

# TRABAJO PRÁCTICO TURNERO

## Prototipo

Pantalla

CODIGO	USUARIO	BOX
AA1	Agustín	BOX 1
AA2	Iván	BOX 2
AA3	Julián	BOX 1
AA4	Tomas	BOX 3
AA5	- - - - -	- - - - -
AA6	Facundo	- - - - -
AA7	Maria	- - - - -

Box

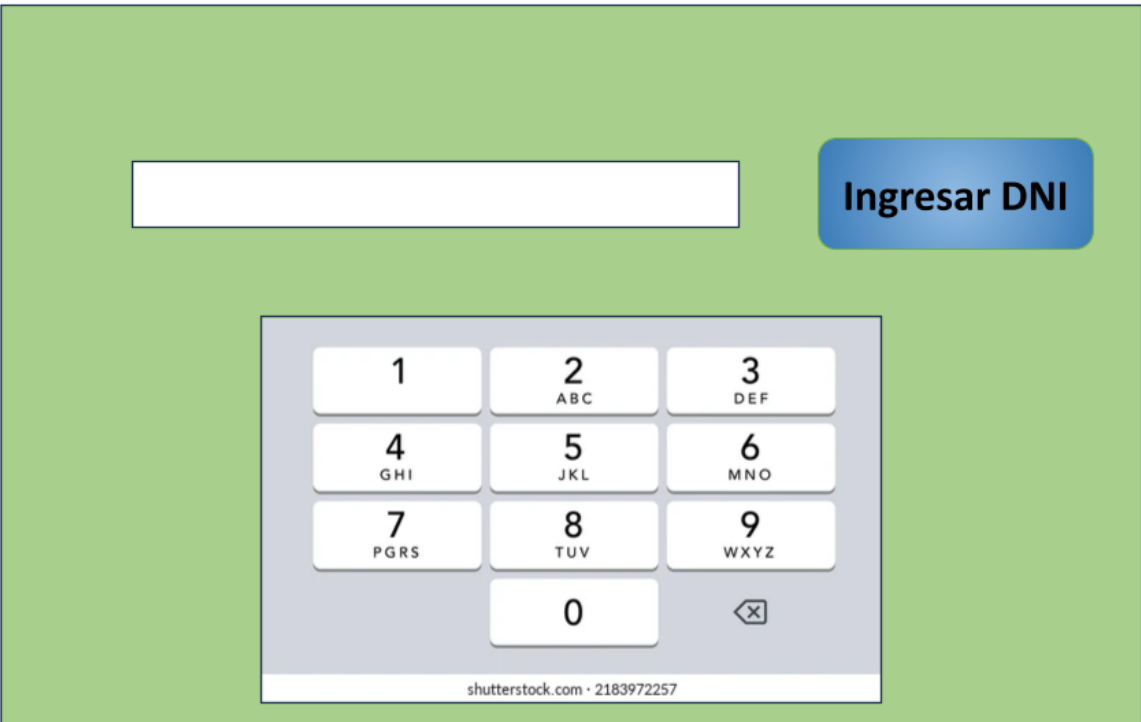
HORA	CODIGO	ESTADO	TURNO	PACIENTE
11:55	AA13	ATENDIDO	Profesional: Hora:	Roberto
12:00	AA14	AUSENTE	Profesional: Hora:	pedro
12:05	AA15	ATENDIDO	Profesional: Hora:	
12:10	AA16	ATENDIDO	Profesional: Hora:	
12:15	BB17	AUSENTE	----- -----	----- -----
12:20	BB18	ATENDIENDO	----- -----	----- -----
12:25	AA19	ATENDER	Profesional: Hora:	
12:30	AA20	ATENDER	Profesional: Hora:	

Tótem de ingreso (tablet)

Paso 1



Paso 2



Paso 3

<b>10:00</b> Doctor ...	<b>Elegir</b>
<b>12:00</b> Doctor ...	<b>Elegir</b>
<b>18:00</b> Doctor ...	<b>Elegir</b>
<b>20:00</b> Doctor ...	<b>Elegir</b>

