

# *Lenguajes de Marcado y Estilo Web*



## **Desafío 1 LME**

- Mockups: Definición y Uso
- GitHub y su Papel en el Desarrollo Web
- Integración de Mockups y GitHub en el Desarrollo Web

Catedrático: Ing. Edwin Bonilla.

### **INTEGRANTES**

**Carlos Mauricio López García**  
**Jennifer Michelle Gamboa Gutiérrez**  
**David Isaac Segura Sánchez**  
**Moisés Alexander Lara Guevara**

### **CARNET**

**LG151035**  
**GG202300**  
**SS231442**  
**LG220949**

## Introducción

Este documento informa acerca de los conceptos y la visualización de los mockups en pc, a través de nuestra investigación se describe la forma en la que se aplican los códigos para formar el diseño de nuestra página ya que es de suma importancia conocer estas herramientas que nos proporciona facilidades en el diseño web y a la hora de trabajar nuestros proyectos de desarrollo, más sin embargo dentro de la creación de páginas web podemos encontrar diferentes software y plataformas que nos ayudaran a realizar estos procesos y así aplicar funciones para el contenido que deseamos crear, además dicho documento describe el proceso de creación de este mostrando el resultado en imágenes referentes a este tema desarrollado, por otro lado, el uso de GitHub que nos permiten guardar nuestros proyectos de forma segura en la nube, crear modificaciones y además es una gran herramienta que nos facilita el control de versiones para trabajar cada uno de nuestros proyectos, acá te mostraremos como se realiza dicho proceso que permite la colaboración entre múltiples desarrolladores web.

## I. Mockups: Definición y Uso

### Importancia de los Mockups:

#### **Explique por qué los mockups son importantes en el proceso de desarrollo web.**

Son importantes en el proceso de desarrollo web porque nos sirve para transformar ideas en funcionalidades y ayuda al cliente a exteriorizar y comprender lo que necesita. Con ello ganamos una expectativa clara respecto a los detalles de desarrollo, como por ejemplo la ubicación de los botones y del contenido, o cómo las páginas y funcionalidades van a interactuar entre sí.

Al trabajar con un mock-up, el cliente expone ante el profesional qué es lo que pasa por su cabeza, y consigue comprender mejor el nivel de complejidad del trabajo, lo cual también facilita la comunicación y desarrollo del proyecto luego de la contratación.

#### **Mencione cómo los mockups ayudan a los equipos a alinear sus objetivos y expectativas.**

Los mockups se convierten en una herramienta esencial para ayudar a los equipos a alinear sus objetivos y expectativas en el desarrollo web y el lenguaje de marcado, y esto se debe a varios factores clave:

**Representación visual concreta:** Las maquetas proporcionan una representación visual concreta de cómo se verá y se comportará el sitio web. Esto elimina la ambigüedad y permite a todos los miembros del equipo, incluyendo diseñadores, desarrolladores, gerentes de proyectos y clientes, tener una comprensión clara de cómo será el resultado final.

**Definición de estructura y contenido:** Las maquetas permiten definir la estructura del sitio web, incluyendo la disposición de elementos como encabezados, navegación, secciones de contenido, formularios y más. Esto asegura que todos estén de acuerdo sobre la organización y la jerarquía de la información.

**Aclaración de la navegación:** A través de las maquetas, se puede definir la navegación y la arquitectura de información, lo que ayuda a establecer cómo los usuarios navegarán por el sitio. Esto es esencial para garantizar una experiencia de usuario intuitiva y eficiente.

