



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.
- Ten presente avanzar en el valor agregado de tu proyecto. Si resuelven las historias de usuario en poco tiempo, podrán concentrar sus esfuerzos en los componentes diferenciales.
- Este sprint tiene como propósito hacer una transición hacia las tecnologías del back-end, por ello se hará uso de los conocimientos adquiridos en JavaScript para analizar y resolver los requisitos solicitados en este enunciado.

Recomendaciones

1. Esta entrega se realizará a través del autocalificador codegrade.
2. Verifica los nombres de los archivos de entrega y la extensión.
3. Ingresen al espacio en plataforma “Actividad: Sprint 2” y sigan las instrucciones.
4. Esta actividad busca que analices nuevamente la estructura de datos que debes manejar; una aplicación funciona con base en eventos, lectura de datos y análisis de información.

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 funciones, parámetros y argumentos haciendo uso del lenguaje de programación JavaScript.
- Analizar de forma preliminar el comportamiento y la implementación que da lugar a la separación del front-end y del back-end.

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-03	Título:	Obtener citas disponibles
Descripción	COMO:	Usuario	
	QUIERO:	Ingresar los datos de citas	
	PARA:	Ver la lista de citas disponibles que se acoplen a mi tiempo	
Criterios de aceptación	I. DADO que un usuario se encuentra en el sistema CUANDO se va a escoger la búsqueda de citas disponibles ENTONCES el sistema le retorna las citas disponibles de acuerdo a sus preferencias. * En el ambiente de pruebas hay un array llamado CITAS que ya viene precargado con unas citas ficticias, tendrá que usar este para ejecutar la prueba, OJO sólo está en el ambiente de pruebas. Un ejemplo de este está en la sección Entrega		
	1. Crear una función 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. Filtre el array CITAS 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. 4. Exporte la variable empleando la siguiente sentencia de código: <ul style="list-style-type: none">• module.exports. obtenerCitasDisponibles=obtenerCitasDisponibles;		



Identificador Historia#:	HU-04	Título:	Obtener citas disponibles por jornada
Descripción	COMO:	Usuario	
	QUIERO:	Ingresar la jornada que quiero la cita	
	PARA:	Ver la lista de citas disponibles que se acoplen a mi tiempo	
Criterios de aceptación	I. DADO que un usuario se encuentra en el sistema CUANDO se va a escoger la búsqueda de citas disponibles ENTONCES el sistema le retorna las citas disponibles de acuerdo a sus preferencias. * En el ambiente de pruebas hay un array llamado CITAS que ya viene precargado con unas citas ficticias, tendrá que usar este para ejecutar la prueba, OJO sólo está en el ambiente de pruebas. Un ejemplo de este está en la sección Entrega		
	<div>1. Crear una función que se llame obtenerCitasPorJornada que reciba 4 parámetros, especialidad, fecha_inicio, fecha_final y jornadaPreferida y debe retornar las citas que concuerden con esos 4 parámetros.</div> <div>2. Especialidad será “odontología” , “medicina” y “optometría” , fecha_inicio y fecha_final son el rango de fechas a buscar, jornadaPreferida será “mañana” o “tarde”.</div> <div>3. Filtre el array CITAS que concuerde como el punto 3 de la historia de usuario anterior, (hint) puede llamar la función hecha en la historia de usuario anterior, para que reúses código.</div> <div>4. Filtra el resultado del punto 3 de la siguiente manera, si la jornada es mañana sólo se van a filtrar las horas de 8 a 12 y si es tarde se va a filtrar las citas que tengan hora de 1 a 5. (ver la siguiente sección para ahondar más)</div> <div>5. Podrás notar que esta historia de usuario usa dos filtros.</div> <div>6. Exporte la variable empleando la siguiente sentencia de código:<ul style="list-style-type: none">• module.exports. obtenerCitasPorJornada=obtenerCitasPorJornada;</div>		

Sprint backlog

1. Construya un tablero de apoyo para la asignación y estado de las actividades.
2. Recordar que para comparar fechas en js se utiliza los operadores normales excepto ==, mirar material de apoyo.
3. Revisar muy bien los strings de retorno, nombre de funciones y variables, con un gran nivel de detalle.
 - Estar muy seguro de los strings “mañana” , “tarde” están bien escritos.
4. Revisar la función filter propuesto en el material de apoyo.
5. Extraer la hora de la propiedad hora de la cita, para eso puedes usar la función Split propuesta en el material de apoyo.
6. Transformar la hora extraída en entero para que no haya problemas con la comparación, no es lo mismo comparar strings que enteros.
7. Probar por aparte su código, para su comodidad (opcional).
8. Establezcan una revisión periódica de los avances.



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**, de lo contrario no se calificará.

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

```
const obtenerCitasDisponibles=(especialidad,fecha_inicio,fecha_final)=>{
  return CITAS.filter(.....
);
}
const obtenerCitasPorJornada=(especialidad,fecha_inicio,fecha_final,jornadaPreferida)=>{
  let resultados= obtenerCitasDisponibles(especialidad,fecha_inicio,fecha_final);
  return resultados.filter(...);
}
module.exports.obtenerCitasDisponibles=obtenerCitasDisponibles;
module.exports.obtenerCitasPorJornada=obtenerCitasPorJornada;
```

```
global.CITAS = [
  {
    especialidad: "odontología",
    nombre: "Dr. Juan",
    fecha: "2020-01-01",
    hora: "08:00",
  },
  ...
  {
    especialidad: "medicina",
    nombre: "Dr. Mario",
    fecha: "2020-01-01",
    hora: "15:00",
  },
  .....
  {
    especialidad: "optometría",
    nombre: "Dr. Mario",
    fecha: "2020-01-01",
    hora: "16:00"
  }
]
```


Material de apoyo:

<https://desarrolloweb.com/articulos/comparar-fechas-javascript.html>

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

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/String/includes

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

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/String/split