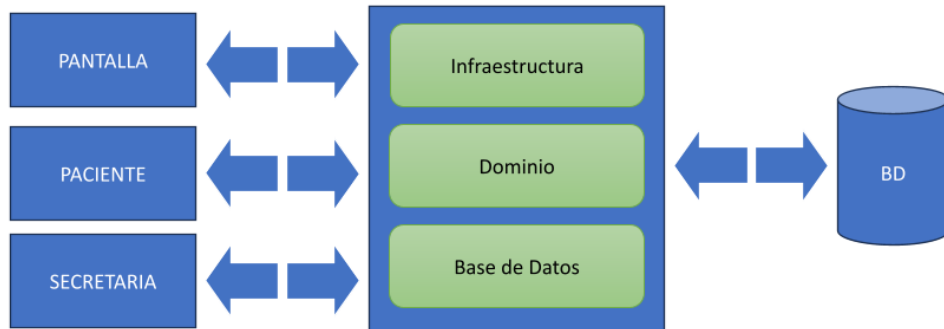
Paso 4

**Tu Codigo es:**  
**AA1**

**Salir**

# Arquitectura

Pensamos nuestro sistema como una arquitectura Cliente - Servidor, donde se accede a la app a través de un navegador, y en el cual el código de la interfaz se encuentra separado del backend. Todo el sistema se instala en un servidor interno de la clínica, y sólo se puede acceder al sistema desde una red local.



## Backend

El backend, que se puede implementar con tecnologías como Node o Java, está a su vez dividido en capas:

1. **Infraestructura:** Esta sección se encarga del procesamiento primario de las requests HTTP, validación de parámetros, parseo de los datos recibidos, y serialización de los datos a entregar.
2. **Dominio:** Aquí se encuentran las reglas de dominio de nuestro sistema, las clases denominadas Services se encargan de ejecutar los casos de uso de la aplicación.
3. **Persistencia / Base de Datos:** Esta capa tiene la responsabilidad de comunicar al sistema con la base de datos, y provee una interfaz reutilizable y concreta a la capa de dominio.

Cabe aclarar que el orden de comunicación entre estas capas es Infraestructura -> Dominio -> Persistencia.

## Frontend

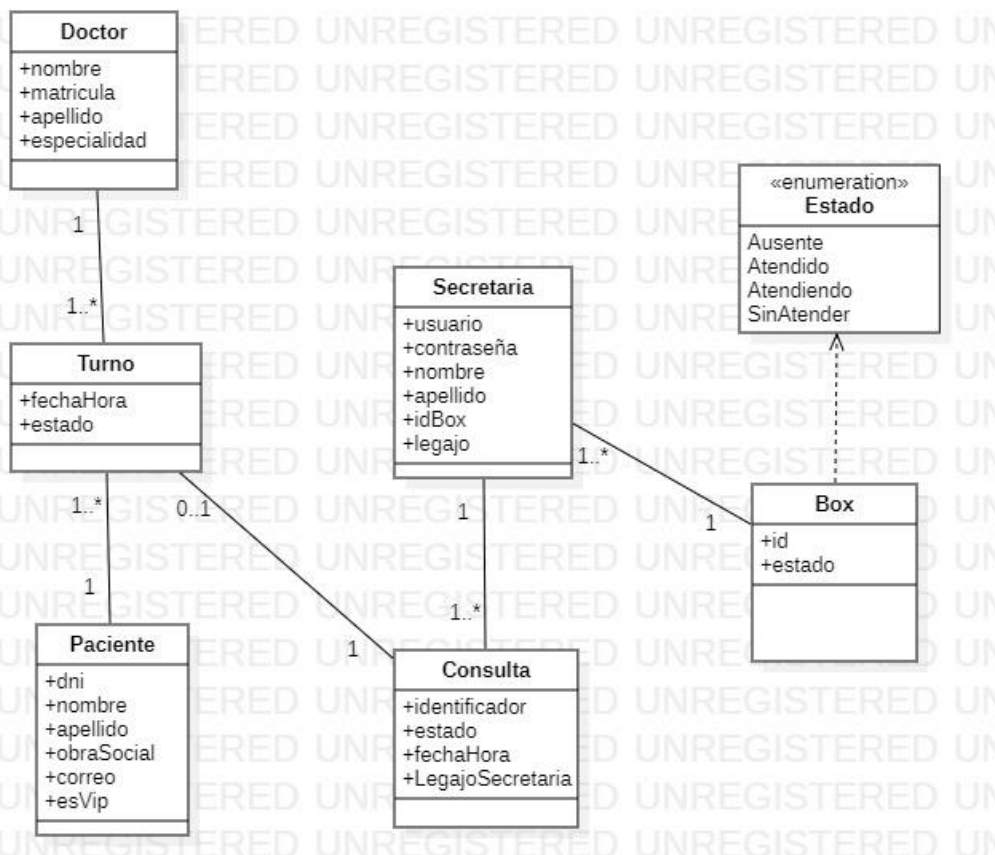
Nuestro sistema ofrece tres experiencias de usuario muy diversas, la de la pantalla de la sala de espera, la del turnero que ve el paciente al ingresar, y la vista del box, que utiliza la secretaria. Entre otras diferencias, la pantalla y el

