6-1-2024

Jorge Blanco Sánchez

Uo293697

Módulo

Nos vamos de Vacaciones

Contenido

[Introducción 2](#_Toc155534319)

[Estructura 2](#_Toc155534320)

[Lógica 3](#_Toc155534321)

[Paquete booker 3](#_Toc155534322)

[Paquete Discount 8](#_Toc155534323)

[Paquete Game 9](#_Toc155534324)

[Paquete Service 10](#_Toc155534325)

[Interfaz 10](#_Toc155534326)

[Pruebas 18](#_Toc155534327)

# Introducción

El objetivo de este módulo es crear un aplicación que represente la página de reserva de habitaciones de diferentes castillos por Europa, para ello se presenta en una ventana dichos castillos donde el usuario puede elegir el que mejor le encaje, además se proporciona una opción de filtrado, tanto de precio como de características de los castillos.

En este documento se explicará detalladamente la lógica y la interfaz que conforma la aplicación y las diferentes pruebas de personas que se utilizaron para validarla, además he incluido la internacionalización, es decir, la aplicación esta traducida y localizada en Ingles y en Español, también se permite al usuario consultar su descuentos, de esta manera no se le obliga a jugar para que le índice el sistema que ya posee un descuento, por último consideré añadir la generación de recibos, en caso de que el usuario lo necesitara.

# Estructura

En primer lugar se explicara la estructura de los paquetes:

* Src
  + castleBooker
    - model
      * booker
        + Booker.java
        + Castle.java
        + Persona.java
        + Reserva.java
      * discount
        + Discount.java(Clase enumerada)
        + DiscountData.java
      * game
        + Casilla.java
        + Dice.java
        + Game.java
    - service
      * App.java
    - ui
      * frame
        + JScrollPaneWithPanel.java(classe propiá que herada de JPanel)
      * GameUi.java
      * VentanaDescuento.java
      * VentanaEdad.java
      * VentanaGuardadoDescuento.java
      * VentanPrincipal.java
      * VentanReserva.java
    - Main.java
  + img
  + rcs
  + útil
    - CastleFileUtil.java
    - DiscountFileUtil.java
    - FileUtil.java

Dentro de la carpeta img se encuentran todas las imágenes que necesita la app, y en rcs los archivos de idioma

# Lógica

Toda la lógica de la aplicación se encuentra en el paquete model, es decir todos los archivos dentro de este forman parte de la lógica

Este paquete está divido en tres subpaquetes, que se procederán a describir a continuación:

## Paquete booker

En este paquete principal se encuentran 4 archivos java

En el primero, con nombre Booker.java, se puede considera el más importante, algunos de los métodos más relevantes de esta clase son:

Texto

Descripción generada automáticamente

El constructor de esta clase se encarga de leer los castillos, crear una estructura de datos con los diferentes tipos de encantamientos que tienen los castillos y de establecer la localización de la aplicación según la localización de la máquina virtual de Java

Texto

Descripción generada automáticamente

Este método es uno de los más importante ya que es el encargado de leer el fichero de los castillos gracias al método estático que se encuentra en la clase CastleFileUtil, a su vez compruebo que si la lectura produce un erro este lance la excepción NullPointerException con un mensaje detallado



Este método se encarga de leer de cada castillos sus diferentes encantamientos y guardarlos en un TreeSet, decidí utilizar esta estructura por el orden de los elementos y también para asegurarme de que un tipo de encantamiento no aparezca repetido, tras esta comprobación y haciendo uso del toString del TreeSet se puede obtener una String ordenada con los diferentes encantamientos, a su vez está pensado para que en el futuro, si aparece algún encantamiento nuevo, no necesite modificarse el código.

Texto

Descripción generada automáticamente

Uno de los método que me gustaría resaltar es el setCastles(), el cual establece como array principal de la aplicación los castillos que se van a mostrar, este recibe los filtros que se desena aplicar y la cantidad máxima de dinero que el cliente desea gastar, y a partir de esto datos se genera la lista, si el valor del dinero es máximo y no se proporciona ningún filtro, se devuelve la lista con todos los castillos, siendo la variable totalCastillos el array donde se almacena todas la información de los castillos

Clase Castle.java

Esta clase está pensada para guardar la información de cada castillo, haciendo que cada línea del fichero de castillos corresponda con un objeto Castillo

La gran mayoría de esta clase se compone de getters y setters, aunque hay dos métodos que me gustaría resaltar

Texto

Descripción generada automáticamente

Este método recibe la string con el formato de texto que se va a presentar al usuario, el método se encarga de recibirla y poner en cada @ que se encuentra la información correspondiente, para que posteriormente la interfaz la reciba y la muestre al usuario, el formato es obtenido por la clase Booker tras leerlo del fichero de propiedades correspondiente, según la localización de la aplicación

Texto

Descripción generada automáticamente

Este método realiza algo parecido al anterior, lee los encantamiento del castillo y los pasa a texto, para que cuando se soliciten se muestre el encantamiento de manera explicativa, por ejemplo, si un castillo posee el encantamiento De, este método lo transforma a descenso de las temperaturas

