

El sistema debe estar en la capacidad de:

Req1. Registrar un nuevo usuario con tipo de documento de identidad, núm. De documento, nombre, apellidos, teléfono y dirección. No permitirá que se agreguen usuarios con el mismo documento de identidad. La dirección y el teléfono son información opcional.

Req2. Añadir un nuevo tipo de turno con nombre y duración. No permitirá que se añada si la duración es menor o igual a 0, o si ya existo uno con ese nombre.

Req3. Asignar un turno a un usuario. El turno está formado por una letra y un número entre 00 y 99. Cuando se llegue al último número de una letra (el 99), se cambiará al primer número de la letra siguiente y así continuará. Por ejemplo, del A99 sigue el B00. No se permitirá asignar un turno a un usuario que ya tenga un turno y no haya sido atendido. Además, al turno se le asignara un tipo turno.

Req4. Llamar los turnos que han sido registrados antes de la fecha y hora actual del sistema. El sistema automáticamente informará si la persona fue atendida o no estaba presente.

Req5. Cambiar la fecha y hora del sistema manualmente. No permitirá cambiarla por una fecha y hora menor a la actual del sistema.

Req6. Generar un reporte de los turnos que ha solicitado un usuario. Este reporte se mostrará en la consola o en un archivo.

Req7. Guardar el estado actual del programa para la luego poder ser cargado.

Req8. Cargar el estado del programa previamente guardado.

Diseño casos de prueba

Configuración de los Escenarios

Nombre	Clase	Escenario
setup1	CompanyTest	Una lista de objetos de tipo User vacía, users<User> = null Una lista de objetos de tipo TurnType, turnTypes<TurnType> = {"anciano", 1.5f}
setup2	CompanyTest	Una lista de objetos de tipo User, users<User> = {"CC", "1068895", "Pedro", "Hernandez", "3165577", "cra 23 #20-36"}, {"CC", "63883786", "Sandra", "Gutierrez", "", ""}, {"CC", "10097733", "Alfonso", "Bustamante", "", ""}
setup3	CompanyTest	Una lista de objetos de tipo User, users<User> = {"CC", "1068895", "Pedro", "Hernandez", "3165577", "cra 23 #20-36"}, {"CC", "63883786", "Sandra", "Gutierrez", "", ""}, {"CC", "10097733", "Alfonso", "Bustamante", "", ""}, {"CC", "44695237", "Andres", "Lopez", "", ""} Una lista de objetos de tipo TurnType, turnTypes<TurnType> = {"anciano", 1.5f}, ("embarazada", 1f), ("adulto", 2.5f)}
setup4	CompanyTest	Una lista de objetos de tipo User, users<User> = {"CC", "1068895", "Pedro", "Hernandez", "3165577", "cra 23 #20-36"} Una lista de objetos de tipo TurnType, turnTypes<TurnType> = {"adulto", 2.5f}

Diseño de Casos de Prueba

Objetivo de la Prueba: Verificar que el método addUser de la clase Company funcione correctamente si el usuario que se desea añadir no existe, asignando apropiadamente los valores de los parámetros a sus respectivos atributos.				
Clase	Método	Escenario	Valores de Entrada	Resultado

Company	addUser	setup1	("CC", "1068895", "Pedro", "Hernandez", "3165577", "cra 23 #20-36")	Se ha creado correctamente un objeto de la clase User con los valores pasados por parámetro al método.
---------	---------	--------	--	--

Objetivo de la Prueba: Verificar que el método addUser de la clase Company funcione correctamente si el usuario que se desea añadir ya existe.

Clase	Método	Escenario	Valores de Entrada	Resultado
Company	addUser	setup2	("CC", "44695237", "Andres", "Lopez", "", "")	Se ha creado correctamente un objeto de la clase User con los valores pasados por parámetro al método.

Objetivo de la Prueba: Verificar que el método addUser de la clase Company funcione correctamente si no hay usuarios en el programa, asignando apropiadamente los valores de los parámetros a sus respectivos atributos.

Clase	Método	Escenario	Valores de Entrada	Resultado
Company	addUser	setup3	("CC", "44695237", "Andres", "Lopez", "", "")	Se espera la excepción UserExistsException

Objetivo de la Prueba: Verificar que el método addUser de la clase Company funcione correctamente si no hay usuarios en el programa, asignando apropiadamente los valores de los parámetros a sus respectivos atributos.

Clase	Método	Escenario	Valores de Entrada	Resultado
Company	addUser	setup1	("CC", "6455443", "Gabriel", "Martinez", "", "")	Se ha creado correctamente un objeto de la clase User con los valores pasados por parámetro al método.

Objetivo de la Prueba: Verificar que el método selectionSortUsersById de la clase Company funcione correctamente.

Clase	Método	Escenario	Valores de Entrada	Resultado
Company	selectionSortUsersById	setup2		Se ha ordenado correctamente la lista de usuarios