turnero no necesitan autenticación, mientras que el box sí. Por ello pensamos en utilizar tres aplicaciones frontend separadas, cada una con su propia lógica, pero conectadas al mismo servidor backend. Todas ellas pueden ser implementadas como SPAs (Single Page Applications) usando herramientas como React o Vue.

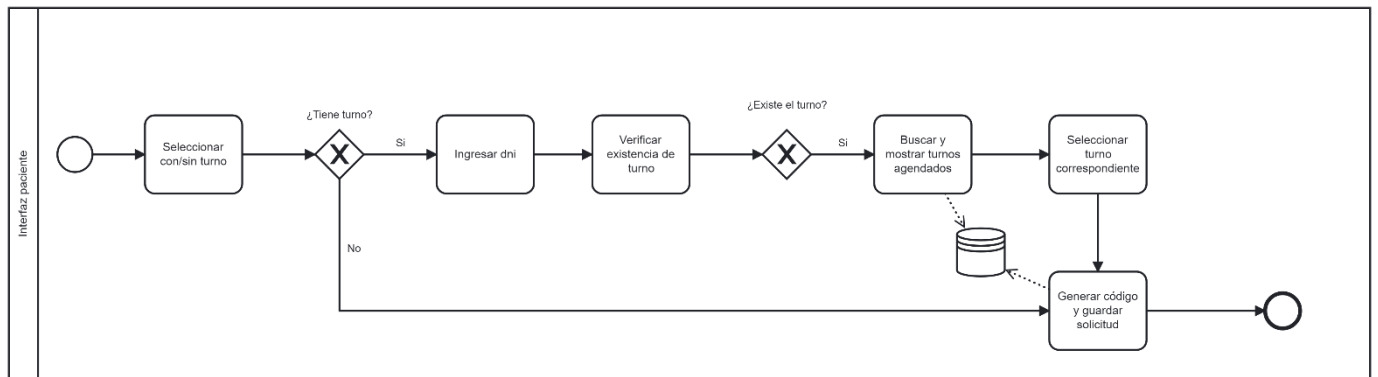
Descripción de cada frontend:

1. Pantalla: En ella se muestra el listado de turnos ya atendidos, y los próximos a ser atendidos. Para actualizar dicho listado se utiliza la técnica de Polling, donde se realizan peticiones al backend cada 15 segundos.
2. Turnero (paciente): Se trata de una vista responsiva, descrita en el prototipo, que no requiere autenticación.
3. Secretaria (box): La parte quizás más compleja del frontend, ya que además de la vista donde se presentan los turnos y sus estados, también se agrega la funcionalidad de login.

