

## 1. Requerimientos funcionales

El sistema debe estar en la capacidad de:

**Req1. Agregar** un nuevo usuario con nombre, dirección, teléfono y código. No permitirá que se agreguen usuarios con el mismo código y con el mismo tipo de id.

**Req2. Asignar** un turno a un usuario, para ser atendido, el turno está formado por una letra y un número entre 00 y 99, siendo el primero A00; cuando llegue al turno A99, debe cambiar al B00, y así sucesivamente, letra por letra, hasta Z99.

**Req3. Permitir** ir avanzando el turno en la medida en que otro empleado va atendiendo a cada cliente de acuerdo con el turno que tiene asignado.

**Req4. Buscar** un usuario, por medio de su documento de identidad, y el tipo de este, donde se le asignara un turno, para ser atendido, mostrando así en consola el tipo de documento, el numero de id, el teléfono y el nombre.

**Req5. Indicar** si el usuario con el turno asignado fue atendido realmente, o si no lo atendió ya que el usuario no se encontraba en el lugar cuando fue llamado para atenderlo, cambiando así el estado del turno.

**Req6. Permitir** que el sistema de turnos se reinicie cuando llegue al Z99, o pase al otro día.

**Req7. Generar** un nuevo turno, cuando un usuario registrado lo este pidiendo.

**Req8. Mostrar** el turno actual.

**Req9. Generar** una lista de usuarios con el estado de los turnos, si fue atendido o no tiene turno asignado.

Sebastian morales  
Laboratorio gestion de turnos

Clase	Metodo	Escenario	Valores Entrada	Resultado
ControlShift	AdduserTest		1, "1234", "James", "Rodriguez", "31333313", "ST 40 Madrid"	<b>Agrega un nuevo usuario funciona correctamente:</b>  i.Si no hay usuarios en el programa
ControlShift	<b>addUserTest2</b>	Setup1	1, "12345", "James", "Rodriguez", "31333313", "ST 40 Madrid"	i.Si hay usuarios en el programa i.Si el usuario no existe
ControlShift	<b>addUserTest3</b>	Setup2	1, "12345", "James", "Rodriguez", "31333313", "ST 40 Madrid"	Si el usuario existe
ControlShift	findUserIdTest	Setup1	"1234567", 1  "1234567", 2	<b>Busca un usuario con base en su número de documento de identidad: Si ya existe el usuario a buscar</b>  i.Si hay usuarios en el programa
ControlShift	findUserIdTest2		"123456", 2	i.Si no existe el usuario a buscar i.Si no hay usuarios en el programa
ControlShift	AssignementShiftTest1	Setup2	"123456", 1 "123456", 1 "123456", 1 "123456", 1 "1234567", 2 "1234567", 2	<b>Asigna un turno a un usuario:</b> Si el usuario tiene un turno activo, entonces debe probar que retorne el turno activo y no genere otro

ControlShift	AssignementShiftTest2	Setup1	"123456", 1	i.Si el usuario no tiene un turno activo entonces le genera uno nuevo
ControlShift	serveShiftTest	Setup3	A00	a. Genera un nuevo turno: Consecutivo al último turno asignado i.Si es el primer turno genera el turno A00
ControlShift	serveShifTest2	Setup4	D99	E00
ControlShift	serveShiftTest3	Setup5	Z99	i.Si el último turno generado es el Z99 entonces genera el turno A00
ControlShift	serveShifTest4	Setup6	E19	Consulta correctamente el siguiente turno por atender: i. Habiendo turnos por atender
ControlShift	serveShiftTest5	Setup6	"1234567", 1 "123456", 1	Atiende un turno asignado a una persona: i. Habiendo turnos por atender

## ESCENARIOS

### SETUP1

USUARIO (1, "123456", "julian", "perez", "553131", "calle 6 # 21-21");  
USUARIO2 (1, "1234567", "juan", "morales", "52123642", "calle 1 # )  
USUARIO3(2, "1234567", "pedro", "perez", "2126732", "calle 5 # 19-2");  
USUARIO4(3, "1234568", "david", "ramirez", "2126732", "calle 5 # 19-2");  
USUARIO5(1, "1234561", "ana ", "sebastian", "2324232", "calle 3 # 22-2");

### SETUP2

USUARIO (1, "12345", "James", "Rodriguez", "31333313", "ST 40 Madrid");

### SETUP3

A00

### SETUP4

D99

### SETUP5

Z99

### SETUP6

E19

USUARIO (1, "123456", "julian", "perez", "553131", "calle 6 # 21-21");  
USUARIO2 (1, "1234567", "juan", "morales", "52123642", "calle 1 # 1-11");