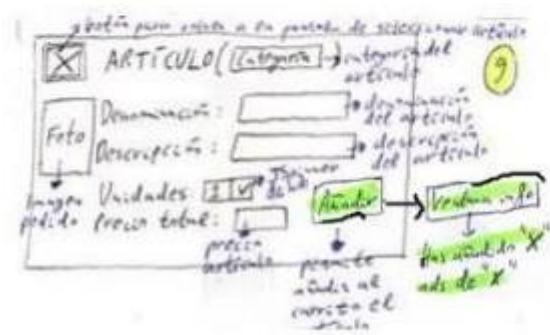
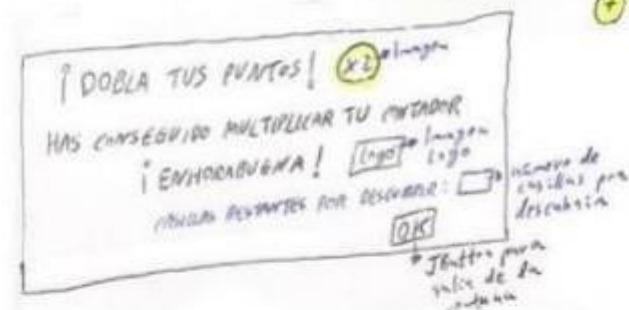
Las pantallas presentadas a continuación están incluidas en el Wireframe V2 presentado en el seminario 3 de la asignatura de Comunicación Persona-Máquina. Se han realizado una serie de pequeños cambios con respecto al diseño final de la aplicación, que son los siguientes:

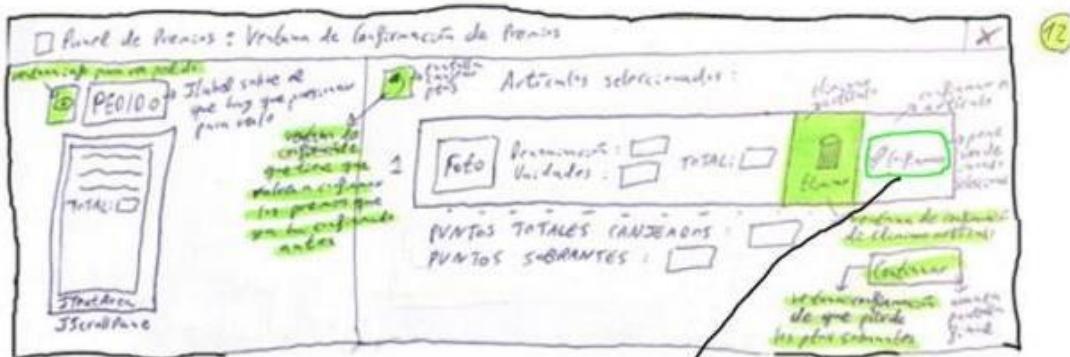
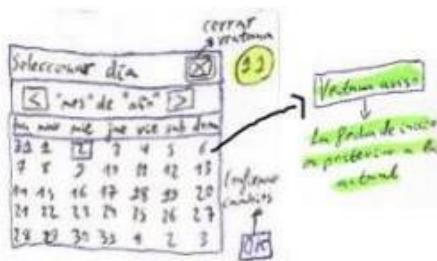
- En la ventana 1 (ventana de registro) se ha cambiado el botón “¿No tienes tarjeta?” por el botón “¿Cómo Jugar?” (este último abrirá otra ventana que presenta un pequeño resumen sobre cómo jugar en 5 sencillos pasos). La causa del cambio ha sido que en el diseño pensaba que la aplicación te ofrecía la posibilidad de obtener tu tarjeta de cliente en el caso de que no lo fueras, pero contamos con identificadores de clientes válidos para probar la aplicación.
- En la ventana 8 (ventana de regalos) se ha añadido una categoría más de regalos, se ha cambiado la distribución en el panel de los regalos y se ha añadido un JSlider para filtrar los regalos por la puntuación que el usuario desee. Distribuir los regalos de esta manera fue debido a que he añadido los regalos en un panel de manera dinámica, se añadió una categoría más debido a que esta (“Juguetes”) no aparecía en la documentación inicial del módulo, y se ha incorporado un JSlider para facilitar el filtrado de puntos al usuario y hacer que la aplicación sea más familiar. También se ha suprimido el menú de arriba.
- En la ventana 13 (ventana de confirmación de regalos) se ha eliminado el código para recoger los premios, ya que en un principio no sabía que la aplicación no se encargaba de enviar los regalos al domicilio del cliente y que no era objeto de este módulo de prácticas.
- Respecto a las ventanas de aviso, en la ventana 10 (compra de un viaje) será obligatorio introducir observaciones para el viaje; por lo tanto, en caso de que no haya, lanzará un aviso en caso de que el usuario quiera añadir este artículo sin observaciones. Y en la ventana 12 (confirmación de entrega) se han suprimido los avisos cuando confirmamos el artículo de falta de puntos, ya que la ventana de compra de regalos (8, y en consecuencia la 9 y la 10 también) se encarga de controlar que el usuario no gaste más puntos de los que tiene en su contador.

WIREFRAME VERSIÓN 2

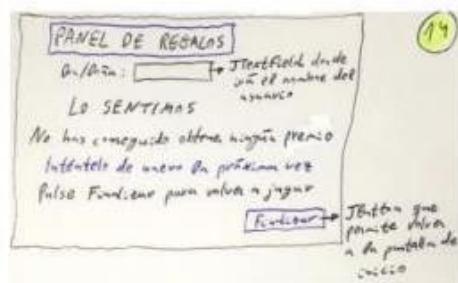
- VENTANAS DE LA APLICACIÓN (1-14)







Ventana de aviso → No puedo canjear los premios debido a la puntuación



2.2. FASE DE IMPLEMENTACIÓN

2.2.1. LÓGICA

1. Clase que controla la partida (casillas, puntuación e intentos)

- **Clase Juego**

La clase Juego es la clase más importante de la lógica del módulo, controla tanto el panel de las casillas que elige el jugador para ganar puntos, los propios puntos que se van acumulando en el contador y los intentos que el usuario dispone para ello.

Por ello contiene un array de casillas (la clase Casilla será explicada a continuación), el número de intentos y el número de puntos del jugador. El número de casillas del panel por defecto es 25, y los intentos 3 (aunque luego se verá que pueden ser más).

```
public class Juego {  
    private Casilla[] casillas;  
    private int intentos;  
    private int puntos;  
  
    public static final int TAM = 25;  
    public static final int INTENTOS = 3;  
  
    public Juego() {  
        casillas = new Casilla[TAM];  
        this.intentos = INTENTOS;  
        this.puntos = 0;  
        rellenarCasillas();  
    }  
}
```

A continuación, el método (rellenarCasillas()) que se encarga de colocar todos los tipos de casilla aleatoriamente por el panel. En primer lugar, rellena todo el array (rellenarSinPremio()) con casillas que no tienen premio (dan 0 puntos si te tocan); y luego va distribuyendo las casillas de 50, 250 y 1000 puntos; y las especiales y las que doblan la puntuación (colocaTipoCasilla(numeroCasillas, tipo)).

```
private void rellenarCasillas() {  
    llenarSinPremio();  
    colocaTipoCasilla(1, new Casilla(1000));  
    colocaTipoCasilla(5, new Casilla(250));  
    colocaTipoCasilla(8, new Casilla(50));  
    colocaTipoCasilla(1, new CasillaX2Puntos(-1));  
    colocaTipoCasilla(2, new CasillaEspecial(-1));  
}  
  
private void llenarSinPremio() {  
    for(int i=0; i < TAM; i++) {  
        casillas[i] = new Casilla(0);  
    }  
}
```

```

private void colocaTipoCasilla(int numeroCasillas, Casilla casilla) {
    int posicionCasilla;
    for (int i = 0; i < numeroCasillas; i++) {
        do {
            posicionCasilla = lanzar(TAM);
        } while (! (casillas[posicionCasilla].getPuntos() == 0));
        casillas[posicionCasilla] = casilla;
    }
}

```

El método lanzar simplemente devuelve un valor aleatorio entre 0 y el recibido por parámetro, hacemos uso del método de la librería Math (random()).

```

public int lanzar(int max) {
    return ((int) (Math.random() * max));
}

```

El último método de la clase se encarga de seleccionar una casilla, simula que el usuario ha elegido la casilla “x” del panel de regalos. Por lo que tendrá que gestionar los intentos que le quedan al usuario y modificar el contador de puntos según el tipo de casilla.

```

public void seleccionaCasilla(int i){
    intentos--;
    if (casillas[i] instanceof CasillaX2Puntos) {
        this.puntos = puntos * 2;
        return;
    } else if (casillas[i] instanceof CasillaEspecial) {
        intentos++;
        return;
    }
    puntos += casillas[i].getPuntos();
}

```

2. Clases para distribuir los elementos del panel de casillas

- **Clase Casilla**

La clase Casilla (definida como abstracta) representa cada una de las casillas del panel de la aplicación, y que puede ser de diferentes tipos: aquellas que contienen puntos, las especiales y las que doblan el contador de los puntos.

```

public class Casilla {
    private int puntos;

    public Casilla(int puntos) {
        this.puntos = puntos;
    }

    public int getPuntos() {
        return puntos;
    }

    public String toString() {
        return "Premio : " + getPuntos() + " puntos";
    }
}

```

- **Clase CasillaEspecial**

La clase hereda de la clase Casilla, ya que es un tipo de casilla del panel. Por lo que cuando el usuario seleccione la casilla y vea si la casilla es una instancia de esta clase, podrá aumentar el número de intentos en una unidad.

```
public class CasillaEspecial extends Casilla {  
  
    public CasillaEspecial(int puntos) {  
        super(puntos);  
    }  
  
    public String toString() {  
        return "Premio : " + "Casilla Especial";  
    }  
}
```

- **Clase CasillaX2Puntos**

La funcionalidad de esta clase es similar a la anterior, la única diferencia es que este tipo de casilla dobla los puntos del contador.

```
public class CasillaX2Puntos extends Casilla {  
  
    public CasillaX2Puntos(int puntos) {  
        super(puntos);  
    }  
  
    public String toString() {  
        return "Premio : " + "Puntos X2";  
    }  
}
```

3. Clases para presentar los regalos

- **Clase Regalo**

La clase Regalo tiene una serie de propiedades que definen a cada objeto del fichero “regalos.dat”, por lo que tendrá los siguientes atributos.

```
public class Regalo {  
    private String codigo;  
    private char seccion;  
    private String denominacion;  
    private String descripcion;  
    private int puntos;  
  
    public Regalo(String codigo, char seccion, String denominacion, String descripcion, int puntos) {  
        this.codigo = codigo;  
        this.seccion = seccion;  
        this.denominacion = denominacion;  
        this.descripcion = descripcion;  
        this.puntos = puntos;  
    }  
}
```

- **Clase Regalos**

Esta clase se encarga de cargar los regalos del fichero “regalos.dat” en una lista, también tiene otras propiedades como una listaRegalosTipo y una listaRegalosFiltroPrecio que nos ayudarán a gestionar los regalos en la aplicación.

```
public class Regalos {  
    private static final String FICHERO_REGALOS = "files/regalos.dat";  
    private List<Regalo> listaRegalos = null;  
    private List<Regalo> listaRegalosTipo = new ArrayList<Regalo>();  
    private List<Regalo> listaRegalosFiltroPrecio = new ArrayList<Regalo>();  
  
    public List<Regalo> getListaRegalos() {  
        return listaRegalos;  
    }  
  
    /**
     * Constructor sin parámetros, crea una lista de regalos y carga los usuarios del fichero "regalos.dat"
     */  
    public Regalos(){  
        listaRegalos = new ArrayList<Regalo>();  
        cargarRegalos();  
    }  
  
    /**
     * Carga los regalos del fichero en la lista
     */  
    private void cargarRegalos(){  
        FileUtil.loadFileRegalos(FICHERO_REGALOS, listaRegalos);  
    }  
}
```

El método getRegalos(tipo, total) se encarga de guardar en una lista aquellos regalos que tengan esa categoría y que su precio (de una unidad) sea menor o igual que el total, el método filtrarRegalosPrecio(total) solamente filtra los regalos respecto al precio por unidad como se hizo en el anterior método.

```
public List<Regalo> getRegalos(String tipo, int total) {  
    listaRegalosTipo.clear();  
    for (Regalo regalo : listaRegalos) {  
        if (regalo.getCodigo().contains(tipo) && regalo.getPuntos() <= total) {  
            this.listaRegalosTipo.add(regalo);  
        }  
    }  
    return listaRegalosTipo;  
}  
  
public List<Regalo> filtrarRegalosPrecio(int total) {  
    listaRegalosFiltroPrecio.clear();  
    for (Regalo regalo : listaRegalos) {  
        if (regalo.getPuntos() <= total) {  
            listaRegalosFiltroPrecio.add(regalo);  
        }  
    }  
    return listaRegalosFiltroPrecio;  
}
```

4. Clases para comprar los regalos

- **Clase Entrega**

La utilidad de esta clase es crear objetos que el usuario ha escogido respecto a su puntuación en el panel de casillas, y de alguna manera almacenar esa información para guardarla en un fichero de salida. Las propiedades del objeto Entrega son: el identificador del cliente, el código del regalo, la fecha, las observaciones, unidades, precio, nombre y la más importante, un booleano que indica si el artículo está confirmado o no.

```
public class Entrega {  
    private String idCliente;  
    private String codigo;  
    private String fecha;  
    private String observaciones;  
    private int unidades;  
    private int precio;  
    private String nombre;  
    private boolean confirmado;  
  
    public Entrega(String idCliente, String codigo, String fecha, String observaciones, int precio, String nombre) {  
        this.idCliente = idCliente;  
        this.codigo = codigo;  
        this.fecha = fecha;  
        this.observaciones = observaciones;  
        this.precio = precio;  
        this.nombre = nombre;  
        this.confirmado = false;  
    }  
}
```

El método `toString()` presenta la información del entrega para guardarla en el fichero “entregas.dat” y el `toStringPedido()` para mostrarla en el pedido de la ventana de confirmación de los pedidos.

```
@Override  
public String toString() {  
    if (fecha == null && observaciones == null) {  
        return idCliente + "@" + codigo;  
    }  
    return idCliente + "@" + codigo + "@" + fecha + "@"  
        + observaciones;  
}  
  
public String toStringPedido() {  
    if (fecha == null) {  
        return nombre + " - " + precio + " puntos - (Uds: " + unidades + ")";  
    }  
    return nombre + " - " + precio + " puntos (Uds: " + unidades + ")" + " Fecha : " + fecha;  
}
```

- **Clase Entregas**

La función de la clase es almacenar los puntos que el usuario ha ganado en el panel de casillas, su nombre, identificador y una lista de las entregas que el usuario ha escogido en función de la puntuación.

```
public class Entregas {
    // atributos de la clase
    private List<Entrega> listaEntregas = null; // lista de entregas
    private String id = null; // id del usuario
    private String nombre = null; // nombre del usuario
    private int puntosJuego = 0; // puntos del usuario en el juego
    private int puntosEntregas = 0; // puntos del usuario según sus entregas

    /**
     * Crea una lista de entregas y borra los artículos anteriores
     */
    public Entregas(){
        listaEntregas = new ArrayList<Entrega>();
    }

    /**
     * Borra las entregas
     */
    public void inicializar(){
        listaEntregas.clear();
        this.id = null;
        this.nombre = null;
        this.puntosEntregas = 0;
        this.puntosJuego = 0;
    }
}
```

