



Escuela de
Ingeniería
Informática
Universidad de Oviedo



Panel de Regalos – CPM (2021/22)

Realizado por *Omar Teixeira González*,
DNI: 32892095W
UO281847

Índice:

| | |
|------------------------------|----|
| ÍNDICE: | 2 |
| INTRODUCCIÓN: | 3 |
| DESARROLLO DE LA APLICACIÓN: | 4 |
| 1. FASE DE DISEÑO: | 4 |
| 2. FASE DE IMPLEMENTACIÓN: | 6 |
| • <i>Lógica:</i> | 6 |
| • <i>Interfaz:</i> | 9 |
| • <i>Pruebas:</i> | 14 |
| ○ Escenario 1: | 14 |
| ○ Escenario 2: | 14 |
| ○ Escenario 3: | 15 |

Introducción:

Tal y como se ha pedido en este módulo, se debía realizar una aplicación en la que el usuario tuviera que introducir su código de cliente, una vez validado dicho código, el usuario será llevado a la ventana del panel de juego.

Donde, en un primer momento, podrá pulsar 3 de los 25 botones del panel, con la excepción de que, si encuentra una casilla especial (representada en este módulo como una estrella) podrá presionar en otra casilla extra; de esta manera, el cliente puede llegar a pulsar un máximo de 5 casillas, pudiendo obtener un máximo de 2500 puntos.

Una vez, ha conseguido sus puntos, el usuario deberá canjearlos en el panel correspondiente, el cual, contiene 5 secciones de productos, tales como: *Alimentación, Deportes, Electrónica, Juguetes y Viajes/Experiencias*. Cuando haya canjeado los productos solicitados, el usuario habrá terminado su curso en la aplicación, y quedará preparada para el uso del siguiente.

Se ha realizado la internacionalización de la aplicación, de modo que se tenga la posibilidad de optar a la valoración extra que esto conlleva, pero, para hacer más completa esta opción, se ha permitido al usuario cambiar el idioma en cualquier momento de la aplicación, incluyendo la ventana de canjeo, actualizando así la lista de regalos completa y la de los seleccionados.



Desarrollo de la aplicación:

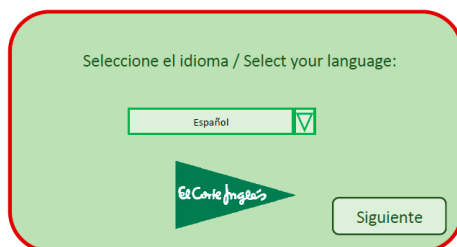
1. Fase de Diseño:

En primer lugar, se han escogido como colores principales de la aplicación colores verdes (muy claros y muy oscuros) y rojos oscuros. Los colores verdes cumplen con la función de respetar la guía de estilo de la empresa escogida como ejemplo para el desarrollo de la aplicación, en este caso, El Corte Inglés. Sumado a esto, los colores rojo unidos a los verdes guardarían la similitud con la Navidad, periodo escogido para presentar la aplicación al público.

Cabe destacar, que debido a que los colores verdes y rojos son los que pueden llevar a una mayor confusión a usuarios con daltonismo, se ha llevado a cabo ese contraste entre el muy claro y el muy oscuro para que así no haya problemas.

Pasando a la coherencia con el wireframe realizado en las prácticas, se ha intentado guardar la mayoría de las apariencias, realizando pequeños cambios en ventanas concretas de manera que cumplan con los requisitos finales del módulo.

De esta forma:

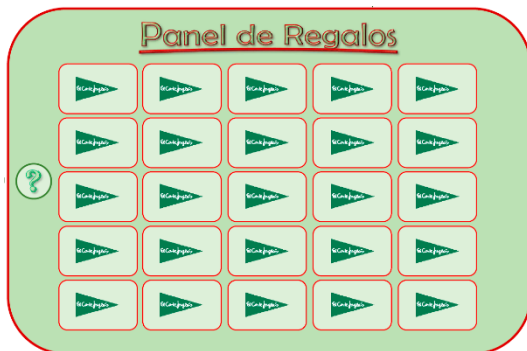


La ventana de selección de idioma se ha cambiado el JComboBox por dos botones con las banderas respectivas, ya que así sería más fácil de ver para los usuarios.