Clase Persona

Esta clase representa al usuario que actualmente está utilizando la aplicación, con esta clase se pueden conservar datos temporales sobre el usuario, así como el DNI o el nombre, y si en algún momento se le pide datos que el sistema ya conoce no se lo preguntarán, ya que será posible recuperarlos

En su gran mayoría también son setters y getters aunque si tuviera que resaltar uno de ellos sería el que se encarga de inicializar a la persona, ya bien cuando este decide salir o cuando termina la reserva

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Clase Reserva

Esta en la última clase del paquete y sirve para representar el castillo que el usuario desea reservar

Texto

Descripción generada automáticamente

El método valida() se encarga de comprobar que las habitaciones sean correctamente ocupadas por una o dos personas

Texto

Descripción generada automáticamente

El método serialize que se encarga de crea una línea con todos los datos de la reserva para posteriormente guardarlo en el fichero de reservas

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Por último el getReservaToSave crea la String con el recibo, este método solo se invocará si el usuario decide guardar dicho recibo y funciona de la misma manera que el método info de la clase castillo

## Paquete Discount

El paquete discount posee dos clases

La primera es una clase enumerada con nombre Discount, la cual posee lo descuento que se dan en la aplicación

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

La clase DiscountData, la cual se utilizará para almacenar los descuentos de cada usuario, posee un HashMap para poder almacenarlos y asegurarme de que no se dupliquen entradas, a su vez esta estructura proporciona acceso rápido para poder saber si un usuario posee o no descuento

Algunos método a detallar son:

Texto

Descripción generada automáticamente

El cual lee del fichero todos los descuentos y los carga

Texto

Descripción generada automáticamente

Guarda todos los descuento con el formato correspondiente al fichero, el parámetro boleano define el tipo de escritura, el false indica que se sobrescribirá todo el archivo

Texto

Descripción generada automáticamente

Guarda en el fichero una nueva entrada, a diferencia del anterior este solo añade al final

## Paquete Game

En este paquete se representa toda la lógica que está relacionada con el juego consta de cuatro clase aunque la principal es la clase Game

Texto

Descripción generada automáticamenteEsta clase posee métodos importantes como fillBoard el cual crea en una matriz, y la rellana con los difernetes tipos de casillas, los cuales pueden ser pared, fantasmas( de 4 tipos distintos) fantasma jefe y cazafantasmas, este método crea dicha matriz con forma piramidal, siendo la base los cazafanstasmas y en cada piso se añaden los diferentes tipos de fantasmas, en el último piso, el cual será de una única casilla, se coloca al jefe, y el resto de casillas que se consideran vacías, serán llenadas por pared.

Para asegurarme que el método creara el tablero correctamente, realice un teste que se puede encontrar en la carpeta test

Texto

Descripción generada automáticamente

Este método también se puede considerar importante se encarga desplazar el cazafantasma tantas posiciones como han salido en el dado, recibe la posición respecto a la columna

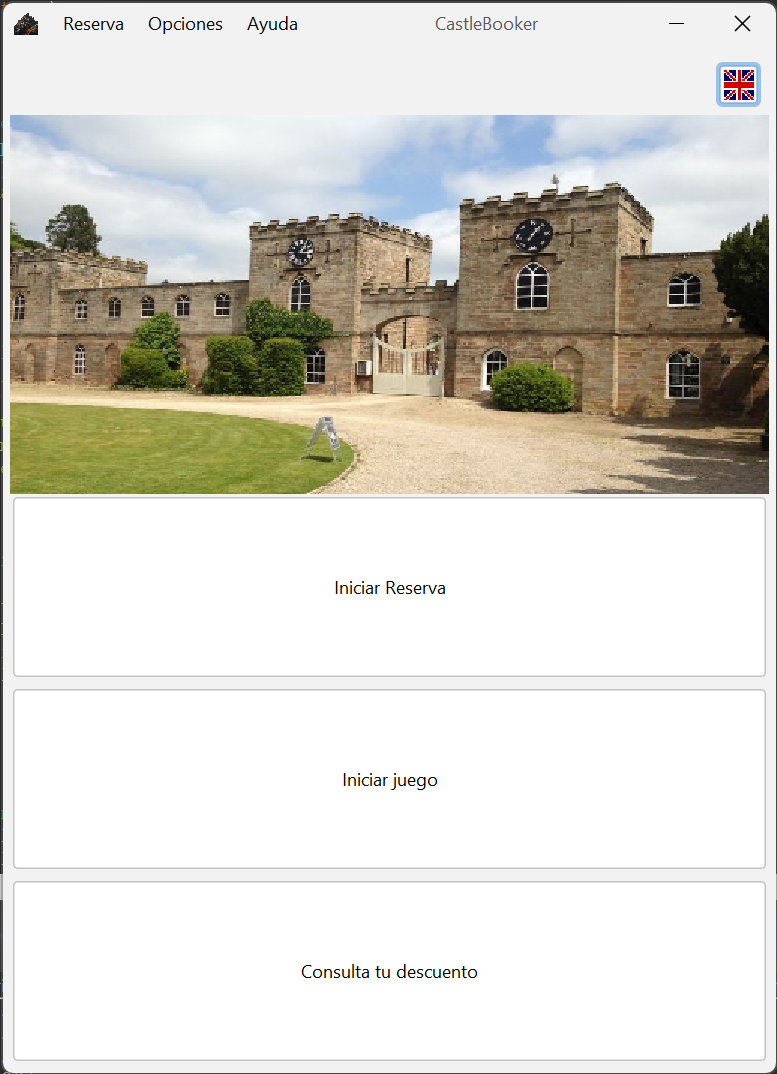
## Paquete Service

Este paquete incluye la clase que servirá de puente entre la interfaz y la lógica

