

# Proyecto final Sprint I

## Actividad

Desarrollar un sitio web que conste de 5 secciones, cada sección deberá ser realizada por un estudiante. Los diferentes elementos del sitio web contarán con estilos personalizados creados por los estudiantes y deberán estar condensados en una sola hoja de estilos.

Para iniciar con el desarrollo del sitio web, los estudiantes deberán definir previamente aspectos como ancho del sitio (mínimo 1200px), la información a presentar en el sitio web, diseñar un mockup, borrador o bosquejo, también descargar, diseñar o crear los recursos a utilizar como imágenes, iconos, logos, entre otros y finalmente asignar responsable a cada sección.

El proyecto, deberá manejarse versionado en un repositorio remoto en github, en el que los diferentes miembros del equipo deben tener acceso, no se permite el push directo a las ramas Dev o Master.

## Secciones

1. Sección superior, cabecera o header: esta sección deberá contar con el menú de navegación a tres páginas internas del sitio web y su respectivo enlace al inicio. Debe contar con una imagen que ocupe el ancho del sitio.  
El menú debe de estar en un tag `<nav></nav>`.
2. Sección de servicios: aquí los estudiantes deberán dividir en ancho del sitio en 3 partes y en cada parte se deberá incluir un servicio. Cada servicio debe contar con una breve descripción, un logo, ícono o imagen que lo identifique y un enlace al detalle de los servicios. El elemento donde están los servicios debe de tener un `id="services"`
3. Sección de noticias: El estudiante encargado de la sección de noticias, deberá dividir el ancho del sitio web en 2 filas y dos columnas para presentar 4 noticias. Cada noticia debe contar con un resumen, una imagen y un enlace a leer la noticia completa.  
El elemento donde están los servicios debe de tener un `id="news"`
4. Sección de equipo: El estudiante deberá dividir el ancho del sitio entre el número de participantes del equipo e incluir junto con cada uno una foto o imagen, el nombre y algún dato como la institución educativa, edad, pasatiempo, entre otros.  
El elemento donde están los servicios debe de tener un `id="team"`
5. Sección Footer o pie del sitio: el estudiante deberá crear el pie del sitio web en donde se incluye información de contacto, deberá estar dividido en dos partes, en una se tendrá la información del sitio web como el motivante del desarrollo y el enlace al repositorio de github, y en la otra los contacto de los miembros del equipo con sus nombres y roles. La sección debe de estar en un tag `<footer></footer>`



## Implementación

1. Luego de contar con los recursos listos, imágenes, fotos, logos, iconos y textos procedemos a crear la carpeta del proyecto.
  - a. Clic derecho en la carpeta del curso -> Nueva carpeta, yo la voy a nombrar "workshop1". Ustedes la deben nombrar semana-1-#{#deGrupo}, la palabra semana seguido de un guión, seguido del número de la semana, seguido de un guión, seguido del número de su grupo. Esto también aplica para el nombre del repositorio remoto en github.
  - b. Descargar (**NO** clonar) el código fuente base desde el repositorio <https://github.com/Tecnalia-Cilco-3/semana-1>
    - i. Se descarga un .zip que deben descomprimir en la carpeta del curso o nuestro Work Directory.
  - c. Debe ir al editor de código, VS Code para el caso actual, luego clic en Archivo -> Abrir, y se selecciona la carpeta completa que se acaba de crear.

Ahora ya se cuenta con el espacio de trabajo para iniciar el desarrollo.

2. Debido a que el proyecto debe ser versionado, se debe iniciar una nueva instancia del git en el directorio.
  - a. Para ello desde VS Code, debe dar clic sobre Ver -> Terminal, luego verificar que la terminal o consola haya abierto en el directorio correcto, esto se hace con el comando \$: pwd, verificamos que la ruta que imprime en pantalla corresponde con la carpeta deseada.

```
(base) DdMAC:final usuarioutp$ pwd
/Users/usuarioutp/Desktop/MisionTic/material/sprint1/final
```

Si es correcto pasar al siguiente paso, si no se debe navegar hasta la carpeta deseada utilizando el comando \$: cd.

- b. Una vez ubicados en la carpeta del proyecto inicializar git con el comando \$: git init, se obtendrá un mensaje como el siguiente:

```
(base) DdMAC:final usuarioutp$ git init
Initialized empty Git repository in /Users/usuarioutp/Desktop/MisionTic/material/sprint1/final/.git/
```

Ya se ha iniciado un nuevo repositorio local de git.