En cambio, las ventanas correspondientes a la introducción del código y la bienvenida a la aplicación se han mantenido fieles al diseño original, con alguna serie de añadidos, como botones para ir a la ventana anterior y a la siguiente, en el caso de código, o una JLabel dedicada a mostrar el nombre del usuario.

La ventana del panel de botones ha mantenido su estructura en su mayoría, pero también se han añadido varias cosas para adaptarlas a los requisitos.

Tal y como se ve en las siguientes imágenes, se han añadido adornos navideños y modificado el logo de El Corte Inglés, para hacer que la interfaz recuerde a dicha etapa del año. Además, se ha añadido un contador de puntos (requisito del módulo) y, otro contador de las tiradas restantes del usuario.



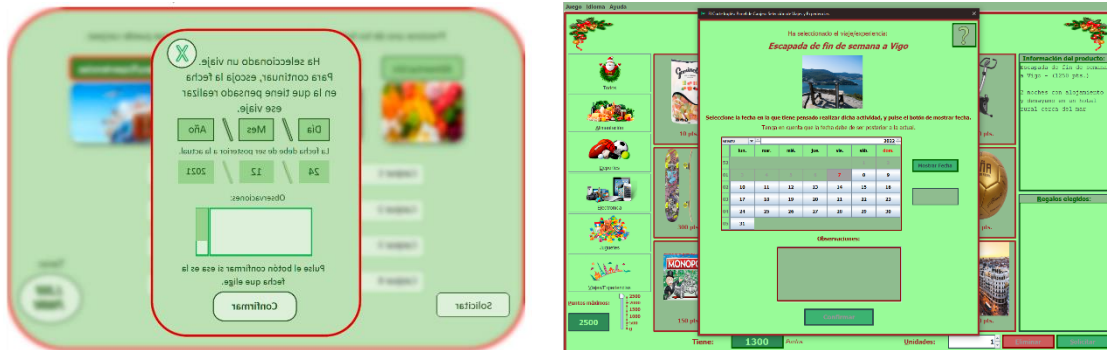
Respecto a la ventana de canjeo, se han modificado bastantes cosas, para cumplir tanto los requisitos como con mayor simplicidad a la hora del desarrollo.



En primer lugar, los botones de filtro se han pasado al lateral, dejando espacio para el título y los adornos correspondientes. También se ha añadido un JSlider para filtrar por el precio y un JTextArea que muestra la información del producto sobre el que se posa el ratón, ambos requisitos del módulo.

Para mostrar los productos seleccionados se ha escogido un JList, de forma que si quiere eliminar un producto elegido, se presione sobre dicho producto y posteriormente el botón de eliminar (también añadido en la versión final). Por último, se ha dado al usuario la opción de añadir o eliminar más de una unidad, para eso está el JSpinner de las unidades, donde el usuario debe seleccionar las unidades a añadir y posteriormente presionar sobre dicho producto.

A la ventana de selección de viajes/experiencias se le ha añadido un JCalendar, de forma que hiciera más sencilla la selección de la fecha, y se ha incorporado el nombre y foto de la actividad seleccionada.



El resto de la aplicación ha mantenido su diseño en líneas generales, ya que temas de estilo, como el fondo de los mensajes no se ha podido personalizar.

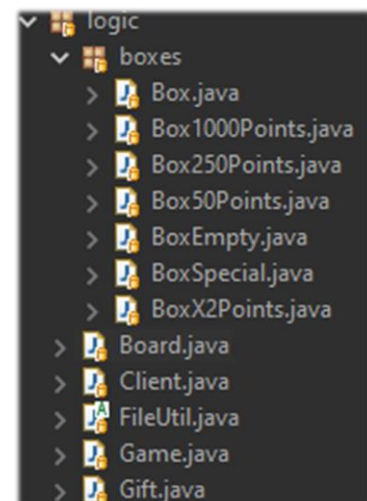
Todos los cambios realizados han sido para cumplir los requisitos del módulo, o mejorar y simplificar la experiencia del usuario. Entre ellos se encuentran la inclusión de una barra de menús, que se explicará en la sección de *Interfaz*.

