

Turnero

Consigna

Para trabajo en clase y de manera grupal: se deben conformar dos equipos de trabajo, cada equipo debe presentar una solución que permita representar la situación planteada tanto en el punto "A" como en el punto "B". Luego deberán exponer la solución y justificar cada decisión de diseño tomada.

Descripción

El turnero permite a las personas que asisten a una clínica ser atendidos de manera ordenada respetando el orden de llamado.

Para ser atendidas, las personas se deben dirigir al turnero, puede que las personas tengan turno o no, se deberán identificar primero.

A. Se requiere generar un prototipo, para la pantalla que muestra los turnos, que permita a las personas conocer el orden de llamado y en que Box será atendido. Además de mostrar cuales fueron los turnos atendidos anteriormente y cuáles serán los próximos. Existen diferentes letras y números que permiten direccionar según cada especialidad.
¿Qué información sería relevante mostrar?

B. Por otro lado, cada Box de atención debería tener una interfaz con los turnos registrados (desde el turnero) y deberán poder tomar los turnos a atender respetando el orden de llegada. Desde cada box se puede tomar un turno para ser atendido. Una vez que el turno es tomado por algún box no debería estar disponible para ser tomado por los demás. Cuando una persona tenga turno con algún médico debería mostrar dicha información a la persona que está en el box de atención.

Suele ocurrir eventualmente, que la persona que saca el turno tenga alguna atención prioritaria, por ejemplo: cuando asiste quien "paga los sueldos al personal de la clínica", en dicho caso, las personas que están en el box podrán dar una atención prioritaria saltando el orden de llegada.

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:	

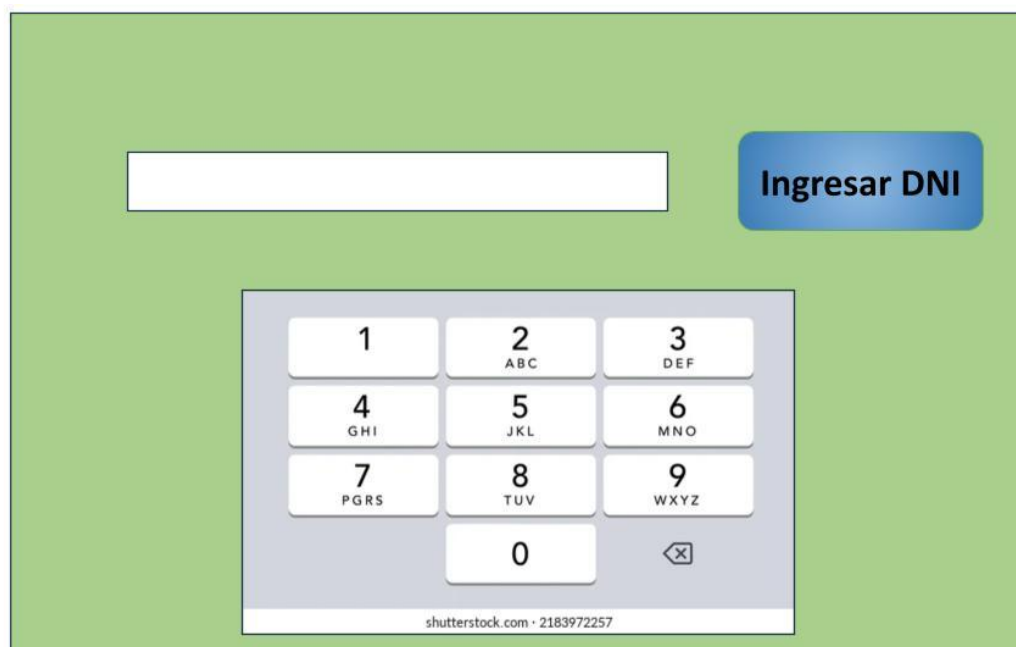
Tótem de Ingreso (tablet)

Paso 1



Screen 1: Selection of queue status. The screen has a light green background and two blue buttons with white text. The top button is labeled "Con Turno" and the bottom button is labeled "Sin Turno".

