



DOCUMENTACION DE PRUEBAS REALIZADAS

Yannick Baratte Forner

PROYECTO FINAL ESTACION DE ESQUI 1ro DAW



1. Prueba de funcionamiento correcto del método crearUsuario()

Este método pretende crear un nuevo usuario en la base de datos, haciendo antes diversas comprobaciones como si el DNI es válido, si la fecha de nacimiento es válida o si no existe ya en la base de datos. Si todo esta correcto deberían imprimirse los datos del cliente.

Asimismo, también guardamos en un log (archivo) todos los usuarios que creamos con su fecha de creación y otros datos.

- ✓ Se muestran los datos del usuario y queda guardado correctamente en la base de datos

```
---- Crear usuario ----
Dime tu DNI con este formato 12345678Q, si no tienes escribe 0
23415671T
Dime tu nombre
Juan
Dime tus apellidos
Sanchez Garcia
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990
15/04/1980

---- Datos Usuario ----
Identificador: 31
DNI: 23415671T
Nombre: Juan
Apellidos: Sanchez Garcia
Fecha de nacimiento: 15/04/1980
Proceso completado
```

☐  Editar  Copiar  Borrar 31 23415671T Juan Sanchez Garcia 1980-04-15

- ✓ El usuario queda guardado en el log

```
logUsuarios: Bloc de notas
Archivo Edición Formato Ver Ayuda
---- Datos Usuario ----
Identificador: 0
DNI: 47750189G
Nombre: Yannick
Apellidos: Baratte Forner
Fecha de nacimiento: 21/05/1993
Hora y fecha: 17:40:58 18/05/2020
---- Datos Usuario ----
Identificador: 31
DNI: 23415671T
Nombre: Juan
Apellidos: Sanchez Garcia
Fecha de nacimiento: 15/04/1980
Hora y fecha: 19:51:16 30/05/2020
```

2. Prueba de funcionamiento correcto del método comprobarDni()

Este método pretende comprobar que cuando insertamos un DNI su formato es correcto.

- ✓ Si introducimos bien el formato de DNI devuelve true y no hay problema

```
---- Crear usuario ----  
Dime tu DNI con este formato 12345678Q, si no tienes escribe 0  
45612347Q  
Dime tu nombre  
|
```

- ✓ Si introducimos mal el formato de DNI devuelve false y en este caso nos genera una excepción

```
---- Crear usuario ----  
Dime tu DNI con este formato 12345678Q, si no tienes escribe 0  
1234  
Has introducido mal el DNI. Vuelve a introducir los datos
```

3. Prueba de funcionamiento correcto del método comprobarFechaNacimiento()

Este método pretende comprobar que cuando insertamos una fecha de nacimiento su formato es correcto.

- ✓ Si introducimos bien el formato de la fecha devuelve true y no hay problema

```
---- Crear usuario ----  
Dime tu DNI con este formato 12345678Q, si no tienes escribe 0  
12345678Q  
Dime tu nombre  
Yannick  
Dime tus apellidos  
Baratte Forner  
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990  
21/05/1993
```



- ✓ Si introducimos mal el formato de la fecha devuelve false y en este caso nos genera una excepción

```
---- Crear usuario ----  
Dime tu DNI con este formato 12345678Q, si no tienes escribe 0  
12345678Q  
Dime tu nombre  
Yannick  
Dime tus apellidos  
Baratte Forner  
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990  
1234  
Has introducido mal la fecha. Vuelve a introducir los datos
```

4. Prueba de funcionamiento correcto de los métodos comprarForfaits(), comprarForfaitElegido() y imprForfTick()

comprarForfaits() pretende mostrar un menú en el que el usuario elige que tipo de forfait quiere comprar. Según el forfait que elija llamamos al método comprarForfaitElegido(), con un parámetro que indica el tipo de forfait, este se ocupa de realizar las operaciones necesarias para completar la compra de dicho forfait. comprarForfaitElegido() llama a imprForfTick() que se encarga de imprimir el forfait y su ticket.