2. Fase de Implementación:

- **Lógica:**

La lógica de la aplicación se ha realizado mediante el uso de diferentes clases. En esta memoria se procederá a explicarlas en profundidad según el orden en las que la muestra el Eclipse.

- Comenzando por el paquete *boxes*, donde se encuentra todo lo relacionado con las diferentes casillas que integran el panel de botones. La clase principal corresponde a *Box*, la cual es la clase padre del resto, donde se crean las funciones de añadir los puntos correspondientes al panel (los puntos que se añaden serían los que cada clase lleva en el nombre), y, en el caso de las casillas vacías, x2 y especiales, no se añaden puntos, ya que la actualización de estos se realiza en un método de otra clase.



- La clase *Board* corresponde al tablero de casillas del panel de botones, aquí se crea el Array de casillas, haciendo que en un primer momento esté vacío, posteriormente, mediante métodos que crean las casillas pertinentes en posiciones aleatorias se añaden el resto de tipos, a su vez, es aquí donde se realiza la impresión por consola de la posición de las casillas, con el objetivo de facilitar las pruebas del programa.
- La clase *Client* es la encargada de guardar todos los atributos del mismo, tales como el código (o ID), el nombre del cliente y si este puede o no jugar. Es empleada para cargar los ficheros de clientes y reescribirlos, mediante métodos como el *fileToString()*, que guarda los valores del cliente en el formato del fichero solicitado, es decir: *ID@NOMBRE@PUEDE_JUGAR*.
- La clase *FileUtil* es la que se encarga de cargar los ficheros de clientes y regalos en listas, para su posterior uso en el programa. También se encarga de sobrescribir los clientes actualizados (cuando un cliente juega, modificar el archivo para que no pueda volver a jugar), y de escribir los regalos seleccionados por un usuario en el fichero de *entregas.dat*.
- La clase *Game* cumple la funcionalidad principal de la aplicación, cuando se ejecuta, su constructor llama al método *startGame()*, que se encarga de inicializar las listas y las casillas del tablero.

Es utilizada al principio para cargar los clientes del fichero *clientes.dat*, y de comprobar que el id introducido en la ventana de código corresponde con el de un cliente o no, y si corresponde, si puede jugar o no. También se encarga de actualizar si el cliente ya ha jugado.

La segunda función corresponde al panel de casillas, es donde se realiza la suma de los puntos obtenidos y donde se van reduciendo las tiradas correspondientes, también comprueba si el juego ha terminado cuando los disparos lleguen a cero.

Por último, también se encarga de los regalos, primero, carga los regalos en español por defecto *regalos.dat*, pero si el cliente selecciona el idioma a inglés, dicho fichero se cambiaría. El método *reloadSelectedGifts()* se encarga de traducir la lista de regalos seleccionados en caso de que se

haya cambiado el idioma en plena ejecución (ya que es una opción extra de este módulo).

Dentro de la sección de regalos, se encarga de añadir a la lista de seleccionados las unidades del producto elegido, comprobando si tiene los puntos suficientes y mostrándolo por pantalla (la lista de regalos de la interfaz gráfica); a su vez, también se encarga de eliminar de la misma el producto y las unidades que haya seleccionadas.

Y para finalizar en la sección de regalos, se encarga de, una vez que el usuario confirme que quiere los regalos seleccionados, guarde dichos regalos (con el formato pedido en el módulo) en el fichero *entregas.dat*.

- La clase *Gift* es la encargada de guardar todos los atributos del mismo, tales como el código (o ID), la sección del regalo, la denominación del regalo, la descripción del regalo y los puntos que cuesta ese regalo.

