

# **Trabajo Practico N.º 3**

## **Juan Manuel García 2º C**

### **Atención al Cliente Clínica Odontológica**

El proyecto genera un acceso público para los clientes y un acceso privado acciones administrativas.

Desde el acceso público los clientes pueden reservar un turno. El mismo les muestra los turnos disponibles para la fecha seleccionada y deben ingresar el tipo de turno que desean de una lista de tipos, su nombre, DNI y número de teléfono.

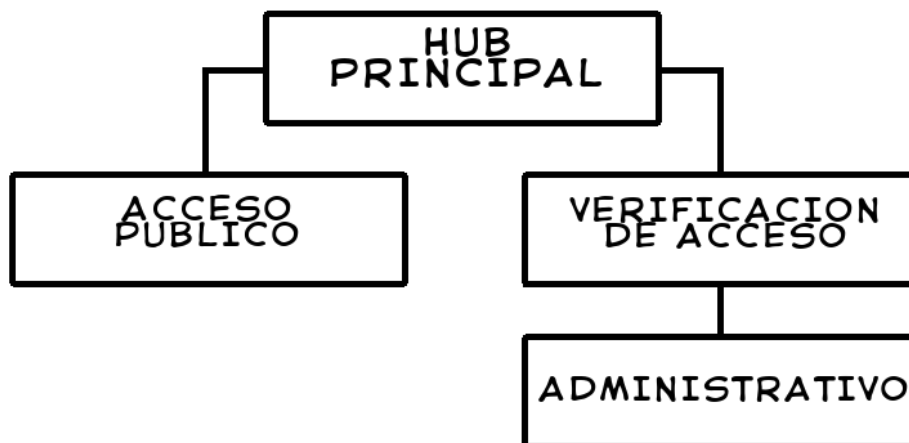
Desde el acceso privado, primero uno debe ingresar un usuario y contraseña. En el mismo uno puede ver los turnos solicitados en las distintas fechas, modificar un turno, eliminarlo, indicar que fue atendido y generar nuevos turnos con los datos correspondientes. Además puede modificar los precios de los distintos tratamientos, exportar los turnos a un archivo y obtener los turnos almacenados en un archivo seleccionado.

La base del proyecto es la clase Turno, la cual maneja la información respecto a los turnos de la clínica y la clase ManejadorDeTurnos que se encarga de leer y guardar los turnos en archivos, poder obtener que horarios están disponibles, obtener turnos de un día específico, modificar, agregar y eliminar turnos.

Usuario: Odontologia

Contraseña: SatoshiNakamoto

Posee la siguiente estructura:



Utilización de los temas de las respectivas clases de la materia:

- Tema 10 – Excepciones:

La clase DatosInvalidosException genera una excepción nueva.

La misma es lanzada por Turno cuando se intentan crear un Turno con datos inválidos.

También es lanzada por ManejadorDeTurnos al intentar asignar un precio negativo o inválido.

- Tema 11 – Pruebas Unitarias:

Un proyecto de pruebas unitarias llamado TestUnitarios.

En el mismo se realizan pruebas a métodos de las clases Turno, ManejadorDeTurnos y Autenticador.

- Tema 12 – Tipos Genericos | Tema 13 – Interfaces:  
 La interfaz IArchivoDTO utiliza un tipo generico T que debe ser una clase.  
 Posee un metodo que toma un objeto de tipo T, con el fin de que deba ser convertido a un DataTransferObject para poder ser serializado sin perder el encapsulamiento en el objeto de tipo T  
 Posee un metodo para devolver una instancia del tipo T, cuando uno necesite leer de un DTO y volver al tipo original
- Tema 14 – Archivos | Tema 15 – Serialización:  
 El ManejadorDeTurnos se encargara de leer los turnos almacenados en un path por defecto asignado al momento de instanciar el mismo (si no existe crea el archivo con una lista vacia) o en uno especificado. Utiliza serializacion JSON.  
 El ManejadorDeTurnos se encargara de guardar la lista de turnos actualmente en memoria. Puede ser guardada en el path por defecto o una eleccion. Se crea una lista de TurnoDTO, el cual tiene los requisitos para poder ser serializado como JSON.  
 El ManejadorDeTurnos se encargara de leer los precios de los tipos de tratamientos de un archivo serializado en XML, utiliza un path por defecto y si no existe el archivo lo crea con valores por defecto.  
 El ManejadorDeTurnos se encargara de guardar nuevamente los precios, ya que los mismos pueden ser modificados y se requiere que perduren al terminar el programa.