Los métodos más importantes de la clase y aquellos relacionados con los regalos que escoge el usuario son el add(entrega, unidades) y el remove(entrega, unidades). El primero añade a la lista las unidades de esa entrega (si tiene suficientes puntos) y retorna true, false si no tiene los suficientes puntos. El remove elimina de la lista las unidades de esa entrega si esta se encuentra en el pedido (en la lista de entregas).

```
public boolean add(Entrega entrega, int unidades){
    if (sePuedeComprar(entrega, unidades)) {
        Entrega entregaPedido = null;

        for (Entrega a : listaEntregas){
            if (a.getCodigo().equals(entrega.getCodigo())){
                entregaPedido = a;
                entregaPedido.setUnidades(entregaPedido.getUnidades() + unidades);
                puntosEntregas = puntosEntregas - entregaPedido.getPrecio() * unidades;
                break;
            }
        }

        if (entregaPedido == null){
            Entrega entregaAPedido = new Entrega(entrega);
            entregaAPedido.setUnidades(unidades);
            listaEntregas.add(entregaAPedido);
            puntosEntregas = puntosEntregas - entregaAPedido.getPrecio() * unidades;
        }

        return true;
    } else {
        return false;
    }
}

public boolean sePuedeComprar(Entrega entrega, int unidades) {
    return entrega.getPrecio() * unidades <= puntosEntregas;
}
```

```

public void remove(Entrega entrega, int unidades){
    Entrega entregaPedido = null;
    for (Entrega a : listaEntregas){
        if (a.getCodigo().equals(entrega.getCodigo())) {
            entregaPedido = a;
        }
    }
    if (entregaPedido != null) {
        if (entregaPedido.getUnidades() > unidades) {
            entregaPedido.setUnidades(entregaPedido.getUnidades() - unidades);
            puntosEntregas = puntosEntregas + entregaPedido.getPrecio() * unidades;
        } else {
            listaEntregas.remove(entregaPedido);
            puntosEntregas = puntosEntregas + entregaPedido.getPrecio() * entregaPedido.getUnidades();
        }
    }
}

```

El método `getTotal()` devuelve el precio total de todos los artículos confirmados de la lista de entregas, el `grabarPedido()` guarda en un fichero (en disco) la información de cada una de las entregas confirmadas de la lista, utiliza el método `toString()` que ha sido redefinido según se ha indicado en la documentación del módulo de prácticas.

```

/**
 * Devuelve el precio total de todos las entregas del pedido
 * @return precio
 */
public int getTotal() {
    int precio = 0;
    for (Entrega a : listaEntregas){
        if (a.isConfirmado()) {
            precio += a.getPrecio()* a.getUnidades();
        }
    }
    return precio;
}

/**
 * Guarda en un fichero de texto las entregas
 */
public void grabarPedido() {
    List<Entrega> listaEntregasConfirmadas = new ArrayList<Entrega>();
    for (int i=0; i < listaEntregas.size(); i++) {
        if (listaEntregas.get(i).isConfirmado()) {
            listaEntregasConfirmadas.add(listaEntregas.get(i));
        }
    }
    FileUtil.saveToFile("entregas", listaEntregasConfirmadas);
}

/**
 * Devuelve el método toString de cada artículo del pedido
 * @return cadena, de tipo String
 */
public String toString() {
    String result = "";
    for(int i=0;i<listaEntregas.size();i++) {
        if (listaEntregas.get(i).isConfirmado()) {
            result = result + listaEntregas.get(i).toStringPedido() + "\n";
        }
    }
    return result;
}

```

5. Clases para registrar al usuario

- **Clase Usuario**

La clase Usuario contiene las propiedades que vienen en el fichero de texto: el identificador del cliente, su nombre y apellidos, y un parámetro entero (0 o 1) que indica si un usuario tiene derecho a jugar o no.

```
public class Usuario {  
    private String idCliente;  
    private String nombreYApellidos;  
    private int puedeJugar; // 0 si no puede - 1 si puede
```

El único método relevante es el que indica que un usuario ya no puede jugar, puesto que ya se ha registrado, por lo que se pone un 0 en el atributo puedeJugar.

```
public void actualizarJugador() {  
    this.puedeJugar = 0;  
}
```

- **Clase Usuarios**

La clase Usuarios se encarga de leer los usuarios del fichero “clientes.dat” y los almacena en una lista. También tiene que comprobar si un identificador existe en la lista de usuarios de la aplicación y si ese usuario puede jugar o no. El último método (actualizaFichero()) se encarga de actualizar el fichero de “usuarios.dat”, para que el jugador que se ha registrado no pueda volver a jugar.

```
public class Usuarios {  
    private static final String FICHERO_USUARIOS = "files/clientes.dat";  
    private List<Usuario> listaUsuarios = null;  
  
    /**  
     * Constructor sin parámetros, crea una lista de usuarios y carga los usuarios del  
     */  
    public Usuarios(){  
        listaUsuarios = new ArrayList<Usuario>();  
        cargarUsuarios();  
    }  
  
    /**  
     * Carga los usuarios del fichero en la lista  
     */  
    private void cargarUsuarios(){  
        FileUtil.loadFileUsuarios(FICHERO_USUARIOS, listaUsuarios);  
    }
```

```
public boolean compruebaIdUsuario(String idCliente) {  
    for (int i=0; i < listaUsuarios.size(); i++) {  
        if (idCliente.equals(listaUsuarios.get(i).getIdCliente())) {  
            return true;  
        }  
    }  
    return false;  
}  
  
public boolean compruebaPuedeJugarUsuario(String idCliente) {  
    if (compruebaIdUsuario(idCliente)==false) { return false; }  
    for (int i=0; i < listaUsuarios.size(); i++) {  
        if (idCliente.equals(listaUsuarios.get(i).getIdCliente())) {  
            if (listaUsuarios.get(i).getPuedeJugar()==1) {  
                listaUsuarios.get(i).actualizarJugador();  
                return true;  
            } return false;  
        }  
    }  
    return false;  
}  
  
public void actualizaFichero() {  
    FileUtil.saveToFileUsuarios(FICHERO_USUARIOS, listaUsuarios);  
}
```

6. Clase que se encarga el manejo de ficheros

- **Clase FileUtil**

La clase FileUtil definida como abstracta, simplemente se encarga de gestionar tanto la lectura de ficheros como la escritura en ellos (serán de tipo .dat). Los ficheros que se gestionan son los siguientes:

- clientes.dat -> contienen los datos de los clientes que pueden jugar
- regalos.dat -> presenta los regalos que se pueden adquirir
- entregas.dat -> guarda la entrega que se tiene que realizar al usuario

La clase FileUtil se encarga de la lectura de los dos primeros ficheros, y los guarda en una lista de tipo Usuario y en otra de tipo Regalo, respectivamente. Respecto al penúltimo se encarga de escribir en un fichero las propiedades de cada uno de los objetos de tipo Entrega de la lista que se le pasa por parámetro. Es decir, va escribiendo en el fichero las entregas de todos los usuarios.

Y el último se ocupa de actualizar el fichero de “clientes.dat”, para que cuando un jugador se ha registrado no pueda volver a jugar (se actualiza en memoria y en disco) poniendo un 0 en el campo puedeJugar.

```

public static void loadFileUsuarios(String nombreFicheroEntrada, List<Usuario> listaUsuarios) {
    String linea;
    String[] datosUsuarios= null;

    try {
        BufferedReader fichero = new BufferedReader(new FileReader(nombreFicheroEntrada));
        while (fichero.ready()) {
            linea = fichero.readLine();
            datosUsuarios = linea.split("@");
            listaUsuarios.add(new Usuario(datosUsuarios[0],datosUsuarios[1], Integer.parseInt(datosUsuarios[2])));
        }
        fichero.close();
    }
    catch (FileNotFoundException fnfe) {
        System.out.println("El archivo no se ha encontrado.");
    }
    catch (IOException ioe) {
        new RuntimeException("Error de entrada/salida.");
    }
}
/** 
 * Guarda en un fichero con nombre "nombreFicheroSalida" los usuarios de la lista
 * @param nombreFicheroSalida
 * @param listaUsuarios
 */
public static void saveToFileUsuarios(String nombreFicheroSalida, List<Usuario> listaUsuarios){
    try {
        BufferedWriter fichero;
        fichero = new BufferedWriter(new FileWriter(nombreFicheroSalida));
        String linea = "";
        for (int i=0; i < listaUsuarios.size(); i++) {
            linea += listaUsuarios.get(i).toString() + "\n";
        }
        fichero.append(linea.substring(0, linea.length()-1));
        fichero.close();
    }

    catch (FileNotFoundException fnfe) {
        System.out.println("El archivo no se ha podido guardar");
    }
    catch (IOException ioe) {
        new RuntimeException("Error de entrada/salida");
    }
}

public static void saveToFileEntregas(String nombreFicheroSalida, List<Entrega> listaEntregas){
    try {
        BufferedWriter fichero;
        fichero = new BufferedWriter(new FileWriter("files/" + nombreFicheroSalida + ".dat", true));
        String linea = "";
        for (int i=0; i < listaEntregas.size(); i++) {
            linea += listaEntregas.get(i).toString() + "\n";
        }
        fichero.append(linea);
        fichero.close();
    }

    catch (FileNotFoundException fnfe) {
        System.out.println("El archivo no se ha podido guardar");
    }
    catch (IOException ioe) {
        new RuntimeException("Error de entrada/salida");
    }
}

public static void loadFileRegalos(String nombreFicheroEntrada, List<Regalo> listaRegalos) {
    String linea;
    String[] datosRegalos= null;

    try {
        BufferedReader fichero = new BufferedReader(new FileReader(nombreFicheroEntrada));
        while (fichero.ready()) {
            linea = fichero.readLine();
            datosRegalos = linea.split("@");
            listaRegalos.add(new Regalo(datosRegalos[0],datosRegalos[1].charAt(0),datosRegalos[2],datosRegalos[3],Integer.parseInt(datosRegalos[4])));
        }
        fichero.close();
    }
    catch (FileNotFoundException fnfe) {
        System.out.println("El archivo no se ha encontrado.");
    }
    catch (IOException ioe) {
        new RuntimeException("Error de entrada/salida.");
    }
}

```

2.2.2. INTERFAZ

El conjunto de pantallas (ventanas y/o paneles) que conforman la aplicación son las siguientes (situadas en el paquete `igu` del proyecto):

- > `PanelEntrega.java`
- > `VentanaCalendario.java`
- > `VentanaComoJugar.java`
- > `VentanaConfirmacionPremios.java`
- > `VentanaConfirmacionSinPremios.java`
- > `VentanaPanelCasillas.java`
- > `VentanaPanelEntregas.java`
- > `VentanaPanelRegalos.java`
- > `VentanaRegistro.java`
- > `VentanaSeleccionArticulo.java`
- > `VentanaSeleccionViajes.java`
- > `VentanaTarjetaFidelizacion.java`

A continuación, serán detallados los componentes más relevantes de cada una de las pantallas, así como la justificación acerca de la elección de cada uno.

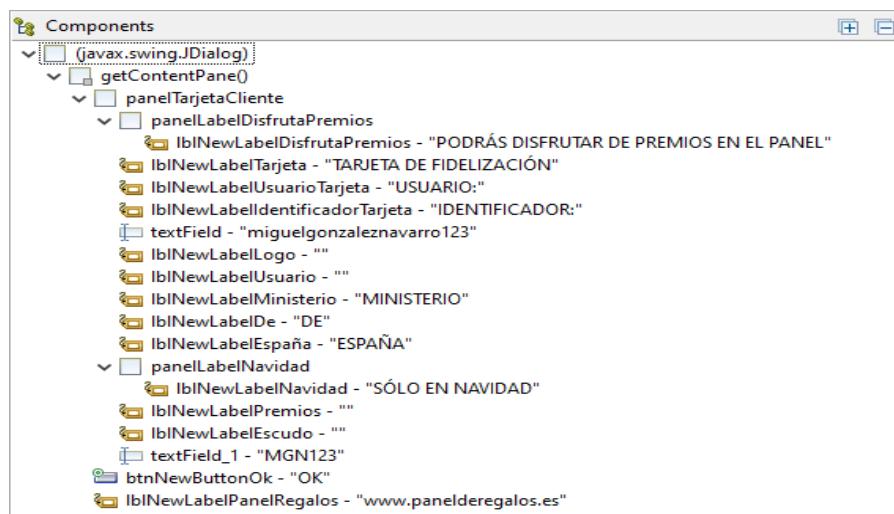
1. VentanaTarjetaFidelizacion: Es un `JDialog`, se despliega cuando se selecciona el botón de la `VentanaRegistro` del interrogante y del botón del menú (de la misma ventana) de localizar el identificador de la tarjeta (Jugar > Localiza tu identificador).

La funcionalidad de esta ventana es presentar una captura del reverso de la tarjeta de cliente, para que el usuario pueda localizar el identificador de la suya, y así poder registrarse en la web y jugar la partida.



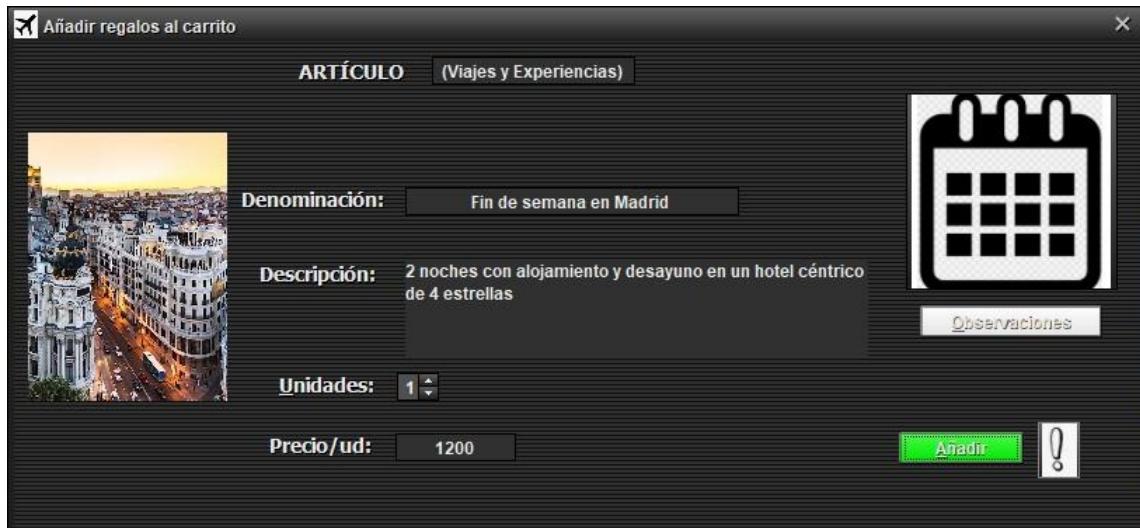
Los componentes que utiliza esta ventana son:

- Un panel principal (absolute layout) que contiene a su vez otro panel (la tarjeta), un botón “OK” que permite cerrar la ventana y un JLabel (link de la web).
 - El panel de la tarjeta tiene otros dos paneles (absolute layout) que contienen (cada uno una etiqueta) las etiquetas de texto “PODRÁS DISFRUTAR DE PREMIOS EN EL PANEL” y “SÓLO EN NAVIDAD”, respectivamente.
 - El resto de los componentes del panelTarjetaCliente son JLabel y JTextField. Los JTextField se utilizaron para mostrar un usuario y un identificador modelo para que el cliente pueda visualizar sus datos en el reverso de su tarjeta. Los JLabel se utilizaron para poner texto o para poner imágenes, las imágenes que se utilizaron son la de los premios, el logo de la aplicación (logo del regalo), el escudo de España y el usuario; y las etiquetas que se utilizaron son los títulos del usuario, el identificador, la tarjeta de fidelización y el texto de Ministerio de España.