## Diagrama de clase

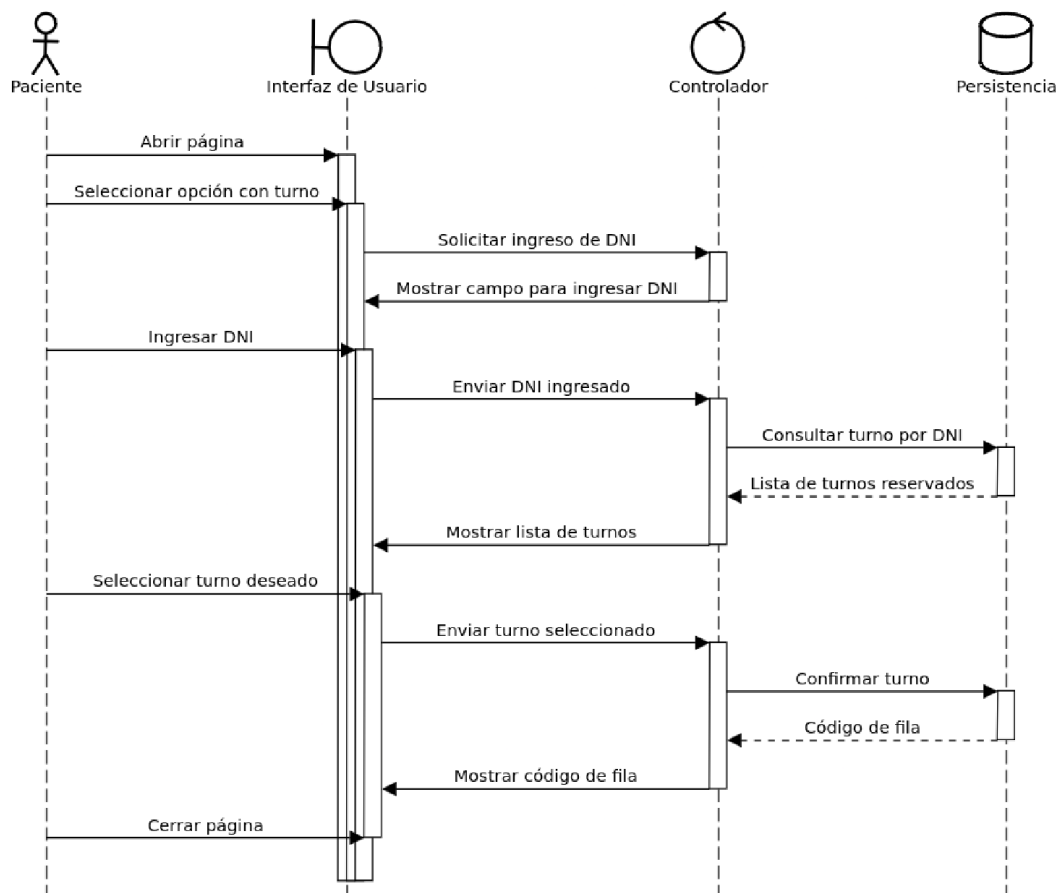


# Proceso de negocio

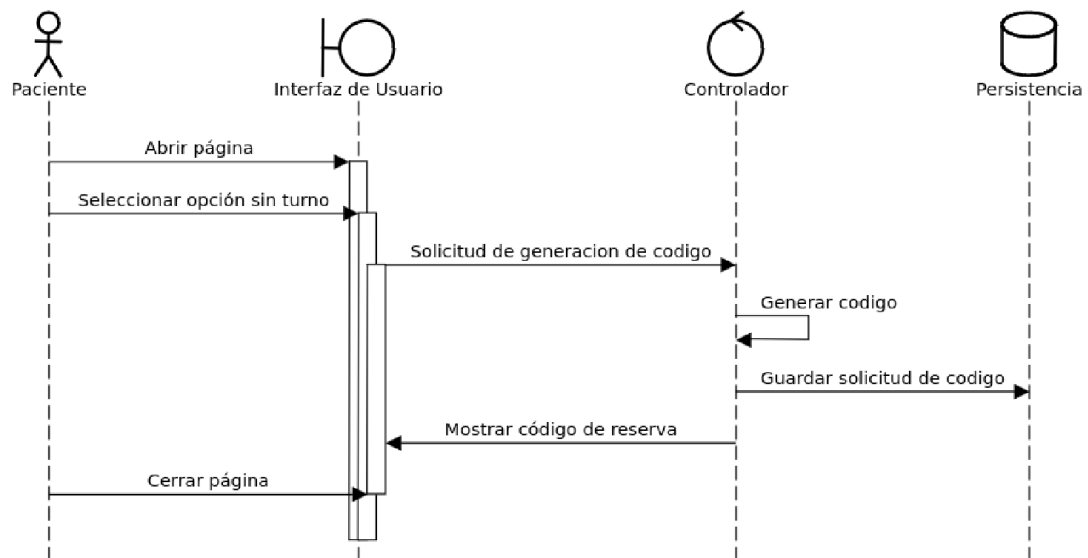


## Diagramas de secuencia

### Proceso de turnero con turno previo



## Proceso de turnero sin turno previo



## Mapecto de objetos a tablas