## **Herramientas para Crear Mockups:**

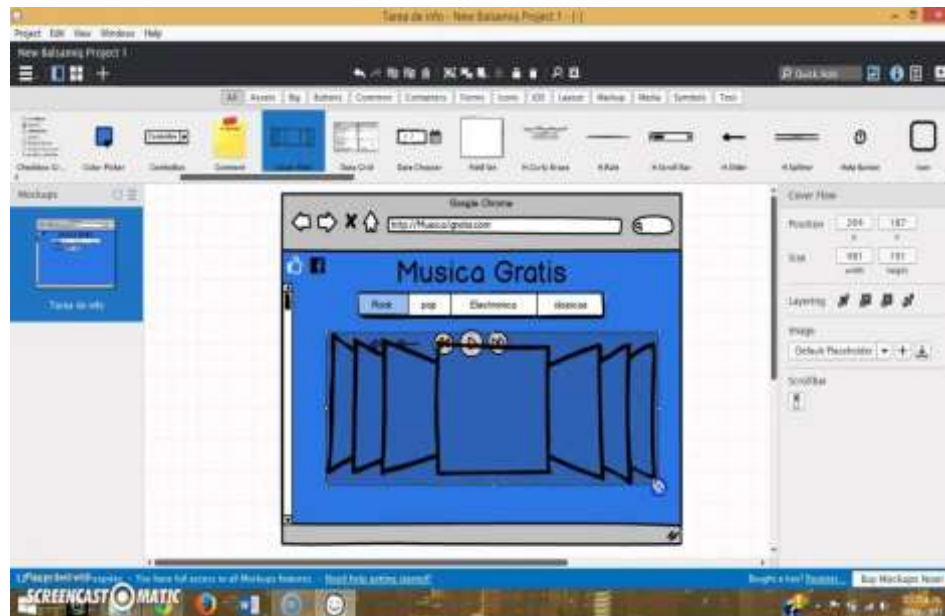


### **Enumere y describa algunas herramientas populares para crear mockups.**

1. Balsamiq: es una de las para crear mockups. Es fácil y sencillo de usar para tener un buen mockup preparado más rápido de lo que uno podría llegar a pensar a primeras. No se necesita interacciones con el prototipo es la elección perfecta.
2. Lucidhart: se caracteriza por una interfaz que opera al arrastrar y soltar los elementos para crear un mockup. Se puede compartir el proyecto con otros y en un poco de tiempo puede pasar de tener un proyecto rápido a uno que sería el producto final.
3. MarvelApp: Se sincroniza con Dropbox y ofrece equipos completos. Otra de sus características distintivas es que se pueden exportar el prototipo como si fuera un apk para desarrolladores Android.
4. Moqups: Esta herramienta está bajo HTML5 y lo convierte en una herramienta rápida y sencilla. A destacar de miembros en el proyecto es limitada
5. Uxpin: Una gran herramienta con un montón de opciones para wireframing y que entre las cuales destaca la posibilidad convertir un boceto en un wireframe digital.
6. Protoshare: Es una herramienta profesional que permite discutir todos los proyectos en tiempo real con colaboradores. Con una gran cantidad de opciones y al ser una herramienta bien clara en la interfaz le obliga a uno casi a tener que probarla.
7. Proto io: tiene un diseño simple y esto mismo es su gran valía. Una herramienta de gran valor del prototipado con numerosas opciones para Android y Windows.
8. Wirify: ofrece un script que convierte un sitio web en un wireframe. Es de gran ayuda cuando uno mismo se encuentra en una web que le llama mucho la atención y quiere usarla como base para su proyecto.
9. Pidoco: Se puede usar esta herramienta para probar los prototipos al enviar un enlace a los usuarios y así empezar a tomar los datos del mismo.

- ✚ Proporcione ejemplos de cómo se pueden utilizar estas herramientas para diseñar una página web.

