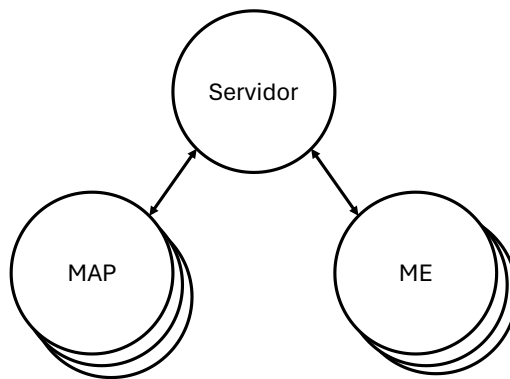


	Grado en Ing. Biomédica
	Sistemas Distribuidos
	Curso 2023-2024
	Práctica (proyecto)
	<i>Gestión de la medicación</i>

Crear una plataforma web distribuida mediante estándares (HTML, CSS y Javascript) para gestionar **interconsultas** entre médicos de atención primaria (**MAP**) y médicos especialistas (**ME**). La aplicación permitirá a un MAP crear expedientes asociados a sus pacientes para que un ME pueda dar su asistencia (pruebas, diagnósticos, opiniones...). Para ello se realizarán dos aplicaciones web, una para el MAP y otra para el ME, interconectadas ambas mediante un servidor.



Por lo tanto, el sistema distribuido estará formado por tres módulos:

- Un **servidor** basado en Javascript mediante la plataforma NodeJS.
- Una **aplicación web para el MAP**.
- Una **aplicación web para el ME**.

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

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. Existirá un **UNICO** servidor con todos los datos y que implementará las TRES tecnologías.

En el servidor, los datos serán almacenados como colecciones en memoria, simples arrays de objetos 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 los siguientes (cada uno es un array):

