



El futuro digital
es de todos

MinTIC

Ciclo 3:

Sprints de evaluación.



Universidad
Pontificia
Bolivariana

Vigilada MinEducación

Misión
TIC 2022



IMPORTANTE:

La información contenida en esta presentación es de carácter pedagógico, el material presentado obedece a situaciones que simulan el desarrollo de un proyecto en un entorno organizacional a partir de una metodología ágil como lo es SCRUM.

En este sentido, no se exigirá dominar los artefactos, herramientas y cargos de la metodología, pero si establecer acuerdos para llevar a buen fin el proyecto.

¿Cómo afrontar los sprints durante este ciclo 3?



Recuerda...

¿Cuál es el alcance del proyecto que se debe desarrollar durante todo el ciclo?

1. Un sitio web, para un sector económico de relevancia actual, en el que se aplique el conocimiento adquirido durante ciclo 3
2. El sitio debe estar soportado por una base de datos en la que se pueda gestionar la información asociada al proyecto.
3. El sitio debe cumplir con una interfaz gráfica para ser usada en diferentes navegadores.
4. El sitio debe ser desplegado en un servicio cloud para el acceso durante la feria de proyectos.

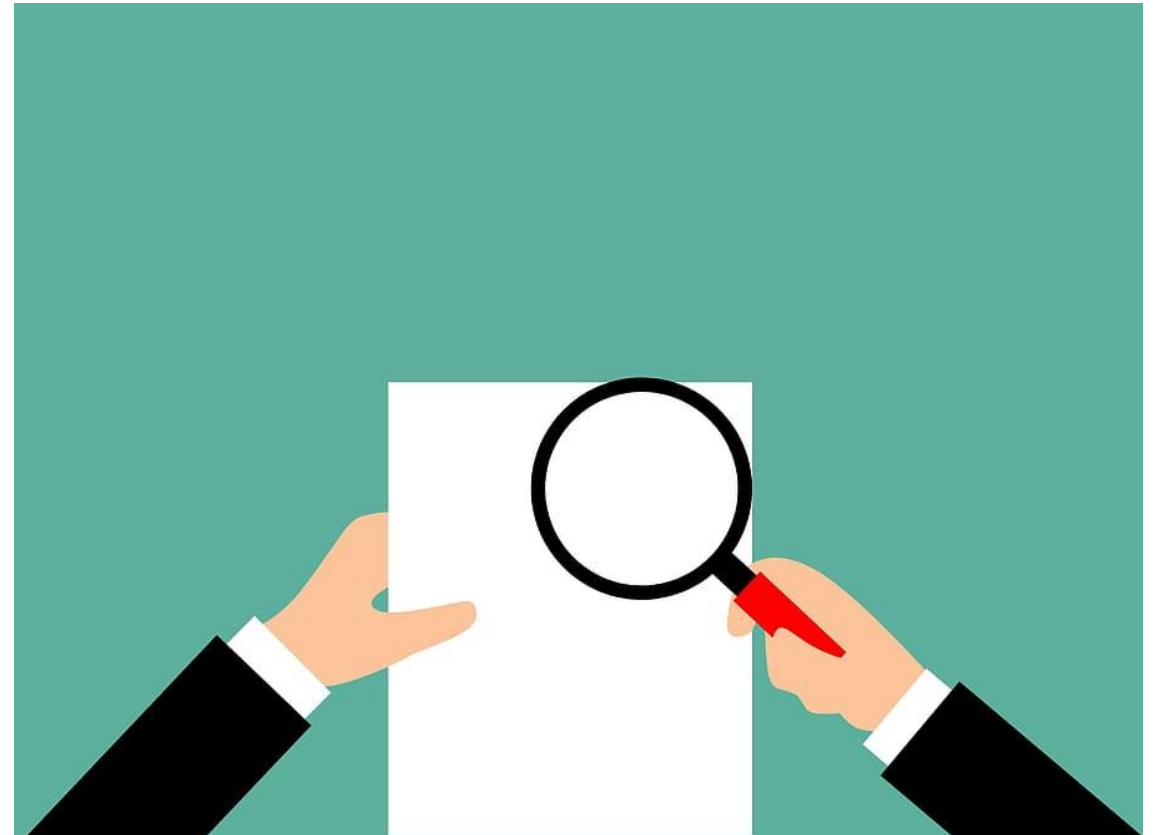
Los sprints están definidos bajo historias de usuario con requerimientos genéricos de sitios web. Su propósito es validar los conocimientos en los temas centrales del ciclo y se publicarán semanalmente.

