

	Grado en TIS
	Sistemas Distribuidos
	Curso 2018-2019
	Práctica 2
	<i>Deshabituación al tabaco</i>

Crear una aplicación web distribuida mediante estándares (HTML, CSS y Javascript) para la deshabituación al tabaco. La aplicación permitirá a un profesional médico realizar un estudio sobre los hábitos de un paciente respecto al tabaco, mediante el cual poder recetar medicamentos orientados a eliminar su consumo. El estudio se fundamenta en la realización de unos test por parte del paciente, que se evaluarán en función a unas puntuaciones que permitirán al médico saber si se puede financiar el medicamento para cada paciente. Posteriormente se realizará un chat entre médico y paciente.

El sistema distribuido estará formado por tres módulos:

- Un servidor basado en Javascript mediante la plataforma NodeJS.
- Una aplicación web para el médico
- Una aplicación web para el paciente

Para la realización de la práctica se utilizarán distintos mecanismos de comunicación: Servicios Web tipo REST, RPC y WebSockets

Tests

Los tests que realizará el paciente son los siguientes.

Test de Fageström

Se trata de un test para valorar el nivel de adicción del paciente.

Pregunta	Respuesta	Puntuación
¿Cuánto tiempo pasa entre que se levanta y se fuma su primer cigarrillo?	Hasta 5 minutos	3
	De 6 a 30 minutos	2
	De 31 a 60 minutos	1
	Más de 60 minutos	0
¿Encuentra difícil no fumar en lugares donde está prohibido (hospital, cine, biblioteca)?	Sí	1
	No	0
¿Qué cigarrillo le desagrada más dejar de fumar?	El primero de la mañana	1
	Cualquier otro	0
¿Cuántos cigarrillos fuma al día?	Menos de 10 cigarrillos/día	0
	Entre 11 y 20 cigarrillos/día	1

	Entre 21 y 30 cigarrillos/día	2
	31 o más cigarrillos	3
¿Fuma con más frecuencia durante las primeras horas después de levantarse que durante el resto del día?	Sí	1
	No	0
¿Fuma aunque esté tan enfermo que tenga que guardar cama la mayor parte del día?	Sí	1
	No	0

Valoración del resultado:

- Entre 0 y 4: El fumador es poco dependiente de la nicotina
- Entre 5 y 6: El fumador tiene una dependencia media
- Entre 7 y 10: El fumador es altamente dependiente de la nicotina.

Test de Richmond

Se trata de un test para valorar la motivación del paciente.

Pregunta	Respuesta	Puntuación
¿Le gustaría dejar de fumar si pudiera hacerlo fácilmente?	No	0
	Sí	1
¿Cuánto interés tiene usted en dejarlo?	Ningún interés	0
	Algún interés	1
	Bastante interés	2
	Mucho interés	3
¿Intentará dejar de fumar en las próximas 2 semanas?	Definitivamente No	0
	Quizás	1
	Sí	2
	Definitivamente Sí	3
¿Cuál es la posibilidad de que usted dentro de los próximos 6 meses sea un no fumador?	Definitivamente No	0
	Quizás	1
	Sí	2
	Definitivamente Sí	3

Valoración del resultado:

- Menor o igual a 4: Motivación Baja
- Entre 5 y 6: Motivación Media
- Más de 7: Motivación Alta

Medicación

En función del resultado de los test, el médico podrá recetar un medicamento para ayudar al paciente a dejar su adicción.

CIE10	Producto	C.N.
F17.210	CHAMPIX 0,5 mg y 1 mg comprimidos recubiertos con película 11 x 0,5 mg + 14 x 1 mg + 28 x 1 mg comprimidos	697864
F17.211	CHAMPIX 1 mg comprimidos recubiertos con película 56 comprimidos	697866
F17.211	CHAMPIX 0,5 mg comprimidos recubiertos con película 56 comprimidos	656184
F17.210 F17.211	ZYNTABAC 150 mg 30 comprimidos de liberación prolongada	893875

F17.210	ZYNTABAC 150 mg 60 comprimidos de liberación prolongada	893891
F17.211		

Servidor y sistema de información

Toda la información del sistema estará centralizada en el servidor. Las dos aplicaciones obtendrán toda la información del servidor, utilizando para ello las distintas tecnologías que se proponen.

En el servidor los datos serán almacenados como simples arrays de javascript (no será necesario utilizar una base de datos), por lo que cada vez que se reinicie el servidor se perderán los cambios realizados.