# Interfaz

En este apartado se mostraran las diferentes ventanas que conforman la aplicación y se compararan con el diseño del wireframe

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

En esta venta se han realizado dos cambios, el botón del interrogante(ayuda) que se sustituyó por acceso con el botón F1 y dicho botón también puede ser encontrado en la barra de menú, el botón de accesibilidad fue eliminado ya que procure que al aplicación fuera accesible y no hubiera otra versión más accesible

De esta ventana destacaría los tres botones principales y el icono del idioma; estos botones principales considero que es la mejor manera de dar pasos a las respectivas ventanas que se activan tras su pulsación, además se proporciona acceso por teclado, por último el icono del idioma, el cual es un botón, me pareció una forma rápida para el usuario no hispanohablante, se diese cuenta de que existe una versión en Inglés, de esta forma no tendría que estar buscando entre los elemento del menú para saber si existe dicha versión, teniendo en cuenta que igual no es capaz de leer el menu

Texto

Descripción generada automáticamenteUna captura de pantalla de un celular con texto e imágenes

Descripción generada automáticamente con confianza media

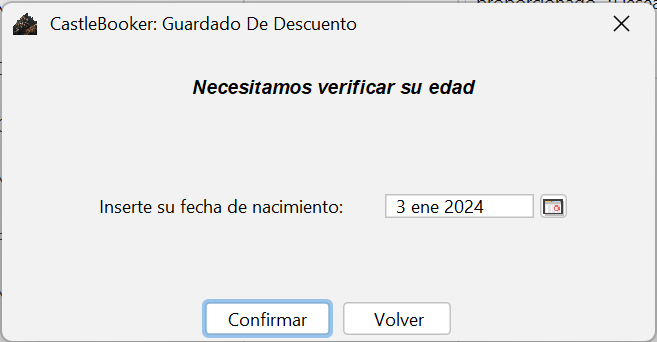
Respecto al diseño original solo se ha eliminado el botón de ayuda(?), pero este está incluido en la barra de menú y posee ayuda con el botón F1, esta ventana esta compuesta por botones, considero que son la mejor opción ya que poner cualquier otro componente no sería adecuado

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteTabla

Descripción generada automáticamente con confianza media

En estas ventana no se ha realizado cambios, los componentes principales son botones los cuales al pulsar abren las diferentes ventanas asociadas a ellas

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

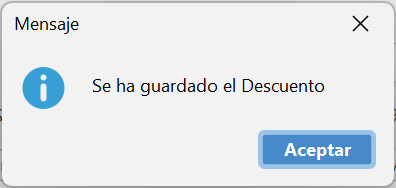
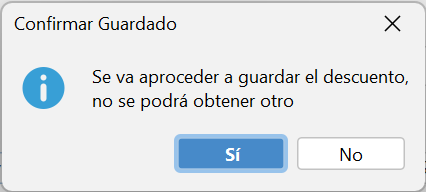
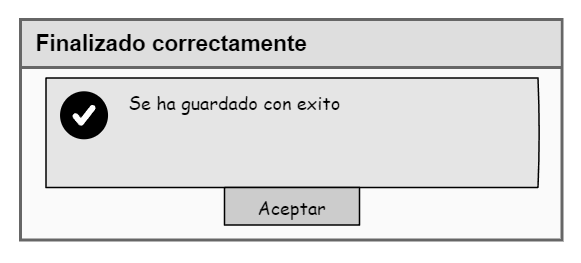
En esta ventana se cambió la forma de solicitar los datos de la fecha, es poco accesible utilizar combobox, para ello utilice la librería Jcalendar, la cual proporciona al usuario poder seleccionar más cómodamente la fecha

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente Interfaz de usuario gráfica

Descripción generada automáticamente

En esta ventana no sea modificado respecto al diseño original, el principal componente es el JTextField el cual permite al usuario inserta su DNI, este componente se adapta mucho mejor que un JTextArea, los otros 3 botones permiten guarda el descuento o volver a pantallas anteriores



Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En estas dos ventanas no se han modificado respecto al diseño original, su uso es informativo, con ello se pretende avisar del estado de la aplicación y confirmar el guardado o pedirle confirmación al usuario

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Esta ventana no se ha modificado, su componente mas relevante es el JTextField el cual permite al usuario inserta su DNI para que la aplicación pueda comprobar si posee o no descuento, este componente me pareció el más adecuado para solicitar texto al usuario