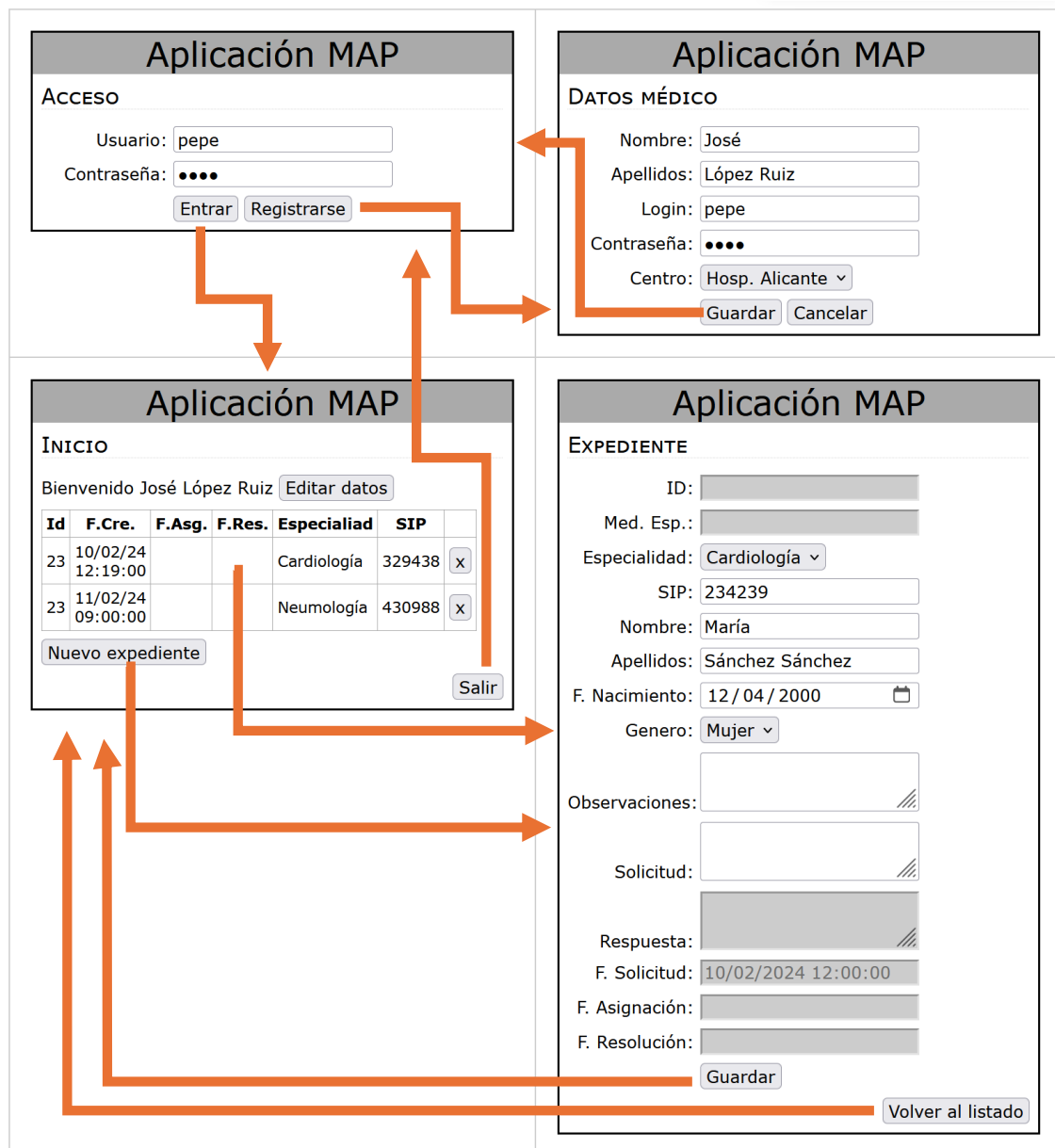
- **especialidades:** Conjunto de especialidades a las que pertenecen los ME.
 - **id:** Identificador único de la especialidad (1,2,3...)
 - **nombre:** Nombre de la especialidad
- **centros:** Conjunto de centros (hospitales, clínicas...) donde trabajan los médicos.
 - **id:** Identificador único del centro (1, 2,3...)
 - **nombre:** Nombre del centro.
- **medicos:** 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.
 - **apellidos:** Apellidos del médico.
 - **login:** Texto corto utilizado para realizar la entrada al sistema.
 - **password:** Contraseña asociada al login para realizar la entrada al sistema.
 - **especialidad:** Identificador de la especialidad (si es un ME) o 0 si es un (MAP).
 - **centro:** Id del centro del médico.
- **expedientes:** Conjunto de expedientes de interconsulta entre MAP y ME.
 - **id:** Identificador numérico único para cada expediente (1,2,3...).
 - **map:** Identificador del MAP que crea el expediente.
 - **me:** Identificador del ME que asiste el expediente (0 cuando aún no tiene un ME asignado).
 - **especialidad:** Id de la especialidad asociada al expediente.
 - **sip:** SIP del paciente
 - **nombre:** Nombre del paciente.
 - **apellidos:** Apellidos del paciente.
 - **fecha_nacimiento:** Fecha de nacimiento del paciente.
 - **genero:** Género del paciente ('H' o 'M').
 - **observaciones:** Texto de observaciones e historial realizadas por el MAP.
 - **solicitud:** Texto con la solicitud realizada por el MAP a un ME.
 - **respuesta:** Texto con la respuesta realizada por el ME al MAP.
 - **fecha_creacion:** Fecha y hora de creación del expediente.
 - **fecha_asignacion:** Fecha y hora de asignación de un ME.
 - **fecha_resolucion:** Fecha y hora de resolución del expediente.

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

Parte 1: App de gestión del MAP

La primera parte de la práctica consiste en la realización de la aplicación de gestión para el MAP.

Se trata de una aplicación web que está alojada y trabaja con el servidor, e implementada con servicios web de tipo **REST asíncronos**.