Los datos a gestionar son (cada uno es un array):

- **médicos:** Conjunto de médicos dados de alta en el sistema. Por cada médico se tendrán los siguientes datos:
 - **id:** Identificador del médico, número único por cada médico (1,2,3...)
 - **nombre:** Nombre del médico
 - **login:** Texto utilizado para realizar la entrada al sistema
 - **password:** Contraseña asociada al login para realizar la entrada al sistema
- **pacientes:** Conjunto de pacientes que el médico irá dando de alta
 - **id:** Identificador numérico único para cada paciente (1,2,3...)
 - **nombre:** Nombre del paciente
 - **medico:** identificador del médico
 - **codigo_acceso:** Código de acceso alfanumérico para este paciente (creado por el médico)
 - **observaciones:** Texto de observaciones realizadas por su médico
 - **estado:** 1 (inicial, sin test realizados) 2 (test realizados) 3 (finalizado)
 - **medicacion:** Medicación asignada por el médico (o null si no hay)
- **tests:** Conjunto de tests que el paciente debe realizar
 - **id:** identificador numérico único de cada test (1 y 2)
 - **nombre:** Nombre del test
- **preguntas:** Conjunto de preguntas de cada test
 - **id:** identificador numérico único de cada pregunta por test (1, 2, 3...)
 - **test:** identificador del test al que pertenece la pregunta
 - **texto:** Texto de la pregunta
- **respuestas:** Conjunto de posibles respuestas de las preguntas
 - **id:** Identificador numérico único de cada respuesta por pregunta (1, 2, 3...)
 - **test:** Identificador del test
 - **pregunta:** Identificador de la pregunta para esta respuesta

- **texto:** Texto de la respuesta
- **puntuación:** Puntuación de esta respuesta
- **contestaciones:** Conjunto de contestaciones realizadas por cada paciente a cada pregunta
 - **paciente:** Paciente que ha respondido
 - **pregunta:** Identificador de la pregunta contestada
 - **respuesta:** Identificador de la respuesta
- **medicación:** Conjunto de medicaciones que el médico puede recetar a cada paciente
 - **id:** Identificador único del medicamento
 - **cie10:** Código(s) cie10
 - **nombre:** Nombre del producto
 - **cn:** Código numérico del producto
- **mensajes:** Conjunto de mensajes enviados entre pacientes y médicos
 - **emisor:** Identificador del emisor (médico o paciente)
 - **receptor:** Identificador del receptor (médico o paciente)
 - **origen:** "M" (si el origen es un médico) o "P" (si el origen es un paciente)
 - **texto:** Texto del mensaje
 - **hora:** Fecha y hora de la creación del mensaje

A continuación, se describen los requerimientos funcionales de cada uno de los módulos

Parte 1: Gestión del paciente

En una primera parte de la aplicación se realizará la parte relativa al paciente

Esta gestión se implementará mediante la tecnología de Servicios Web tipo REST entre el servidor y la aplicación del paciente.

Requerimientos funcionales

El paciente tendrá que poder hacer las siguientes funcionalidades:

- Entrar al sistema mediante un código de acceso que el médico le habrá facilitado. (para realizar esta parte, y dado que aún no se habrá hecho la parte del médico, crear entradas en el array de pacientes con datos rellenos, incluyendo un código de acceso). Si el paciente introduce un código que no se

corresponde con ningún paciente se le indicará mediante un mensaje y no se le dejará continuar.

- Una vez dentro del sistema se le mostrará un mensaje de bienvenida con su nombre, el nombre de su médico y:
 - Si el paciente aún no ha hecho los test, se le permitirá realizarlos. Para ello se le mostrarán las preguntas con las posibles respuestas que tendrá que ir seleccionando. Una vez tenga todas elegidas (si sólo cuando estén TODAS elegidas) las respuestas se enviarán al sistema
 - Si el paciente ya ha realizado los test, se le mostrará:
 - Si el médico ya ha valorado su expediente: La medicación que el médico le ha recetado o un mensaje que de que se le ha denegado
 - Si el médico aún no ha valorado su expediente: Un mensaje indicando que su resolución está en proceso
- El paciente podrá salir del sistema

Al paciente no se le mostrará en ningún momento la puntuación de cada respuesta, ni la puntuación obtenida en los test.

Servidor REST

Para que la aplicación del paciente obtenga toda la información necesaria, el servidor ofrecerá un conjunto de servicios REST con Node/Express, para acceder a la información almacenada en los arrays del servidor.

Los servicios a implementar son:

URL	Método	Descripción
<code>/api/paciente/login</code>	POST	Realiza un login para pacientes. En body se indica un objeto con el código de acceso. Por ejemplo: <pre>{ "código_acceso": "sdf3kdc9" }</pre> Si va bien se obtiene el id del paciente: <pre>5</pre>
<code>/api/paciente/:id</code>	GET	Obtiene los datos del paciente indicado (no devolver el código de acceso) Por ejemplo, si se realiza <pre>GET /api/paciente/5</pre> Se obtiene <pre>{ "id": 5, "nombre": "Diego", "medico": 2, "observaciones": "", "estado": 2, "medicacion": null }</pre>

		}
<code>/api/medico/:id</code>	GET	Obtiene los datos del médico (no devolver la contraseña)
<code>/api/tests</code>	GET	Obtiene un array con todos los test
<code>/api/preguntas</code>	GET	Obtiene un array con todas las preguntas
<code>/api/respuestas</code>	GET	Obtiene un array con todas las respuestas
<code>/api/paciente/:id/contestaciones</code>	POST	<p>Envía las respuestas realizadas al test. El body es un array de respuestas:</p> <pre>[{ "pregunta": 1, "respuesta": 1 }, { "pregunta": 2, "respuesta": 3, } ...]</pre> <p>Cuando se registran las preguntas, el estado del paciente pasa automáticamente a 2.</p>

Todos los servicios retornarán como código de estado:

- 200: Si todo ha ido bien
- 404: Si se solicita un elemento que no existe (p.e. GET `/api/paciente/7`, y no existe un paciente con código 7)
- 403: Si la autenticación no es correcta

Parte 2: Gestión del médico

La segunda parte de la práctica consiste en la realización de la aplicación de gestión para el médico.

