

# Solucionario

## Ejercicio Propuesto 1

Tenemos la siguiente estructura de carpetas:

```
alumno |--- apuntes      |--- modulo1.doc
        |--- modulo2.doc
        |--- modulo3.doc
        |--- documentos
        |--- ejercicios  |--- modulo1.html
                        |--- modulo2.html
                        |--- modulo3.html
                        |--- modulo4.html
                        |--- modulo5.html
```

1. Se pide crear la estructura de carpetas y archivos por medio de consola.
2. Se pide crear una carpeta por cada módulo, manteniendo la estructura de subcarpetas, de la siguiente forma:

```
alumno |--- modulo1      |--- apuntes      |--- modulo1.doc
        |--- documentos
        |--- ejercicios   |--- modulo1.html
|--- modulo2 |--- apuntes  |--- modulo2.doc
        |--- documentos
        |--- ejercicios   |--- modulo2.html
|--- modulo3 |--- apuntes  |--- modulo3.doc
        |--- documentos
        |--- ejercicios   |--- modulo3.html
|--- modulo4 |--- apuntes  |
        |--- documentos
        |--- ejercicios   |--- modulo4.html
|--- modulo5 |--- apuntes  |
        |--- documentos
        |--- ejercicios   |--- modulo5.html
                        |--- proyecto1
```

3. Se pide eliminar las carpetas obsoletas:

- alumno/apuntes.
- alumno/documentos.
- alumnos/ejercicios.

## Solución

1. Se pide crear la estructura de carpetas y archivos por medio de consola.

- **Paso 1:** creamos la estructura de las carpetas:

```
mkdir alumno
mkdir apuntes
mkdir documentos
mkdir ejercicios
touch modulo1.doc
touch modulo2.doc
touch modulo3.doc
touch modulo1.html
touch modulo2.html
touch modulo3.html
touch modulo4.html
touch modulo5.html
```

- **Paso 2:** movemos los directorios a su estructura correspondiente:

```
mv apuntes alumno
mv documentos alumno
mv ejercicios alumno
```

- **Paso 3:** movemos los archivos a su directorio correspondiente:

```
mv modulo1.doc alumno/apuntes
mv modulo2.doc alumno/apuntes
mv modulo3.doc alumno/apuntes
mv modulo1.html alumno/ejercicios
mv modulo2.html alumno/ejercicios
mv modulo3.html alumno/ejercicios
mv modulo4.html alumno/ejercicios
mv modulo5.html alumno/ejercicios
```

2. Se pide crear una carpeta por cada módulo, manteniendo la estructura de subcarpetas, de la siguiente forma:

- **Paso 1:** creamos las nuevas carpetas (hacemos lo mismo con el resto de los módulos):

```
mkdir modulo1
mkdir apuntes
mkdir documentos
mkdir ejercicios
```

- **Paso 2:** movemos los directorios a su estructura correspondiente (hacemos lo mismo con el resto de los módulos):

```
mv modulo1 alumno
mv apuntes modulo1
mv documentos modulo1
mv ejercicios modulo1
```

- **Paso 3:** movemos los archivos a su estructura correspondiente:

```
mv modulo1.doc alumno/modulo1/apuntes
mv modulo2.doc alumno/modulo2/apuntes
mv modulo3.doc alumno/modulo3/apuntes
mv modulo1.html alumno/modulo1/ejercicios
mv modulo2.html alumno/modulo2/ejercicios
mv modulo3.html alumno/modulo3/ejercicios
mv modulo4.html alumno/modulo4/ejercicios
mv modulo5.html alumno/modulo5/ejercicios
```

3. Se pide eliminar las carpetas obsoletas:

```
rm -r alumno/apuntes
rm -r alumno/documentos
rm -r alumnos/ejercicios
```

## Ejercicio Propuesto 2

Toma un proyecto anterior o incluso la estructura de carpetas que hemos creado en el ejercicio 1 y realiza los pasos para iniciar git y versionar los cambios de tu proyecto.

1. Inicializa git en la carpeta contenedora.
2. Agrega los archivos al stage.
3. Genera el primer commit.
4. Muestra el status.

## Solución

Tomaremos como ejemplo la carpeta con la estructura que generamos en el ejercicio 1.

