



---

# **INSTITUTO TECNOLÓGICO SUPERIOR P'URHÉPECHA**

DIVISIÓN DE ESTUDIOS PROFESIONALES  
DEPARTAMENTO DE INGENIERÍA BIOMÉDICA

## **Práctica 1- unidad 2 INFORMÁTICA MÉDICA**

### **“TOPICOS AVANZADOS DE BIOMEDICA”**

Presenta:

Pedro Ambrocio Juárez

ASIGNATURA:

Tecnologías de bases de datos

Profesor:

Mtro. Carlos Eduardo López Valencia.

CHERÁN, MICHOACÁN

---

---

---

# Índice

Capítulo 1 .....	1
Introducción.....	1
1.1    Objetivos.....	1
1.2    Justificación .....	2
Capítulo 2 .....	3
Marco Teórico .....	3
Capítulo 3 .....	5
Capítulo 4 .....	12
Conclusiones.....	12
Bibliografía.....	13



# Capítulo 1

## Introducción

En el ámbito de la informática y el desarrollo de software, el control de versiones es una herramienta fundamental que permite gestionar cambios en archivos y proyectos de manera eficiente. GitHub, una de las plataformas más utilizadas a nivel mundial, facilita esta tarea al ofrecer un entorno colaborativo para almacenar, compartir y controlar versiones de código y documentos. Esta práctica tuvo como objetivo principal familiarizarse con los conceptos básicos de GitHub, desde la creación de un repositorio hasta la gestión de archivos y el registro de cambios mediante commits. Además, se exploraron funcionalidades clave como la subida de archivos, la modificación de documentos y el análisis del historial de versiones, habilidades esenciales para cualquier profesional en el área de la informática.

### 1.1 Objetivos

#### Objetivo General:

- Crear y gestionar un repositorio en GitHub, aplicando los principios básicos del control de versiones para el desarrollo de proyectos colaborativos.

#### Objetivos Específicos:

- **Crear una cuenta en GitHub** y configurar un repositorio público para alojar archivos relacionados con la práctica.
  - **Aprender a realizar commits** para registrar cambios en los archivos del repositorio, acompañados de descripciones claras.
  - **Subir y modificar archivos** en el repositorio, incluyendo documentos de texto e imágenes.
  - **Explorar el historial de commits** para comprender cómo GitHub almacena y permite recuperar versiones anteriores de los archivos.
-

- 
- 
- **Generar un reporte final** que documente el proceso realizado y el enlace al repositorio creado.

## 1.2 Justificación

El uso de sistemas de control de versiones como GitHub se ha vuelto indispensable en el desarrollo de software y la gestión de proyectos digitales. Esta práctica es relevante porque:

- **Fomenta el trabajo colaborativo**, permitiendo que múltiples usuarios contribuyan a un mismo proyecto de manera organizada.
- **Proporciona seguridad y respaldo**, ya que GitHub guarda un historial de cambios, evitando la pérdida de información por errores humanos o fallos técnicos.
- **Desarrolla habilidades técnicas esenciales** para estudiantes y profesionales, como la gestión de repositorios, el uso de comandos básicos de Git y la documentación de proyectos.
- **Prepara para entornos laborales reales**, donde el control de versiones es una competencia básica requerida en equipos de desarrollo, investigación y gestión de datos.

Además, GitHub no solo se limita al ámbito de la programación; también es útil para diseñadores, escritores y cualquier profesional que necesite organizar y versionar archivos de manera eficiente.



# Capítulo 2

## Marco Teórico

### 1. Control de Versiones

Un sistema de control de versiones es una herramienta que registra los cambios realizados en archivos a lo largo del tiempo, permitiendo recuperar versiones específicas, comparar modificaciones y trabajar en equipo sin conflictos. Existen dos tipos principales:

- **Sistemas locales:** Los cambios se almacenan en una base de datos local (ejemplo: RCS).
- **Sistemas distribuidos:** Cada colaborador tiene una copia completa del repositorio y su historial (ejemplo: Git).

### 2. Git y GitHub

- **Git:** Creado en 2005 por Linus Torvalds, es un sistema de control de versiones distribuido, rápido y eficiente, ampliamente adoptado en la industria.
- **GitHub:** Plataforma en línea que aloja repositorios Git, añadiendo funciones colaborativas como issues, pull requests y herramientas de integración continua.