3. Ahora se debe crear el repositorio remoto en github para sincronizar el proyecto.
  - a. Ir a el perfil de github, clic en el + -> 'Nuevo repositorio', debe ingresar los datos del formulario, marcar la opción "Public" y "add a README file".
  - b. Ahora deberá añadir a sus compañeros como colaboradores del proyecto, para ello es necesario el nombre de usuario. En su nuevo repositorio creado, 'workshop1' para este caso, clic en Setting -> Manage access -> invite a collaborator, e ingrese los nombres de usuario de cada uno.

Ahora ya cuenta con un repositorio remoto creado que cuenta actualmente con un único archivo README.

4. Con los repositorios creados e inicializados es necesario vincular repositorio local con repositorio remoto.

- a. Debe copiar la URL del nuevo repositorio, [workshop1](https://github.com/usuario/workshop1) para este caso y luego ir a la consola de VS Code, en el proyecto local. Para añadir el repositorio remoto debe ingresar el comando `$: git remote add workshop1 https://github.com/usuario/workshop1`. y verificar que se haya añadido correctamente con el comando `$: git remote -v`

```
(base) DdMAC:final usuarioutp$ git remote -v
workshop1      https://github.com/dadunque/workshop1 (fetch)
workshop1      https://github.com/dadunque/workshop1 (push)
```

Si todo está correcto, puede continuar al siguiente paso, en caso contrario revisar el error y probar nuevamente.

- b. Ya se ha vinculado el repositorio remoto al local mediante el enlace del punto anterior, ahora es necesario sincronizarlos, lo primero es traer los cambios del repositorio remoto al local con el comando `$: git pull nombre_remoto rama_remota`

```
(base) DdMAC:final usuarioutp$ git pull workshop1 main
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/dadunque/workshop1
* branch                main                -> FETCH_HEAD
* [new branch]           main                -> workshop1/main
```

El nombre de la rama remota inicial puede visualizarse en el repositorio de github bajo la opción "Code".

- c. Ya se han descargado los cambios del repositorio remoto al local, ahora se probará la carga de los cambios locales al remoto. Para ello se deberá crear un nuevo archivo llamado index.html en el directorio del proyecto local, inicialmente vacío, luego desde la consola se verifica el estado con el comando `$: git status`

```
(base) DdMAC:final usuarioutp$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    index.html

nothing added to commit but untracked files present (use "git add" to track)
```

Se puede observar que el archivo index.html no ha sido agregado al paquete de archivos para enviar al repositorio remoto, dado que se ha creado recientemente o sus cambios no han sido añadidos al commit. Para añadir archivos se debe ingresar el comando `$: git add nombre_archivo ->` para archivos específicos o `$: git add .` para todos los archivos nuevos y/o modificados. Posteriormente se verifica nuevamente el estado con el comando `:$ git status`

```
(base) DdMAC:final usuarioutp$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   index.html
```

Ya se tienen los archivos añadidos, ahora solo resta hacer un commit para agregar una descripción general de los cambios, o de la nueva funcionalidad o de algún pendiente, para ello se usa el comando \$: `git commit -m "mensaje para el registro"`

```
(base) DdMAC:final usuarioutp$ git commit -m "just a index empty"
[master 47a4323] just a index empty
```

Finalmente se deben enviar los cambios locales al repositorio remoto, este será el proceso que se repetirá con el fin de mantener el repositorio remoto actualizado, el comando a utilizar es \$: `git push nombre_remoto rama_remota`

```
(base) DdMAC:final usuarioutp$ git push workshop1 master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 283.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/daduque/workshop1/pull/new/master
remote:
To https://github.com/daduque/workshop1
 * [new branch]      master -> master
```

Como se puede observar, se hizo un push a una rama llamada master ubicada en el repositorio remoto workshop1, y el remoto ha contestado que ha creado una nueva rama master a la cual se le aplicará un pull request con el main desde github.

**Importante:** para el proyecto actual se utilizará una rama de desarrollo la cual recibirá los cambios creados en los Issues, una rama master o main (como esta por defecto en github), que tendrá la versión final y estable del proyecto y una serie de ramas Issue que contarán con cada una de las nuevas funcionalidades. Cada rama de tarea o Issue debe ser revisada antes de ser unida con la rama de desarrollo. Todo el proceso deberá ser documentado en el archivo README, este archivo también debe contener los autores y los pasos para replicar el proyecto.