Requerimientos funcionales

Los requerimientos funcionales de la aplicación del médico son:

- Habrá una primera pantalla desde la que se podrá hacer login o registrarse.
- Un médico podrá registrarse en el sistema. Al registrarse se le pedirán todos sus datos menos el ID que se lo asignará el servidor automáticamente y especialidad que será siempre 0. El campo login no podrá estar repetido con el login de otro médico. El centro se seleccionará de un desplegable.
- 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 en la vista principal:
 - **Su nombre y apellidos**
 - Un **listado con los expedientes** que ha creado. El listado será una tabla con los siguientes datos:
 - Id del expediente
 - Fecha de creación
 - Fecha de asignación
 - Fecha de resolución
 - Especialidad (el nombre, NO el id)
 - SIP del paciente
 - Tendrá una opción que le permitirá **salir** del sistema (la aplicación vuelve a la pantalla de login).
 - Tendrá una opción que le permitirá **editar** sus datos (todos menos el ID).
 - Tendrá una opción que le permitirá **agregar un nuevo expediente**. El expediente requerirá de un especialista, pero el ME en concreto no se indicará. Este será autoasignado por un ME desde su aplicación. La especialidad se elegirá desde un desplegable. Podrá introducir todos los datos del expediente excepto:
 - id: lo creará automáticamente el servidor
 - map: lo rellenará automáticamente el servidor
 - me: lo asignará el ME que “tome” el expediente
 - respuesta: lo rellenará el ME desde su aplicación.
 - fecha_creacion: la asignará automáticamente el servidor
 - fecha_asignacion: se asignará cuando un especialista “tome” el expediente con la aplicación del ME.
 - fecha_resolucion: se asignará cuando un especialista de un resultado.
 - Si pulsa sobre un expediente de la lista **consultará el expediente** donde:
 - Verá todos los datos del expediente
 - Podrá actualizar algunos datos del expediente (no podrá modificar los mismo que no podía introducir al crear un expediente).

- En la lista, cada expediente tendrá un botón eliminar que permitirá eliminar ese expediente.

No será necesario realizar una gestión de especialidades ni centros. El listado de especialidades y centros ya estarán creados en sus arrays al iniciar el servidor.