2. VentanaSeleccionViajes: Se trata de un JDialog que se despliega cuando seleccionamos un artículo de categoría “Viajes y Experiencias” en la ventana denominada VentanaPanelRegalos, su utilidad es añadir un viaje al carro de la compra de la ventana mencionada anteriormente.

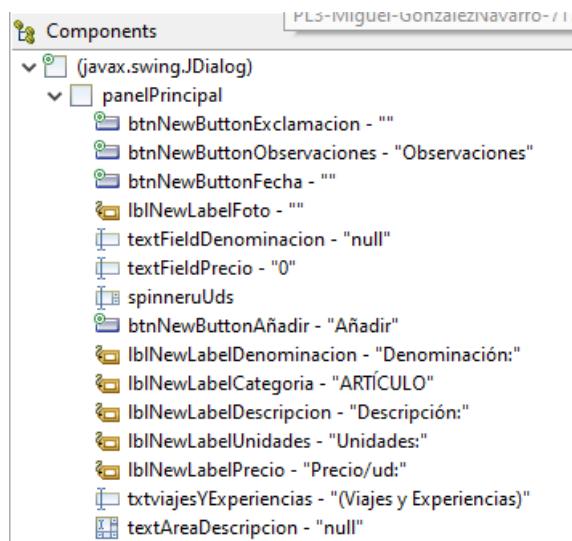




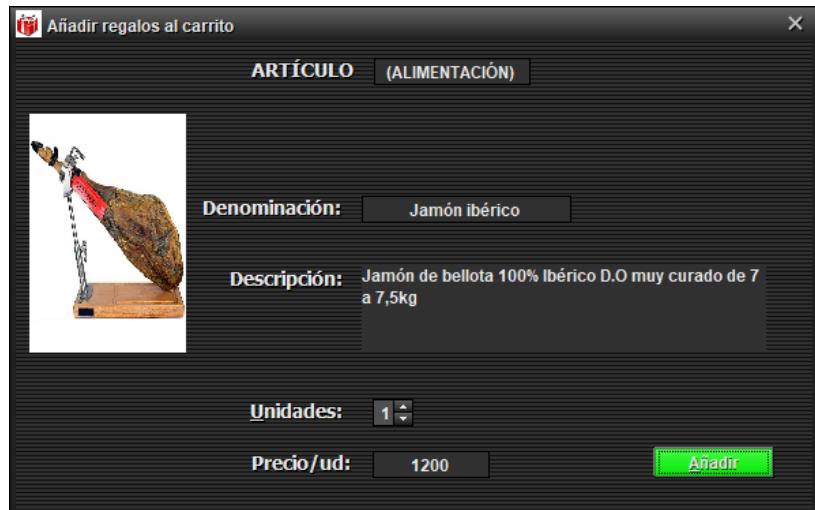
Tiene los siguientes componentes:

El panel principal (absolute layout) que contiene:

- Contiene cuatro botones: el botón “Añadir” (para añadir las unidades de ese viaje al carro, siempre y cuando haya introducido las observaciones y la fecha), el botón “Observaciones” (se despliega una ventana para introducir el texto correspondiente acerca del viaje), el de exclamación (saca una ventana de información acerca de cómo añadir un viaje) y el que tiene una imagen del calendario (se crea una ventana donde el usuario tiene que confirmar la fecha de inicio del viaje en un calendario real).
- Tiene 3 campos de texto (el primero para mostrar la categoría del artículo, el segundo para la denominación del artículo y el último para el precio por unidad), un JTextArea (descripción del artículo) y un JSpinner para indicar las unidades del viaje.
- Los demás elementos son etiquetas de texto (JLabel) que sirven como información acerca de cada uno de los campos de texto, ya sean JTextField o JTextArea. El último JLabel sería aquel que contiene la imagen del artículo.



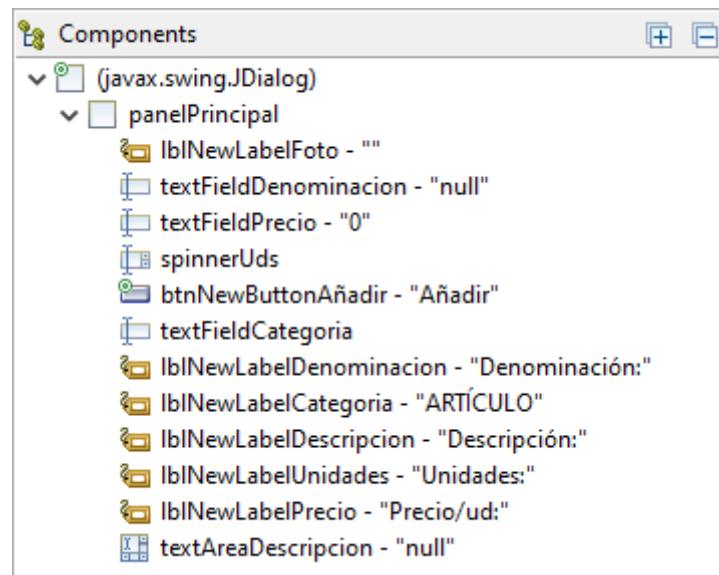
3. Ventana Selección Artículo: Se trata de un JDialog que se despliega cuando seleccionamos un artículo de categoría que no sea “Viajes y Experiencias” en la ventana denominada VentanaPanelRegalos, su utilidad es añadir un artículo al carro de la compra de la ventana mencionada anteriormente.



Tiene los siguientes componentes:

El panel principal (absolute layout) que contiene:

- Contiene un botón: el botón “Añadir” (para añadir las unidades de ese artículo al carro).
 - Tiene 3 campos de texto (el primero para mostrar la categoría del artículo, el segundo para la denominación del artículo y el último para el precio por unidad), un JTextArea (descripción del artículo) y un JSpinner para indicar las unidades del viaje).
 - Los demás elementos son etiquetas de texto (JLabel) que sirven como información acerca de cada uno de los campos de texto, ya sean JTextField o JTextArea. El último JLabel sería aquel que contiene la imagen del artículo.



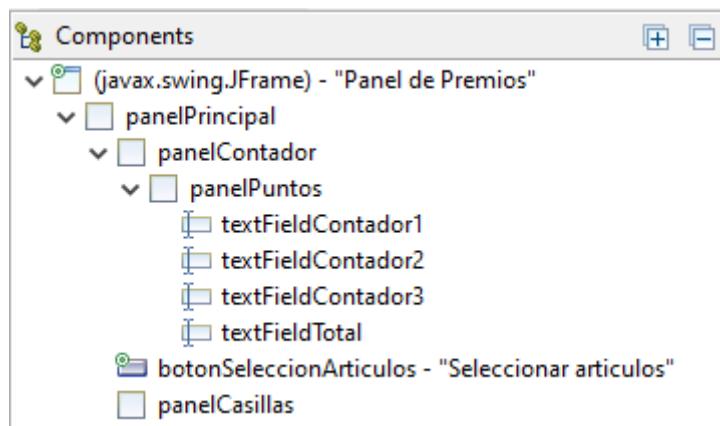
4. VentanaPanelCasillas: Es un JFrame que se ejecuta cuando el usuario se registra en la VentanaRegistro y pasa a la siguiente ventana, su utilidad es hacer que el usuario interactúe con el panel de las casillas, de manera que pueda ir eligiendo casillas y destapando premios para ganar puntos (y canjearlos en regalos).



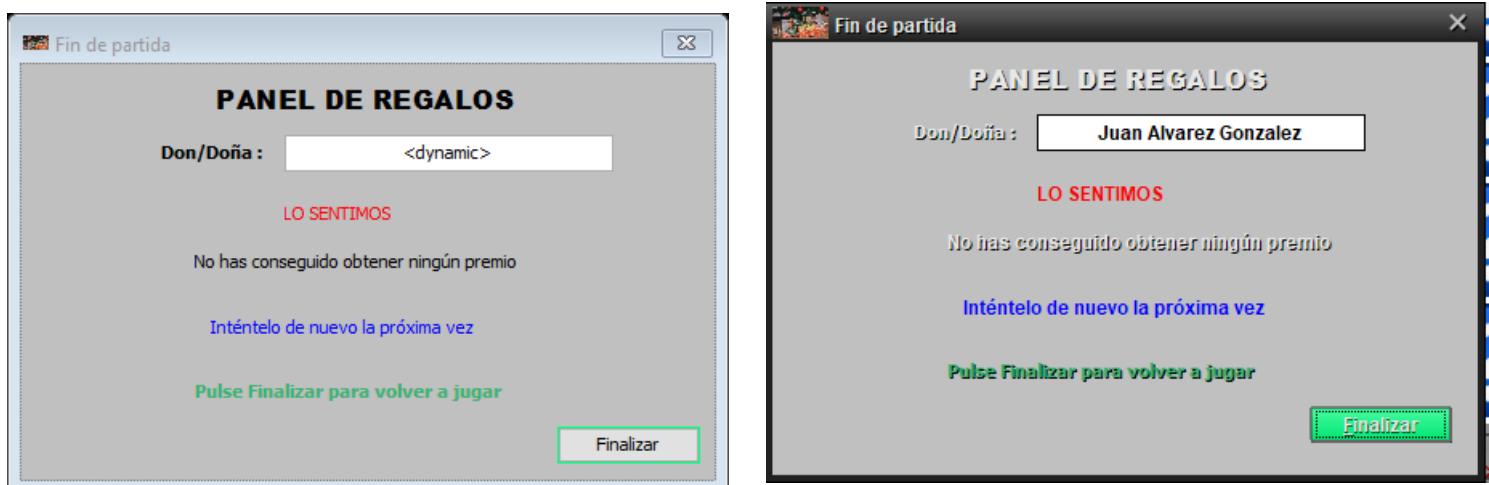
Consta de los siguientes componentes:

El panel principal (border layout, debido a que esta pantalla tiene que redimensionarse adecuadamente) que contiene:

- Un panel central, el de las casillas (propiedad constraints Center), al cual se irán añadiendo 25 botones de manera dinámica (cada uno con su código de evento convenientemente optimizado). Estos botones simularán las casillas de un tablero, que cuando las destapas generan premios. El layout que se ha utilizado es un GridLayout, ya que nos interesa colocar los botones en forma de matriz, distribuidos en 5 filas y en 5 columnas de igual tamaño.
- Un botón “Seleccionar artículos” (propiedad constraints South), que sirve para avanzar a la siguiente ventana (VentanaPanelRegalos) y se habilita cuando el cliente ha gastado todos sus intentos (en principio 3, aunque pueden ser más).
- Un panel (propiedad constraints East) que se utilizará como contador de puntos (se ha utilizado un BorderLayout para que se redimensione adecuadamente y TitleBorder para colocar el título “Contador”). Y este a su vez tiene otro panel (panelPuntos), se ha utilizado el BoxLayout (es un Layout simple que nos ha permitido poner los componentes de forma vertical y de igual tamaño, similar al FlowLayout); este panel contiene 4 campos de texto, los tres primeros serán para poner los puntos de las 3 tiradas y el último para poner el total de ellas.



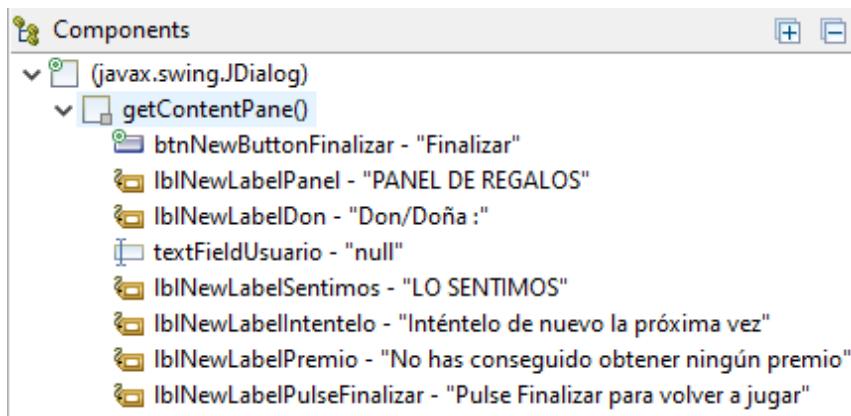
5. VentanaConfirmacionSinPremio: Este JDialog se lanza cuando el usuario selecciona el botón “Seleccionar artículos” del panel de casillas y no ha conseguido ganar ningún punto. La función de esta pantalla es indicar al usuario que se ha acabado la partida, que no ha conseguido ningún premio y animarle a participar la próxima vez.



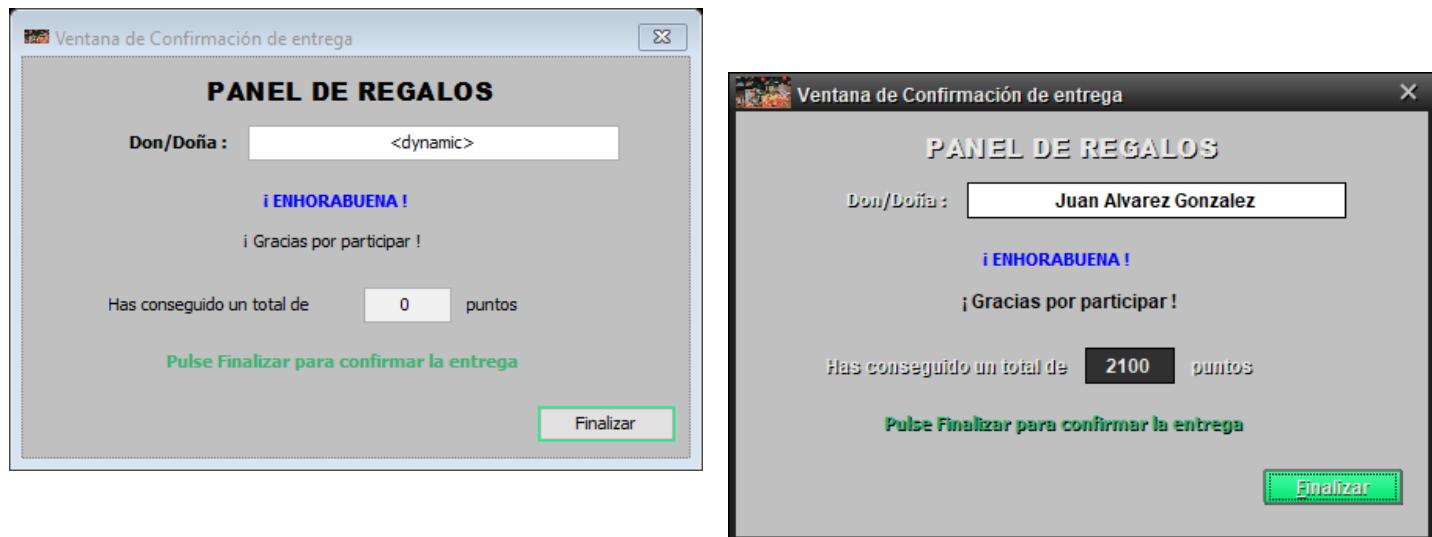
Tiene los siguientes componentes:

El panel principal (absolute layout) que contiene:

- El botón finalizar para iniciar de nuevo la aplicación.
- Un campo de texto, para mostrar el nombre del usuario.
- Los demás elementos son etiquetas de texto (JLabel) que sirven para guiar al usuario a finalizar el juego, y darle las gracias por participar e intentar que se anime a jugar la próxima vez.



