

# Taller: Integración de Funcionalidades Básicas de JavaScript en el Sitio Web del Hospital

## Contexto:

En este taller, los estudiantes deben continuar incorporando funcionalidades de **JavaScript básico** en el proyecto del hospital. Además de manipular el DOM, validar datos, y manejar errores, se pedirá la entrada de información del usuario a través de un **prompt**, mostrando los resultados por consola o como alertas en el navegador.

**Duración: 2 horas**

---

## Requisitos:

### 1. Manejo del Entorno de Ejecución y Consola (1.5 puntos)

- Configura y utiliza correctamente el **entorno de ejecución de JavaScript** en el proyecto del hospital.
  - Implementa el uso de **console.log()** en varias partes del proyecto para depuración y mensajes informativos.
  - Utiliza **import** o **require** si es necesario para la estructura del programa JavaScript.
  - Explica en el archivo **README** cómo funciona el **event loop** (stack, heap, queue) en JavaScript.

### 2. Manejo de Variables, Operadores y Prompt (2 puntos)

- Solicita información al usuario mediante **prompts** (nombre, email, teléfono).
  - Los datos deben mostrarse tanto en la **consola** como en una **alert** del navegador para darle retroalimentación al usuario.
  - Usa variables (**let**, **const**) para almacenar la información y explica el **scope** de estas variables en el README.
  - Utiliza operadores (**==**, **===**, **typeof**) para validar los datos ingresados por el usuario. Por ejemplo, verifica que el campo de email contenga un "@".

### 3. Ciclos e Iteraciones: Uso en la Interfaz (1.5 puntos)

- Implementa un ciclo o iteración (for, while, forEach) que manipule el **DOM** de forma dinámica.
  - Ejemplo: muestra dinámicamente una lista de **servicios médicos** o **doctores** en la página de inicio o en la página de equipo médico, utilizando datos estáticos.
  - Utiliza **condicionales** dentro del ciclo para mostrar solo elementos específicos, como doctores con más de 5 años de experiencia o servicios médicos disponibles.

### 4. Manejo de Errores con try/catch y Debugging (2 puntos)

- Añade una funcionalidad que use un bloque **try/catch** para manejar errores.
    - Ejemplo: manejo de errores al ingresar un teléfono con un formato incorrecto o al intentar acceder a una variable que no existe.
  - Utiliza el **debugger** para rastrear posibles errores dentro del código y explica su uso en el **README**.
- 

### Herramientas a Utilizar:

- **JavaScript** para crear las funcionalidades solicitadas.
  - **README**: Actualiza el archivo **README** con:
    - Descripción de la integración de **prompts** y cómo se validan los datos ingresados por el usuario.
    - Explicación del **scope** de variables y el uso de operadores.
    - Instrucciones sobre cómo usar el **debugger** y el manejo de errores con **try/catch**.
- 

### Entrega:

- **Formato de entrega**:
  - Opción 1: Enviar un **enlace al repositorio de GitHub** con el proyecto actualizado.
  - Opción 2: Entregar un archivo **ZIP comprimido** con el proyecto completo (HTML, CSS, JS, README, etc.).