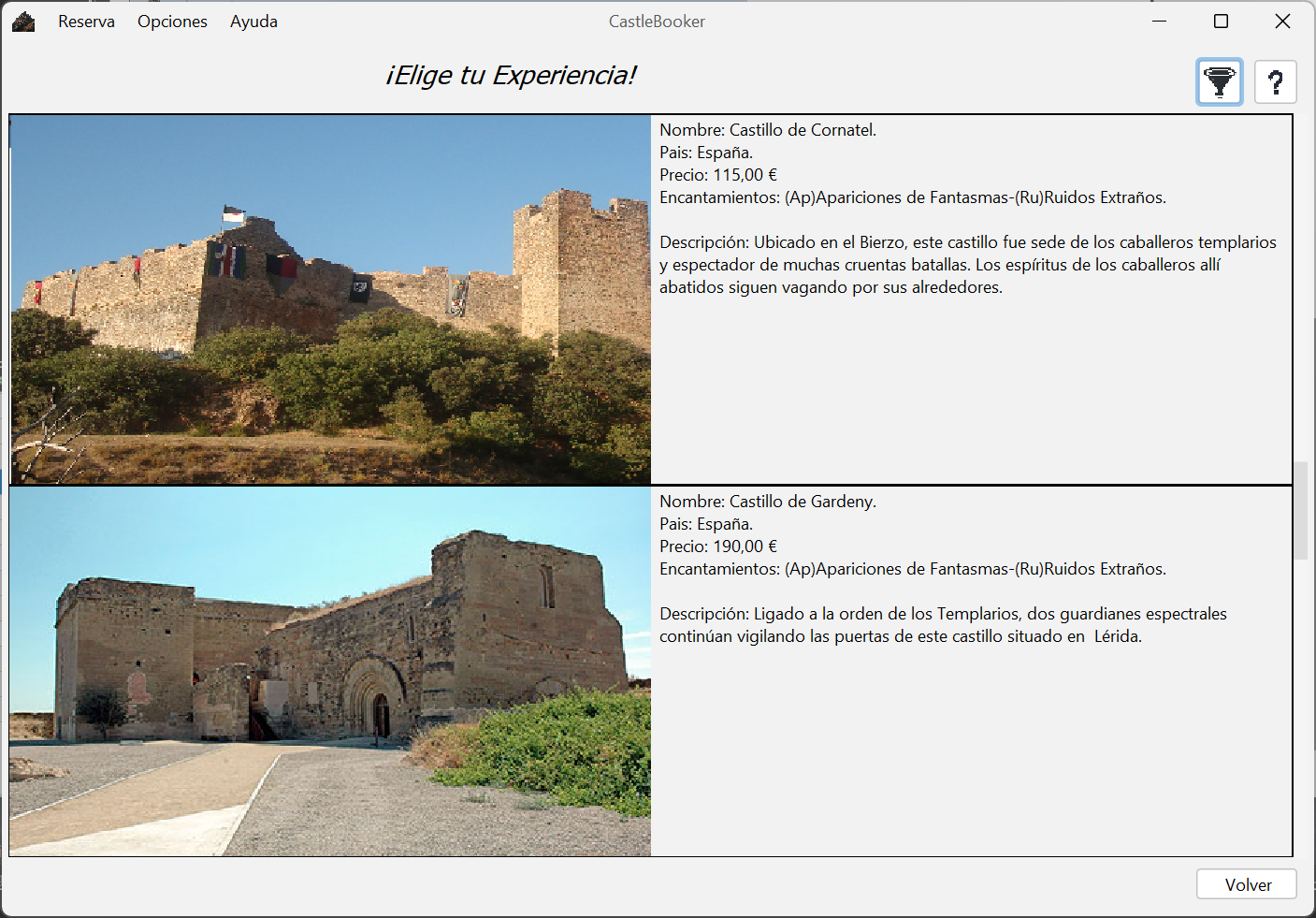
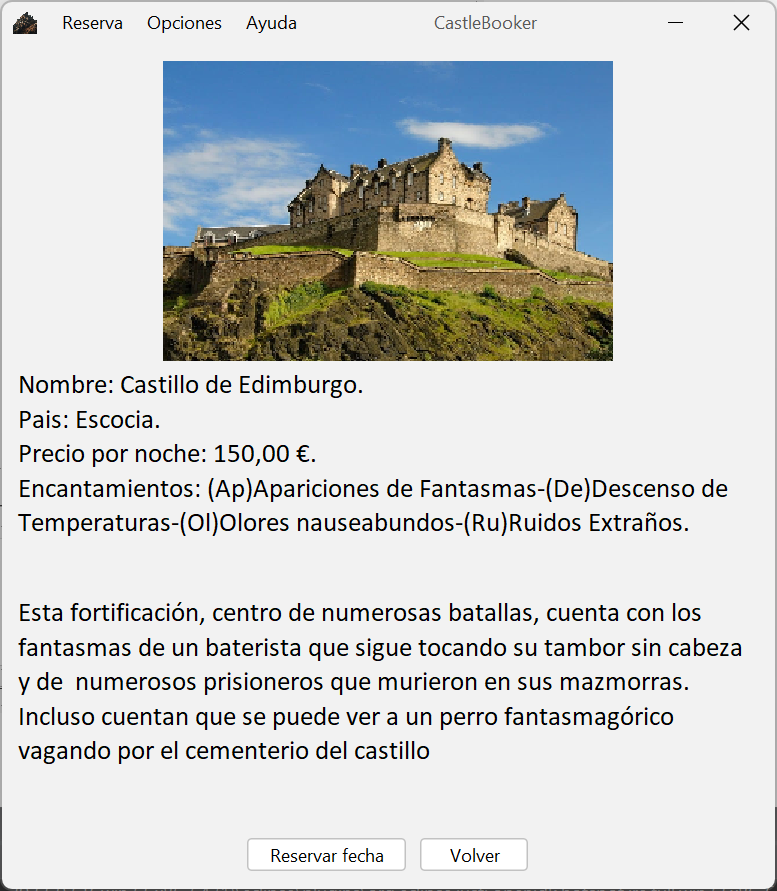