- **Paso 1.** Vamos a la carpeta contenedora:

```
cd alumno
```

- **Paso 2.** Inicializa git en la carpeta contenedora.

```
git init
```

- **Paso 3.** Agrega los archivos al stage.

```
git add .
```

- **Paso 4.** Genera el primer commit.

```
git commit -m "initial commit"
```

- **Paso 5.** Muestra el status.

```
git status
```

## Ejercicio Propuesto 3

Github, desde agosto de 2020 nos permite crear un perfil personalizado a través de un archivo readme.md, en un repositorio con tu nombre de usuario, y es lo que haremos en el siguiente ejercicio.

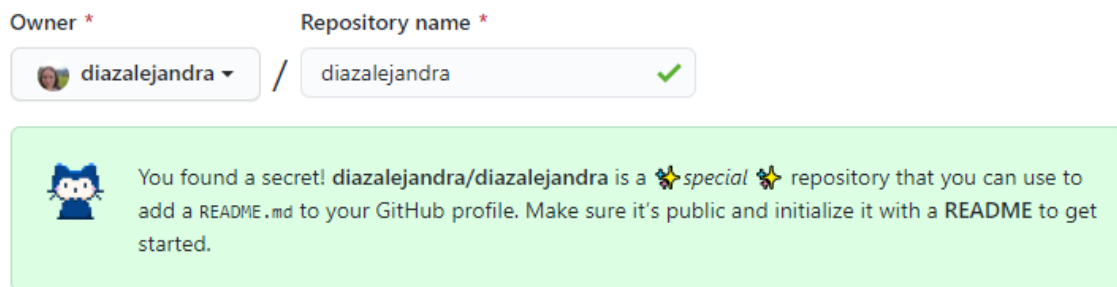
Mira el siguiente [video](#), donde se explica cómo debes crear el repositorio y el archivo.

## Solución

- **Paso 1:** Crear un repositorio con el nombre de usuario correspondiente:

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)



Owner \*      Repository name \*

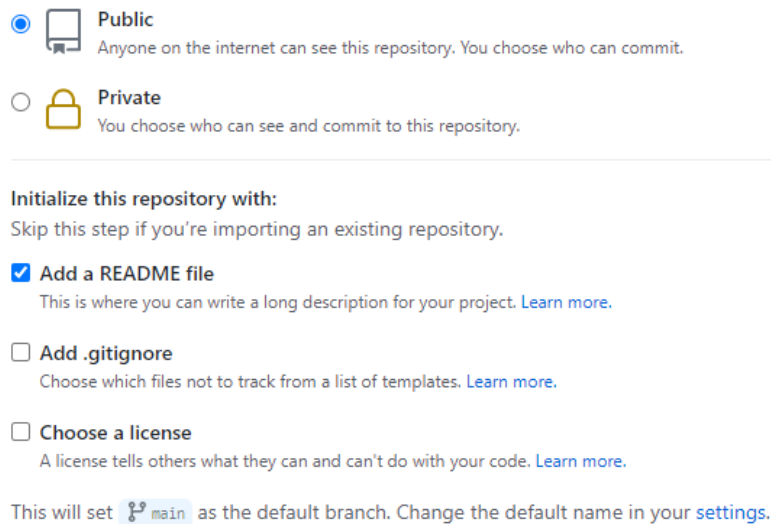
diazalejandra / diazalejandra ✓

You found a secret! diazalejandra/diazalejandra is a special repository that you can use to add a README.md to your GitHub profile. Make sure it's public and initialize it with a README to get started.

Imagen 1. Crear repositorio.

Fuente: Desafío Latam.

Seleccionamos el repositorio como público y generamos el archivo README.md.



☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Imagen 2. Configuración repositorio.  
Fuente: Desafío Latam.

- **Paso 2.** Podemos editar el archivo directamente en Github o llevarlo a un directorio local. Haremos lo segundo:

```
git clone [dirección del repositorio]
```

- **Paso 3.** Podemos editar el archivo README.md desde Atom y agregar los cambios globales en el repositorio (por los menos un cambio).

```
git commit -a -m "primer commit"
```

- **Paso 4.** Guardar los cambios en Github.

```
git push origin main
```