### 3. Términos Clave

- **Repositorio:** Directorio que almacena todos los archivos de un proyecto, junto con su historial de cambios.
  - **Commit:** "Instantánea" de los cambios realizados en un momento determinado, acompañada de un mensaje descriptivo.
-

- 
- 
- **Branch (rama):** Línea de desarrollo independiente, útil para trabajar en nuevas funciones sin afectar la versión principal.
  - **Pull Request:** Solicitud para fusionar cambios de una rama a otra, facilitando la revisión de código.

#### 4. Búsquedas Avanzadas en Internet

Como complemento, se exploraron técnicas para optimizar búsquedas en Google, como el uso de:

- **Comillas (")** para frases exactas.
- **Operadores** como site:, filetype:, intitle: y define:.
- **Funciones matemáticas y gráficas** directamente en la barra de búsqueda.

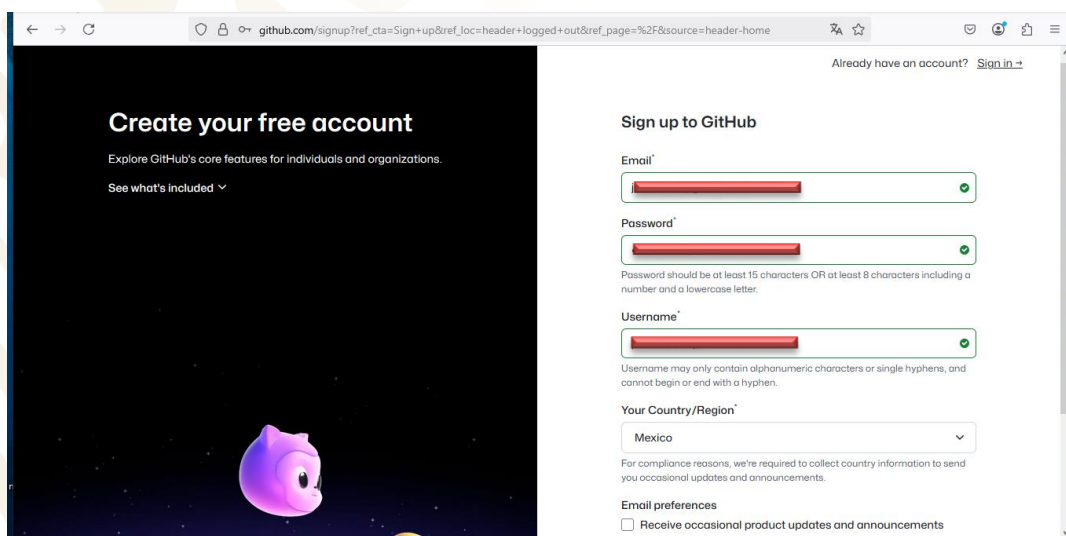
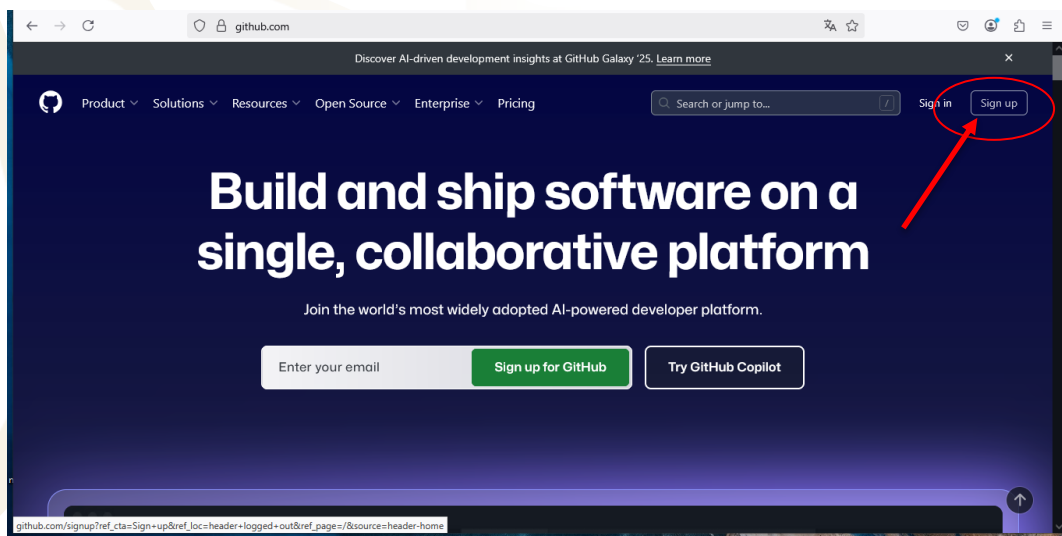


