



El futuro digital
es de todos

MinTIC

Ciclo 3:

Sprints de evaluación.



Universidad
Pontificia
Bolivariana

Vigilada MinEducación

Mision
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.

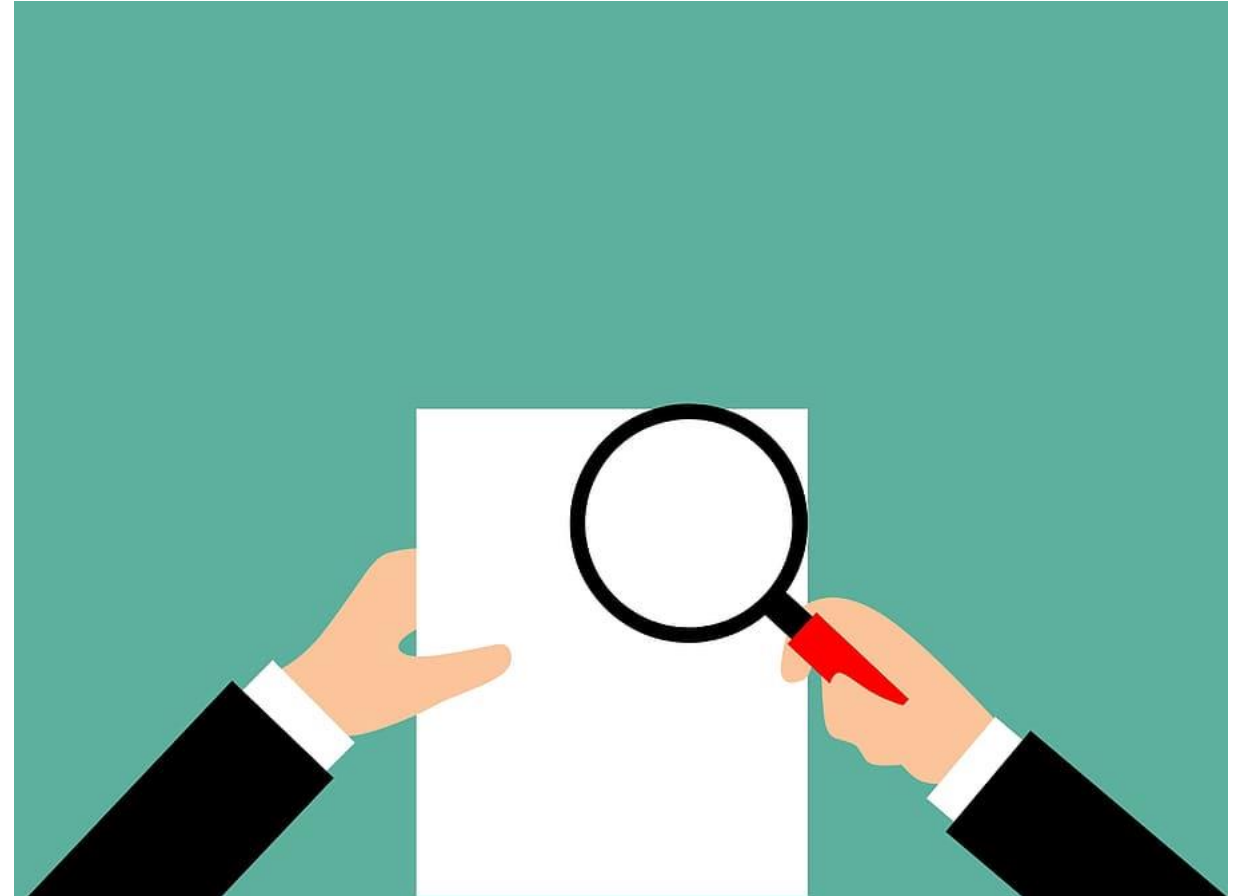
Los sprints están definidos bajo historias de usuario con requerimientos genéricos de sitios web. Su propósito es validar los conocimientos en los temas centrales del ciclo y se publicarán semanalmente.

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 requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.



Sprint Backlog:

Historia de usuario

Yo como administrador requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.

Actividades

1. Validar el campo nombre evitando que se ingresen caracteres numéricos, restringiendo la longitud entre 4 y 30 caracteres máximo y que el campo no se deje vacío.

Instrucciones:

- Crear la función de JS **checkNombre(valor)** en donde se verificarán las validaciones del campo nombre del usuario registrado.
- La función debe de retornar verdadero, si el usuario cumple con las políticas para el ingreso de su nombre en el registro, o falso en caso de no cumplir con las políticas.
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports = checkNombre;**

Sprint Backlog:

Historia de usuario

Yo como administrador requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.

Actividades

2. Validar que campo género se encuentre seleccionado, es decir, el usuario no podrá proseguir con el registro, si no escoge alguna de las opciones, dado que, es un campo requerido.

Sprint Backlog:

Historia de usuario

Yo como administrador requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.

Actividades

3. Validar el campo teléfono, este campo debe tener una longitud de 7 dígitos únicamente, solo puede contener caracteres numéricos y no puede dejarse vacío.

Instrucciones:

- Implementar la función de JS **checkTelefono(valor)**, en la cual, se verificarán las validaciones del campo.
- La función debe de retornar verdadero, si el usuario cumple con las políticas o falso en caso de no cumplirlas.
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports = checkTelefono;**

Sprint Backlog:

Historia de usuario

Yo como administrador requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.

Actividades

4. Validación del campo dirección. Este campo debe de contener caracteres alfanuméricos y únicamente los caracteres especiales de # y -. Así mismo, el campo deberá de contar con una longitud no mayor a 50 caracteres y al ser un campo requerido no puede quedar vacío.

Instrucciones:

- Implementar la función de JS **checkDir(valor)**, en la cual, se verificarán las validaciones del campo.
- La función debe de retornar verdadero, si el usuario cumple con las políticas o falso en caso de no cumplirlas.
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports = checkDir;**

Sprint Backlog:

Historia de usuario

Yo como administrador requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.

Actividades

5. Validar el campo correo, este campo de presentar la estructura de un correo, es decir, debe verse de la siguiente manera:

[usuario@correo.com](#)

Por lo que debe incluir @ seguido del dominio del correo. Así mismo, el campo no puede estar vacío.

Instrucciones:

- Implementar la función de JS **checkCorreo(valor)**, en la cual, se verificarán las validaciones del campo correo.
- La función debe de retornar verdadero, si el usuario cumple con las políticas o falso en caso de no cumplirlas.
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports = checkCorreo;**

Sprint Backlog:

Historia de usuario

Yo como administrador requiero validar los campos que componen el formulario de registro, cumpliendo con las buenas prácticas de desarrollo y las reglas del negocio.

Actividades

6. Validación de la contraseña, en donde el usuario deba de ingresar por lo menos una letra mayúscula, una minúscula, un número y con una longitud mayor o igual a 8 dígitos. Este campo es requerido, por lo que, no se puede dejar vacío.

Instrucciones:

- Implementar la función de JS **checkContrasena(valor)**, en donde se valide el cumplimiento de las políticas para el campo contraseña.
- La función debe de retornar verdadero, si el usuario cumple con las políticas o falso en caso de no cumplirlas.
- Exporte la función empleando la siguiente sentencia de código:
 - **module.exports = checkContrasena;**

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 **form.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 1. De igual manera, nombre las funciones **exactamente igual** a como se nombran en el instructivo de cada actividad del sprint actual.



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