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.

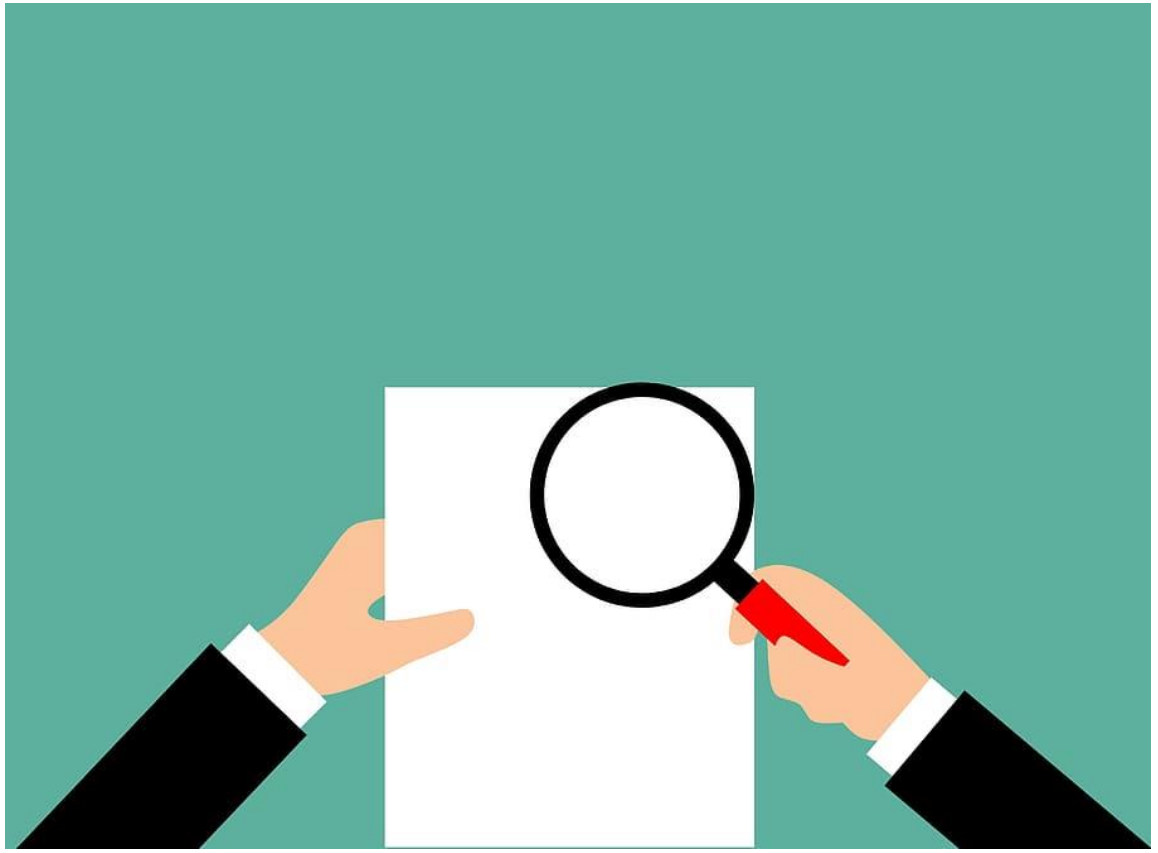
Los sprints están definidos bajo historias de usuario con requerimientos genéricos de sitios web. Su propósito es validar los conocimientos en los temas centrales del ciclo y se publicarán semanalmente.

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 validar los campos del formulario que los usuarios usan para diligenciar sus datos de registro.



Sprint Backlog:

Historia de usuario

Yo como administrador quiero validar los campos del formulario que los usuarios usan para diligenciar sus datos de registro.

Actividades

- El formulario cuenta con un <input> cuyo id es **“dato_nombre_usuario”**. Debe implementar una función de JS **validar_nombre_usuario** que recibe un parámetro tipo string, la función realiza las siguientes validaciones para el string:
    - Solo puede contener solo letras de la A a la Z y espacios.
    - Debe validar que el string no inicie ni termine con espacios.
    - Debe validar que la primera letra después de cada espacio sea mayúscula.

  En caso de cumplir las condiciones retorna verdadero, de lo contrario retorna falso.

Sprint Backlog:

Historia de usuario

Yo como administrador quiero validar los campos del formulario que los usuarios usan para diligenciar sus datos de registro.

Actividades

- El formulario cuenta con un <input> cuyo id es **“dato_edad_usuario”**. Debe implementar una función de JS **validar_edad_usuario** que recibe un parámetro, la función realiza las siguientes validaciones:

  - El valor del parámetro es un número.
  - El número debe ser positivo.
  - El número debe ser igual o mayor a 13 y menor que 110.

  En caso de cumplir las condiciones retorna verdadero, de lo contrario retorna falso.

Sprint Backlog:

Historia de usuario

Yo como administrador quiero validar los campos del formulario que los usuarios usan para diligenciar sus datos de registro.

Actividades

- El formulario cuenta con un <input> cuyo id es **“dato_contrasena”**. Debe implementar una función de JS **validar_contrasena** que recibe un parámetro tipo string, la función realiza las siguientes validaciones para el string:

  - Debe tener 6 o más caracteres.
  - Solo puede contener caracteres alfanuméricos. Es decir, letras de la A a la Z y los números del 0 al 9.

  En caso de cumplir las condiciones retorna verdadero, de lo contrario retorna falso.

Entrega:

1. Debe entregar un archivo llamado exactamente **validaciones.js**, de lo contrario no se calificará.
2. Es muy importante que las etiquetas y atributos id de los elementos que use sean **exactamente iguales** que los que se indican en las actividades de cada historia de usuario.
3. Los nombres de las funciones y la estructura del archivo **validaciones.js** deben ser exactamente los que se muestran a continuación. No olvide agregar los **module.exports**.

```
function validar_nombre_usuario(string) {
    //Implementación
}

function validar_edad_usuario(edad) {
    //Implementación
}

function validar_contrasena(string) {
    //Implementación
}

module.exports.validar_nombre_usuario = validar_nombre_usuario;
module.exports.validar_contrasena = validar_contrasena;
module.exports.validar_edad_usuario = validar_edad_usuario;
```

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

#MisiónTICSomosTodos