Los sprints por si solos no completan el proyecto, por lo tanto, el trabajo de cada subgrupo será refinar al máximo su propuesta, durante las semanas de formación, para la postulación y participación en el evento de cierre de ciclo.

Product Backlog:

Historias de usuario:

1. Yo como administrador quiero un login para los usuarios que ya se encuentran registrados.
2. Yo como administrador requiero que el login cuente con un CAPTCHA como filtro adicional para que los usuarios ingresen a la aplicación.



Sprint Backlog:

Historia de usuario

Yo como administrador quiero un login para los usuarios que ya se encuentran registrados.

Actividades

1. En un nuevo documento **login.html**.
Crear un formulario de inicio de sesión que le permita a los usuarios ya registrados ingresar a su cuenta.

El formulario de **login** debe de construirse de la siguiente manera:

- Emplee la etiqueta `<form></form>` para crear el elemento y añada el atributo id igual a "form-login".
- El formulario debe contar con dos **inputs**, uno para el **correo** y otro para la **contraseña**. Además, cada input debe de tener un atributo id igual al nombre del campo, es decir, "**correo**" y "**contrasena**" respectivamente. Estos campos serán requeridos para que el usuario pueda ingresar a su cuenta.
- De igual forma, el formulario debe de tener un input adicional para la validación del CAPTCHA con un atributo id igual a **captcha**, esto con el fin de verificar la autenticidad del usuario.
- **Nota: Sí ya cuenta con un formulario de inicio de sesión omita esta actividad**

Sprint Backlog:

Historia de usuario

Yo como administrador quiero un login para los usuarios que ya se encuentran registrados.

Ejemplo gráfico

Iniciar sesión

Correo

Contraseña

 CAPTCHA

Login

Sprint Backlog:

Historia de usuario

Yo como administrador quiero un login para los usuarios que ya se encuentran registrados.

Actividades

1. Validación de ingreso a la aplicación. Para que un usuario pueda ingresar a la aplicación debe encontrarse registrado, que sus datos de ingreso coincidan y que complete un CAPTCHA.

Instrucciones:

- Implementar la función **login()**, esta función estará encargada de validar los campos ingresados con la información alojada en el arreglo global donde se encuentran cada uno de los registros de los usuarios de la aplicación (arreglo trabajado en la actividad uno del sprint número tres). Así mismo, en esta función se invocará la función **validarCAPTCHA()**, que como su nombre lo indica, añade un filtro adicional al inicio de sesión. Al cumplirse las condiciones antes descritas el usuario podrá ingresar a la aplicación.
 - NOTA IMPORTANTE: Debes exportar el arreglo global declarado con la sentencia
 - **module.exports.registros = registros;**
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports.login = login;**

Sprint Backlog:

Historia de usuario

Yo como administrador requiero que el login cuente con un CAPTCHA como filtro adicional para que los usuarios ingresen a la aplicación.

Actividades

2. Implementar una función estática que por medio de una pregunta, compruebe la autenticidad de usuario y evite que el registro esté siendo diligenciado por un Bot.

Instrucciones:

- Crear la función **validarCAPTCHA(valor)**, en donde se verificará la autenticidad de un usuario.
- La función deberá validar el valor ingresado a partir de la pregunta **¿Qué resultado da el cuarto de un tercio de 12000?**. La función retornará verdadero si el valor ingresado es correcto o falso en caso de no serlo.
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports.validarCAPTCHA = validarCAPTCHA;**

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 **login.js**, de lo contrario no se calificará.

Para la evaluación de este sprint es **muy importante** que los valores de los atributos **id** de los elementos sean **exactamente iguales** a los indicados en las actividades del sprint. De igual manera, nombre las funciones **exactamente igual** a como se nombran en el instructivo de cada actividad del sprint actual.

NOTA IMPORTANTE: Incluya en este archivo la función de agregar registros implementado en el sprint 3.

```
let registros = [];  
  
function login(){  
  
}  
  
function agregarRegistro(){  
  
}  
  
function validarCAPTCHA(valor){  
  
}  
  
module.exports.login = login;  
module.exports.registros = registros;  
module.exports.validarCAPTCHA = validarCAPTCHA;  
module.exports.agregarRegistro = agregarRegistro;
```



www.upb.edu.co/es/mision-tic
#MisiónTICSomosTodos