



El futuro digital
es de todos

MinTIC

Sprints: Desarrollo Aplicaciones Web.



Universidad
Pontificia
Bolivariana

Vigilada MinEducación

Mision
TIC 2022



Generalidades

- Recuerda reunirte con tu equipo de trabajo para determinar los acuerdos en la ejecución de las actividades.
- Implementen la primera versión de su proyecto con algunas funcionalidades básicas y definan los repositorios compartidos para la integración y despliegue continuo.
- Este sprint tiene como propósito realizar una implementación con una API de prueba.
- Hasta este momento debes tener presente las vistas de tu proyecto, la estructura de manejo de datos y la lógica de operación de las funcionalidades.

Recomendaciones

1. Entrega únicamente los archivos solicitados dentro del espacio de la actividad.
2. Esta debe hacerse a través del autocalificador codegrade.
3. Verifica los nombres de los archivos de entrega y la extensión.
4. Ingresen al espacio en plataforma “Actividad: Sprint 3” y sigan las instrucciones.
5. Esta actividad tiene como propósito revisar el concepto de API y su uso desde una fuente externa.

Introducción/Información:

A través de la Resolución 1552 del 14 de mayo de 2013, el Ministerio de Salud y Protección Social, reglamentó que las Entidades Promotoras de Salud (EPS), de ambos regímenes, directamente o a través de la red de prestadores que definan, deberán tener siempre agendas abiertas para la asignación de citas de medicina especializada la totalidad de días hábiles del año. Dichas entidades en el momento en que reciban la solicitud, informarán al usuario la fecha para la cual se asigna la cita, sin que les sea permitido negarse a recibir la solicitud y a fijar la fecha de la consulta requerida.

Las medidas hacen parte de la reglamentación de los artículos 123 y 124 del [Decreto Ley 019 de 2012](#) conocido como Decreto Antitrámites.

Objetivos:

- Resolver las historias de usuario propuestas como requisitos del proyecto de aplicaciones web asignado durante el ciclo.
- Repasar el concepto de microservicios, APIs y su implementación dentro de proyectos.
- Implementar el primer prototipo del proyecto según las vistas, estructura de manejo de datos y funcionalidades básicas.

Contexto:

Como desarrollador de aplicaciones web ha sido contratado para diseñar la nueva interfaz gráfica para la página web de la EPS COLSALUD, donde se podrán agendar las citas medicas por parte de sus afiliados

Historias de usuario:

Identificador Historia#:	HU-04	Título:	Búsqueda citas médicas desde la API
Descripción	COMO:	Usuario	
	QUIERO:	Buscar citas disponibles, de acuerdo a mis preferencias	
	PARA:	Ordenar mejor mi tiempo.	
Criterios de aceptación	1. DADO que un usuario se encuentra en el sistema CUANDO le da click en la sección Solicitar Cita médica ENTONCES el sistema le mostrará un formulario para poder llenar sus datos de preferencia.		
	1. Una empresa de backend fue contratada para construir una api que retorne las citas medicas registrados en el sistema, siéntete libre de explorar su resultado con el navegador de tu preferencia o un cliente http.		
	2. https://misiontic2022upb.vercel.app/api/medical-appointments/appointments		
	1. Crear una función (async) que se llame obtenerCitasDisponibles , que reciba 3 parámetros, especialidad, fecha_inicio, fecha_final y debe retornar las citas que concuerden con esos 3 parámetros.		
	2. Especialidad será “odontología” , “medicina” y “optometría” , fecha_inicio y fecha_final son el rango de fechas a buscar.		
	3. Llamar a la API https://misiontic2022upb.vercel.app/api/medical-appointments/appointments para retornar el array de citasAPI. Use Async/Await.		
	4. Filtre el array citasAPI que concuerde así 3.1 que la especialidad de la cita sea igual al parámetro especialidad 3.2 que la fecha de la cita sea mayor o igual al parámetro fecha_inicio y 3.3 que el la fecha de la cita sea menor o igual al parámetro fecha_final.(se sugiere usar la función filter)		
	5. Retorne el resultado del item 4.		
	6. Exporte la función empleando la siguiente sentencia de código:		
	• module.exports. obtenerCitasDisponibles= obtenerCitasDisponibles;		

Historias de usuario:

Identificador Historia#:	HU-05	Título:	Confirmar Cita
Descripción	COMO:	Usuario	
	QUIERO:	Confirmar la cita escogida	
	PARA:	Poder guardarla y posteriormente hacer uso de ella	
Criterios de aceptación	I. DADO que un Usuario se encuentra en el sistema CUANDO le da click en el botón confirmar cuenta. ENTONCES el sistema envía los datos a la API para que se agende y confirme la cita		
	1. *Una empresa backend fue contratada para construir un API que retorne para hacer el Create de la cita medica. Siéntete libre de explorar su resultado en algún cliente http (usando el método post). https://misiontic2022upb.vercel.app/api/medical-appointments/confirm/1		
	1. Crear una función (async) confirmarCita que reciba un parámetro id, que es el id de la tarea a confirmar. 2. Enviaremos un objeto por método POST a la API https://misiontic2022upb.vercel.app/api/medical-appointments/confirm/[id] , y no le vamos a enviar ningún body, [id] es el id de la cita use ASYNC/AWAIT 3. Retornar el string resultado del item 2. 4. Exporte la función empleando la siguiente sentencia de código: • module.exports.confirmarCita= confirmarCita;		

Sprint backlog

1. Construya un tablero de apoyo para la asignación y estado de las actividades.
2. Buscar la fórmula de depreciación lineal, que está sugerida en los materiales de apoyo de este enunciado.
3. Revisar como llamar una API con fetch (usando async/await), trabajar con objetos, recorrer un Array, clonar objetos (spread operator o Object.assign) y adicionar elementos en un array guiándose de los materiales de apoyo
4. Preparar y codificar las funciones y variables.
5. Revisar muy bien los strings de retorno, nombre de funciones y variables, con un gran nivel de detalle.
6. Probar por aparte su código, para su comodidad (opcional).
7. Establezcan una revisión periódica de los avances.
8. Realicen el montaje del primer prototipo de manera local en sus máquinas.

Entrega:

Suba a la plataforma los archivos JS, que empleó para dar solución a cada una de las actividades de las historias de usuario planteadas en este sprint; el archivo debe de llamarse **src.js**.

Adicionalmente, incluya el archivo en formato JSON disponible en la descripción del sprint (“appointments.json”) de lo contrario no se calificará. Para el fetch hacer uso de ASYNC/AWAIT

Para la evaluación de este sprint es **muy importante** seguir muy detalladamente las instrucciones, los valores de retorno, las variables, strings.

```
const obtenerCitasDisponibles = async (
  especialidad,
  fecha_inicio,
  fecha_final
) => {
  let response = await fetch(
    "https://misiontic2022upb.vercel.app/api/medical-appointments/appointments"
  );
};

const confirmarCita = async (idCita) => {
  let response = await fetch(
    "https://misiontic2022upb.vercel.app/api/medical-appointments/confirm/"+idCita,
    {
  };
};

module.exports.obtenerCitasDisponibles = obtenerCitasDisponibles;
module.exports.confirmarCita = confirmarCita;
```


Material de apoyo:

<https://www.jose-aguilar.com/blog/5-formas-diferentes-para-recorrer-un-array-con-javascript/>

<https://dmitripavlutin.com/javascript-fetch-async-await/#2-fetching-json>

https://www.w3schools.com/js/js_async.asp

https://www.w3schools.com/js/js_objects.asp

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

https://www.w3schools.com/jsref/jsref_tolowercase.asp