Además, si se añade más de uno, se almacenan sus unidades aquí, y, si es un viaje, la fecha seleccionada y la observación escrita por el cliente también se guardan aquí.

Al igual que la clase *Client*, la clase *Gift* cuenta con un método *fileToString()* que guarda los valores del regalo en el formato del fichero solicitado, es decir: *ID_CLIENTE@ID_ARTICULO[@FECHA_VIAJE[@OBSERVACIONES_VIAJE]]*.

- *Interfaz:*

La aplicación cuenta con la siguiente interfaz gráfica:



En la ventana de la selección de idioma se han elegido dos botones con el icono de la bandera de cada país con idioma soportado (España y Reino Unido), para dar un

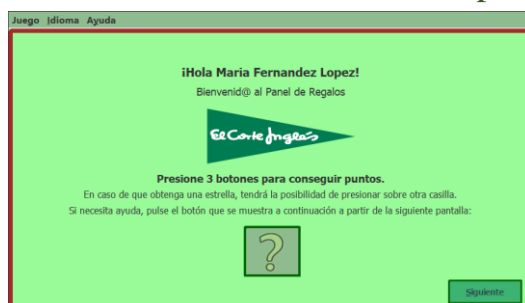
aporte más visual al usuario, ya que, si se hubiera hecho como estaba originalmente previsto, el usuario no tendría tanta información visual como actualmente. El botón siguiente simplemente sirve para ir a la siguiente ventana, la cual sería la introducción del código.

Esta ventana contiene principalmente el JTextField donde introducir el código, se ha programado para que, solo se pueda escribir 3 letras y 3 dígitos, como el código es en mayúsculas, también se ha programado que aunque se escriba en minúscula, se pase a mayúscula, con el fin de simplificarle la experiencia al cliente.



También cuenta con 2 botones, para ir a la ventana anterior (y poder cambiar el idioma) y a la ventana siguiente, que da la bienvenida a la aplicación al usuario.

La ventana de bienvenida se emplea para informar al usuario del



funcionamiento del juego, y le avisa del botón que tendrá que pulsar (en siguientes ventanas), en caso de que necesite ayuda (ya que, le llevará con acceso directo a la ayuda.

Una vez, le da al botón siguiente, el cliente accede a la ventana del panel de botones.

Esta es la ventana del panel de botones, que se divide en 3 zonas principales. En primer lugar, los 25 botones que forman el panel, donde el cliente debe presionar (al menos 3 veces). También es clave el JTextField que muestra los puntos que el usuario va acumulando y el JTextField de las tiradas restante, que se inicializa a 3.



Cuando el usuario ha agotado todas las tiradas disponibles, se le avisa por mensaje, mostrándole todos los puntos conseguidos y una ayuda sobre como canjear sus puntos en la siguiente ventana.



La ventana final de la aplicación corresponde al panel de canjeo del programa, que se divide en varias zonas principales.

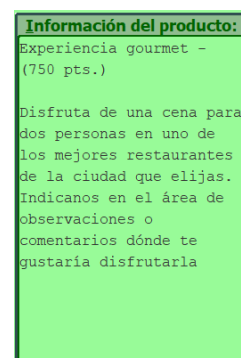
La más destacable es el panel central que contiene a todos los productos disponibles en el fichero, donde, si el cliente quiere añadir un producto, debe presionar sobre este, cabe mencionar que, tal y como se explica en el mensaje de la ventana anterior, si quiere añadir más de un producto debería seleccionar en el JSpinner de las unidades abajo a la derecha, se ha elegido este componente con motivo de poder dejar libre elección al usuario sobre si quiere añadir pocas unidades (presionando los botones que lo integran) o muchas, mediante la escritura en el campo dedicado para ello.



A la izquierda del panel central se encuentra la zona dedicada al filtro de productos, con 5 botones que habilitan los de ese tipo de sección y deshabilitan el resto, y un JSlider que inhabilita todos los que tengan puntos superiores al valor mostrado, se ha empleado un JSlider y botones para cumplir los requisitos del módulo de una manera efectiva.

