



Proyecto: Recetas

Curso de Fundamentos

Introducción

Es hora de poner en práctica todo el conocimiento de HTML que has adquirido. En este proyecto, vas a crear un sitio web básico de recetas.

El sitio web constará de una página de índice principal que tendrá enlaces a algunas recetas. El sitio web no se verá muy bonito cuando haya terminado, pero es importante tener en cuenta que este proyecto es solo para desarrollar las herramientas HTML; volveremos a revisar este proyecto en el futuro para darle estilo con CSS.

Configurar el repositorio de GitHub de su proyecto

Como se mencionó en la [lección Introducción a Git](#), querrás organizar todos tus proyectos como un portafolio y vincularlos a GitHub para que otros puedan verlos.

Tenga cuidado al crear archivos en GitHub

GitHub nos permite realizar cambios directamente en su sitio. Si lo haces después de haber clonado el repositorio en tu máquina, hará que tu código local sea una versión posterior al remoto, lo que creará desafíos adicionales cuando envíes tu trabajo. La mayoría de las veces, deberías crear archivos localmente.

A medida que avances en el curso, aprenderás a manejar estas situaciones, pero por ahora, es importante seguir las instrucciones cuidadosamente para

mantenerte en el camino simple.

Si no sabe cómo configurar un repositorio, siga los pasos de instrucciones 1 y 2 que se encuentran en [Conceptos básicos de Git](#) para aprender cómo hacerlo, antes de continuar con estos pasos:

1. Cree un nuevo repositorio para este proyecto en GitHub.com, llámelo `odin-recipes` y elija la `public` opción en lugar de la privada predeterminada.
2. Clona ese repositorio en tu máquina local, dentro de la `repos` carpeta que creaste previamente en la lección Conceptos básicos de Git. El comando debería verse así `git clone git@github.com:username/odin-recipes.git` (use SSH).
3. Ahora, `cd` vaya al `odin-recipes` directorio del proyecto que se encuentra en su máquina local.
4. Prepara tu `README.md` expediente y escribe una breve introducción que describa en qué consiste el proyecto actual y qué habilidades habrás demostrado una vez que lo hayas completado. (También puedes hacer esto como una autorreflexión al final del proyecto, que es una buena manera de repasar lo que has aprendido).

Si tienes problemas:

- Todos los comandos de Git deben ejecutarse desde dentro de la carpeta de tu proyecto (¿olvidaste `cd` ingresar a la `odin-recipes` carpeta?).
- Asegúrese de haber seguido los pasos sobre cómo [crear una clave SSH](#) para clonar desde GitHub con SSH.
- Consulta el flujo de trabajo en la [lección Conceptos básicos de Git](#).

Consejos sobre cuándo comprometerse

¡No olvides todo lo que repasamos en la [lección Mensajes de confirmación](#) !

Cuando estás construyendo tu proyecto, probablemente terminarás haciendo varios ciclos `git add` + `git commit` antes de estar listo para subirlo a GitHub con `git push origin main`.

When writing code, it's considered best practice to commit early and often. Commit every time you have a meaningful change in the code. This will create a timeline of

your progress and show that your finished code didn't appear out of nowhere.

After you have entered `git push origin main`, switch over to your browser and open your repository on GitHub. You should now see all the files you just pushed.

Okay, that's enough Git for the moment – time to actually build stuff!

Assignment

Iteration 1: initial structure

1. Within the `odin-recipes` directory, create an `index.html` file.
2. Fill it out with the usual boilerplate HTML and add an `h1` heading "Odin Recipes" to the body.

Iteration 2: recipe page

1. Create a new directory within the `odin-recipes` directory and name it `recipes`.
2. Create a new HTML file within the `recipes` directory and name it after the recipe it will contain. For example `lasagna.html`. You can use the name of your favorite dish or, if you need some inspiration, you can find a recipe to use at [Allrecipes](#). Be sure to include the usual boilerplate HTML. This boilerplate code should be in every `.html` file you create.
3. For now, just include an `h1` heading with the recipe's name as its content.
4. Back in the `index.html` file, add a link to the recipe page you just created. Example: Under the `<h1>Odin Recipes</h1>` heading, write out the link like so: `Recipe Title`. The text of the link should again be the recipe name.
5. **Add a link back to the index page** on your recipe page for easier navigation. You can place this link at the top or bottom of your recipe page (e.g., `lasagna.html`). Here's an example:

```
1 | <a href="../index.html">Home</a>
```

This allows users to quickly return to the home page after viewing the recipe.

Iteration 3: recipe page content

Your new recipe page should have the following content:

1. An image of the finished dish under the h1 heading that you added earlier. You can find images of the dish on Google or [Allrecipes](#).
2. Under the image, it should have an appropriately sized "Description" heading followed by a paragraph or two describing the recipe.
3. Under the description, add an "Ingredients" heading followed by an **unordered list** of the ingredients needed for the recipe.
4. Finally, under the ingredients list, add a "Steps" heading followed by an **ordered list** of the steps needed for making the dish.