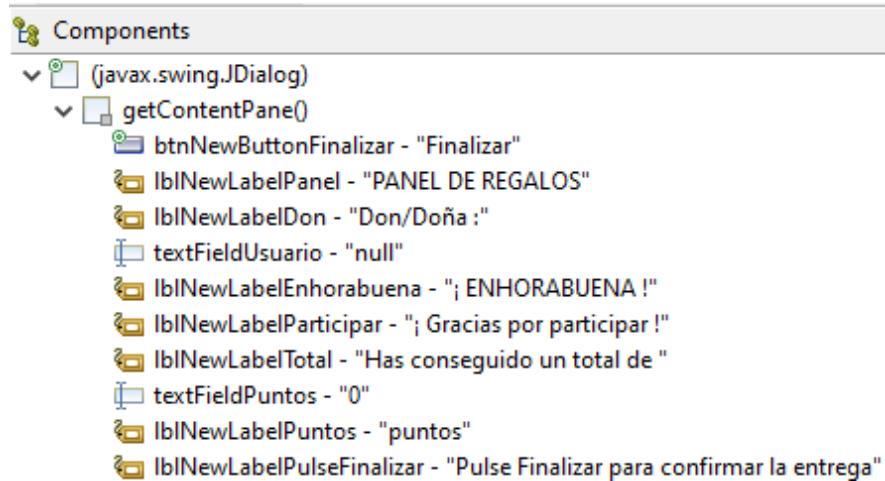
6. VentanaConfirmacionPremios: Este JDialog se lanza cuando el usuario confirma los artículos del pedido (y obviamente el usuario ha ganado algún punto), de alguna manera sirve al usuario a darse cuenta de que se ha acabado la partida y recapitular los puntos que ha conseguido y darle las gracias por participar.



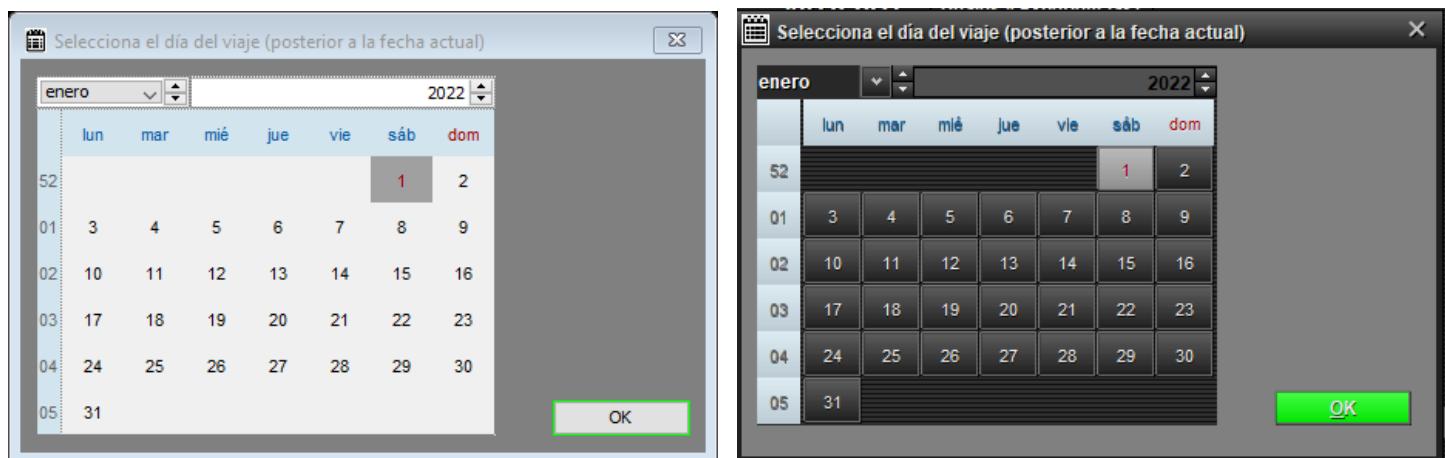
Tiene los siguientes componentes:

El panel principal (absolute layout) que contiene:

- El botón finalizar para iniciar de nuevo la aplicación y generar el fichero de entregas del usuario.
- Dos campos de texto, el primero para mostrar el nombre del usuario y el segundo para los puntos que ha conseguido en el panel de casillas.
- Los demás elementos son etiquetas de texto (JLabel) que sirven para guiar al usuario a finalizar el juego y darle la enhorabuena.

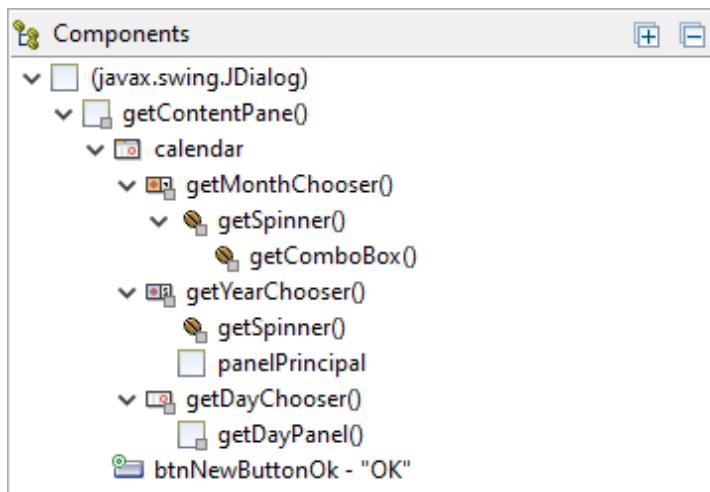


7. VentanaCalendario: Es un cuadro de diálogo (JDialog) que se despliega al pulsar el calendario de la ventana de selección de un viaje, para introducir su fecha de inicio.

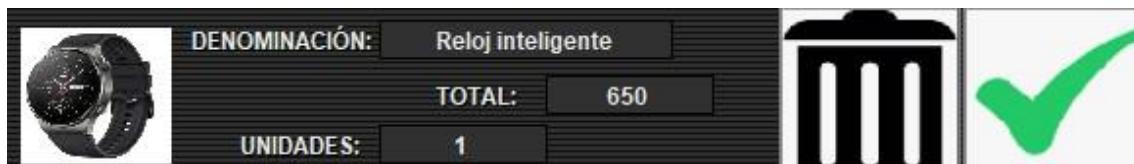


Los elementos que conforman la ventana son:

- Un JPanel (absolute layout) que contiene un calendario (objeto de tipo JCalendar) y un botón “OK” que permite confirmar la fecha y cerrar la propia ventana.
- El calendario contiene tres elementos: un JMonthChooser (para elegir el mes), un JYearChooser (para el año) y un JDayChooser (para el día). El primero de todos contiene a su vez un JSpinner y dentro de este un JComboBox, el segundo un JSpinner y un panel, y el último otro panel.



8. PanelEntrega: Es un JPanel (absolute layout) que contiene las características del artículo a confirmar (o eliminar), es utilizado en el JTabbedPane de la ventana denominada VentanaPanelEntregas. En este JTabbedPane, cada uno de los elementos que lo conforma es el denominado PanelEntrega, tendrá tantos PanelEntrega como entregas (tipos de artículo) tiene el pedido; y su etiqueta es la denominación del artículo.

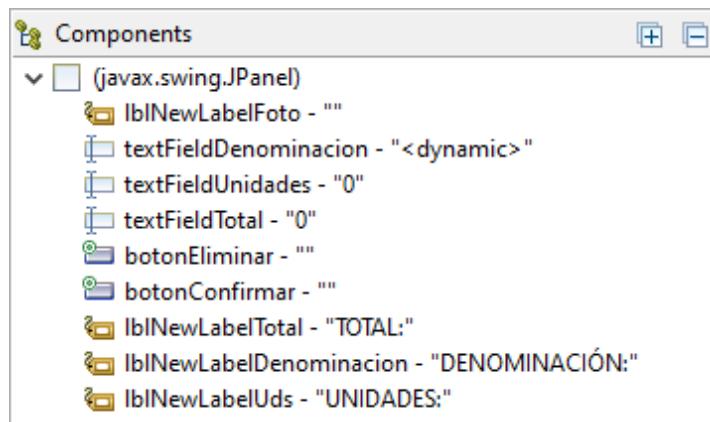


JTabbedPane de la ventana de entregas



Los elementos que conforman este panel son los siguientes:

- Un JLabel que contendrá la foto del artículo, así como otras tres etiquetas más: una para la denominación, otra para el total (del precio), y la última para las unidades.
- Un botón que permite eliminar las unidades que desea el usuario del artículo y un botón confirmar que permite confirmarlo.
- También consta de tres JTextField: el primero será para introducir la denominación del artículo, en el segundo irán las unidades y en el otro el precio total.



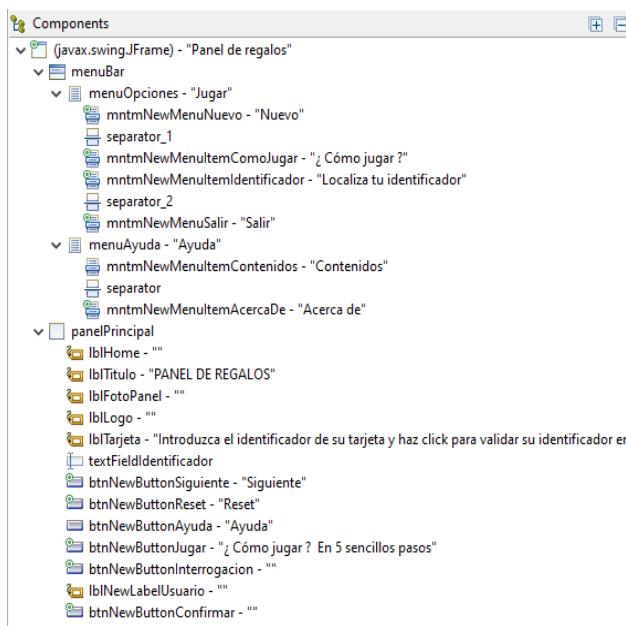
9. Ventana Registro



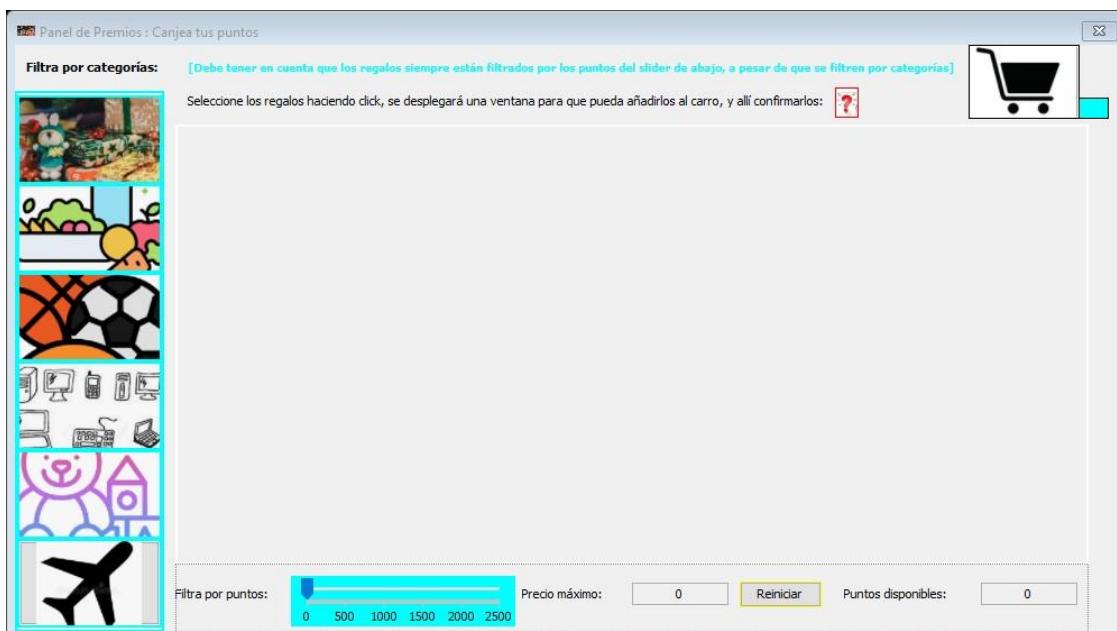
Esta pantalla está formada por dos elementos:

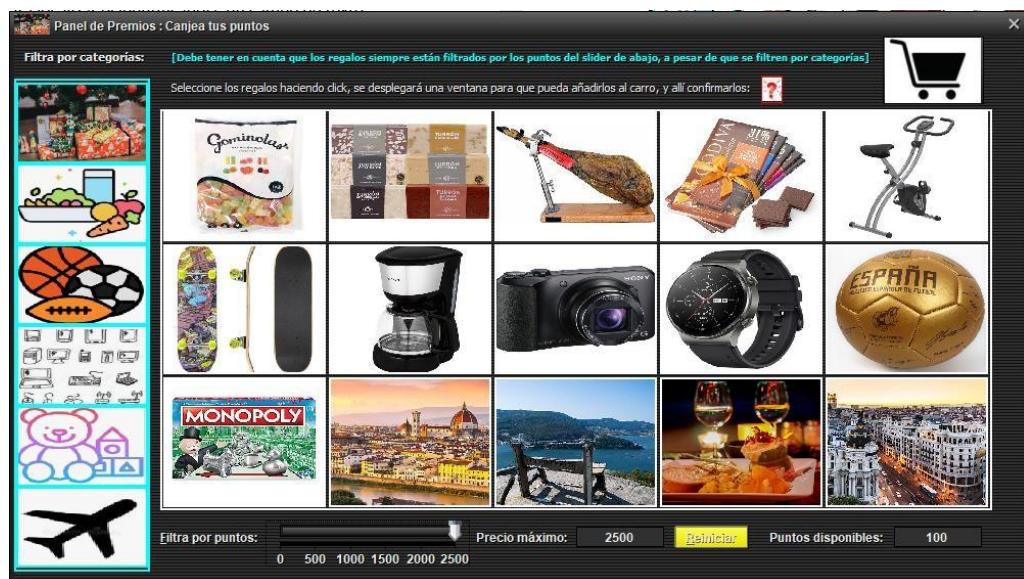
- Un JMenuBar formado por dos JMenu, el menú de jugar y el menú de ayuda. El menú de “Jugar” (menuOpciones) está formado por cuatro JMenuItem (uno para reiniciar la aplicación, el segundo despliega una ventana sobre cómo jugar en 5 pasos, el tercero sirve para identificar el identificador de la tarjeta de fidelización y el último para salir de la aplicación (contiene separadores después del primer item y antes del último item). El menú de ayuda contiene dos JMenuItem (con un separador entre ellos): uno que contiene información acerca de la web y el otro que sirve para que el usuario acceda al manual de ayuda de esta.

