



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:

En Colombia, al ICA le corresponde establecer las medidas sanitarias y fitosanitarias para prevenir, controlar o erradicar las plagas y enfermedades que afectan los productos o subproductos, es por ello que desde los años 2004 y 2008 mediante las resoluciones ICA 1806, 2964 viene fortaleciendo las acciones de inspección, vigilancia y control a los predios productores y exportadores de frutas, hortalizas y aromáticas, y productos de origen vegetal.

Por lo anterior expuesto se hace necesario llevar un control de inventarios de insumos agrícolas aplicados en los predios y lotes que cuentan con esta certificación.

## 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 una interfaz gráfica para el sistema de gestión de inventarios que estarán utilizando los productores agrícolas de fruta de exportación, es necesario que la aplicación sea muy simple y funcional para la gestión de inventarios en el cultivo de fruta.

## Historias de usuario:

Identificador Historia#:	HU-01	Título:	Registrar record de visita
Descripción	COMO:	usuario	
	QUIERO:	Ingresar la información suministrada por el ingeniero agrónomo en una finca determinada	
	PARA:	Verificar el registro realizado por el ingeniero agrónomo	
Criterios de aceptación	I. DADO que un usuario se encuentra en el sistema CUANDO se va a registros ENTONCES el sistema un formulario donde pueda diligenciar la información		
	1. *Una empresa backend fue contratada para construir un API que retorne para hacer el Create de las visitas técnicas. Siéntete libre de explorar su resultado en algún cliente http (usando el método post). <a href="https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit">https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit</a>		
	1. Crear una función (async) insertarVisita que reciba 6 parámetros, fecha, finca, lote, labor, área, tecnico. 2. Enviaremos un objeto por método POST a la API <a href="https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit">https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit</a> y su estructura será: { fecha: "2021-11-12", finca: "Altagracia", lote:"Lote 4", labor: "primera abonada", área: "4 ", técnico: "Anibal Zamora"} usando fetch ASYNC/AWAIT 3. Retornar el string resultado del item 2. 4. Exporte la variable empleando la siguiente sentencia de código: • <b>module.exports.insertarVisitas=insertarVisitas;</b>		

fecha visita	nombre Agronomo	nombre finca	ver detalle
12/10/2021	ANIBAL ZAMORA	ALTAGRACIA	
12/10/2021	ANIBAL ZAMORA	LA PRIMAVERA	
13/10/2021	ANIBAL ZAMORA	EL ENCANTO	

## Historias de usuario:

Identificador Historia#:	HU-02	Título:	Búsqueda del record de visita
Descripción	COMO:	usuario	
	QUIERO:	Buscar la información suministrada por el ingeniero agrónomo en una finca determinada	
	PARA:	Tener un listado de los registros realizados por el ingeniero agrónomo	
Criterios de aceptación	1. DADO que un usuario se encuentra en el sistema CUANDO se va a listar registros ENTONCES el sistema muestra una tabla donde están los registros realizados por el agrónomo.		
Precondiciones	1. Una empresa de backend fue contratada para construir una api que retorne los sitios turísticos registrados en el sistema, siéntete libre de explorar su resultado con el navegador de tu preferencia o un cliente http. <a href="https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit">https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit</a>		
Instrucciones	1. Crear una función (async) que se llame filtrarResultados , que reciba dos parámetros, key, value. 2. Key es el criterio de búsqueda, “fecha”, “finca”, “lote” , “labor”, “área”, “técnico” y value es el valor a buscar eje: “2021-11-13” 3. Llamar a la API <a href="https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit">https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit</a> para retornar el array de visitas. Haga uso del ASYNC/AWAIT. 4. La función debe retornar los resultados filtrados ej. Si key=“finca” y value=“Altagracia” entonces se retornará los registros del array visitas (creado en la anterior historia de usuario) que en su propiedad finca contengan el valor “Altagracia”. Se sugiere usar la función filter 5. En el punto 3 se requiere que AlTaGraCiA haga match con Altagracia, se sugiere usar el método toLowerCase. 6. Exporte la función empleando la siguiente sentencia de código: <ul style="list-style-type: none"><li><b>module.exports. filtrarResultados= filtrarResultados;</b></li></ul>		

## 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 ("technical-visit.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 insertarVisita= async(
  fecha,
  finca,
  lote,
  labor,
  area,
  tecnico
)=>{
  let visita2send={
    fecha,
    finca,
    lote,
    labor,
    area,
    tecnico
  };
  let response = await fetch(
    "https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit",
    {
      method: "POST",
      headers: {
        Accept: "application/json",
        "Content-Type": "application/json",
      },
      body: .....,
    }
  );
  .....;
};

const filtrarResultados = async (key, value) => {
  let response = await fetch(
    "https://misiontic2022upb.vercel.app/api/agricultural-inputs/technical-visit"
  );
  let visitasAPI = await response.json();
  .....
};
module.exports.insertarVisita = insertarVisita;
module.exports.filtrarResultados = filtrarResultados;
```



## Material de apoyo:

<https://www.globalgap.org/es/for-producers/>

<https://www.ica.gov.co/periodico-virtual/prensa/el-registro-de-predios-productores-agropecuarios-f.aspx>

[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)

[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working\\_with\\_Objects](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working_with_Objects)

[https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\\_Objects/String/includes](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/Array/filter)

[https://www.w3schools.com/jsref/jsref\\_tolowercase.asp](https://www.w3schools.com/jsref/jsref_tolowercase.asp)