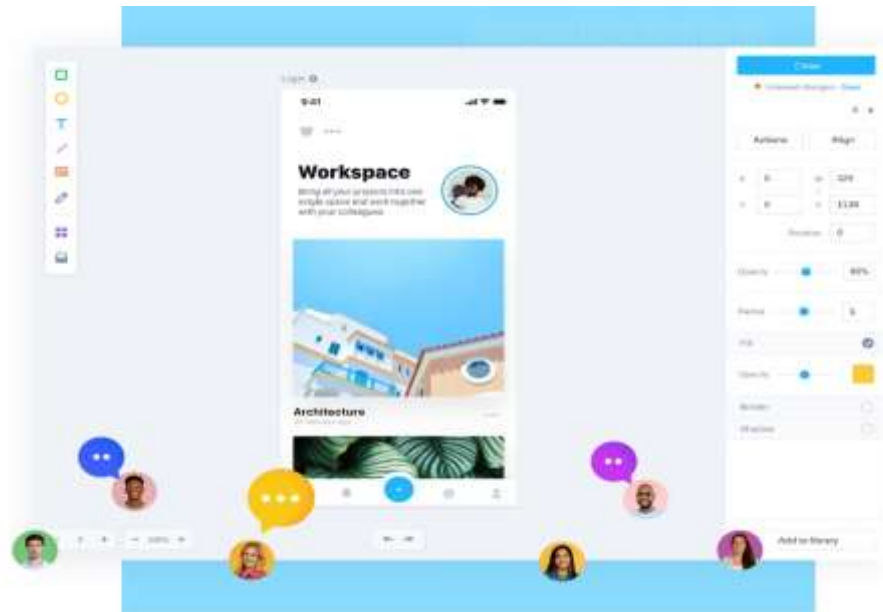
Baslamiq



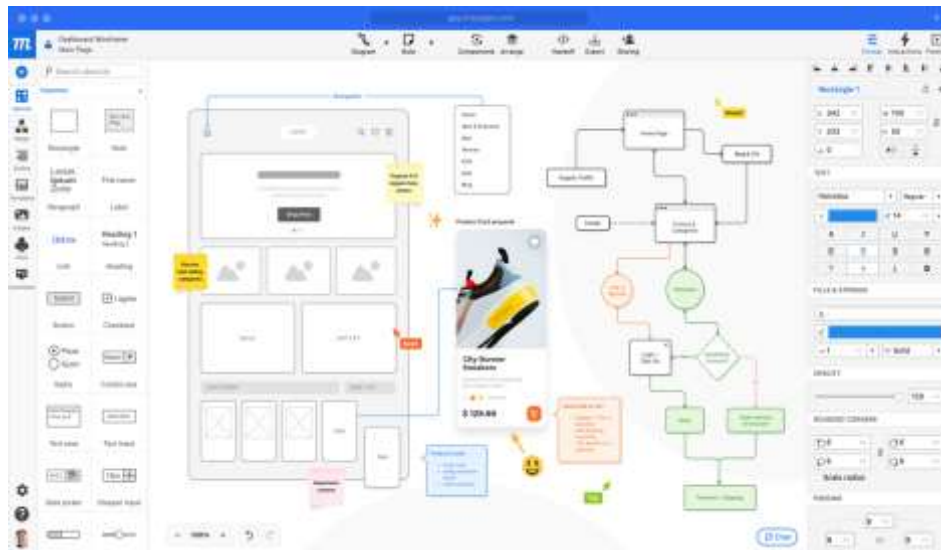
Lucidhart



## Marvelapp



## Mocqur



Uxpín

Destino / Hotel  
Barcelona

Entrada  
Jueves, 02/06/2016

Salida  
Jueves, 09/06/2016

Huéspedes  
2 habitación, 3 adultos, 1 niño

Buscar

Filtros

Nombre de hotel

Escribe el nombre del hotel

Precio total (para todos los huéspedes)

Desde Hasta

50 € 450 €

Tipo de alojamiento

☐ Hoteles (200)

☐ 1 Estrella (2)

☐ 2 Estrellas (8)

☐ 3 Estrellas (58)

☐ 4 Estrellas (11)

☐ 5 Estrellas (13)

☐ Apartamentos (21)

Régimen de alojamiento

☐ Solo habitación (8)

☐ Desayuno buffet (88)

☐ Media pensión (11)