- Un panel principal (absolute layout) que contiene 6 botones (uno para borrar el contenido del campo de texto “Reset”, otro para registrar el usuario “Confirmar”, el tercero para avanzar a la siguiente ventana, el cuarto despliega una ventana con información sobre cómo jugar, el penúltimo para acceder a un manual de ayuda y el último para localizar el identificador), un campo de texto (para introducir el identificador) y etiquetas de texto (para información del usuario e imágenes de la web).



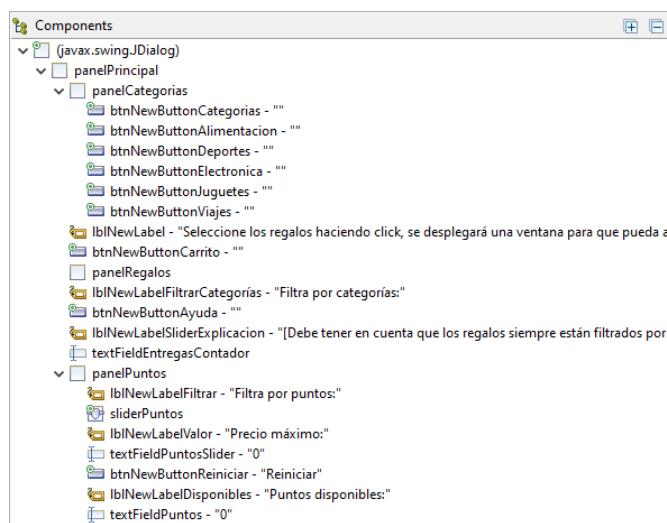
10. VentanaPanelRegalos:



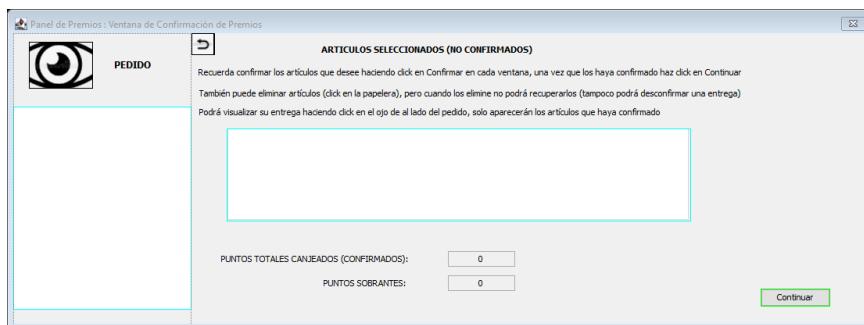


Esta ventana está formada por un panel principal (absolute layout) y contiene:

- Un botón “Ayuda” que muestra información sobre cómo añadir regalos al carro, un botón “Carrito” para acceder a otra ventana de confirmación de regalos, un JTextField que indica el número de regalos (tipo de regalos) hay en el carro y un panelRegalos para mostrar todos los regalos (con gridlayout, para organizar los regalos en columnas de 5 elementos cada una de igual tamaño) que se irán añadiendo de forma dinámica, este panel también tiene etiquetas de texto con información de la ventana.
- Un panelCategorías (gridLayout, para organizar los 6 elementos del panel de una columna de igual tamaño) formado por 6 botones: el primero para eliminar el primer filtrado, y los demás para filtrar por su respectiva categoría (los regalos del panel principal).
- Un panelPuntos (absolute layout) para administrar el sistema de puntos formado por: un JSlider (acompañado de un campo de texto) para filtrar los regalos, un campo de texto para indicar los puntos restantes del usuario, un botón “Reiniciar” para eliminar el filtrado de puntos y categorías, y etiquetas de texto (JLabel) para mostrar información sobre los componentes.

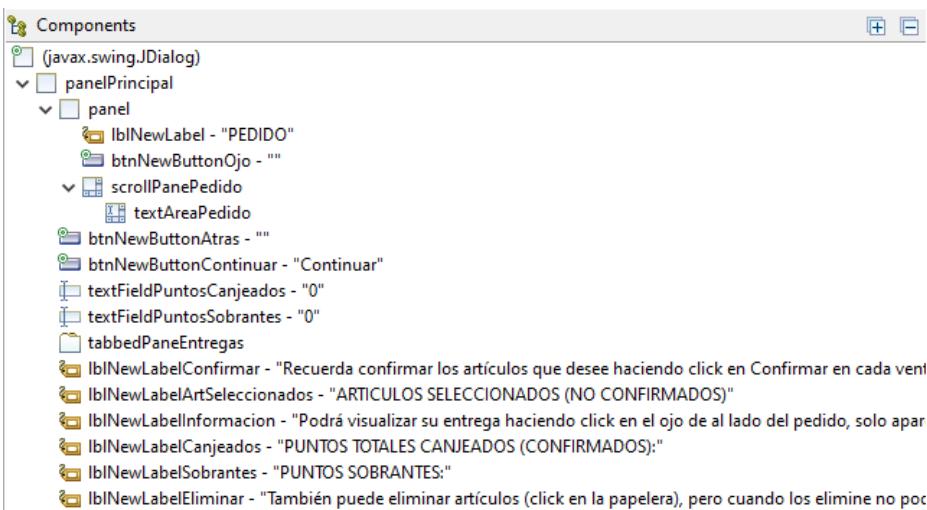


11. VentanaPanelEntregas:

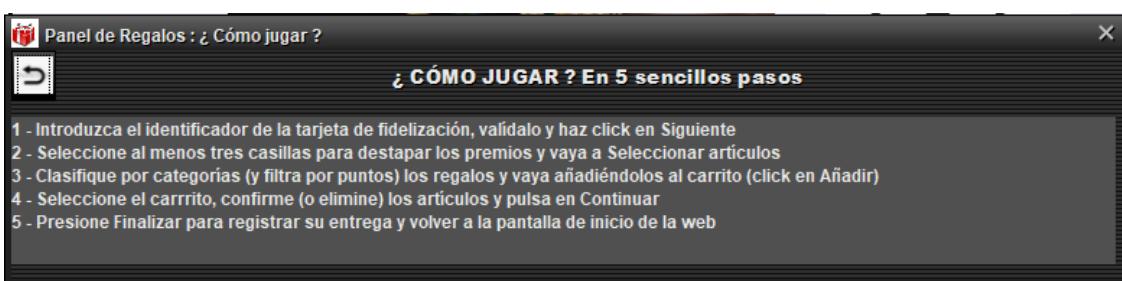
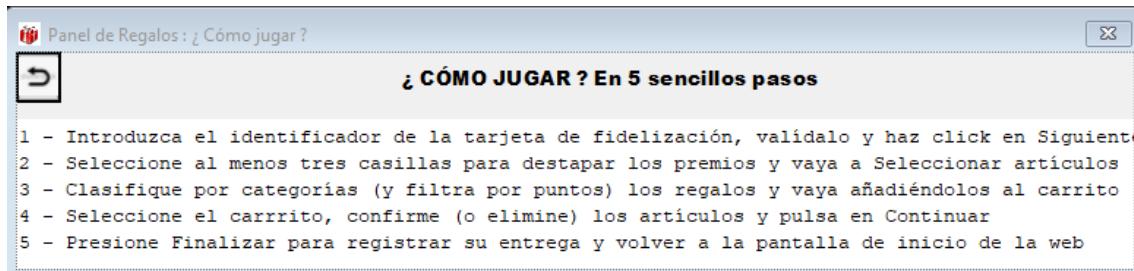


Esta ventana está formada por un panel principal (absolute layout) y contiene:

- Un botón “Atrás” para volver a la pantalla se selección de regalos, un botón “Continuar” para avanzar a la última pantalla de la aplicación (finalización de entrega), un campo de texto para mostrar los puntos canjeados y otro para los puntos sobrantes y el elemento más importante de la ventana, un JTabbedPane (esta ventana también tiene JLabel con explicaciones sobre la confirmación de artículos y demás). Este JTabbedPane tiene tantos paneles como tipos de artículos ha añadido al carro el usuario, la etiqueta de estos JTabbedPane es la denominación de cada regalo y su contenido es un JPanel denominado PanelEntrega (con información acerca del regalo, y botones de eliminación y confirmación del artículo) que ha sido explicado anteriormente.
- Un panelPedido (absolute layout) que contiene un JLabel para el título, un botón que al mantener pulsado sobre él (MouseEvent) hace visible el pedido y un scrollPane (y un JTextArea dentro de este) para mostrar la información del pedido del usuario (de esta forma si no hay suficiente espacio para el pedido, nos podemos servir de las barras de scroll, las verticales en concreto).

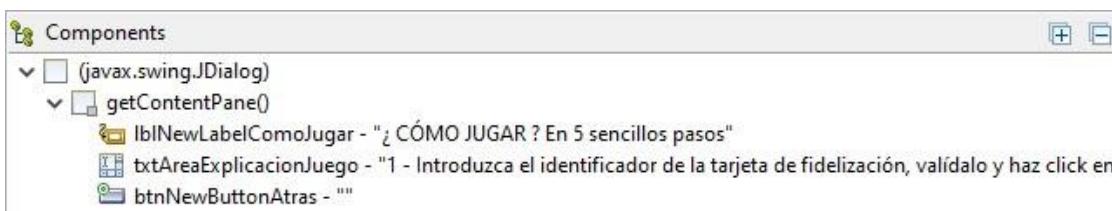


12. VentanaComoJugar:



Esta ventana está formada por un panel principal (absolute layout) y contiene:

- Un botón “Atrás” que permite volver a la ventana principal (la de registro) y cerrar esta ventana.
- Un JLabel para poner el título de la ventana.
- Un JTextArea para escribir información acerca de cómo se juega en 5 sencillos pasos al Panel de Regalos.



2.2.3. PRUEBAS

En este apartado se van a documentar los resultados obtenidos tras la realización de los 3 escenarios planteados por los profesores en el seminario una vez finalizado el desarrollo del módulo. También se tendrá en cuenta algún posible error que haya implicado su corrección, la documentación del escenario, el error encontrado y su solución implementada.

- **Escenario 1: Aña (Edad 41) – Asesora contable.**

Escenario: Ana introduce sin problemas el identificador de cliente y observa el panel. Elige una de las casillas: ¡acaba de ganar 1000 puntos! Parece que no ha terminado el juego, así que elige una segunda casilla. Esta vez es de 250, ya tiene 1250. Prueba una tercera vez y... ¡es su día de suerte! Multiplica por 2, así que acumula un total de 2500 puntos. Ya no puede girar ninguna casilla más. Ve claramente las secciones de los diferentes premios. Es su aniversario, así que va directamente a elegir uno de los viajes. Hay dos regalos de fin de semana que le atraen, uno para Madrid y otro para Vigo. Comprueba que uno es de 1200 puntos y el otro de 1250, así que ¡¡puede quedarse con los dos!! Elige ir a Madrid un fin de semana de febrero y a Vigo en mayo. En este último quiere poner como observación que le den una habitación con vistas al mar, si fuera posible. Al finalizar la selección se le indica que le quedan 50 puntos por canjear. Vuelve a ver los regalos disponibles, pero no encuentra ninguno que le encaje así que decide perderlos.

En primer lugar, podemos observar que Ana es cliente (ya que está en el fichero de clientes) y todavía no ha jugado (tiene un 1 en lugar de un 0 en el campo puedeJugar), por lo que tiene derecho a participar en el Panel de Regalos.

clientes: Bloc de notas	
	Archivo
MFL137@Maria Fernandez Lopez@1	
SGM361@Susana Garcia Menendez@0	
MFL729@Manuel Fernandez Lopez@0	
JAG569@Juan Alvarez Gonzalez@1	
AGG567@Ana Gonzalez Garcia@1	
AFD123@Antonio Fernandez Diaz@1	
GWD768@George Winson Davies@1	

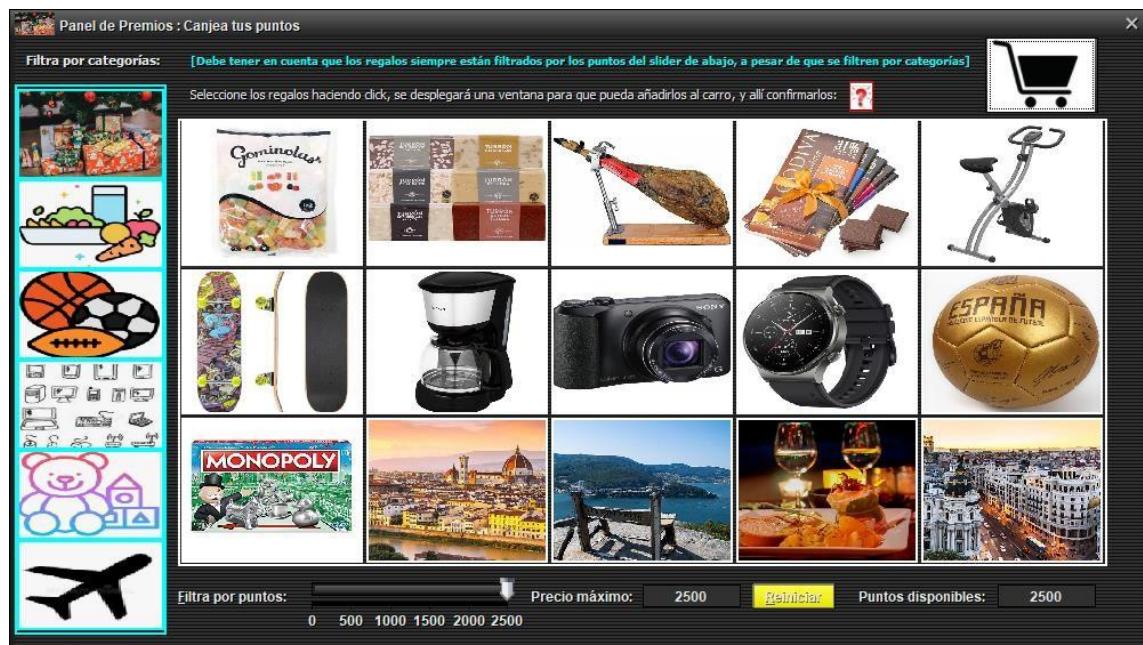
Ana introduce su identificador correctamente, pulsa confirmar y la aplicación le da la bienvenida. Avanza a la siguiente ventana haciendo click en Siguiente.



Una vez en el panel de casillas, escoge una por una, la primera tiene un premio de 1000 puntos, la segunda de 250 puntos y la última dobla el contador de puntos, por lo que ha conseguido la puntuación máxima, 2500 puntos (dobra los 1250 anteriores).