- ✓ Si elegimos las opciones del menú nos lleva a comprarForfaitElegido() con el forfait que hemos elegido, se hacen las operaciones necesarias y se imprime la información

```
---- Comprar forfaits ----  
1- Comprar forfait de dia 20€  
2- Comprar forfait de 2 dias 37€  
3- Comprar forfait de una semana 100€  
4- Comprar forfait de temporada 350€  
Dime un opcion  
1  
  
---- Comprar forfait de un dia ----  
Es necesario estar dado de alta en la base de datos para comprar un forfait. ¿Lo estas? Escribe si
```

```

----Autenticar usuario----
Tienes DNI? (escribe Si o No)
si
Dime tu DNI con este formato 12345678Q
47750189G
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990
21/05/1993
Proceso completado!

----Forfait----
Tipo forfait: forfait de un dia
Nombre: Yannick
Apellidos: Baratte
Id cliente: 30
Fecha expedicion: 31/05/2020

---- Ticket ----
Tipo forfait: forfait de un dia
Id cliente: 30
Fecha y hora: 15:51:30 31/05/2020
Precio: 20.0€

```

- ✓ Queda guardada la compra del forfait en la base de datos

id_forfait	id_cliente	fecha_hora	1	precio
f1d	30	2020-05-31 15:51:30		20

5. Prueba de funcionamiento correcto de los métodos alquilarMaterial() y buscarAlquilarMaterial()

alquilarMaterial() pretende solicitar los datos necesarios para saber quien alquila, que quiere alquilar, que talla, cuantos días y una vez se tienen estos datos se llama al método buscarAlquilarMaterial pasándole por parámetro estos datos y se encarga de si está disponible hacer los cambios en la base de datos e imprimir un ticket.

- ✓ Si alquilamos algo que está disponible nos imprime el ticket correctamente

```

---- Menu principal ----
1- Crear usuario
2- Comprar forfaits
3- Alquilar material de esquí/snow
4- Devolver material de esquí/snow
5- Consultar información de las pistas
6- Consultar rutas por dificultad
7- Tareas de mantenimiento
8- Salir
Dime una opcion
3

----Autenticar usuario----
Tienes DNI? (escribe Si o No)
si
Dime tu DNI con este formato 12345678Q
47750189G
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990
21/05/1993

----Alquilar material----
Que desea alquilar material de esquí o de snow? (escriba esquí o snow)
snow
Que desea alquilar tabla(15€) o botas de snow(20€)? (escriba tabla o botas)
tabla
Dime tu altura en cm
175
Dime tu peso en kg
70
Dime para cuantos dias deseas alquilar la tabla
1
Estas a punto de alquilar: tabla de snow. Talla: 150. Dias: 1. Precio total: 15.0€.
si
Se ha alquilado con exito tabla de snow!

----Ticket----
Material alquilado: tabla de snow
Talla: 150
Id del material: tb11
Id del cliente: 30
Fecha y hora inicio: 31/05/2020 17:34:15
Fecha y hora fin: 31/05/2020 19:00:00
Precio: 15.0€

```

- ✓ Si alquilamos algo que está disponible nos hace cambios en la base de datos

id	talla	tipo_material	precio	disponibilidad
tb11	150	tabla	15	0

id_material	id_cliente	fecha_hora_inicio	fecha_hora_fin	precio
tb11	30	2020-05-31 17:34:15	2020-05-31 19:00:00	15

- ✓ Si intentamos alquilar un material del que no hay existencias no nos deja

No existe tabla de snow para la talla 150

6. Prueba de funcionamiento correcto de los métodos devolverMaterial() y aplicarDevolucionMaterial()

Método con el cual el usuario puede autenticarse. Pide los datos del usuario para autenticarse y si se identifica con éxito devuelve un int con su id, sino devuelve 0

- ✓ Si nos identificamos con datos correctos nos deja seguir

```
----Autenticar usuario----  
Tienes DNI? (escribe Si o No)  
si  
Dime tu DNI con este formato 12345678Q  
47750189G  
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990  
21/05/1993  
  
----Alquilar material----  
Que desea alquilar material de esquí o de snow? (escriba esquí o snow)
```

- ✓ Si nos identificamos con datos erróneos no nos deja seguir

```
Dime tu DNI con este formato 12345678Q  
12345678I  
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990  
20/10/1990  
Parece ser que tu usuario no esta registrado  
No te has identificado correctamente  
  
---- Menu principal ----  
1- Crear usuario
```

7. Prueba de funcionamiento correcto de los métodos devolverMaterial() y aplicarDevolucionMaterial()

devolverMaterial() es un método cuya función es que el cliente pueda devolver el material. Pedimos el id del material, comprobamos que este pendiente de devolver y devolvemos, si el cliente se retrasa con la devolución tendrá que pagar una multa. aplicarDevolucionMaterial(), aplica dicha devolución, es decir, una vez comprobada la devolución y la multa en

caso que la haya, aplicarDevolucionMaterial() nos hace los cambios en la base de datos y nos imprime un ticket.