Paso 2



Screen 2: DNI input and numeric keypad. The screen has a light green background. At the top left is a white rectangular input field. To its right is a blue button with white text labeled "Ingresar DNI". Below the input field is a numeric keypad with a light gray background. The keypad has buttons for digits 1 through 9, 0, and a backspace key (represented by a left arrow and an 'X'). Each digit button also has its corresponding letters (e.g., 2 has ABC, 3 has DEF, etc.). At the bottom of the keypad, there is a small text string: "shutterstock.com · 2183972257".

Paso 3

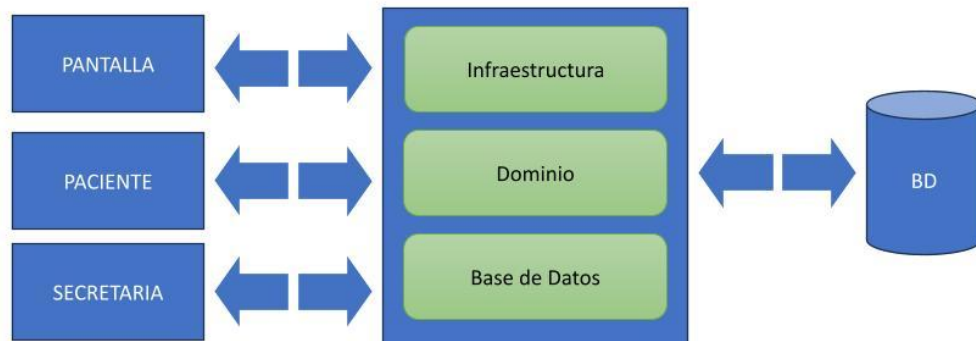
10:00 Doctor ...	<input type="button" value="Elegir"/>
12:00 Doctor ...	<input type="button" value="Elegir"/>
18:00 Doctor ...	<input type="button" value="Elegir"/>
20:00 Doctor ...	<input type="button" value="Elegir"/>

Paso 4

Tu Codigo es:
AA1

Arquitectura

Pensamos nuestro sistema como una arquitectura **Cliente - Servidor**, donde se accede a la app a través de un navegador, y en el cuál 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.

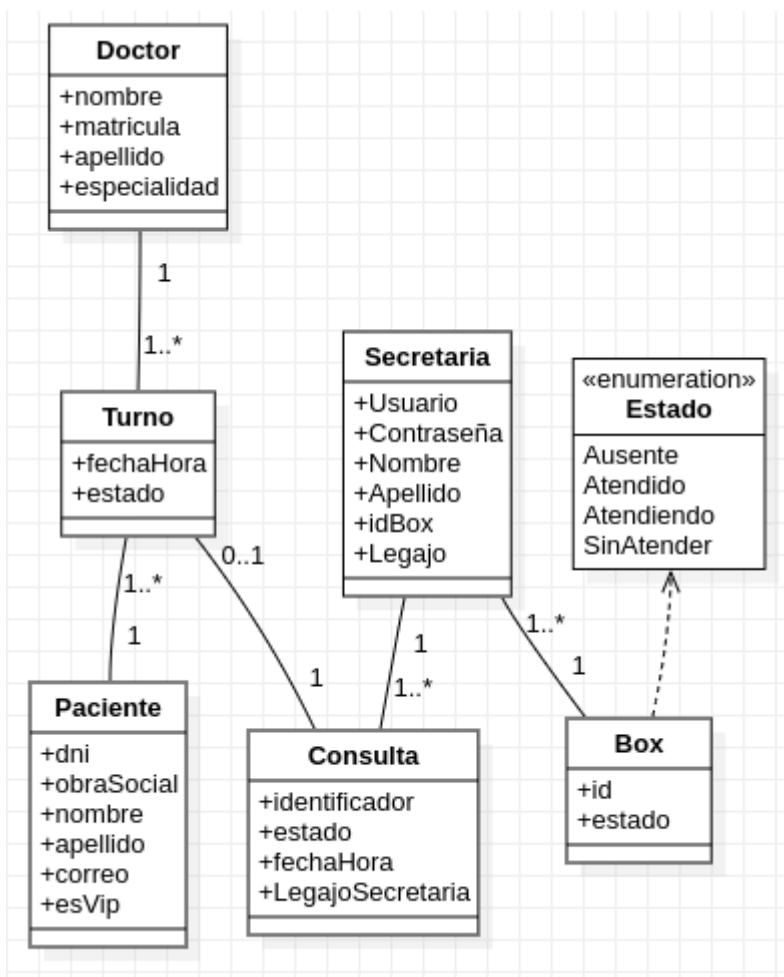
Frontend

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.

