



PRACTICA FINAL POO

Batería de pruebas



DIEGO JIMENO DE LA CALLE
DIEGO BORJA FARINANGO

1. Comprobar que la aplicación no cree excepciones inesperadas por introducir datos en un formato erróneo, como letras en campos de números o valores no permitidos.

Dentro de los menús, si intentamos introducir tanto letras como valores no soportados, se repetirá el menú pidiéndonos introducir otra opción.

- Dentro de 1.1.Añadir habitaciones:

Numero de habitación: comprobamos que el numero introducido no corresponda a ninguna habitación, y se corresponde al formato de habitaciones del hotel(6 plantas con 20 habitaciones cada una en formato [planta, habitación], 105 como planta 1 habitación 5).

Tipo de habitación: comprobamos que el tipo esté entre 1-4, y cualquier otro valor sea numérico o carácter nos repetirá la pregunta.

Precio de habitación: comprobamos que el valor sea válido, si no corresponde con el formato se pedirá de nuevo.

Comparte baño: pide si o no, si se introduce un numero repetirá la pregunta, así como si introduce una palabra distinta.

- Dentro de 1.2.Modificar habitación:

Para introducir el número de habitación sigue siempre el mismo patrón. Si la habitación no existe, da la opción de continuar o de abortar, este patrón también se seguirá en las validaciones de si y no.

Para introducir el precio, realiza lo mismo, si no es válido el valor introducido, realiza de nuevo la pregunta.

Para comparte baño, seguirá repitiendo la pregunta hasta recibir una respuesta válida de si o no.

- En 1.3.Informacion de una habitación:

Valida de la misma forma el número de habitación introducido, pero si no coincide con ninguna habitación registrada, nos lleva de vuelta al menú para evitar crear un bucle infinito si aún no hay ninguna habitación registrada.

- En 1.4.Eliminar habitación:

Por los mismos motivos de 1.3.informacion de una habitación, esta sección realizará el mismo procedimiento.

- Dentro de 1.5.Listado de habitaciones:

El usuario no introduce ningún valor, por lo tanto no requiere de validaciones.

- En 1.6.Filtrar habitaciones disponibles por tipo y fecha:

Se pide el tipo de habitación, que se seguirá pidiendo si el valor introducido no está entre 1 y 4.

En cuanto a la validación de la fecha, comprueba que esté en el formato dd/mm/yy, cualquier otro formato no lo aceptará y repetirá la pregunta. Hará lo mismo si la fecha no existe. Además comprobará que la fecha inicial sea anterior a la final, si no es así volverá a pedir las dos fechas.

- En 1.7. Guardar modificaciones:

El usuario no podrá introducir datos, por lo tanto no hay posibles excepciones.

- En 1.8. Volver atrás:

Nos lleva al menú anterior, sin posibilidad de generar excepciones por parte del usuario.

- En 2.1. Añadir reservas:

Valida el tipo de habitación como anteriormente, del 1 al 4, repitiendo la pregunta si no es así.

Valida la fecha igual que anteriormente, repitiendo la pregunta si no se ajusta a los parámetros.

Después, nos pedirá indicar el número de las habitaciones, no permitiéndonos elegir una de las que no estén en el listado. Si no hubiese habitaciones disponibles, cortaría el proceso con un mensaje de que no hay habitaciones disponibles.

El DNI, se validará según el formato 00000000x, permitiendo la letra en minúscula y mayúscula, y comprobará que sea un DNI válido utilizando el código corrector de la letra con respecto a la secuencia de números.

Para el resto de preguntas de si y no, seguirá los patrones del resto de funciones anteriores.

- En 2.2. Modificar reservas:

Hace las mismas comprobaciones anteriores para el número. Si la habitación no se encuentra, nos devuelve al menú, para evitar bucles infinitos en caso de no haber agregado ninguna habitación todavía.

Al introducir la fecha, si no se encuentra la reserva, nos devuelve al menú.