☐ Pensión completa (13)

5 hoteles disponibles

Listado

Galería

Mapa

Ordenar por Precio Moneda Euro

Nombre hotel

★★★★

Rambla del Parc, s/n - Sabadell, ESPAÑA [Ver mapa](#)

Algunos servicios de este hotel

✓ WiFi

✓ Piscina

✓ Aparcamiento

[Ver ficha de hotel](#)

Desde

400,00€

estancia

Elegir habitación >

Nombre hotel

★★★★

Rambla del Parc, s/n - Sabadell, ESPAÑA [Ver mapa](#)

Algunos servicios de este hotel

✓ WiFi

✓ Piscina

✓ Aparcamiento

[Ver ficha de hotel](#)

Desde

400,00€

estancia

Elegir habitación >

Prohtoshare

View Cart | Order History

Products, Parts, Keyword Search

Search

Home | About Us | Products | Blog | News | Support | Contact

800-888-8888

Rainfall Sensors

Wind Anemometer

Weather Stations

Controllers

Indicators/Displays

Temp/Humidity Sensors

Barometric Pressure

Solar Radiation

Replacement Parts

Featured Products

Product Title

SKU

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

More Info >

Product Title

SKU

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

More Info >

Product Title

SKU

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

More Info >

Product Title

SKU

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

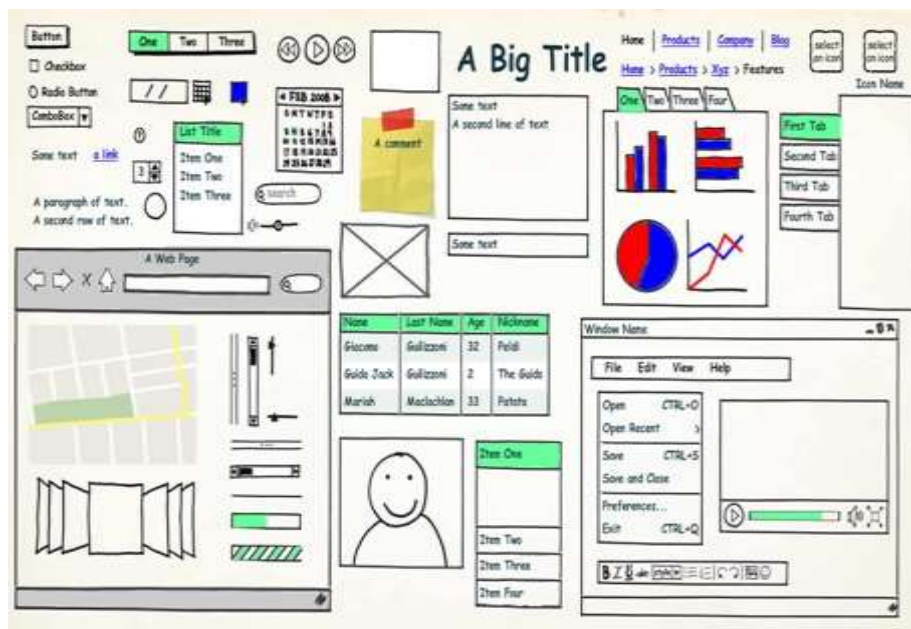
More Info >



Proto io

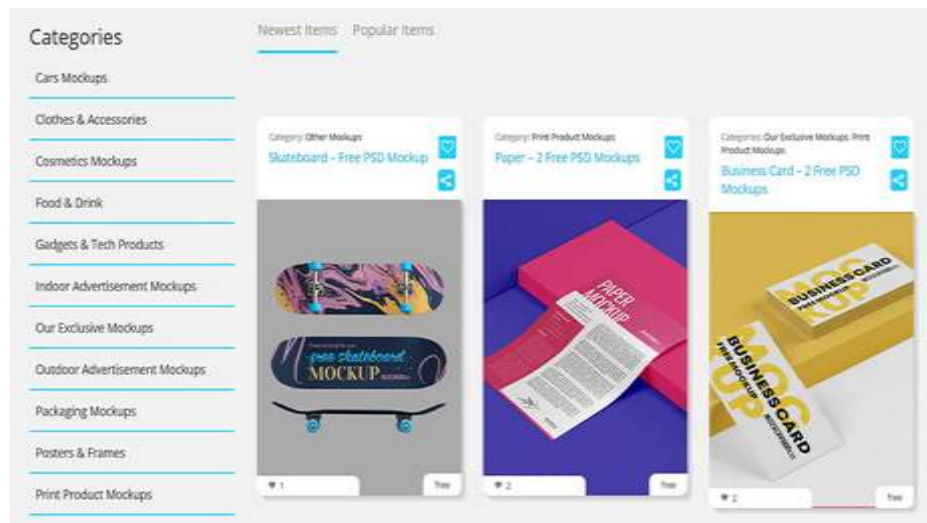


Wirify





Pidoco



### **Creación y Diseño de Mockups:**

✚ **Describe el proceso de creación de un mockup para una página web.**

Como primer paso se debe crear la arquitectura de la plataforma donde queden bien definidas las y categorías y páginas web. Esto se hará en base al contenido que se quiere comunicar al cliente.

Una vez definida la estructura, pasaras a establecer las partes comunes del mismo, es decir el pie de página, la barra lateral y cabecera.

Luego definirás cada uno de los modelos de la página que debes crear, pie de página, inicio, términos de condiciones, etc.

Como siguiente paso se construirán sus maquetas de contenido digital. Para ello, se evaluará en detalle el objetivo de cada una de ellas, es decir lo que busca transmitir, para luego definir así, su estructura final.

Como último paso para crear el mockup en el sitio web, debemos definir los colores del sitio, la tipografía la imagen principal, el logo de la empresa y todo lo que incluye el diseño final de la plataforma.