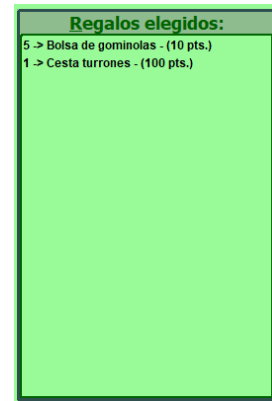
Al igual que en la anterior ventana, existe un JTextField de puntos, que cumple la misma función, salvo que ahora muestra los restantes, ya que, cada vez que canjea dichos puntos se irán restando de este campo.

A la derecha se observan dos JScrollPane que contienen un JTextArea y un JList, respectivamente.



El JTextArea muestra por pantalla la información (denominación, puntos que cuesta y descripción) del producto sobre el que se ha posado el ratón, se eligió este componente por la versatilidad que daba ante la posibilidad de escribir y reescribir simplemente cuando el ratón se posase sobre un botón en concreto.

En el caso del JList, mostraría la lista de productos que se van añadiendo a la aplicación, se ha elegido este componente porque da la opción a eliminar productos de forma mucho más sencilla, para ello se debe seleccionar el producto que quiera eliminar y posteriormente presionar el botón del mismo nombre. Si quiere eliminar más de una unidad del mismo producto, debe seleccionar en el JSpinner las unidades que quiere quitar y posteriormente hacer lo descrito anteriormente, si por algún casual selecciona más unidades de las que tiene de dicho producto, se le avisará al usuario por pantalla.

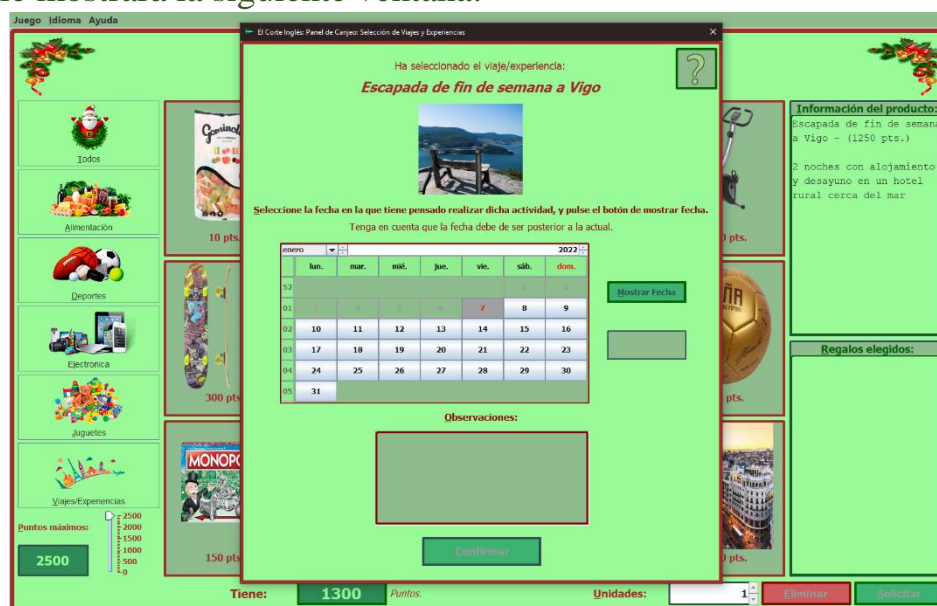


Por último, los botones de eliminar y solicitar, la función del botón eliminar ya se ha descrito en la explicación del JList. La función del botón solicitar es que los productos que se encuentran en el JList (es decir, aquellos elegidos por el cliente) se pasen al fichero de *entregas.dat*, y terminando el curso del cliente en la aplicación.

Se dan 2 casos posibles cuando se da a este botón:

- Cuando el cliente tiene puntos pendientes, se le avisa de que los perderá si prosigue, y se muestra una lista de los productos que ha elegido, preguntándole si está seguro de ello.
- Cuando el cliente no tiene puntos pendientes, se le muestra la lista de los productos seleccionados y se le pregunta si está seguro de que esos productos son los que quiere canjear.