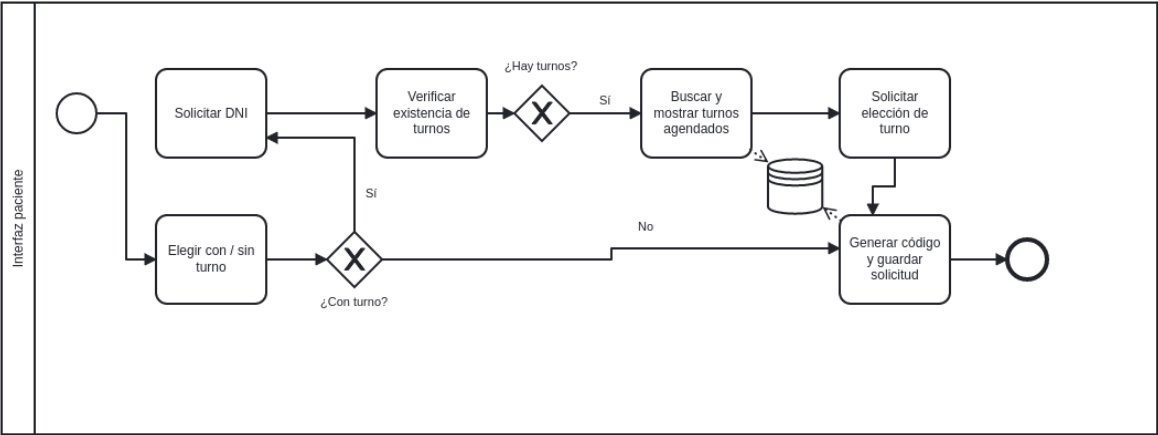
Descripción de cada frontend:

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.

