



**Programación Visual**  
Analista Programador Universitario  
Facultad de Ingeniería

Trabajo Práctico N° 1  
HTML – CSS - Git -GitHub  
Semana 1 – Semana 2

### ***Trabajo Práctico grupal de presentación obligatoria.***

#### **Parte 1: HTML (HyperText Markup Language) – Semana 1**

El objetivo de este trabajo práctico es que el grupo de estudiantes logre diseñar y maquetar la estructura base de un conjunto de páginas web haciendo uso de etiquetas básicas, etiquetas semánticas y el uso de atributos. Está prohibido el uso de CSS o estilos en línea. Es necesario el uso de Git y repositorio Github para implementar el versionamiento del código.

**Para iniciar el Trabajo Práctico N° 1, es necesario lo siguiente:**

1. Haber instalado Visual Studio Code. Descargar e instalar desde [Visual Studio Code](#). Revisar con el docente las extensiones necesarias para trabajar con html y css.
2. Haber instalado Git. Descargar e instalar desde [Git](#).
3. Contar con una cuenta en GitHub, si no posee una, registrarse en [GitHub](#).
4. Crear un repositorio en GitHub y clonarlo en la computadora. El nombre del repositorio: **pv\_tp1\_grupo999**, donde 999 es el número de grupo.

#### **Uso de GitHub:**

Configurar el entorno colaborativo en el repositorio de Github, se deben visualizar los commits con una descripción breve de lo que se está subiendo y con identificación clara del usuario que realiza el commit.

#### **Uso de Visual Studio Code**

Crear el **Layout base** de una plataforma de "Gestión de Proyectos Educativos". El sitio debe ser estático por el momento (HTML/CSS).

Dentro de la carpeta raíz crear los siguientes archivos: **index.html, proyectos.html, detalle.html y perfil.html**.

Uso de Semántica: Implementar correctamente etiquetas de bloque (<main>, <nav>, <header>, <footer>) y de contenido (<article>, <section>, <h1>-<h6>, <ul>, <table>, <form>).

Navegación: Los archivos deben estar vinculados entre sí mediante etiquetas <a>.

**1. Dashboard Principal (index.html):** Página de inicio donde se muestra información relacionada con el sitio, parte de la información que se muestra:

Encabezado Principal (<h1>): Nombre de la plataforma y bienvenida al usuario.

Sección de Estadísticas: Al menos 3 bloques de datos (ej: Proyectos totales, tareas para hoy, mensajes pendientes).

Sección de Novedades: Una lista de las últimas 3 o 4 acciones realizadas en el sistema (ej: "María subió un archivo", "Se creó el proyecto X").

## **2. Explorador de Proyectos (proyectos.html Listado)**

Barra de Filtros: Un <aside> o una sección superior con selectores para filtrar por Categoría, año, o estado (En curso/Finalizado).

Listado de Proyectos: Un contenedor de proyectos (visualizar 5). Cada proyecto debe mostrar título, imagen, descripción breve, categoría, enlace "Ver detalle".

## **3. Detalle de Proyecto (detalle.html Vista Individual)**

Encabezado de Proyecto: Título, fecha de inicio y autores.

Descripción Extendida: Un texto de al menos dos párrafos explicando los objetivos.

Lista de Recursos: Una lista (<ul>) con enlaces a documentos (PDF, Drive, GitHub).

Sección de Equipo: Nombres de los integrantes y sus roles dentro de ese proyecto.

## **4. Perfil de Usuario (perfil.html)**

Información Personal: imagen, nombre, rol (Docente/Alumno).

### **Importante:**

Evitar el uso de "Lorem Ipsum" (solo los docentes pueden usarlo): El contenido debe ser coherente con la temática educativa.

### **Validación:**

Acceder al siguiente sitio para realizar la validación del html creado, copiar y pegar en el sitio todo el html.

**<http://validator.w3.org/>**

## Entregable de Parte 1 (HTML)

- Enlace al repositorio público (github).
  - Archivo **README.md** con los nombres de los integrantes y su respectivo usuario de Github. También una breve descripción de la construcción de las páginas html.
- 

## Parte 2: Aplicación de estilos en HTML con CSS (Cascading Style Sheets) – Semana 2

Continuando con la construcción de las páginas html, proceda a aplicar estilos a cada uno de los documentos utilizando las diferentes formas para aplicar estilos.

- **Estilo en línea (*Inline*):** utilizando el atributo `style`.
- **Estilo interno (*Internal*):** usando un elemento `"style"`.
- **Estilo externo (*External*):** empleando un archivo CSS (*Cascading Style Sheets*) externo al documento HTML.

Dentro de la carpeta raíz crear una carpeta con nombre **css**, en esta carpeta crear un archivo css con el nombre `styles.css`.

Implementar CSS Grid para la estructura general y Flexbox para componentes internos (nav, cards, forms).

Consistencia Visual: Aplicar las variables CSS (`:root`) para colores y tipografías.

Interactividad CSS: Definir estados `:hover`, `:focus` y `:active` para todos los elementos interactivos.

Utilizar selectores CSS para aplicar reglas de estilo utilizando clases (`class`) e identificadores (`id`) para los elementos html.

Asociar el archivo CSS creado (`styles.css`) con el archivo HTML utilizando la etiqueta `<link>` dentro de la sección `<head>` del archivo `index.html` y los demás archivos html.

*Ejemplo:*

```
<link rel="stylesheet" href="../css/styles.css">
```

## Entregable de Parte 2 (CSS)

- Enlace al repositorio público (github).

- Archivo README.md con los nombres de los integrantes y su respectivo usuario de Github. También una breve descripción de lo utilizado en las páginas (css).