

# HACKATHON SEMANAL

## Módulo 3: Maquetación Web (Semana 3)

**LOGRO:** Creación y validación de formularios. Definición y agrupación de selectores y reglas básicas de CSS.

### I. Es hora de demostrar lo aprendido:

Demostrarás todo lo aprendido en este reto que se basará en las clases dictadas durante la semana.

### II. Insumos para resolver el Reto:

- Materiales de clase de la semana 1, 2 y 3.
- [https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5\\_lista\\_elementos#Formularios](https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos#Formularios)
- <https://www.figma.com/file/DmxA6mUgZJts6T8pJ01qww/Project-basic?node-id=0%3A1>

### III. Descripción del reto

El gran reto está formado por 5 pequeños retos.

1. El primer reto trata de resolver un caso y responder ciertas preguntas
2. El segundo reto trata de Implementar etiquetas de formulario y entender su funcionalidad.
3. El tercer reto trata sobre Validar formularios.
4. El cuarto reto es continuar con el proyecto integrador. En esta oportunidad analizaremos, estructuraremos y validaremos el formulario del proyecto.
5. El quinto reto se trata de definir reglas css del proyecto.

### IV. Pasos a seguir para resolver los retos:

- El docente indicará si este reto se resolverá de manera individual o grupal

## Primer reto

### TÍTULO: Maquetación Web

#### ¿Por qué validar formularios?

##### EL PROBLEMA E INSUMOS:

Los formularios son Interfaces (capa de comunicación entre el usuario y la computadora) que permiten enviar información del usuario al sistema. Pudiéndose enviar, por ejemplo, un nombre, un apellido, una fecha de nacimiento, un número de cuenta, etc.

Cuando se envían los datos, el sistema que los recibe espera los formatos correspondientes por cada tipo de dato. Si enviamos un nombre esperará un string, si enviamos una fecha esperará un formato de fecha, si enviamos un número esperará un número. El error se genera cuando enviamos información incorrecta en los campos, o sea, si en un campo que tienes que enviar un dato tipo string envías un número. También evitaremos las inyecciones SQL (SQL Injection es una falla en la codificación de una aplicación cualquiera (web o local) que posibilita por medio de un input cualquiera, la manipulación de una consulta SQL.).

##### PREGUNTAS:

- ¿Para qué sirve la validación de formularios?
- ¿Cuáles son los problemas al no validar un formulario?
- ¿Cuáles son los beneficios?

## Segundo reto:

- Ingresar al siguiente enlace -  
[https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5\\_lista\\_elementos#Formularios](https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos#Formularios)
- Crear rama **feature-tags-form** a partir de master
- Crear archivo **tags-form.html**
- Escribir etiquetas de la lista de elementos html5
- Agregar cambios al stage
- Crear versión y definir mensaje corto
- Subir cambios a repositorio remoto

### Tercer Reto:

- Crear archivo **form.html** en la rama **feature-tags-form**
- Crear formulario con los siguientes campos : **Nombre, Apellido, Edad, Sexo, DNI, Email**
- Los campos **Nombre, Apellido** solo aceptarán valores de **texto**
- El campo **Edad** solo soportara el rango de **18 a 80**
- El campo **Dni** solo aceptara **textos y números y como minimo tiene q tener 8 números**
- El campo **Email** solo soportara **correo**
- Agregar padding, border y también la pseudo clase focus
- Agregar cambios al stage
- Crear versión y definir mensaje corto
- Subir cambios a repositorio remoto
- Cambiarse a rama master, bajar, cambiar y combinar cambios

### Cuarto Reto:

- Ingresar al siguiente enlace -  
<https://www.figma.com/file/DmxA6mUgZJts6T8pJ0lqww/Project-basic?node-id=0%3A1>
- Crear rama **feature-html** a partir de master
- Dentro del archivo **index.html** crear la estructura html del formulario
- Agregar cambios al stage
- Crear versión y definir mensaje corto
- Subir cambios a repositorio remoto
- Cambiar a rama master y combinar cambios de rama feature-html

### Quinto Reto (BONUS):

- Ingresar al siguiente enlace  
<https://www.figma.com/file/DmxA6mUgZJts6T8pJ0lqww/Project-basic?node-id=0%3A1>
- Crear rama feature-css a partir de master
- Crear archivo reset.css dentro de la carpeta styles

- Crear archivo main.css dentro de la carpeta styles
- Crear reglas css y clases donde sea necesario
- Agregar cambios al stage
- Crear versión y definir mensaje corto
- Subir cambios a repositorio remoto

## **V. Solución del reto**

- Para que el reto esté cumplido al 100%, se deben haber respondido las preguntas planteadas y se deben haber resuelto los ejercicios

## **VI. Presentación del Reto**

- El documento debe ser presentado de manera individual o grupal (según se coordine con el docente)
- El tiempo de cada presentación lo definirá el docente a cargo

## **VII. Feedback**

- El docente dará feedback a los estudiantes sobre los ejercicios realizados