El resto de funciones, funcionan de la misma manera que añadir reservas, dando las mismas soluciones para evitar las excepciones.

- En 2.3. Obtener información de una reserva:

Valida el número de habitación como anteriormente, pero si no encuentra la habitación nos devuelve al menú.

Para la fecha, se usan las mismas validaciones anteriores, y nos devuelve al menú si no coincide con ninguna reserva de esa habitación.

- 2.4. eliminar reserva:

Hace las comprobaciones habituales para la fecha y número de habitación, si no coinciden con ninguna reserva nos devuelve al menú.

- 2.5. Lista de todas las reservas:

El usuario no introduce ningún valor, por lo tanto no creará excepciones.

- 2.6. Generar factura:

Valida como es habitual el número de habitación y la fecha inicial de la reserva, si no existe la habitación o la reserva nos devuelve al menú.

- 2.7. Guardar modificaciones:

No lee ningún valor introducido por el usuario, no permite excepciones.

- 2.8. volver atrás:

Nos lleva al menú anterior, no permite excepciones.

2. Comprobar que la aplicación realice correctamente los apartados que se le piden:

Respecto a la parte de gestión de habitaciones:

Al cargar la aplicación, comprobamos que carga las habitaciones y reservas almacenadas, esto lo hacemos viendo el listado de habitaciones y reservas.

Al añadir una habitación comprobamos que se haya añadido correctamente con la función lista de todas las habitaciones, y comprobamos que la habitación creada aparece con los datos añadidos o los que tiene por defecto.

En modificar habitación, comprobamos que nos pide nuevos datos para atributos que tenga la habitación según su tipo, por ejemplo no pide el precio de camas supletorias en habitaciones individuales ni dobles, y solo pide indicar si comparte o no baño en dobles. Después, comprobamos que los datos han sido modificados correctamente con la función información de una habitación, que nos mostrara los nuevos valores de la habitación, además así comprobamos que información de habitación nos muestra correctamente los datos de esa habitación. Para habitaciones individuales y suite no debe mostrar el precio de las camas supletorias puesto que no se pueden poner (lo mismo en lista de todas las habitaciones).

Para borrar habitación, seleccionaremos la habitación que queremos borrar, y con la lista de habitaciones comprobaremos que esa habitación ya no existe.

En filtrar habitaciones por tipo y fecha, si añadimos una reserva en cierto tipo de habitación en una fecha, y pedimos en filtrar habitaciones una habitación del mismo tipo y en las mismas fechas, comprobamos que la habitación en la que se hizo la anterior reserva no aparecerá en el listado.

Respecto a guardar modificaciones, podemos guardar, y posteriormente cerrar el programa, al abrirlo y seleccionar el listado de habitaciones, comprobamos que es igual al que había antes de cerrar el programa.

Respecto a la parte de gestión de reservas:

Al añadir una reserva, podemos comprobar que las habitaciones ofertadas están libres si vamos a añadir dos reservas del mismo tipo de habitación y la misma fecha de entrada en la misma habitación, comprobamos que al intentar crear la segunda reserva, entre las habitaciones ofertadas, que son todas del tipo seleccionado, la habitación de la primera reserva no aparece como opción.

También, podemos comprobar que los datos de la reserva han sido agregados correctamente haciendo uso de la función obtener información de una reserva, así también comprobamos que nos muestre correctamente los datos de las reservas.

Con la función modificar reserva, comprobamos que nos da a elegir en función del tipo de habitación en la que esté la reserva las camas supletorias y si es vip, y al cambiar ciertos atributos comprobamos que se muestren los nuevos datos en lista de todas las reservas. De este modo comprobamos que esta función también esté bien mostrándonos todas las reservas.

Con la función generar factura, y en función del tipo de habitación comprobaremos que se hayan hecho los cálculos correspondientes según los datos de la reserva, como incrementar el precio por las camas supletorias, si la reserva es vip... También comprobamos que nos muestre todos los datos necesarios a la hora de generar la factura.