Imagen que contiene Texto

Descripción generada automáticamente

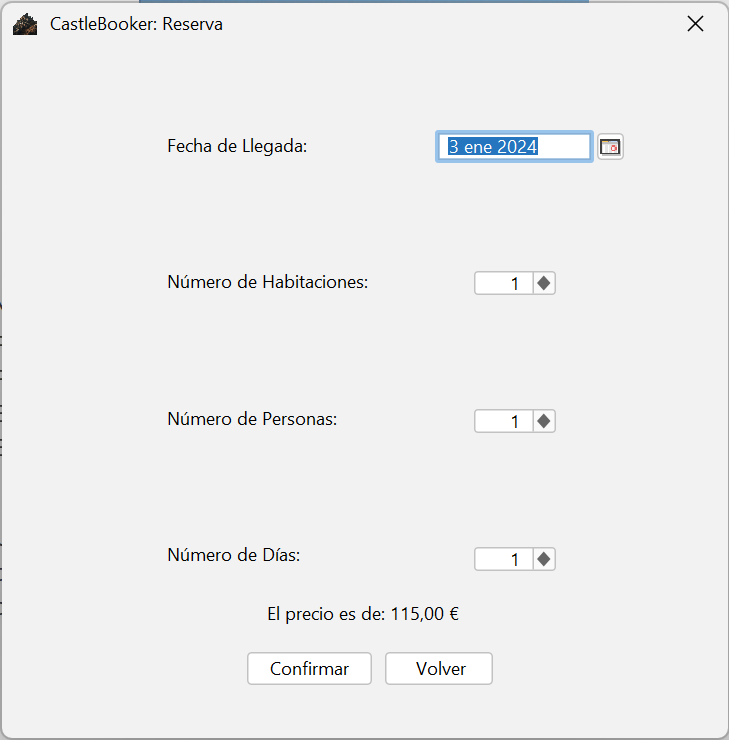
En esta ventana en vez de mostrar de cuatro en cuatro se muestran de dos en dos; esta ventana la podría considerar la más compleja, el componente principal es un panel central el cual diseñe extendiendo de la clase JPanel, dentro de este panel existen dos componentes, el central un panel con un Grid de 2 filas y una columna, donde en cada fila existe un panel con toda la información del castillo, y en la zona este una JScrollBar, la cual permite al usuario ver y desplazarse por los diferentes castillos, esta ventana también posee un panel con todos los filtros posibles, el cual se muestra cuando se presiona el botón de filtro, dicho panel está formado por checkBox, los cuales son la mejor manera de permitir múltiple selección, si fuera de selección única el componente que mejor se adapta sería un radioBoton por esta razón no empleé este componente para el filtro, por último la selección de precio funciona con un JSlider y es acompañado de una etiqueta, de esta manera se hace de manera más cómoda la selección de precio máximo