✚ **Incluya consejos sobre cómo representar la estructura, los elementos visuales y las interacciones en un mockup.**

Tener claridad de los objetivos del cliente, tomando en cuenta cada detalle.

Enlistar cada uno de los requerimientos y funcionalidades que tendrán lugar en la página web.


Estructurar de manera ordena y precisa cada parte de la página tomando en cuenta cada elemento detalladamente como lo son: el encabezado, la barra de navegación, la sección principal, las barras laterales y el pie de página.

Tener en cuenta la importancia de la función que cumple cada sección con respecto a la interacción que tendrá con el usuario final.

Crear los mockups lo más cercanos posibles al diseño final de la página web, para que la experiencia para el usuario y el cliente sea satisfactoria.

## II. GitHub y su Papel en el Desarrollo Web

### Control de Versiones con Git:

 **Explique qué es el control de versiones y cómo Git se utiliza para gestionar cambios en proyectos de desarrollo web.**

El control de versiones es un método para registrar los cambios en los proyectos. Mediante el control de versiones es posible hacer un seguimiento de las diferentes etapas de desarrollo y volver a versiones anteriores si es necesario. Así, el proceso hacia el código terminado no solo se ordena, sino que a menudo se reduce. De este modo también se facilita el trabajo en equipo.

La tarea principal del software de control de versiones (GitHub) es **registrar** los desarrollos en varios proyectos de programación. Tales proyectos incluyen el software de aplicación, así como el desarrollo web. Cualquier cambio en el código puede subirse a **la nube** y guardarse como una versión independiente mediante GitHub, que ya está convenientemente integrado en la mayoría de los entornos de desarrollo.

Así, un proyecto de GitHub contiene diferentes versiones del mismo código. Por un lado, esto te permite seguir el proceso de desarrollo de tus proyectos. Por otro lado, GitHub te permite **volver a versiones anteriores de tu código** en cualquier momento, lo que es especialmente ventajoso si se han colado errores en tu proyecto.