Iteration 4: add more recipes

1. Add two more recipes with identical page structures to the recipe page you've already created.
2. Don't forget to link to the new recipes on the index page. Also, consider putting all the links in an unordered list so they aren't all on one line.

Example:

```
1  <ul>
2    <li><a href="recipes/yourrecipe.html">Recipe Title 1<
3    <li><a href="recipes/yourrecipe.html">Recipe Title 2<
4    <li><a href="recipes/yourrecipe.html">Recipe Title 3<
5  </ul>
```

Your links won't be flashy, but for now, just focus on building them out.

Viewing your project on the web

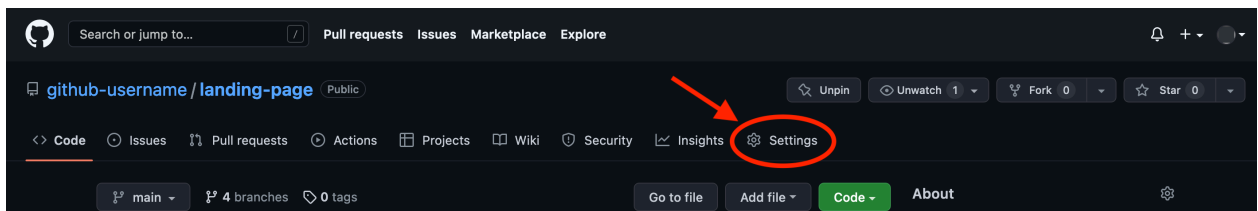
Si desea mostrar su trabajo (el proyecto) a otras personas o enviar una solución a continuación, deberá publicar su sitio para que otros puedan acceder a él desde la web, en lugar de hacerlo solo desde su máquina local. La buena noticia es que, si tiene su proyecto en GitHub (como se describe anteriormente), hacerlo es sencillo.

GitHub te permite publicar proyectos web directamente desde un repositorio de GitHub. Al hacerlo, podrás acceder a tu proyecto desde `your-github-username.github.io/your-github-repo-name`.

Se requiere una cuenta paga de GitHub para publicar proyectos web desde un repositorio privado. Las cuentas gratuitas solo pueden publicar desde repositorios públicos.

Hay un par de formas de hacer esto, pero la más sencilla es esta:

- Asegúrate de que el archivo HTML principal de tu proyecto se llame `index.html`. Si no es así, tendrás que cambiarle el nombre.
- Vaya a su repositorio de GitHub en la web y haga clic en el botón **Configuración** como se muestra en la captura de pantalla a continuación.



- Haga clic en **Páginas** en la barra lateral izquierda.
- Cambie la **rama** de *ninguna* a *rama principal* y haga clic en **Guardar**.
- Puede tomar unos minutos (el sitio web de GitHub dice hasta 10, pero hemos visto que puede tomar hasta una hora. No agregue un "tema" a su proyecto, o puede tener conflictos de Git, en cambio, sea paciente), pero su proyecto debería ser accesible a través de la web desde `your-github-username.github.io/your-github-repo-name` (obviamente sustituyendo sus propios detalles en el enlace).
- Si tu proyecto no se publica después de 1 hora, asegúrate de tener un archivo llamado `index.html` en la raíz de tu repositorio y de que todas las configuraciones se hayan establecido correctamente. Ve a tu repositorio en GitHub y haz clic en Acciones. Si no hay entradas, vuelve a las configuraciones, cambia la **Rama** de

rama principal a ninguna y haz clic en **Guardar** . Luego, cambia la **Rama** de *ninguna a rama principal* y haz clic en **Guardar** .

Cuando miras los proyectos presentados a continuación, es posible que te preguntes: "¿Por qué se ven tan bien? ¿Mi proyecto debería verse así?".

Aquí está la respuesta:

1. La razón principal por la que se ven tan bien es que no los hicieron principiantes. Probablemente los enviaron personas que tenían algo de experiencia en programación o que ya completaron el Proyecto Odin y están regresando para crear mejores sitios web.
2. Tu proyecto no debería verse así. Concéntrate principalmente en los requisitos del proyecto.

Esto también se aplica a proyectos futuros. Para obtener más información, lea ["Aprender a programar" en "Cómo convertirse en una historia de éxito TOP"](#) .

[Mejorar en GitHub](#)[Informar un problema](#)[Ver registro de cambios de la lección](#)

Tu solución

[Ver soluciones comunitarias](#)

Envíe su solución

[Ver curso](#)

[Marcar como completado](#)[→ Próxima lección](#)

¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únase a nosotros para ayudar a estudiantes de todo el mundo apoyando el Proyecto Odin!

[Más información](#)[Dona ahora](#)

THE ODIN PROJECT

Educación en codificación de alta calidad mantenida por una comunidad de código abierto.



Sobre nosotros

[Acerca de](#)[Equipo](#)[Blog](#)[Casos de éxito](#)

Apoyo

[Preguntas frecuentes](#)

Guías

[Guías de la comunidad](#)[Guías de instalación](#)

Legal

[Términos](#)[Privacidad](#)

[Contribuir](#)

[Contáctenos](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.