- ✓ Si devolvemos el material a tiempo no hay problema, nos imprime el ticket
- ✓ Si devolvemos el material nos hace el cambio en la base de datos.

id	talla	tipo_material	1	precio	disponibilidad
tb11	150	tabla		15	1

- ✓ Si devolvemos material con retraso tendremos una multa según el retraso

```
----Devolver material----
Dime el id del material, lo encontraras en el tiket o escrito en tu material
tb11
Como has devuelto el material antes de las 9:00 del dia siguiente solo tienes que pagar una multa
Devolucion completada con exito
----Ticket----
Id material: tb11
Fecha y hora devolucion: 2020-05-31 20:17:45.0
Multa: 5€
```

- ✓ Si intentamos devolver un material que ya ha sido devuelto o que no existe nos saldrá un mensaje informando

```
----Devolver material----
Dime el id del material, lo encontraras en el tiket o escrito en tu material
tb11
No se ha encontrado el id del material o ya ha sido devuelto
```

8. Prueba de funcionamiento correcto del método consultarInfoPistas()

Método para imprimir toda la info de las pistas. Pistas abiertas por colores, temperatura etc.

- ✓ Si ejecutamos el código nos muestra la información de manera correcta concordando con la información de la base de datos

```
----Info pistas----
Pistas abiertas: 9/10
Pistas azules abiertas: 3/3
Pistas verdes abiertas: 2/2
Pistas rojas abiertas: 2/3
Pistas negras abiertas: 2/2

----Info Pistas Ampliada----
Nombre pista: La Font. Nivel: AZUL. Pista abierta: si. Altura inicio: 2000. Altura fin: 1000. Temperatura: 5.0
Nombre pista: La Rasa. Nivel: ROJA. Pista abierta: no. Altura inicio: 2000. Altura fin: 1000. Temperatura: 5.0
Nombre pista: L'Esquiro. Nivel: NEGRA. Pista abierta: si. Altura inicio: 2000. Altura fin: 1500. Temperatura: 5.4
Nombre pista: L'Estadi. Nivel: NEGRA. Pista abierta: si. Altura inicio: 1500. Altura fin: 1000. Temperatura: 5.5
Nombre pista: La Rasa Pauet. Nivel: ROJA. Pista abierta: si. Altura inicio: 1500. Altura fin: 1000. Temperatura: 5.5
Nombre pista: Solei. Nivel: VERDE. Pista abierta: si. Altura inicio: 1000. Altura fin: 0. Temperatura: 5.5
Nombre pista: El Forat. Nivel: ROJA. Pista abierta: si. Altura inicio: 1000. Altura fin: 0. Temperatura: 5.5
Nombre pista: L'Orri. Nivel: AZUL. Pista abierta: si. Altura inicio: 1000. Altura fin: 0. Temperatura: 5.7
Nombre pista: L'Obaga. Nivel: VERDE. Pista abierta: si. Altura inicio: 1000. Altura fin: 500. Temperatura: 5.7
Nombre pista: La Drecera. Nivel: AZUL. Pista abierta: si. Altura inicio: 500. Altura fin: 0. Temperatura: 5.7
```

id	nombre	altura_inicio	altura_fin	pista_abierta	temp	nivel
1	La Font	2000	1000	1	5	AZUL
2	La Rasa	2000	1000	0	5	ROJA
3	L'Esquiro	2000	1500	1	5.4	NEGRA
4	L'Estadi	1500	1000	1	5.5	NEGRA
5	La Rasa Pauet	1500	1000	1	5.5	ROJA
6	Solei	1000	0	1	5.5	VERDE
7	El Forat	1000	0	1	5.5	ROJA
8	L'Orri	1000	0	1	5.7	AZUL
9	L'Obaga	1000	500	1	5.7	VERDE
10	La Drecera	500	0	1	5.7	AZUL

9. Prueba de funcionamiento correcto del método generarObjPistasConBD()

Con este método creamos un array de objetos con la información de las pistas que tenemos en la base de datos y lo retornamos.