Se trata, al igual que la aplicación del paciente, de una aplicación web que está alojada y trabaja con el servidor, pero en este caso está implementada utilizando RPC para la comunicación.

Requerimientos funcionales

Los requerimientos funcionales de la aplicación del médico son:

- El médico entrará al sistema mediante un acceso por usuario y contraseña. Sólo si la combinación usuario/contraseña es correcta podrá continuar.
- Una vez dentro del sistema se le mostrará un mensaje de bienvenida con su nombre y podrá realizar las siguientes operaciones:
 - Nada más entrar verá un listado con los pacientes que tiene asignados, con el nombre y el estado en el que se encuentra (pendiente de test, test realizado o finalizado).
 - Tendrá una opción que le permitirá salir del sistema (la aplicación vuelve a la pantalla de login).
 - Tendrá una opción que le permitirá agregar un nuevo paciente. Esta opción le solicitará el nombre del paciente y un código de acceso que tendrá que dar al paciente para acceder a su aplicación. Al introducir un nuevo paciente se volverá a la lista y se actualizará automáticamente con el nuevo paciente introducido.
 - Si pulsa sobre un paciente de la lista consultará el expediente de ese paciente donde verá:
 - En nombre del paciente y:
 - Si el estado del paciente es 1, le mostrará un mensaje indicando que aun no ha realizado los tests.
 - Si el estado del paciente es 2, le mostrará el resultado del test. Por cada test pondrá un listado de las preguntas con la respuesta elegida (solo la elegida) y el resultado del test (ver tablas al principio). También se mostrará un selector para que el médico recete un medicamento. Habrá una opción que indique que NO se receta nada. Una vez aceptado el estado pasará a 3.
 - Si el estado es 3, simplemente aparecerá el resultado de la receta que se hizo.
 - Una opción para volver a la pantalla principal (a la lista).

Implementación

Definir los procedimientos necesarios para realizar estas operaciones en el servidor. Publicar las funciones mediante la librería RPC para NodeJS que se ha suministrado.

En la aplicación web del médico, y mediante la librería RPC para navegador que también se ha suministrado, realizar la implementación invocando a los procedimientos remotos del servidor.

Parte 3: Chat entre médico y paciente

Para realizar una comunicación directa y fluida entre médico y paciente, en esta tercera parte de la práctica se va a incluir un chat entre ellos.

Dado que la comunicación tiene que ser en tiempo real, se va a utilizar la tecnología de WebSocket para su realización. Para su implementación se integrará en el servidor el módulo de NodeJS llamado "websocket" (ver video explicativo sobre WebSocket).

Los mensajes del chat se almacenarán en el array de mensajes de forma que:

- Si en el momento de escribir el mensaje, tanto médico como paciente están conectados, el chat de ambas aplicaciones se actualizará automáticamente e instantáneamente.
- Si el destinatario del mensaje no se encuentra conectado en ese momento, el mensaje se mostrará cuando se conecte.

En ambos casos todo mensaje es almacenado, y cuando los usuarios entran en su aplicación ven un listado con TODOS los mensajes escritos, ordenados por la fecha.

Para la realización de esta práctica habrá que:

- Poner en marcha un servidor de WebSocket en el servidor que gestione los mensajes del chat y que establezca la comunicación entre médico y paciente
- Implementar, tanto en la aplicación del paciente como en la del médico (cuando el médico ha seleccionado un paciente), la interfaz del chat. El chat (en ambas aplicaciones) realizará:
 - Listará todos los mensajes escritos ordenados por fecha, mostrando por cada mensaje su fecha/hora, el texto escrito y quien lo escribió.
 - Tendrá una caja de entrada para introducir el texto de un nuevo mensaje.

- Tendrá un botón de enviar que al pulsar mandará el mensaje al destinatario (el médico si es la app del paciente, el paciente si es la app del médico)
- La fecha no será introducida por el usuario, se obtendrá automáticamente en el momento del envío.
- Si el destinatario está conectado, el servidor le enviará automáticamente el mensaje para que actualice el chat.
- En la aplicación del médico existirá un botón de “Limpiar” que borrará el chat con ese paciente (eliminará todos los mensajes). En este caso, si el paciente está conectado, también se borrará su chat en tiempo real.

La práctica se podrá realizar de forma individual o en parejas.

Entregar todos los archivos utilizados para realizar la práctica.