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

Servidor REST

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

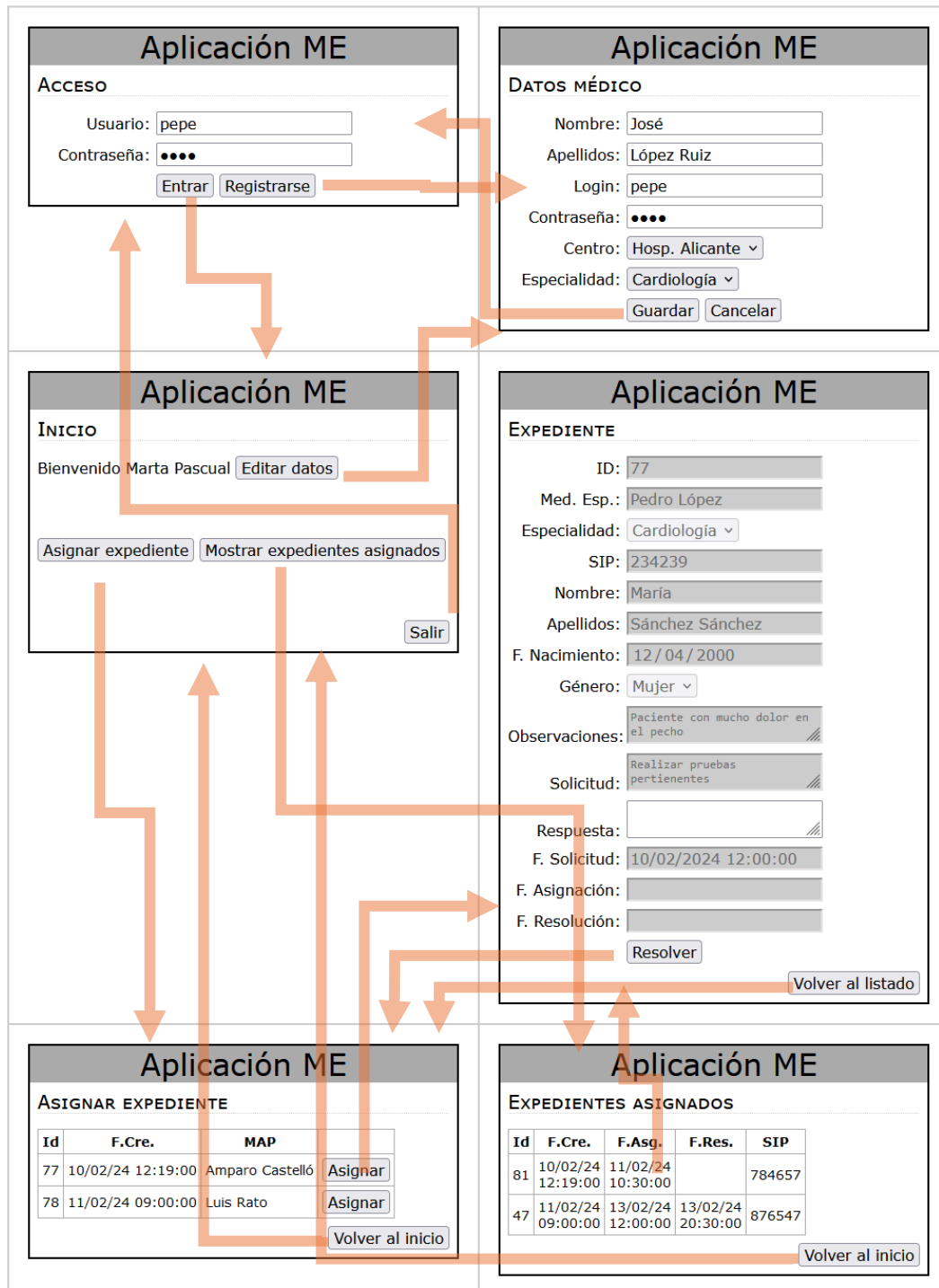
Los servicios a implementar son (SOLO IMPLEMENTAR ESTOS, **NO SE PUEDE IMPLEMENTAR OTROS**):

URL	Método	Descripción
<code>/api/especialidades</code>	GET	Obtiene un array con todos las especialidades
<code>/api/centros</code>	GET	Obtiene un array con todos los centros
<code>/api/medico/login</code>	POST	Realiza un login para médico. En body se indican las credenciales. Por ejemplo: <pre>{ "login": "xxx", "password": "secreto" }</pre> Si va bien se obtiene el id del médico: 5
<code>/api/medico/:id</code>	GET	Obtiene los datos del médico (no devolver la contraseña)
<code>/api/medico</code>	POST	Crea un nuevo médico. Mandar en el body todos los datos excepto ID, que lo creará el servidor. El servidor comprobará que no se repite el campo login con otro médico.
<code>/api/medico/:id</code>	PUT	Cambia la información del médico indicado en el ID. Nuevamente evitar que se repita un login con otro médico.
<code>/api/map/:id/expedientes</code>	GET	Obtiene un array con los expedientes creados por un MAP
<code>/api/map/:id/expedientes</code>	POST	Crea un nuevo expediente para el id de MAP indicado. No pasar los datos prohibidos previamente. Si va bien devolverá el id del nuevo expediente.
<code>/api/expediente/:id</code>	PUT	Actualiza los datos de un expediente. No pasar los datos prohibidos anteriormente.
<code>/api/expediente/:id</code>	DELETE	Borra un expediente

Todos los servicios retornarán como código de estado:

- 200: Si todo ha ido bien
- 201: Si un servicio POST ha creado correctamente un registro
- 404: Si se solicita un elemento que no existe (p.e. GET /api/medico/7, y no existe un médico con código 7)
- 403: Si la autenticación (login) no es correcta

Parte 2: App de gestión del ME



En esta segunda parte de la práctica se aborda el **desarrollo de la aplicación que usará el médico especialista.**

En este caso, la aplicación utilizará método **RPC asíncronos** para su implementación.