Imagen que contiene Texto

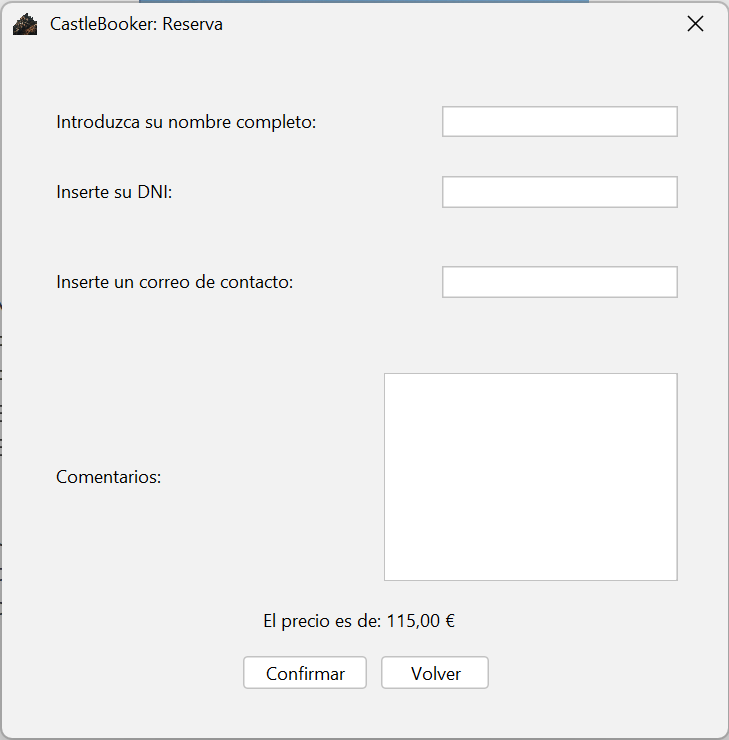
Descripción generada automáticamente

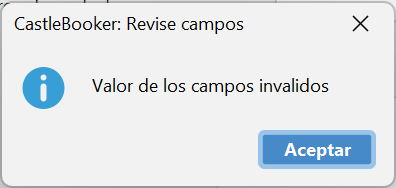
Se ha conservado el diseño original, se emplean dos JTextArea para escribir el texto informativo del castillo, al ser una cantidad de texto grande, este componente se adapta mejor que una JTextField

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica

Descripción generada automáticamenteEn esta ventana he cambiado el formato de calendario, ya que considero que los combobox no son adecuados para esta situación, también cambié los combobox del número de días, personas y habitaciones por spinners, los cuales me parecen que se adaptan más a esta situación, permitiendo que a los usuarios que desean una gran cantidad de habitaciones o días puedan teclear el número sobre la caja del spinner

En estas ventana no se han realizado cambios, permiten al usuario insertar su información personal en un JTextField, como no es una gran cantidad de texto este componente se adapta mejor, por otra parte para los comentarios he empleado un JTextArea el cual si permite la visualización del texto escrito por el usuario en diferentes líneas, en caso de que este ocupe más de una línea

Diagrama

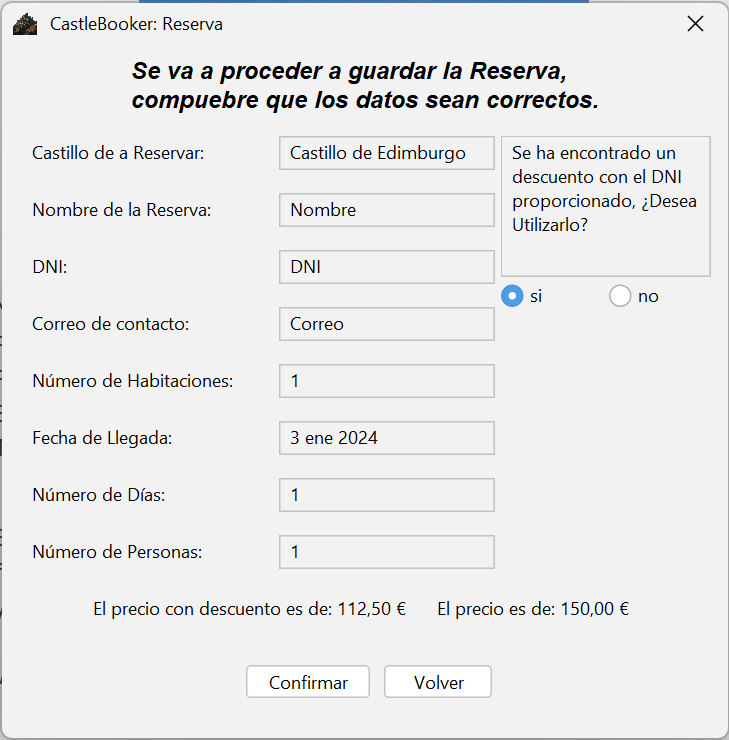
Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Estas dos ventanas con diferentes textos se corresponde con los mensajes informativos cuando los campo sean inválidos, además en el caso del panel con información personal, se resaltará el campo que produjo un error