Además de poner a disposición diferentes versiones, el **trabajo en equipo eficaz** es una de las principales razones para utilizar GitHub en el desarrollo. Cada miembro del equipo puede descargar la versión actual del código como su propio repositorio. Al proceso de copiar un **repositorio** en tu propia cuenta de GitHub se le denomina **fork**.

Un repositorio está formado por distintas ramas, a las que también se conocen como **branches**. Es posible que el proyecto principal se encuentre en Branch Master. Al crear estas ramas, partes del código se separan y pueden desarrolladores individuales pueden trabajar en ellas de forma individual. De este modo, cada integrante del equipo puede ocuparse de una parte del código por separado.

GitHub proporciona el comando **merge** para fusionar de nuevo el código. El usuario propietario del proyecto original en GitHub recibe una solicitud. Si se adoptan los cambios, se produce la fusión del código.

### 🌈 Ilustre cómo se pueden realizar "commits" y "pull requests" en GitHub.

El commit es el elemento central de Git. Es la piedra angular que permite vincular todos los elementos y todos los conceptos de Git. Un commit representa un conjunto coherente de modificaciones.

A screenshot of a terminal window with a dark background. The title bar reads "1 GIT ejemplo práctico". The terminal shows a series of commands and their outputs, demonstrating basic Git workflow. The commands include cloning a repository, pulling updates, creating a new file, adding it to the index, committing it, and pushing it to the remote repository. The output shows the status of the repository at each step.

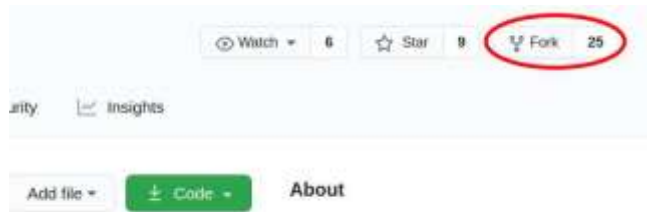
```
1 GIT ejemplo práctico
1 #!/bin/bash
2
3 # GIT ejemplo practico
4
5 git clone https://github.com/pabloba/ctpython.git
6
7
8 #Ver los cambios, si hubo
9 git pull ; git status
10
11 #Crear un archivo
12 touch archivoCreadoPorSUNOMBRE ; git status
13
14 #agregarlo al índice local de git
15 git add archivoCreadoPorSUNOMBRE ; git status
16
17 #Comitar al repositorio local
18 git commit -m "agregado de un archivo para ver que pase" ; git status
19
20 #Enviar los cambios al repositorio remoto
21 git push ; git status
```

Los pull requests son la forma de contribuir a un proyecto grupal o de código abierto.

¡Ejemplo de pull request!

- 1) Realicemos un fork del repositorio.

Realiza un fork del repositorio haciendo un clic en el botón fork de la parte superior de la página. Esto creará una instancia del repositorio completo en tu cuenta.



## 2) Clonar el repositorio

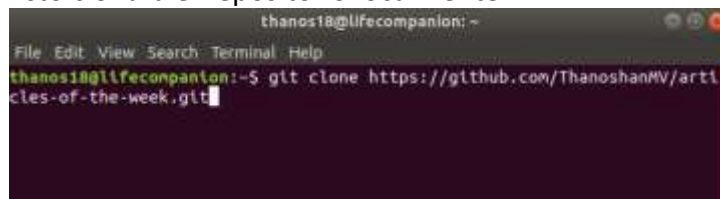
Una vez que el repositorio esté en la cuenta, clónalo a la computadora para trabajarlo localmente.



Abrir la terminal y ejecutar.

```
$ git clone [DIRECCIÓN HTTPS]
```

Esto clonará el repositorio localmente.



Ahora hemos configurado una copia de la rama maestra desde el repositorio principal del proyecto en línea.