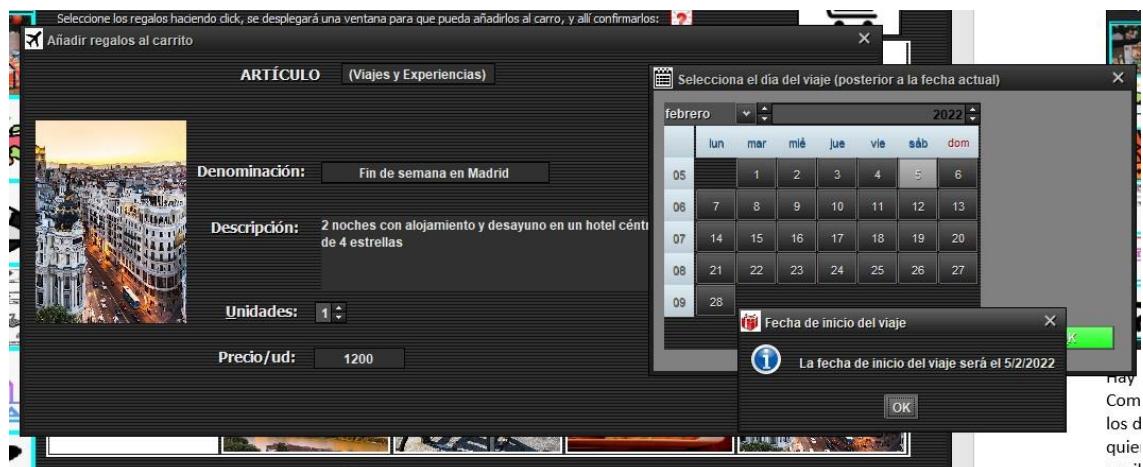


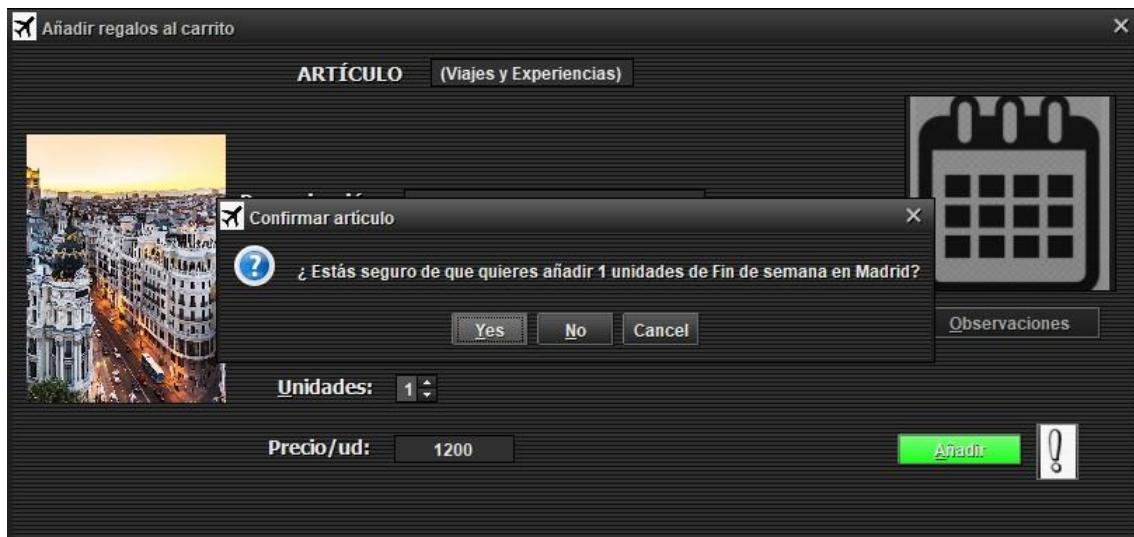
Avanza a la siguiente ventana, como le indica la ventana de información de la anterior ventana, y ve como tiene acceso a todos los artículos.



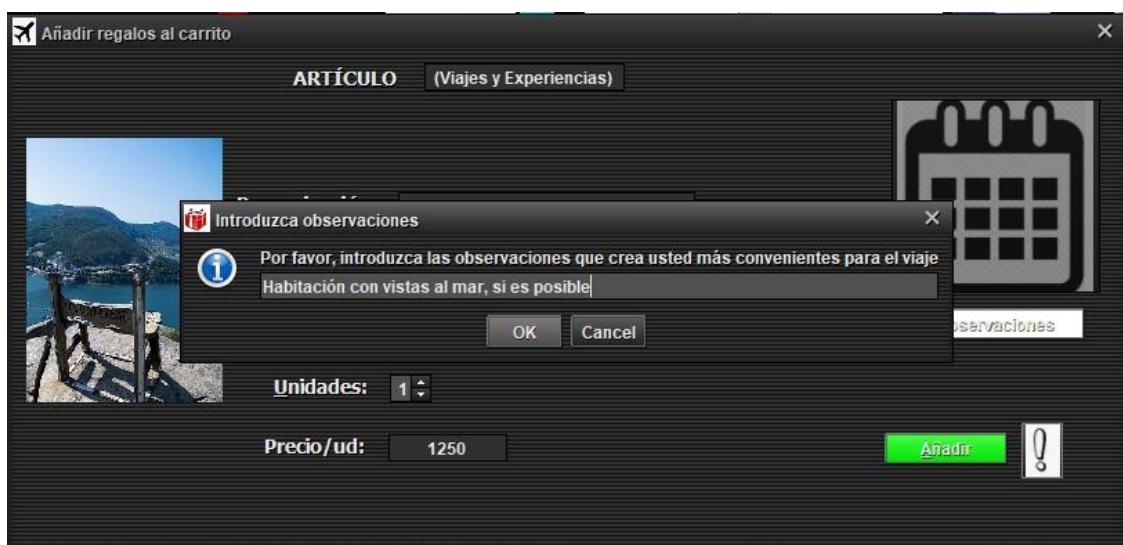
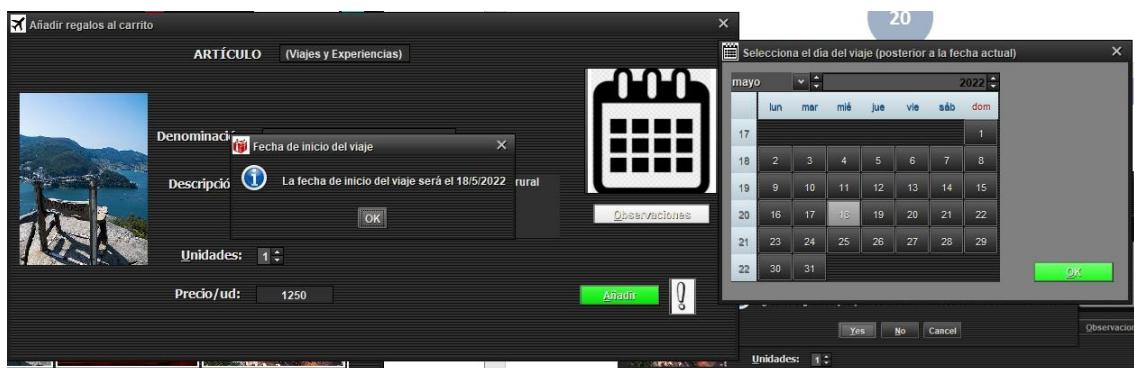
Hay dos regalos de fin de semana que le atraen, uno para Madrid y otro para Vigo. Comprueba que uno es de 1200 puntos y el otro de 1250, así que puede quedarse con los dos. Elige ir a Madrid un fin de semana de febrero y a Vigo en mayo. En este último quiere poner como observación que le den una habitación con vistas al mar, si fuera posible.

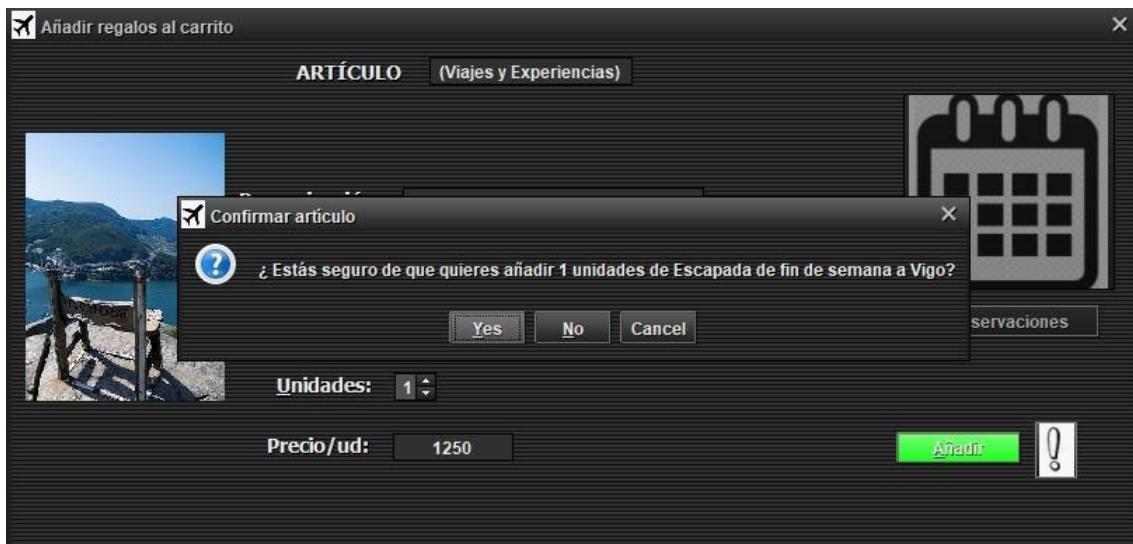
Ana pincha en la imagen del viaje a Madrid (ayudándose del tooltip asociado al botón) y se despliega otra ventana, donde tiene que añadir la fecha, hace click en el calendario y selecciona el 5 de febrero de 2022 para la fecha de inicio del viaje, finalmente introduce las observaciones al viaje y lo añade al carro de la compra.



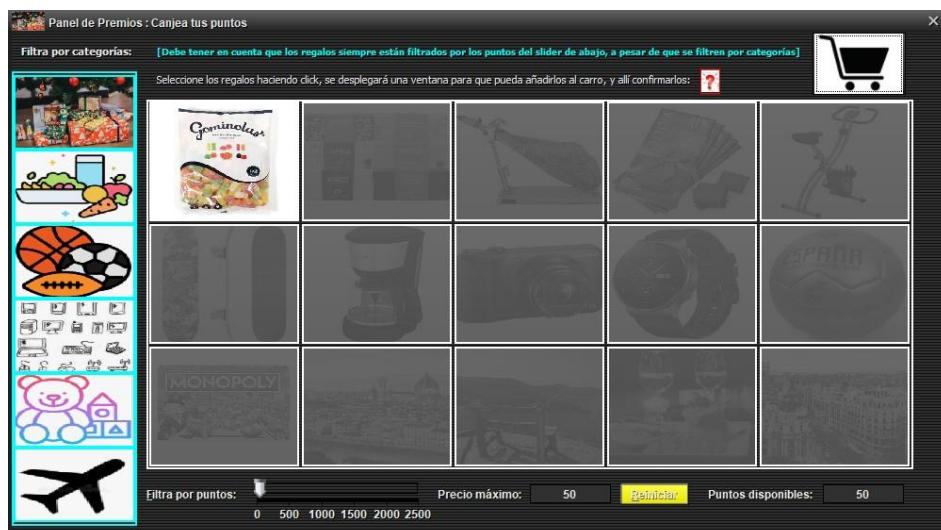


Luego decide irse a por el viaje a Vigo en mayo de 2022 e introduce en observaciones que le den una habitación con vistas al mar (si es posible). Lo añade al carro también.





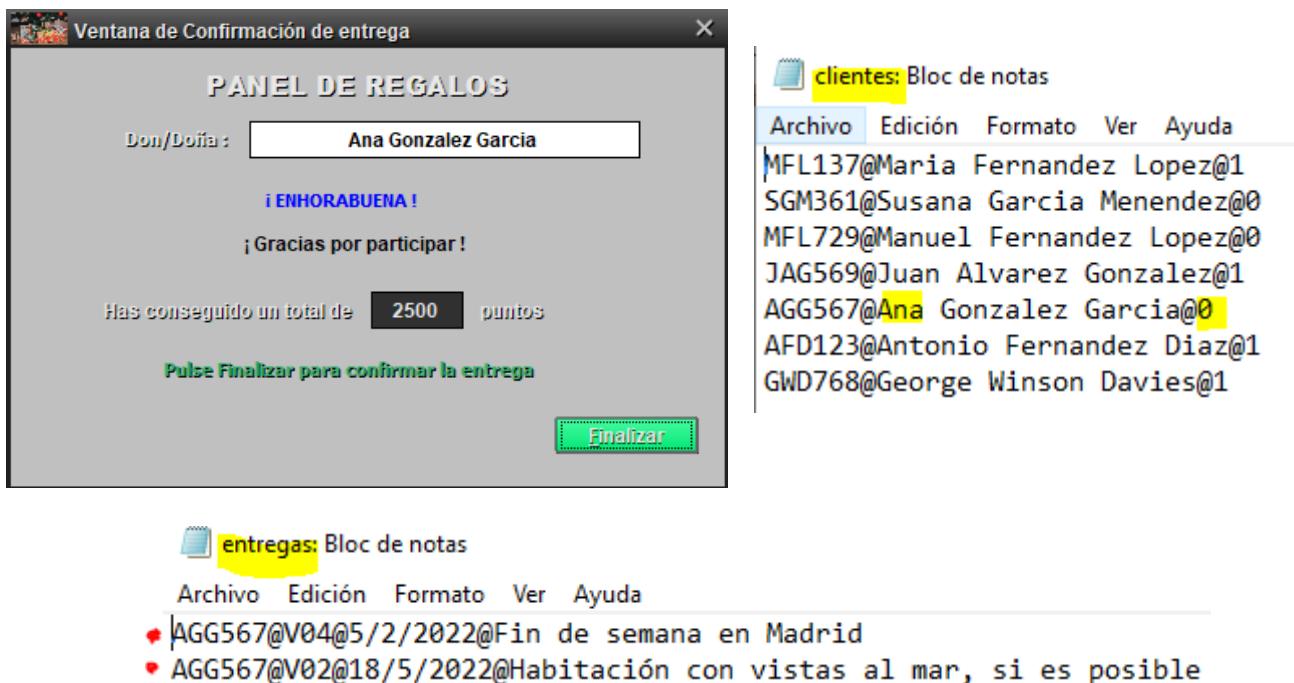
Al finalizar la selección se le indica que le quedan 50 puntos por canjear. Vuelve a ver los regalos disponibles, pero no encuentra ninguno que le encaje así que decide perderlos, ya que solo puede comprar gominolas y no quiere.



Confirma los artículos haciendo click en confirmar en ambos viajes, la aplicación le avisa de que va a perder los 50 puntos y decide avanzar a la pantalla final para terminar la entrega.



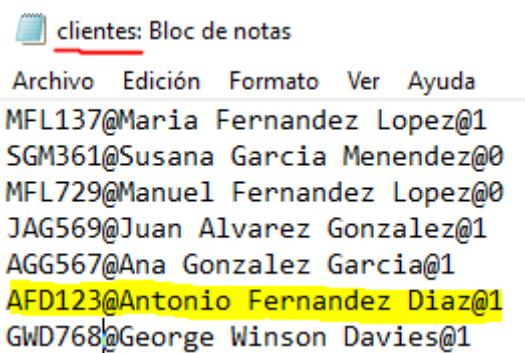
Pulsa finalizar para confirmar la entrega de los premios (los dos viajes) que se guardarán en un fichero de texto con sus características (formato idCliente@codigoArticulo@fecha@observaciones). También se actualizará la aplicación indicando que Ana ya no puede volver a jugar colocando un 0 en el fichero (en lugar del 1).



