Yannick Baratte Forner

Proyecto final estacion de esqui  1ro DAW

Documentacion de pruebas realizadas

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Se muestran los datos del usuario y queda guardado correctamente en la base de datos |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | El usuario queda guardado en el log |

# **2. Prueba de funcionamiento correcto del método comprobarDni()**

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

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si introducimos bien el formato de DNI devuelve true y no hay problema |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si introducimos mal el formato de DNI devuelve false y en este caso nos genera una excepción |

# **3. Prueba de funcionamiento correcto del método comprobarFechaNacimiento()**

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

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si introducimos bien el formato de la fecha devuelve true y no hay problema |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si introducimos mal el formato de la fecha devuelve false y en este caso nos genera una excepción |

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | 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 |
|  |  |

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Queda guardada la compra del forfait en la base de datos |

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si alquilamos algo que está disponible nos imprime el ticket correctamente |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si alquilamos algo que está disponible nos hace cambios en la base de datos  C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Captura11.PNG  C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Captura12.PNG |

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si intentamos alquilar un material del que no hay existencias no nos deja |

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

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

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si nos identificamos con datos correctos nos deja seguir |

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si nos identificamos con datos erróneos no nos deja seguir |

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si devolvemos el material a tiempo no hay problema, nos imprime el ticket |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si devolvemos el material nos hace el cambio en la base de datos. |
|  |  |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si devolvemos material con retraso tendremos una multa según el retraso |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si intentamos devolver un material que ya ha sido devuelto o que no existe nos saldrá un mensaje informando |

# **8. Prueba de funcionamiento correcto del método consultarInfoPistas()**

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

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Si ejecutamos el código nos muestra la información de manera correcta concordando con la información de la base de datos |
|  |  |

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | 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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | El método muestra las pistas disponibles por dificultad sin problemas y ordenadas de más a menos altura |

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | El menú funciona sin problemas |

# **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.

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Cambiamos la temperatura de una pista y si la pista está abierta sin problemas |

# **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

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | Introducimos un nuevo material, comprobamos que se ha introducido en la base de datos |

# **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

|  |  |
| --- | --- |
| C:\Users\Yann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1200px-Yes_Check_Circle.svg.png | 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 |