Debemos ir al repositorio clonado ejecutando el siguiente comando:

```
$ cd [NOMBRE DEL REPOSITORIO]
```



## 3) Crea una rama

Es una buena práctica crear una rama (branch) nueva cuando trabajas con repositorios, ya sea que se trate de un proyecto pequeño o estés contribuyendo en un equipo de trabajo.

El nombre de la rama debe ser breve y debe reflejar el trabajo que estamos haciendo.

Ahora crea una rama usando el comando git checkout:

```
$ git checkout -b [Nombre de la Rama]
```

```
thanos18@lifecompanion:~/articles-of-the-week$ git checkout -b my-article
Switched to a new branch 'my-article'
thanos18@lifecompanion:~/articles-of-the-week$
```

4) Realiza cambios y confírmalos  
Has cambios esenciales al proyecto y guárdalos.

Luego ejecuta git status , y verás los cambios.

```
thanos18@lifecompanion:~/articles-of-the-week$ git status
On branch my-article
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   2020-January/W2/README.md

no changes added to commit (use "git add" and/or "git commit -a")
thanos18@lifecompanion:~/articles-of-the-week$
```

Agrega esos cambios a la rama recién creada usando el comando git add:

\$ git add .

```
thanos18@lifecompanion:~/articles-of-the-week$ git status
On branch my-article
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   2020-January/W2/README.md

no changes added to commit (use "git add" and/or "git commit -a")
thanos18@lifecompanion:~/articles-of-the-week$ git add .
thanos18@lifecompanion:~/articles-of-the-week$
```

Ahora confirma esos cambios utilizando el comando git commit:

\$ git commit -m "Adding an article to week 02 of articles of the week"

```
thanos18@lifecompanion:~/articles-of-the-week$ git commit -m "Adding an article to week 02 of articles of the week"
[my-article ae948c5] Adding an article to week 02 of articles of the week
1 file changed, 1 insertion(+), 1 deletion(-)
thanos18@lifecompanion:~/articles-of-the-week$
```

5) Envía los cambios a GitHub

Para enviar los cambios a GitHub, debemos identificar el nombre del repositorio remoto.

\$ git remote

```
thanos18@lifecompanion:~/articles-of-the-week$ git remote
origin
thanos18@lifecompanion:~/articles-of-the-week$
```

Para este repositorio el nombre es "origin".

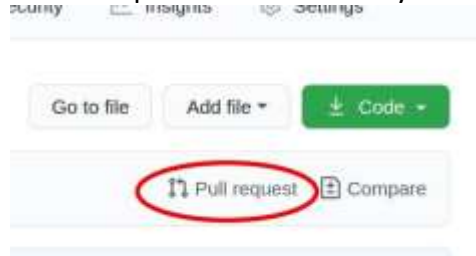
Luego de identificar el nombre podemos enviar en forma segura los cambios a GitHub.

```
$ git push origin [Nombre de la Rama]
```

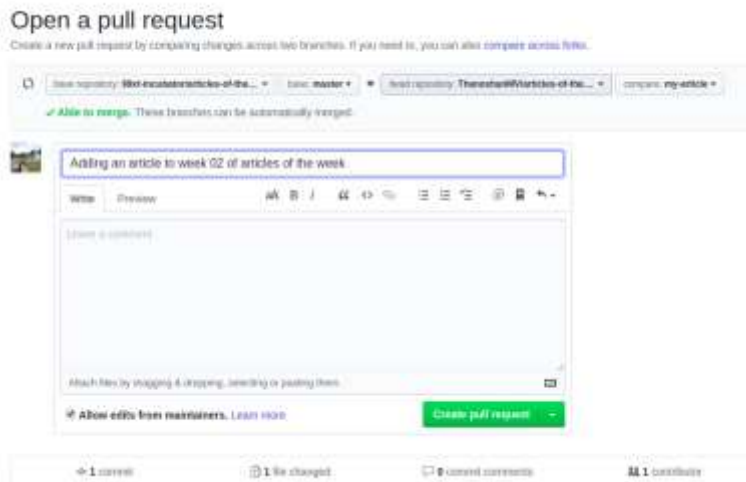
```
thanos18@lifecyclecompanion:~/articles-of-the-week$ git push origin my-article
Username for 'https://github.com': ThanoshanMV
Password for 'https://ThanoshanMV@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 542 bytes | 542.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'my-article' on GitHub by visiting:
remote:   https://github.com/ThanoshanMV/articles-of-the-week/pull/new/my-article
remote:
To https://github.com/ThanoshanMV/articles-of-the-week.git
 * [new branch]      my-article -> my-article
thanos18@lifecyclecompanion:~/articles-of-the-week$
```