Además de esto, en caso de que el cliente seleccione un viaje se le mostrará la siguiente ventana:



En esta ventana, el cliente deberá escoger la fecha en el componente JCalendar, el cual se ha elegido por respetar el principio de familiaridad, ya que es más común entre los usuarios la búsqueda de una fecha en un calendario. Tal y como se explica en el texto predecesor al calendario, una vez elegida la fecha, debe presionar el botón *Mostrar Fecha* (esta decisión, pese a ser más larga, ha tenido que tomarse debido a que el JCalendar no contiene la posibilidad de emplear un *ActionPerformed*). Tras esto, el usuario podrá escribir una observación acerca del viaje, si así lo solicita, en el JTextArea correspondiente, se ha elegido este componente debido a que, el cliente puede redactar la observación que tenga sin miedo a tener un límite de caracteres.

Respecto a la barra de menú, cuenta con varias opciones:

- Sección de *Juego*: Con las opciones de *Nuevo* y *Salir*. Realizadas para respetar el principio de usabilidad en la aplicación.
- Sección de *Idioma*: Que permite el cambio del idioma en plena ejecución de la aplicación.
- Sección de *Ayuda*: Que contiene la pestaña de *Contenidos* (ayuda para el usuario) y la ventana de *Acerca de*, que da información al usuario sobre el desarrollador de la aplicación y el motivo de dicho proyecto.

- **Pruebas:**

- Escenario 1: Se ha pasado la prueba sin problemas, ha sido posible replicar los pasos del caso, introduciendo el código a la primera, obteniendo el mismo valor que en el escenario (gracias a la impresión por consola, lo que da más facilidad para realizar las pruebas). Se ha conseguido canjear ambos viajes con las fechas aproximadas solicitadas en el escenario, y con el comentario correspondiente en la segunda actividad. Se han dejado sin canjear 50 puntos, lo que ha avisado de que los perdería y pediría confirmación, tras presionar “Si”, los regalos se pasan al fichero de *entregas.dat*.

Nombre: Ana

Edad: 41

Ocupación: 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.



- Escenario 2: Se ha pasado la prueba sin problemas, ha sido posible replicar los pasos del caso, debido a las pistas visuales ha logrado introducir el código sin dificultad, se ha podido obtener 300 puntos y en el panel de premios se han elegido 3 unidades del mismo artículo de 100 puntos (en los ficheros dados, sería la cesta de turrone).

Nombre: Antonio

Edad: 82

Ocupación: 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 pq 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 costra 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.




- Escenario 3: Se ha pasado la prueba sin problemas, ha sido posible replicar los pasos del caso, la interfaz se ha adaptado al idioma del usuario, ha fallado el identificador y se ha avisado al usuario, ha conseguido introducirlo correctamente al

segundo intento debido a la información que proporciona la aplicación, en el panel se han obtenido 100 puntos. Los casos de prueba con productos de 80 puntos, 30 puntos, 40 puntos y 60 puntos no se han podido realizar (ya que el fichero de *regalos.dat* no contiene ningún producto con tal coste), pero, probándolo con valores equivalentes (seleccionando 8 unidades del producto de 10 puntos primero, y luego 3 unidades del mismo producto, no se ha podido añadir, sin embargo, cogiendo 6 y 4 unidades respectivamente sí que ha sido posible), tras darle a solicitar, el usuario ha confirmado que esos son los productos que quiere y ha terminado su curso en la aplicación.

De esta forma, no ha sido necesarios realizar cambios en la aplicación tras pasar estas pruebas/escenario. Cabe destacar, que se han realizado más pruebas con miembros de la unidad familiar, de todas las edades posibles y sin ningún conocimiento previo del programa.

A raíz de esas pruebas, se han modificado y añadido textos o ventanas de información de manera que le quede más claro el funcionamiento de la aplicación al futuro usuario que la utilizará.



Nombre: George
Edad: 26
Ocupación: 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, pq 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 que 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.