

## Objetivos

OE2.3 Diseñar pruebas unitarias automáticas que permitan validar el adecuado funcionamiento de las operaciones del sistema desarrolladas para soportar los requerimientos funcionales.

OE2.4 Desarrollar las clases y los métodos necesarios para implementar las pruebas unitarias automáticas, que ayudan a comprobar el correcto funcionamiento de un programa.

## Preparación

- Lea cuidadosamente el enunciado, la documentación suministrada y cada uno de los puntos que debe desarrollar antes de empezar su desarrollo. Pregunte a su profesor cualquier duda respecto al enunciado o a los requerimientos funcionales que debe desarrollar.
- El trabajo debe ser realizado de forma **individual**.
- El trabajo será entregado en la fecha y hora establecida en Moodle.

## Su entrega debe incluir los siguientes elementos

### Diseño de su solución

- (11%) Especificación de Requerimientos Funcionales.  
No utilice el formato (tabla) de AP1, sino que escriba sus requerimientos como párrafos que inician con un verbo después de esta oración (ejemplo):  
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 en el resto del párrafo deben ir todas las especificaciones relacionadas con este requerimiento*).
- (20%) Diseño de casos de prueba.
- (8%) Diseño de diagrama de clases de las pruebas. (Visual Paradigm con FONDO BLANCO).
- (8%) Diseño del diagrama de clases del modelo. (Visual Paradigm con FONDO BLANCO).
- (8%) Diseño del diagrama de clases de las excepciones. (Visual Paradigm con FONDO BLANCO).

### Construcción de la solución (lenguaje de programación)

- (10%) Implementación del modelo y la interfaz de usuario en consola con cada una de sus funcionalidades.
- (5%) Contratos de los métodos y generación de la documentación utilizando javadoc.
- (10%) Implementación de las excepciones.

### Construcción de las pruebas unitarias automáticas (lenguaje de programación)

- (20%) Con base en el diseño de casos de prueba y utilizando JUnit implemente las pruebas unitarias automáticas..

## Enunciado

### SISTEMA DE CONTROL DE TURNOS PARA ATENCIÓN AL USUARIO

Una empresa lo ha contratado para que desarrolle un programa en su departamento de servicio al cliente que resuelva el problema del control y atención de las personas que llegan a sus centros de servicio para ser atendidas.

El programa será utilizado por un empleado de la empresa que se encuentra a la entrada del centro de servicio y con el cual siempre se comunicará, en primera instancia, la persona que ingrese para ser atendida.


El empleado lo buscará con su número de documento de identidad y luego le asignará un turno para ser atendido. El turno está formado por una letra y un número entre 00 y 99. El primer turno es el A00, el siguiente el A01 y así sucesivamente. 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.

El programa también tendrá la posibilidad de registrar a un nuevo usuario en caso de que no se encuentre ya registrado en el sistema. Los datos de un usuario son tipo de documento de identidad, número de documento, nombres, apellidos, teléfono y dirección. Al agregar a un nuevo usuario se debe validar que se diligencien al menos el tipo de documento, el número de documento, los nombres y los apellidos de la persona (es decir son obligatorios).

Por otro lado el programa debe permitir ir avanzando en el turno en la medida en que otro empleado va atendiendo a cada cliente de acuerdo con el turno que tiene asignado. Esa persona que va atendiendo tiene la posibilidad de indicar al programa si 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.

La última funcionalidad explicada debería estar en un programa diferente (o al menos en un perfil diferente) al de las demás funcionalidades ya que es utilizado por un empleado con un rol diferente al de las primeras funcionalidades explicadas, sin embargo por ser ésta la primera versión del programa y con motivos de demostración, todas las funcionalidades serán implementadas en el mismo programa. Cada una en una pestaña diferente, tal como se muestra en los siguientes mockups. Los mockups fueron diseñados para una segunda versión del programa, usted deberá basarse en ellas para guiarse en las opciones que el programa debe tener pero debe desarrollar su interfaz utilizando un menú de opciones en una consola de texto.

Gestión de Turnos :: Turnos Fácil :: Demo



Agregar Usuario    Registrar Turno    Atender Turno

Tipo de Documento: Cédula de Ciudadanía ▼

Número de Documento:

Nombres:

Apellidos:

Teléfono:

Dirección:

Guardar

Las posibilidades en Tipo de Documento son: Cédula de Ciudadanía, Tarjeta de Identidad, Registro Civil, Pasaporte, Cédula de Extranjería

El número de documento debe ser único.

Al guardar el programa verifica que al menos los cuatro primeros campos hayan sido ingresados. En caso de no ser así, o de ingresar un documento que ya existe, muestra un ventana con un mensaje indicando el error.

Campos Obligatorios

Debe digitar la información de todos los siguientes campos: Tipo de Documento, Número de Documento, Nombres y Apellidos

Aceptar

Documento Ya Existe

El documento XXXXXX ya se encuentra registrado en nuestro sistema

Aceptar

Gestión de Turnos :: Turnos Fácil :: Demo



Agregar Usuario

Registrar Turno

Atender Turno

Número de Documento:

56324187

Consultar

Resultado de la búsqueda:

<b>TD</b>	<b>Nombre Completo</b>	<b>Teléfono</b>	Asignar Turno
CC	Pepito Pérez	365 8974521	


El Turno asignado para Pepito Pérez es:

F48

Las únicas posibilidades de búsqueda son: Encontró un usuario y lo muestra o no se encontró un usuario con ese documento.

El campo de texto del turno es de solo lectura. Se utiliza para mostrar el turno asignado internamente por el programa.

Gestión de Turnos :: Turnos Fácil :: Demo



Agregar Usuario   Registrar Turno   Atender Turno

Turno Actual

**B21**

Atendido

Usuario no está

- e. Consulta correctamente el siguiente turno por atender:
- Habiendo turnos por atender
- f. Atiende un turno asignado a una persona:
- Habiendo turnos por atender

Usted debe diseñar e implementar al menos las siguientes pruebas a la operación que:

- Agrega un nuevo usuario funciona correctamente:
  - Si el usuario no existe
  - Si el usuario ya existe
  - Si no hay usuarios en el programa
  - Si hay usuarios en el programa
- Busca un usuario con base en su número de documento de identidad:
  - Si ya existe el usuario a buscar
  - Si no existe el usuario a buscar
  - Si no hay usuarios en el programa
  - Si hay usuarios en el programa
- Asigna un turno a un usuario:
  - Si el usuario tiene un turno activo, entonces debe probar que retorne el turno activo y no genere otro
  - Si el usuario no tiene un turno activo entonces le genera uno nuevo
- Genera un nuevo turno:
  - Consecutivo al último turno asignado
  - Si es el primer turno genera el turno A00
  - Si el último turno generado es el D99 entonces genera el turno E00
  - Si el último turno generado es el Z99 entonces genera el turno A00