El sistema debe estar en la capacidad de:

**Req1**. Registrar un nuevo usuario con tipo de documento de identidad, número de documento, nombres, apellidos, teléfono y dirección. No permitirá agregar usuarios con el mismo código o agregar usuarios que no hayan ingresado como mínimo el tipo de documento de identidad, número de documento, nombres y apellidos.

**Req2**. Asignar un turno a un usuario. Se busca el usuario para asignarle el turno para atenderlo, si se encuentra al usuario le asigna el actual turno libre, si no existe se verifica para registrarlo.

**Req3**. Atender turnos, indicando si se atendió realmente al usuario con ese turno o si no lo atendió porque ya no se encontraba en el lugar cuando fue llamado para ser atendido y avanzar en el turno.

**Req4.** Generar reportes, ya sea de los turnos que ha pedido un usuario o de las personas que han solicitado un turno.

**Req5.** Registrar personas automáticamente por medio de archivos de texto.

**Req6.** Modificar la fecha y hora del programa.

**Req7.** Permitir penalizar a un usuario que ha faltado a dos turnos, penalización de dos días.

<https://github.com/Gevorah/ShiftControlSystem.git>

**Configuración de los escenarios**

|  |  |  |
| --- | --- | --- |
| Nombre | Clase | Escenario |
| Setup1 | ShiftControlTest | Se añade un usuario con documentType=1(CC), documentNumber=”123”, names=”Adolf”,lastNames=”G.L.”. |
| Setup2 | ShiftControlTest | Se añade un usuario con documentType=2(TI), documentNumber=”369”, names=”Ric”,lastNames=”M.L.”.  Se añade un turno, por defecto code=”A00”. |
| Setup2 | ShiftControlTest | Se añaden dos usuarios, el primero con documentType=1(CC), documentNumber=”666”, names=”Gev”,last names=”B.N.” y el segundo con documentType=1(CC), documentNumber=”999”, names=”Bin”,last names=”G.N.”.  Se añade un turno, por defecto code=”A00”.  Se asocia un usuario al turno con code =A00”, por defecto se crea otro turno con code=”A01”.  Se asocia a otro usuario al turno con code=”A01”, por defecto se crea otro turno con code=”A02”. |

**Diseño de casos de prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el constructor de la clase Shift funcione correctamente. Teniendo en cuenta que el atributo code del objeto coincida con el ingresado. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| Shift | Shift | Ninguno | code = “A00” | El objeto Shift ha sido creado correctamente con los valores de entrada. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el constructor de la clase User funcione correctamente. Que los atributos del objeto coincidan con los valores de entrada. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| User | User | Ninguno | documentType=”CC”, documentNumber=”123”, names=”Adolf”, lastNames=”G.L.”, phone=”3693”, address=”agl” | El objeto User ha sido creado correctamente con los valores de entrada. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método addUser de la clase ShiftControl funcione correctamente. Asignando los valores ingresados en los parámetros a sus respectivos atributos. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| ShiftControl | addUser | setup1 | documentType=1(CC), documentNumber=”321”, names=”Leon”, lastNames=”M.C.”, phone=”3693”, address=”agl”.  documentType=3(RC), documentNumber=”123”, names=”Marc”, lastNames=”M.Z”. | Los usuarios han sido añadidos sin problemas, no lanza ninguna excepción y los atributos coinciden con los datos ingresados. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método selectUser de la clase ShiftControl funcione correctamente. Busca un usuario creado y en caso de que no exista lanza una excepción. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| ShiftControl | selectUser | setup1 | Ninguno | Se busca al usuario que ya ha sido creado correctamente y lanza una excepción si busca un usuario que no existe, además la maneja. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método registerShift de la clase ShiftControl funcione correctamente. Siendo que se asigne el usuario a un objeto Shift y se maneje las excepciones correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| ShiftControl | registerShift | setup2 | Ninguno | Se le asigna al usuario un turno; que ya han sido creados en el setup2. Maneja la excepción al intentar asignar de nuevo el mismo usuario y al intentar intentar asignar un usuario que no existe. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método addShift de la clase ShiftControl funcione correctamente. Se tiene en cuenta que añade los turnos con el código indicado para cada caso. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| ShiftControl | addShift | setup2 | Ninguno | Los turnos se añaden correctamente. El siguiente al turno con code=”A99” es el turno con code=”B00”. Y, cuando no hay turnos se crea el turno con code=”A00”. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método selectToAttendShift de la clase ShiftControl funcione correctamente. Tiene en cuenta que encuentre un Shift con status=”Not attended” al encontrarlo, en caso de que no lo encuentre debe lanzar excepción. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| ShiftControl | selectToAttendShift | setup3 | Ninguno | Selecciona correctamente el Shift con status=”Not attended”. En caso de que no hayan turnos por atender maneja la excepción. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método attendShift de la clase ShiftControl funcione correctamente. Teniendo en cuenta que atiende sin problemas cuando hay turnos para atender y se manejan los dos casos de atendido. Además lanza las excepciones correspondientes. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
| ControlShift | attendShift | setup3 | Ninguno | Atiende correctamente, para los dos casos; status=“Attended” y status”No user”. Cuando no hay turnos para atender lanza y maneja la excepción. |