Diagramas de clase

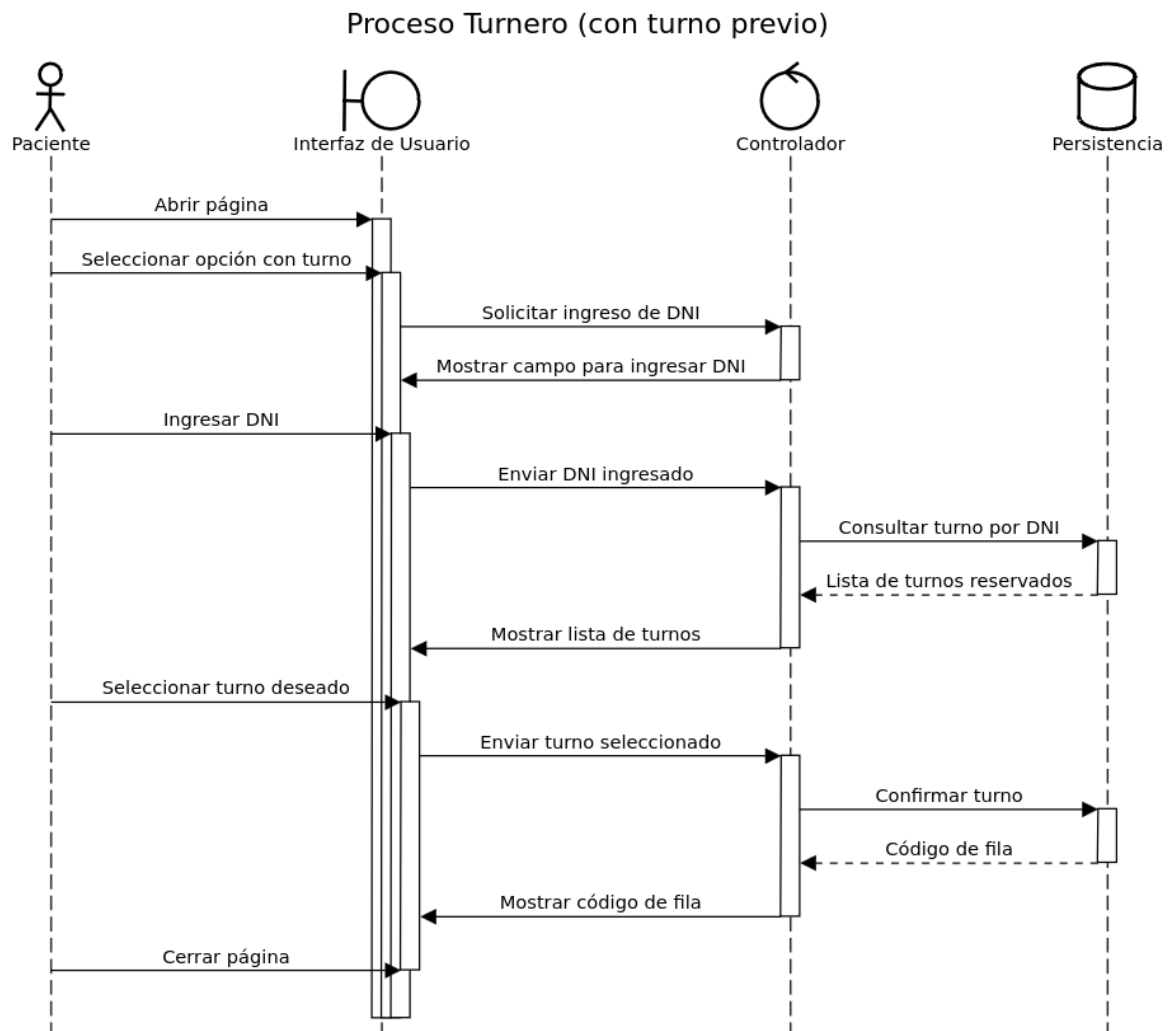


BPMN

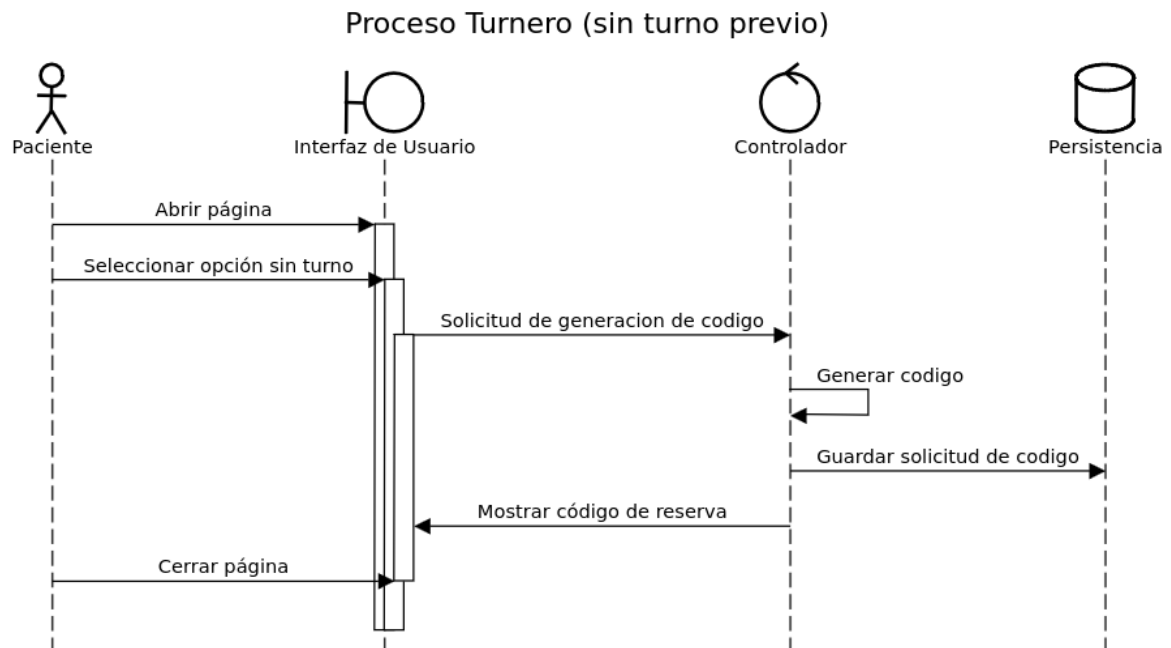


Diagramas de secuencia

Proceso de Turnero (con turno previo)



Proceso turnero (sin turno previo)



Mapecto de Objetos a tablas