## 6) Crea un pull request

Ve a tu repositorio en GitHub y verás un botón llamado "Pull request", has clic en él.



Provee todos los detalles necesarios de lo que has hecho (puedes referenciar problemas utilizando "#"). Ahora, envía el pull request.



Si tu pull request es aceptado recibirás un mail.

## Colaboración y Gestión de Proyectos:

- ✚ Describa cómo GitHub permite la colaboración entre múltiples desarrolladores en un proyecto web.

GitHub permite la colaboración entre múltiples usuarios dando acceso a los colaboradores para que pueda trabajar con el mismo repositorio que estemos desarrollando, los pasos son los siguientes:

Vamos al apartado de **Settings** y luego en **Collaborators**, luego aparecerá un campo de texto donde escribimos el nombre del usuario con el que se registró en GitHub nuestro colaborador, luego damos clic sobre el botón **Add Collaborators**.

Al colaborador recibirá un correo el cual tendrá que abrir con su git activado para que pueda ser parte del repositorio. Dando clic en **Accept invitation**. De esa manera los dos usuarios pueden trabajar en el mismo repositorio.

El colaborador tendrá que **clonar** el repositorio para trabajar en su máquina local, donde tendrá que crear una rama para realizar sus cambios, nos vamos al área de ramas y buscamos la opción **New Branch**, asignamos un nombre a la rama y finalmente en **Create Branch**. Ahora el colaborador está listo para realizar los cambios que considere oportuno, GitHub detectará el cambio luego realizamos un **commit**, en ese momento el cambio se habrá realizado en la máquina local. Para subirlo a la nube damos en la opción **Publish branch** para realizar un **push**, automáticamente se subirá la rama y el commit nuevo en el proyecto que está trabajando el colaborador.

El master o administrador tendrá la nueva rama a su disposición para decidir si fusionar los cambios haciendo un **merge** al proyecto.



## **Conclusión**

Gracias a la investigación desarrollada podemos comprender el uso esencial de los mockups en la realización de páginas web ya que con este podemos previsualizar el resultado que nos proyectamos al hacer nuestra página, vimos las diferentes opciones que puede utilizar el usuario al desarrollar su página web, en conjunto con los pasos que se proporcionan en nuestro documento, además un proyecto de GitHub una gran herramienta que debemos incluir para futuros proyectos de diseño y desarrollo que contiene diferentes versiones del mismo código. Por un lado, esto te permite seguir el proceso de desarrollo de tus proyectos, se puede definir la navegación y la arquitectura de información, lo que ayuda a establecer cómo los usuarios navegarán por el sitio. Esto es esencial para garantizar una experiencia de usuario intuitiva y eficiente.

## **Bibliografía**

- ✚ <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/github/#:~:text=El%20principio%20de%20GitHub%20%2D%20Una%20mirada%20r%C3%A1pida%20a%20Git,-Mientras%20que%20GitHub&text=El%20control%20de%20versiones%20es,versiones%20anteriores%20si%20es%20necesario.>
- ✚ <https://www.freecodecamp.org/espanol/news/como-hacer-tu-primer-pull-request-en-github/>

Trabajar en equipo en Github.com

- ✚ <https://www.youtube.com/watch?v=g-coH1XtqrQ>