- **Escenario 2: Antonio (Edad 82) – Jubilado.**

Escenario: A Antonio le acaban de comentar al pagar su compra que tiene opción a jugar en el panel de regalos. Muy contento y algo asustado porque nunca interactuó con algo similar, se acerca al terminal. Tiene que meter un identificador: por las pistas visuales que le da la interfaz entiende perfectamente dónde localizarlo en su tarjeta y lo introduce sin problemas. Llega a la pantalla donde le aparece el panel de premios y elige una a una las 3 casillas a las que tiene derecho. Ha acumulado un total de 300 puntos. Tiene 3 nietos así que le gustaría elegir 3 premios iguales. Ve claramente las diferentes categorías, pero cree que le va a costar mucho trabajo navegar por todas para localizar artículos de 100 puntos. Así que le gustaría que hubiese una opción para que, en base a su puntuación, viese todos los regalos a los que podría optar, esto es, los que valiesen igual o menos que los puntos acumulados y así poder hacer la selección de una manera más sencilla. Así elegiría 3 unidades de alguno de los premios de 100 puntos o menos.

En primer lugar, podemos observar que Antonio es cliente (ya que está en el fichero de clientes) y todavía no ha jugado (tiene un 1 en lugar de un 0 en el campo puedeJugar), por lo que tiene derecho a participar en el Panel de Regalos.



Selecciona la opción de ayuda para localizar el identificador, y entiende perfectamente donde está situado, lo introduce correctamente y lo valida, avanza a la siguiente ventana.



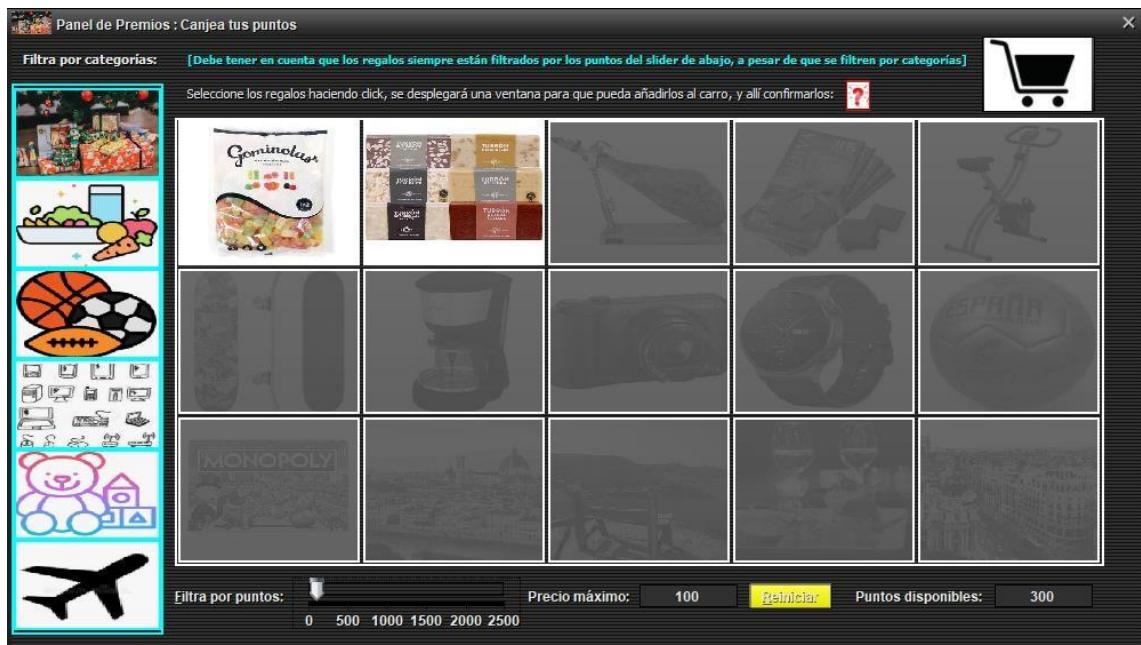


Llega a la pantalla donde le aparece el panel de premios y elige una a una las 3 casillas a las que tiene derecho. Ha acumulado un total de 300 puntos.



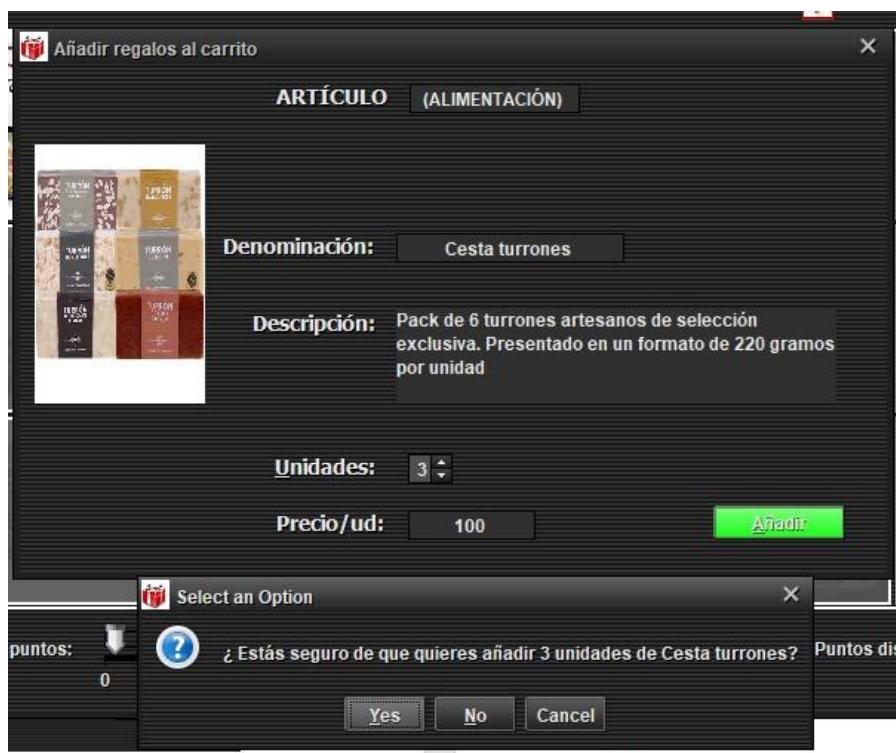
Tiene 3 nietos así que le gustaría elegir 3 premios iguales. Ve claramente las diferentes categorías, pero cree que le va a costar mucho trabajo navegar por todas para localizar artículos de 100 puntos. Así que le gustaría que hubiese una opción para que, en base a su puntuación, viese todos los regalos a los que podría optar, esto es, los que valiesen igual o menos que los puntos acumulados y así poder hacer la selección de una manera más sencilla. Así elegiría 3 unidades de alguno de los premios de 100 puntos o menos.

Filtrá los regalos por slider, deslizando a la izquierda este hasta 100 puntos, por lo que solo aparecerán como disponibles aquellos cuyo valor sea menor o igual a este. Antonio se sirve también de ayuda con el campo de texto del JSlider, ya que no controla muy bien las nuevas tecnologías.

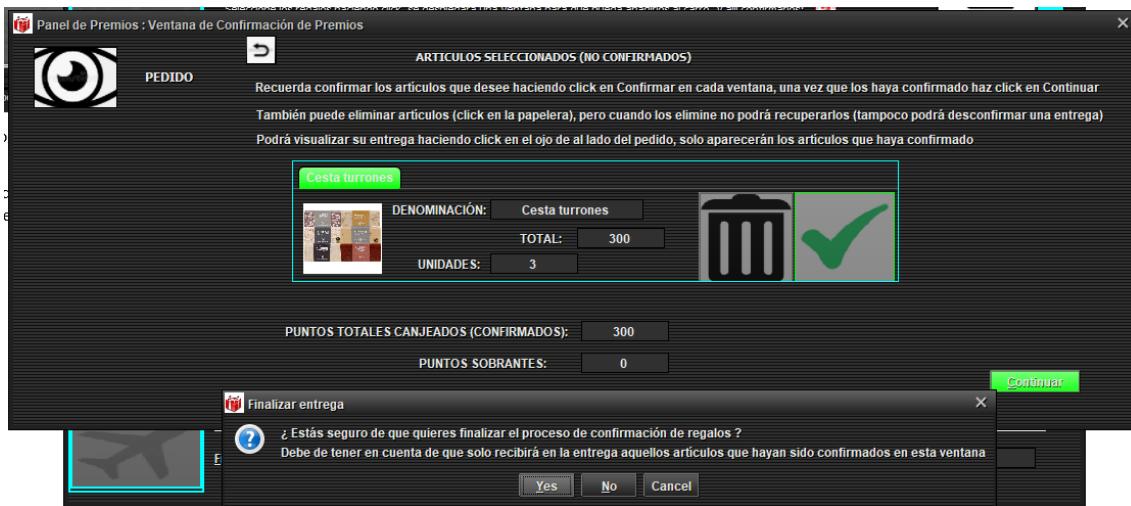


Puede observar que puede coger tres cestas de turrones, con valor de 100 puntos cada una, una para cada nieto.

Antonio pincha en la imagen de la cesta (ayudándose del tooltip asociado al botón) y se despliega otra ventana, donde tiene que añadir las unidades (marca 3), pulsa Añadir.



Confirma los artículos haciendo click en confirmar, y avanza a la pantalla final para terminar la entrega (y pulsa Finalizar).



Los artículos se guardarán en un fichero de texto con sus características (formato idCliente@codigoArticulo). También se actualizará la aplicación indicando que Antonio ya no puede volver a jugar colocando un 0 en el fichero (en lugar del 1).

clientes: Bloc de notas
MFL137@Maria Fernandez Lopez@1
SGM361@Susana Garcia Menendez@0
MFL729@Manuel Fernandez Lopez@0
JAG569@Juan Alvarez Gonzalez@0
AGG567@Ana Gonzalez Garcia@0
AFD123@Antonio Fernandez Diaz@0
GWD768@George Winson Davies@1

*entregas: Bloc de notas
Archivo Edición Formato Ver Ayuda
AFD123@A02

- Escenario 3: George (Edad 26) – Profesor de inglés.

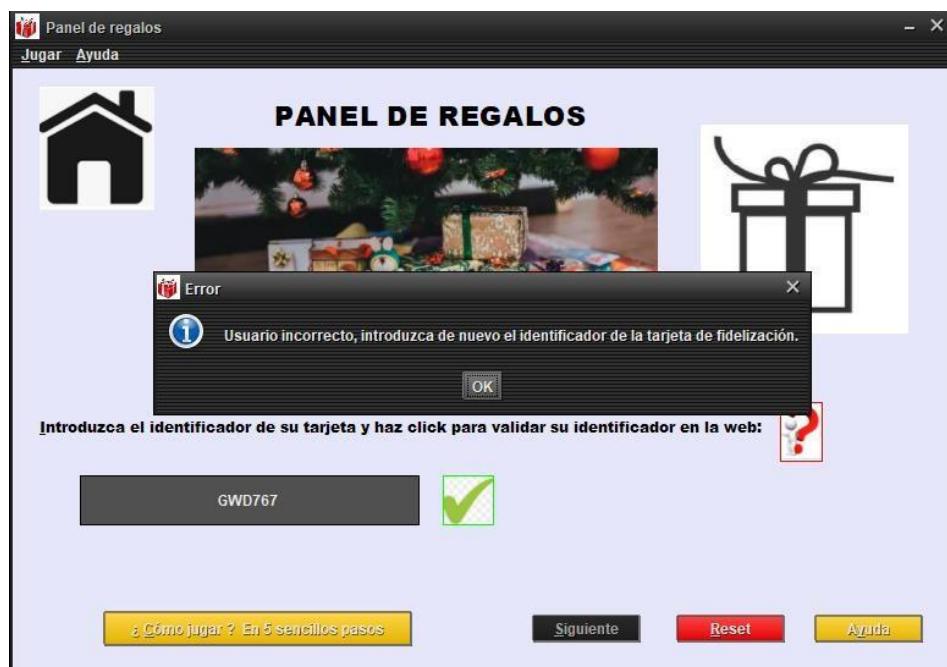
Escenario: George acaba de hacer la compra y le indican que puede acercarse al terminal del panel de regalos a probar suerte. Como acaba de llegar a España y todavía no domina el idioma, espera que, o bien la interfaz sea lo suficientemente intuitiva por los controles e imágenes utilizados o bien que sea posible cambiar al idioma inglés. Comienza a jugar e introduce un identificador que le piden, así que saca su tarjeta e introduce el número que aparece en la misma. No parece que sea ese, porque no le deja continuar. Pero mediante la información que le proporciona la aplicación comprende cual es el número que le piden y lo localiza en la tarjeta. Ahora sí que le aparece un panel delante ¿qué tiene que hacer ahora? ¿darle a algún sitio para que se destape una casilla al azar? No, le queda claro que tiene que elegir él. Destapa una casilla y no tiene premio. Ya piensa que se va sin regalos, pero ve claramente que puede seguir destapando. La siguiente está premiada con 50 puntos. Destapa otra más, la última. Es distinta; en principio no entiende muy bien qué es lo que ocurre con esa casilla, pero entiende que le da opción a destapar otra más. Se lleva otros 50 puntos, así que con los 100 acumulados pasa a canjear sus regalos. Va directamente a la sección de alimentación, que identifica claramente, para elegir varios productos por valor de los 100 puntos exactos. Cuando lleva productos por valor de 80 intenta añadir uno de 30, pero no es posible. Así que devuelve uno que había añadido de 40 puntos y coge otro de 60. La aplicación le confirma que se han procesado la selección, así que se va muy contento.

En primer lugar, podemos observar que George es cliente (ya que está en el fichero de clientes) y todavía no ha jugado (tiene un 1 en lugar de un 0 en el campo puedeJugar), por lo que tiene derecho a participar en el Panel de Regalos.

clientes: Bloc de notas	
Archivo	Edición
Formato	Ver
Ayuda	
MFL137@Maria Fernandez Lopez@1	
SGM361@Susana Garcia Menendez@0	
MFL729@Manuel Fernandez Lopez@0	
JAG569@Juan Alvarez Gonzalez@1	
AGG567@Ana Gonzalez Garcia@1	
AFD123@Antonio Fernandez Diaz@1	
GWD768@George Winson Davies@1	

Como acaba de llegar a España y todavía no domina el idioma, espera que, o bien la interfaz sea lo suficientemente intuitiva por los controles e imágenes utilizados o bien que sea posible cambiar al idioma inglés. En este caso, la aplicación no ha sido internacionalizada (ni localizada) pero es muy intuitiva, ya que se da cuenta de que tiene que introducir algo en el campo de texto (ya que tiene el foco), el identificador.

George se da cuenta donde está el identificador de su tarjeta y lo en el campo de texto, pero observa que la aplicación no le deja continuar, le pone que el usuario no es válido.



Selecciona la opción de ayuda para localizar el identificador, y entiende perfectamente donde está situado, también se da cuenta de que se había confundido en el último número.





Ahora sí, introduce el identificador correctamente, pulsa confirmar y la aplicación le da la bienvenida. Avanza a la siguiente ventana haciendo click en Siguiente.