# Capítulo 3

## Desarrollo de la Práctica:

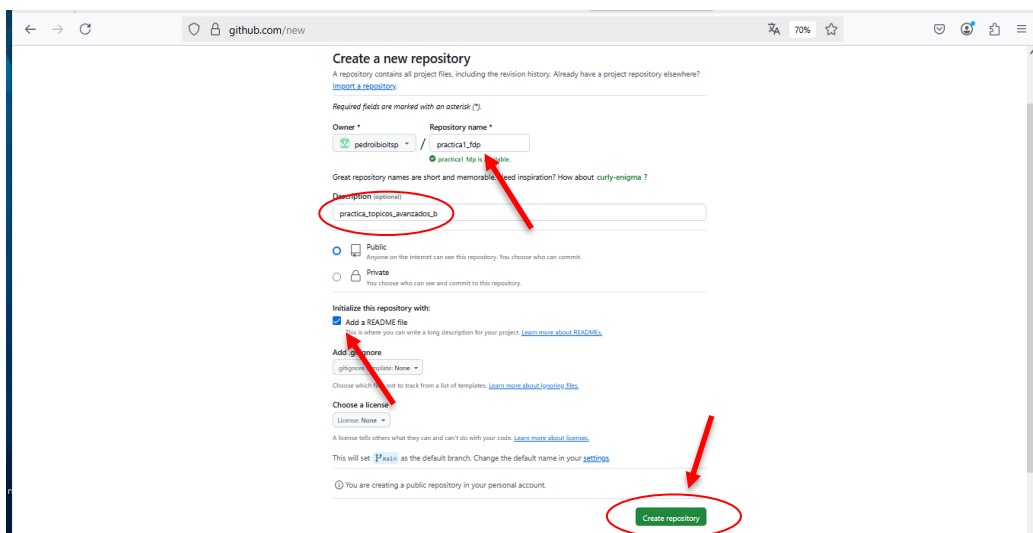
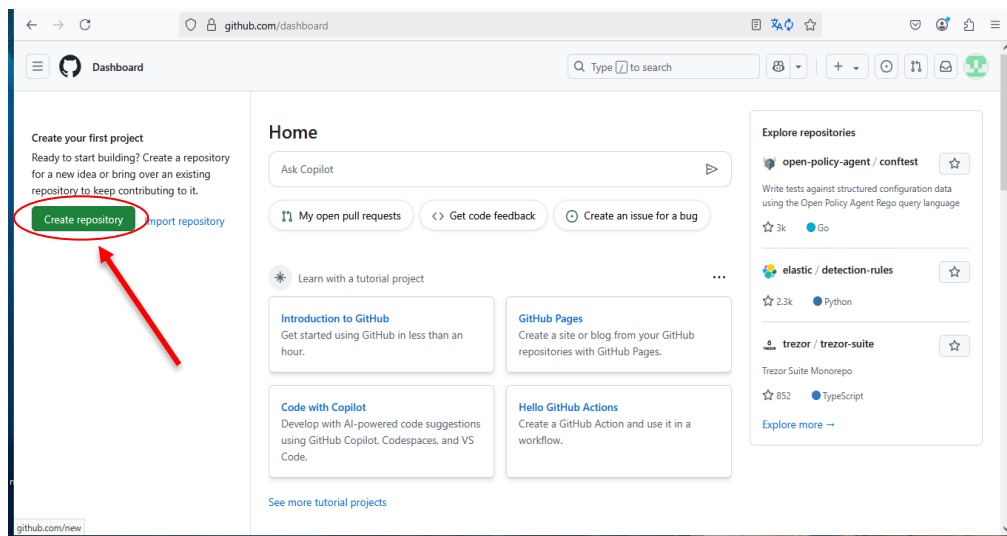
### 1. Creación de una cuenta en GitHub:

- Se accedió al sitio web <https://github.com> y se completó el proceso de registro proporcionando un nombre de usuario, correo electrónico y contraseña.
- Se seleccionó el plan gratuito y se verificó la cuenta mediante el correo electrónico recibido.