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente



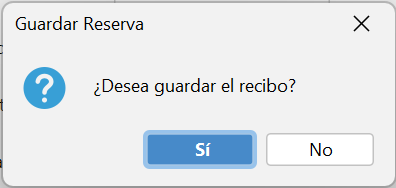
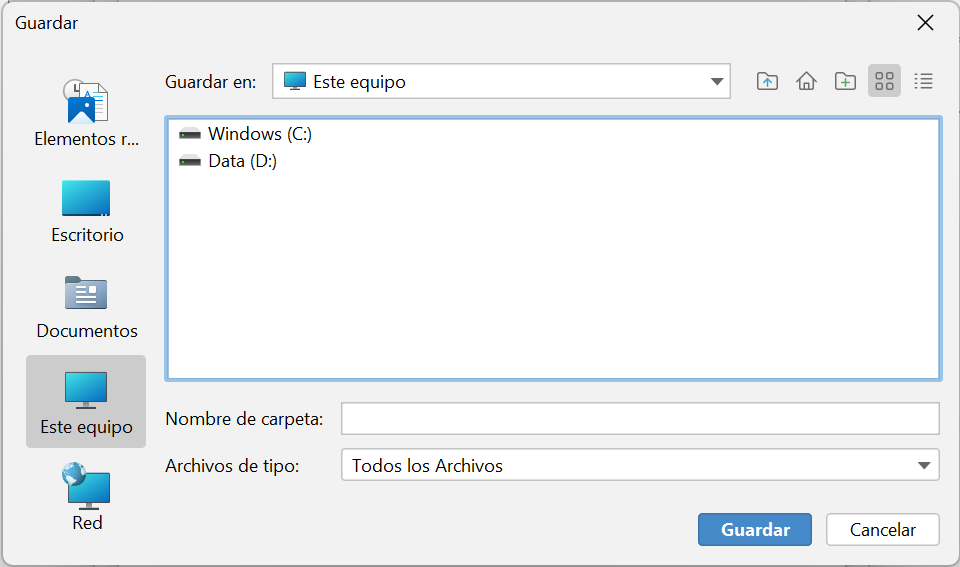
En esta ventana se ha añadido más información a mostrar, como el nombre del castillo, correo de contacto o número de habitaciones, esta información la he añadido ya que la considero importante en una ventana de confirmación, si dicha información no estuviese tendría que obligar al usuario a volver a pasos anteriores para que se pudiera comprobar, haciendo que esta ventana sea poco usable, por otra parte elimine los comentarios en esta ventana ya que no los he considerado información importante. Respecto a los componentes mas relevantes me gustaría destacar el uso de JTexFields no editables, los cuales proporcionan ver la información que el usuaria a introducido en los campos de ventanas anteriores, por ultimo el uso de dos radiobuttons, al ser opciones excluyentes considero que es el componente que mejor se adapta a esta situación

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

No se han realizado cambios, son mensajes informativos



Estas dos ventanas no se habían tenido en cuenta en el diseño original, decidí incluirlas ya que revisando mis escenarios, una de las personas solicitaba la posibilidad de tener un comprobante de la reserva que había hecho, para ellos lo primero que hice fue preguntarle si deseaba dicho comprobante con un JOptionPane, si lo confirmaba le permitía elegir la carpeta donde quisiera guardarlo con un JFileChooser, si no hubiera añadido el JFileChooser la ubicación donde se guardaría sería entre los archivos del proyecto, haciendo que usuarios pocos expertos, no supieran donde se realizó la instalación de la aplicación y no pudieran encontrar dicho comprobante

# Pruebas

Persona

**Forma, Cuadrado

Descripción generada automáticamente**

Nombre: María

Edad: 25

Ocupación: Estudiante

Escenario:

María y sus amigos están en búsqueda de una nueva experiencia para poder hacer todos juntos en verano para este año.

Tras una larga búsqueda encontraron una página web en la que puedes reservar una experiencia única en un castillo encantado, es decir, con experiencias paranormales.

Deciden todos juntos reservar para este verano la experiencia completa, pero justo cuando van a reservar se fijan que pueden jugar a un juego para obtener un descuento, tras dos intentos deciden quedarse con el descuento conseguido aunque no sea el mayor posible

Cuando se realizó la evaluación con esta persona no se habían encontrado fallos, el diseño ya había contemplado la opción de filtrado y también se informa si es o no el descuento máximo

**Forma, Cuadrado

Descripción generada automáticamenteForma, Cuadrado

Descripción generada automáticamente**Con esta persona tampoco encontré fallos, ya que se da opción de reservar sin tener que pasar por el juego, como en el caso anterior el sistema ya contempla el uso de filtros.

Nombre: Roberto

Edad: 43

Ocupación: Autónomo

Escenario:

Roberto quiere reservar un viaje que le ha prometido a su hijo en verano, en un castillo encantado, ya que si aprobaba todas las asignaturas le daban eso a cambio.

Tras ir a la agencia de viajes se dispone a reservar, pero justo antes de darle a aceptar, su hijo le dice que hay un juego para poder obtener un descuento para su viaje. Tras muchos intentos no consigue ninguno, entonces le deja paso a su hijo. Como su hijo es muy competitivo en juegos hasta no conseguir el mayor descuento no paro de jugar.

Nombre: Juan

Edad: 81

Ocupación: Jubilado

Escenario:

Juan es una persona jubilada de 81 años, quien está cansado de su rutina y quiere disfrutar de una experiencia única, tras pasar por la agencia de viajes vio un juego el cual ni se molestó a mirar, ya que él no sabe cómo funcionan los juegos de esta época.

A pesar de ello le genera intriga y se percata de que puede saltar el juego y ver que puede hacer una reserva a un castillo en el cual hay distintos niveles paranormales. Como tiene una arritmia en el corazón evitara los sucesos que produzcan sustos seleccionando en su reserva, como objetos cambiando de lujar y olores nauseabundos

Con Roberto no se han realizado cambios, aunque de paso al hijo a obtener el descuento, si se desea guardar tendrá que ser Roberto el que proporcione su DNI ya que no se permiten a usuarios menores de edad la obtención de descuento o la formalización de la reserva

**Forma, Cuadrado

Descripción generada automáticamenteForma, Cuadrado

Descripción generada automáticamente**Al igual que en el caso anterior Mara podrá jugar al juego, pero tendrá que ser su madre la que proporcione su datos para guardar el descuento

**Nombre**: Marga

**Edad:** 45

**Ocupación:** Abogada

**Escenario:** Marga quiere regarle a su pareja

un viaje por su aniversario. Mientras espera a

que le atiendan en la agencia de viajes, se

acerca a un terminal en el que se anuncia una

estancia inolvidable y la posibilidad de obtener

descuentos en la reserva.