Se abre el panel de las casillas con el texto (en español) de cómo se eligen las casillas, como no entiende muy bien el español cierra la ventana de información. No le queda claro que tiene que elegir él. Destapa una casilla y no tiene premio. Ya piensa que se va sin regalos, pero ve claramente que puede seguir destapando. La siguiente está premiada con 50 puntos. Destapa otra más, la última. Es distinta; en principio no entiende muy bien qué es lo que ocurre con esa casilla, pero entiende que le da opción a destapar otra más. Se lleva otros 50 puntos, así que con los 100 acumulados pasa a canjear sus regalos.



George se ha servido también del número que aparecía en las ventanas cuando destapaba las casillas y ha supuesto que serían los números de tiradas que le quedaban, se había quedado un poco mosqueado cuando había seleccionado las tres casillas y el botón de avanzar no estaba disponible (y los dos últimos campos de texto del contador estaban vacíos). Pero cuando ha seleccionado la cuarta casilla y la aplicación le ha indicado que ha ganado 100 puntos se ha quedado tranquilo. También se había ayudado de la ayuda de la aplicación, seleccionando F1 y ayudándose de las pantallas.

Help - Panel de Regalos

Help - Panel de Regalos

¿Qué es el Panel de Regalos?
Acceder como cliente a la web
■ ¿Cómo registrarse en la web?
■ ¿Cómo obtener el identificador?
■ ¿Cómo jugar?
Seleccionar casillas del panel
■ ¿Cómo añadir regalos al carrito?
■ Agregar regalos al carrito
■ ¿Cómo filtrar los regalos?
■ Añadir un artículo
■ Añadir un viaje
■ Introducir observaciones a un v...
■ Seleccionar una fecha de inicio
■ ¿Cómo confirmar la entrega?
■ Finalizar el proceso de confirmación
■ Confirmar un artículo
■ Eliminar un artículo
■ Visualizar el pedido
Finalizar el proceso de entrega
¿Qué pasa si no conseguimos ning...

Panel de Casillas

Para seleccionar una casilla del panel solamente tenemos que hacer click izquierdo sobre la casilla que queremos pulsar, y dependiendo de la casilla que hemos pulsado se desplegará una ventana de información que tendrá diferente contenido. Disponemos de tres intentos inicialmente, esto es decir, podemos descubrir tres casillas (no significa que luego podamos tener más intentos al desvelar casillas especiales). Hay tres tipos de casillas: las que dan premio (pueden dar 0, 50, 250 o 1000 puntos), las que más aparecen son las de 0 puntos (hay 8 en total), existen otros dos tipos más de casillas: las especiales (que te dan una tirada más, hay dos casillas de este tipo) y las que doblan el contador (las X2, solamente hay una casilla de este tipo). En la imagen siguiente, hemos seleccionado la casilla número 8, cuyo premio son 50 puntos, todavía nos quedan 2 casillas por descubrir. Cuando pulsemos OK, la primera posición del contador cambiará de color negro a gris, y aparecerá el premio de esta casilla (50 puntos).

Panel de Premios

Premio de la Casilla
Esta casilla contiene un premio de:
50 PUNTOS
CASSILLAS RESTANTES PARA DESCUBRIR: 2

Contador

CASILLA 1 = 0
CASILLA 2 = 50
CASILLA 3 = 50
TOTAL = 100

Seleccionar artículos

Cuando hemos seleccionado todas las casillas que nos permite la aplicación (tantas como el número de intentos), se habilitará el botón verde de abajo y aparecerán los puntos totales que hemos conseguido en el contador de abajo a la derecha.

Panel de Premios

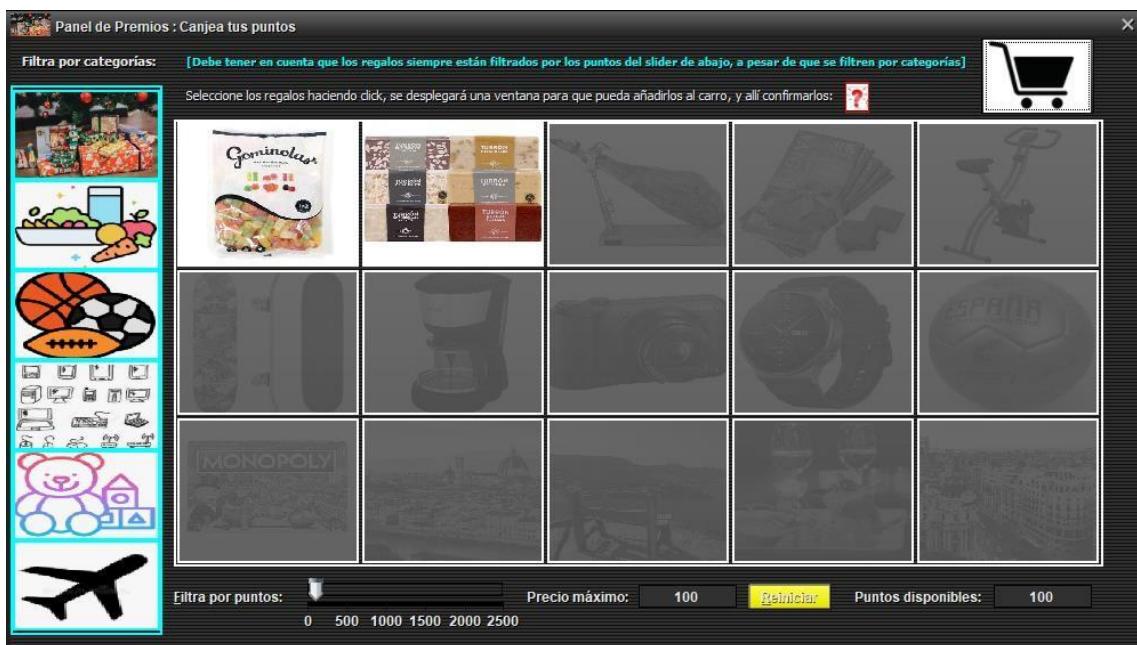
Fin de tiradas
Acaban de terminar sus tiradas, ha conseguido un total de 100 puntos
Avanza a la siguiente ventana, presionando Seleccionar artículos

Contador

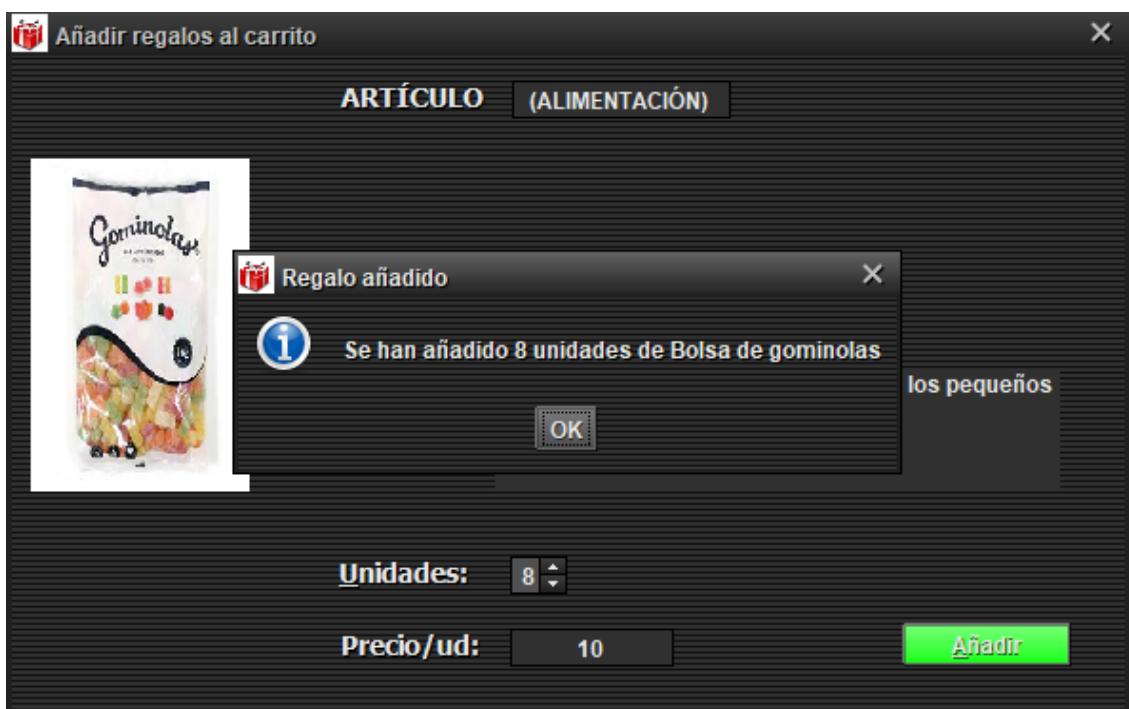
CASILLA 1 = 0
CASILLA 2 = 50
CASILLA 3 = 50
TOTAL = 100

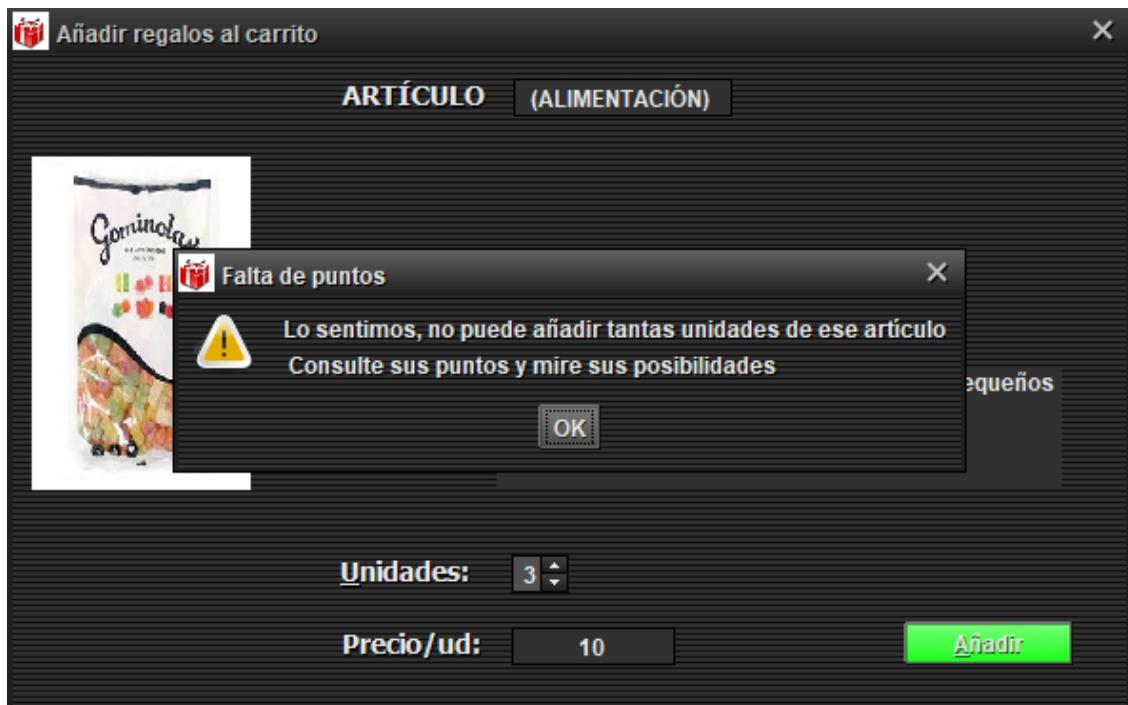
Seleccionar artículos

Avanza a la ventana de regalos y va directamente a la sección de alimentación, que identifica claramente, para elegir varios productos por valor de los 100 puntos exactos. La aplicación por defecto ya filtra los regalos por los puntos disponibles (solo aparecen disponibles gominolas y cesta de turrones). Por lo que cuando ha seleccionado la categoría "Alimentación" se han mantenido los mismos artículos como disponibles.

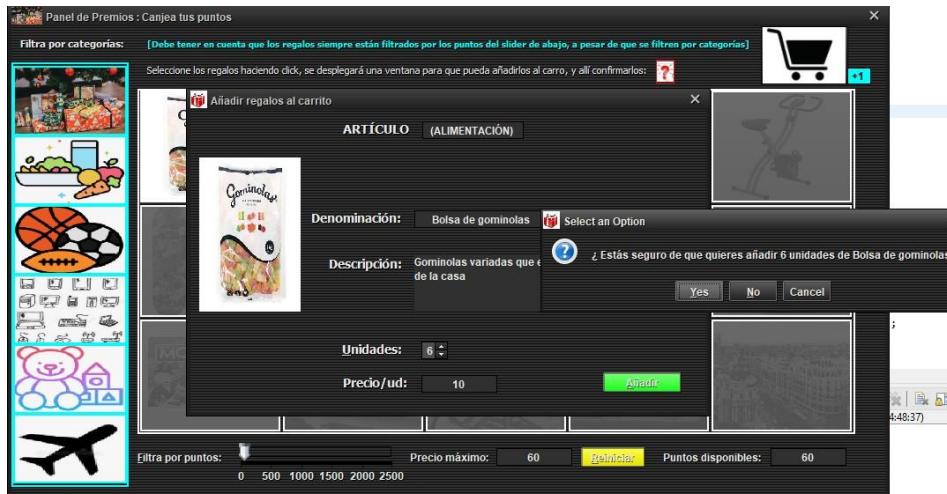
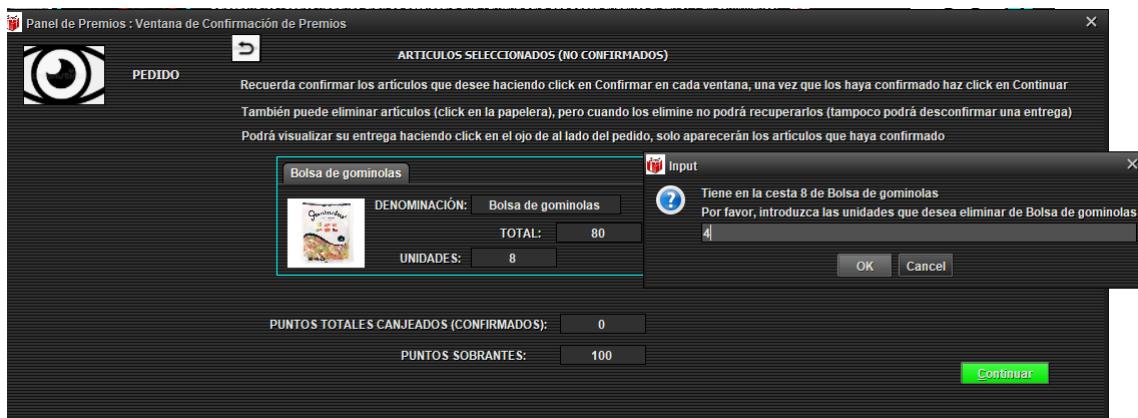


Cuando lleva productos por valor de 80 (añade 8 bolsas de gominolas) intenta añadir uno de 30 (tres bolsas más), pero no es posible.





Decide devolver uno que había añadido de 40 puntos (4 bolsas de gominolas), vuelve atrás a la sección de regalos y coge otro de 60 (6 bolsas de gominolas), y así completa los 100 puntos.





La aplicación le confirma que se han procesado la selección, así que se va muy contento.

Pulsa finalizar para confirmar la entrega de los regalos que se guardarán en un fichero de texto con sus características (formato idCliente@codigoArticulo). También se actualizará la aplicación indicando que George ya no puede volver a jugar colocando un 0 en el fichero (en lugar del 1).