- ✓ El método devuelve sin problemas un array de objetos Pista con las pistas que hay en la base de datos y su información

En el método anterior generamos un array de objetos Pista con `generarObjPistasConDB()` y luego lo leemos, como se puede observar funciona sin problemas

10. Prueba de funcionamiento correcto del método `consultarRutasPorDificultad()`

Con este método podemos indicar nuestro nivel (principiante, intermedio, experto) y nos mostrara las pistas que podemos coger. Además, se muestran de manera que podemos elegir una ruta uniendo varias pistas.

- ✓ El método muestra las pistas disponibles por dificultad sin problemas y ordenadas de más a menos altura

```
----Consultar pistas por dificultad----
Dime tu nivel, escribe principante, nivel medio, experto
experto

----Tus pistas----

-----
L'Esquiro
-----
Altura inicio 2000  Altura fin 1500
-----

-----
L'Estadi
-----
Altura inicio 1500  Altura fin 1000
-----

-----
La Rasa Pauet
-----
Altura inicio 1500  Altura fin 1000
-----

-----
El Forat
-----
Altura inicio 1000  Altura fin 0
-----
```

11. Prueba de funcionamiento correcto del método tareasMantenimiento()

Este método sirve para mostrar el menú de tareas de mantenimiento y que cuando elijas una opción te mande a su respectivo método.

✓ El menú funciona sin problemas

```
----Tareas de mantenimiento----
1- Cambiar informacion de pistas
2- Agregar nuevo material de esquí/snow
1

----Cambiar informacion de pistas----
Dime el nombre de la pista
.

----Tareas de mantenimiento----
1- Cambiar informacion de pistas
2- Agregar nuevo material de esquí/snow
2

----Añadir nuevo material----
Dime el id del material
.
```

12. Prueba de funcionamiento correcto del método cambiarInfoPistas()

Con este método solicitamos el nombre de una pista y cambiamos su temperatura y si está abierta o cerrada.

✓ Cambiamos la temperatura de una pista y si la pista está abierta sin problemas

```
----Cambiar informacion de pistas----
Dime el nombre de la pista
La Font
Dime la temperatura de la pista
6
La pista esta abierta (escribe Si o No)
no
Cambios realizados!
```

id	nombre	altura_inicio	altura_fin	pista_abierta	temp	nivel
1	La Font	2000	1000	0	6	AZUL

13. Prueba de funcionamiento correcto de los métodos añadirNuevoMaterial() y añadirMaterialNuevoDB()

Con añadirNuevoMaterial() creamos un objeto tipo NuevoMaterial y se lo pasamos al método añadirMaterialNuevoDB() en la clase NuevoMaterial que se encarga de rellenar el objeto NuevoMaterial con los datos que vamos rellenando y luego plasmarlo en la base de datos

- ✓ Introducimos un nuevo material, comprobamos que se ha introducido en la base de datos

```

----Añadir nuevo material----
Dime el id del material
eq15
Dime la talla del material
150
Dime el tipo de material (tabla, esquis, botas_snow, botas_esqui)
esquis
Dime el precio de alquiler del material por 1 dia
10
Material añadido!

```

id	talla	tipo_material	precio	disponibilidad
eq15	150	esquis	10	1

14. Prueba de funcionamiento correcto del método ponerMayusculasNombreApellido()

Con ponerMayusculasNombreApellido() lo que hacemos es que cuando creamos un nuevo usuario, creamos un objeto Usuario y en el setter llamamos a este método, de forma que da igual cómo nos escriba el

nombre y los apellidos, ponemos la primera letra en Mayúscula de cada nombre/apellido ya sea un nombre/apellido compuesto o no

- ✔ Registramos un nuevo usuario, y vemos como nos cambia las primeras letras por mayúsculas en el objeto, el cual luego mostramos atributos, y guardamos en la base de datos

```
---- Crear usuario ----
Dime tu DNI con este formato 12345678Q, si no tienes escribe 0
123456780
Dime tu nombre
yannick
Dime tus apellidos
baratte forner
Dime tu fecha de nacimiento con el siguiente formato dd/mm/aaaa ej 01/01/1990
21/05/1993

---- Datos Usuario ----
Identificador: 33
DNI: 123456780
Nombre: Yannick
Apellidos: Baratte Forner
Fecha de nacimiento: 21/05/1993
Proceso completado
```