Ve claro por las explicaciones en el terminal

que jugar no le compromete a reservar, así que

prueba suerte. No obtiene ningún descuento,

pero la aplicación le ofrece jugar de nuevo sin

compromiso de reserva. Repite el juego y esta

vez obtiene un descuento del 10%. La

aplicación le deja claro que podría jugar de

nuevo para obtener un descuento mayor, pero

como ya le van a atender en el mostrador,

prefiere irse y no guardar el descuento

obtenido. Así, si no reserva un viaje tradicional,

volverá para tratar de obtener el descuento

máximo y reservar en un castillo encantado.

Nombre: Mara

Edad: 15

Ocupación: Estudiante

Escenario:

Mara y su madre están buscando un plan de viaje para este verano, por lo que entran a una agencia de viajes de su barrio para informarse de los distintos viajes que pueden hacer este año.

Tras un rato hablando con la dependienta mara deja a su madre ya que se aburria de estar ahí sentada escuchando la conversación, tras dar una vuelta vio un viaje a un castillo encantado, el cual había un minijuego que ella podía jugar hasta que su madre terminara. Se pasa toda el rato que esta su madre buscando viajes con la agencia hasta que se va, pero ve que no puede guardar su descuento ya que no es mayor de edad.

Con Marga, no se realizaron cambios en el wireframe evaluado, la aplicación permite jugar, reservar y jugar o reservar, también a la hora de guarda el descuento se informa del tipo de descuento y si este es el máximo se indicará, también en la instrucciones del juego se indica que este no compromete a la reserva

**Forma, Cuadrado

Descripción generada automáticamente**

**Nombre**: Antonio

**Edad**: 79

**Ocupación**: Jubilado

**Escenario**: Antonio está a punto de cumplir 80

años y quiere celebrarlo con su familia con un

viaje especial que recuerden todos. Se acerca a

uno de los terminales de la agencia y ve

claramente que antes de reservar le compensa

jugar para tratar de conseguir un descuento.

Tras leer las instrucciones básicas del juego,

juega y gana un descuento del 25%. Le queda

claro que es el mayor que puede conseguir, así

que proporciona su DNI para usarlo después.

Pasa a reservar y analiza las posibilidades:

quiere ver los castillos que ofrezcan todos los

encantamientos menos los olores, que eso le da

mucho asco. Encuentra uno que le encaja. Indica

que va a reservar 5 habitaciones para 12

personas, pero el sistema le avisa que no es

posible, así que añade una habitación más.

Antonio indica que quiere utilizar el descuento

obtenido. En el resumen de la reserva

comprueba que está todo correcto, con el precio

final ya actualizado, así que formaliza la reserva

introduciendo los datos que le solicitan.

Los cambios que se había detectado en la evaluación del wireframe fueron aplicados, ya que la aplicación muestra si es el máximo descuento y si existe algún problema respecto a las habitaciones y los datos personales, tras la finalización de la aplicación no se encontraron más problemas

**Forma, Cuadrado

Descripción generada automáticamente**

**Nombre**: Maya

**Edad**: 11

**Ocupación**: Estudiante

**Escenario**: Maya ha acompañado a su madre

a una agencia de viajes, que tiene pensado

reservar para pasar juntas unos días en

Disneyland. Mientras espera, a Maya le llaman

la atención los terminales que ofrecen jugar

para obtener descuentos en una estancia muy

especial. Juega varias veces hasta obtener el

descuento máximo. Al pedirle el DNI para

guardar el descuento, le indica claramente que

ha de corresponder a una persona mayor de

edad, así que no lo guarda. Por curiosidad,

revisa la lista de hoteles disponibles y ve el tipo

de experiencias que se ofrecen. Va corriendo a

buscar a su madre para que no reserve en

Disneyland si no que cambie el viaje para ir a

uno de los castillos, que le mola mucho más.

Con Maya no se han encontrado ningún problema, ya que la aplicación cuenta con verificación de edad

Con estos 7 escenario se había evaluado el diseño del wireframe, aunque me gustaría destacar uno de mis escenarios personales que no se habían contemplado durante la evaluación

Forma, Cuadrado

Descripción generada automáticamente

**Nombre: Juan Luis**

**Edad: 83**

**Ocupación: Jubilado**

Juan Luis es un hombre desconfiado, no se fía de las tecnologías, le gusta tener comprobantes de pago por seguridad, ya que ha tenido algunos problemas con el banco.

A su edad, tampoco le gusta mucho este tipo de juegos, y prefiere una aplicación sencilla que le permita reservar su viaje sin pasar por muchas pantallas. No está interesado en el descuento, únicamente en la función de la aplicación y en unas vacaciones fáciles y cómodas.

Cuando comprobé mis propios escenarios me di cuenta que a Juan Luis le gustaría tener un comprobante, por ello implemente esta venta, también intenté que la aplicación fuera lo más sencilla posible.