Requerimientos funcionales

- Habrá una **primera pantalla desde la que se podrá hacer login o registrarse.**
- Un **médico especialista podrá registrarse en el sistema.** Al registrarse se le **pedirán todos sus datos (incluida la especialidad) menos el ID que se lo asignará el servidor automáticamente.** El campo login no podrá estar repetido con el login de otro médico. El **centro se seleccionará de un desplegable.**
- 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 en la vista principal donde se mostrará su nombre y apellidos**
 - Tendrá una opción que le permitirá **salir** del sistema (la aplicación vuelve a la pantalla de login).
 - Tendrá una opción que le permitirá **editar** sus datos (todos menos el ID).
 - Tendrá un **botón que le permitirá asignarse expedientes libres.** Al pulsar el botón se le **mostrará una tabla donde se mostrarán expedientes creados por MAP.**
 - Sólo se mostrarán expedientes que cumplan las siguientes condiciones:
 - **No estar ya asignados a otro ME**
 - **Que la especialidad solicitada sea la del ME que lo consulta.**
 - Por confidencialidad, de cada **expediente** sólo se podrán ver los siguientes datos:
 - **Id**
 - **Nombre del MAP**
 - **Fecha de creación**
 - Por cada expediente habrá un **botón de ASIGNAR** que, al pulsarlo, **asignará ese expediente al ME.** Al asignarlo se asignaran automáticamente los campos del expediente:
 - **me:** con el id del ME que lo ha tomado
 - **fecha_asignacion:** con la fecha actual
 - **Se abrirá una ficha donde el ME podrá consultar el expediente**
 - Se podrá volver a la vista inicial.

- Tendrá un segundo botón desde el que poder consultar una tabla con todos los expedientes asignados.
 - Se mostraran todos los expedientes asignados por este ME
 - Se mostrarán por cada expediente los siguientes datos:
 - Id
 - Fecha_creacion
 - Fecha_asignacion
 - Fecha_resolucion
 - Sip
 - Al pulsar en cada expediente se abrirá también la ficha del expediente
 - Se podrá volver a la vista inicial.
- Desde la ficha de cada expediente se podrá consultar todos los datos del expediente y:
 - Solo se podrá modificar la respuesta
 - Al guardar se actualizará automáticamente la fecha de resolución
 - Se podrá volver a la lista de expedientes asignados

Servidor

El servidor RPC implementará todas las funciones que podrá invocar el cliente. Utilizar la librería RPC que se aporta.

Hay que implementar los siguientes procedimientos RPC:

Nombre	Argumentos	Descripción y Retorno
obtenerEspecialidades		Array con todas las especialidades
obtenerCentros		Array con todos los centros
login	login password	ID del ME si las credenciales son correctas o null si no
crearME	datos (objeto con todos los datos del ME)	ID del nuevo ME o null si hay error (p.e. se repite login)
actualizarme	id_me datos (objeto con todos los datos del ME)	Actualiza los datos del médico con indentificador id_med
obtenerDatosMedico	id_medico	Datos de un médico (excepto password) o null si el id no se corresponde a un médico
obtenerExpDisponibles	id_especialidad	Array de expedientes sin ME y de esa especialidad (solo id, map, fecha_creacion)
asignarExp	id_exp id_me	Se asignará a un ME el expediente y se rellenará la fecha_asignacion. Retorna un booleano de si ha ido bien.

obtenerExpAsignados	id_me	Array de expedientes asignados a ese ME (todos los datos)
resolverExp	id_exp respuesta	Resuelve un expediente poniendo la respuesta y asignado la fecha_resolucion

Integrar este servidor con el servidor de REST, de forma que se lance un único servidor. Se recomienda separar en archivos JS distintos el servidor principal, el servidor REST, el servidor RPC y los datos.