## 2. Creación del repositorio:

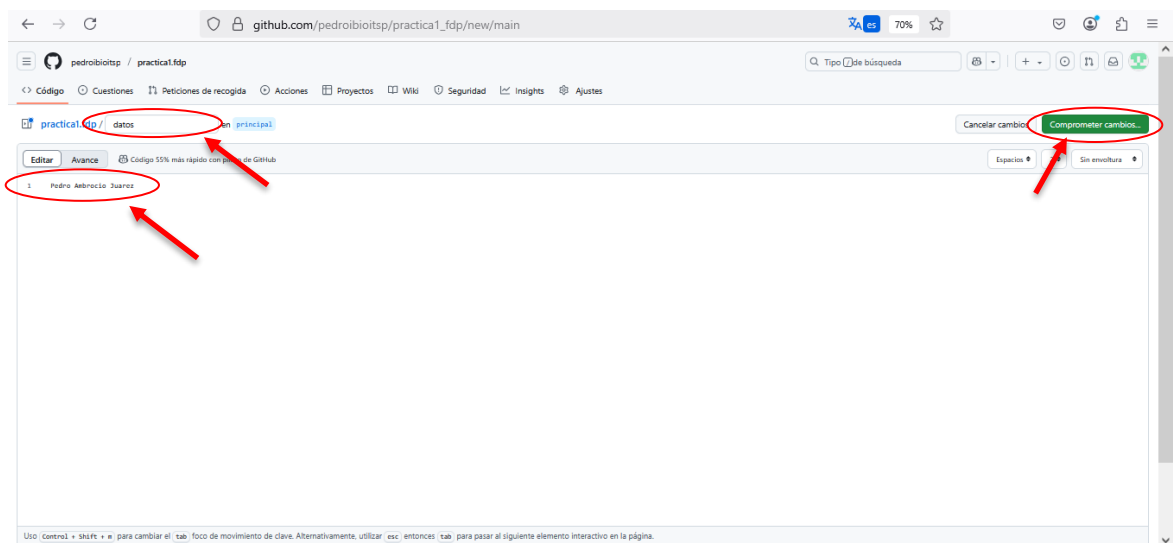
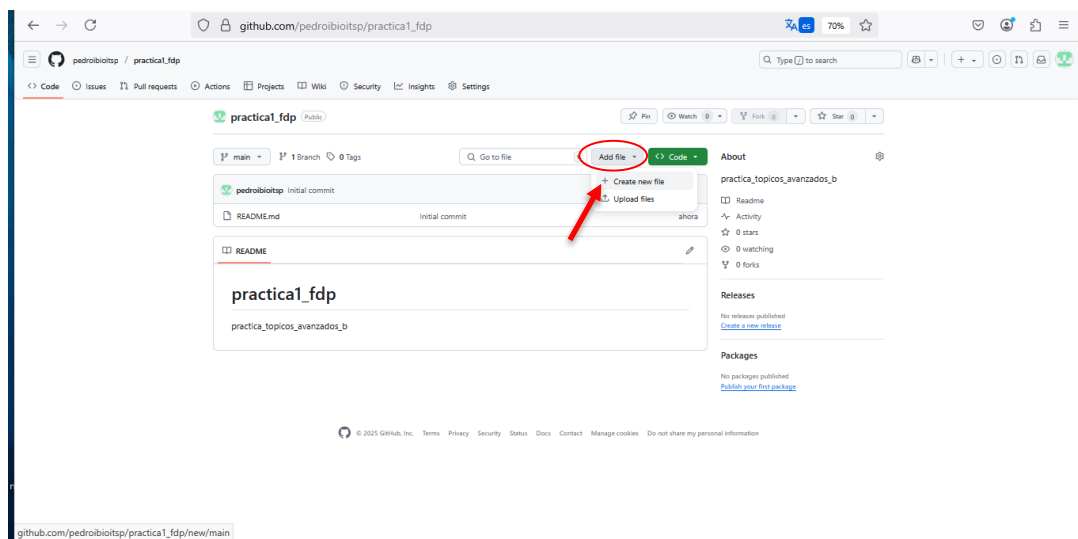
- Se hizo clic en "Start a Project" para crear un nuevo repositorio.
- Se asignó el nombre practical\_fdp al repositorio y se agregó una descripción: "Práctica\_topicos\_avanzados\_b".
- Se inicializó el repositorio con un archivo README.md para documentar el proyecto.

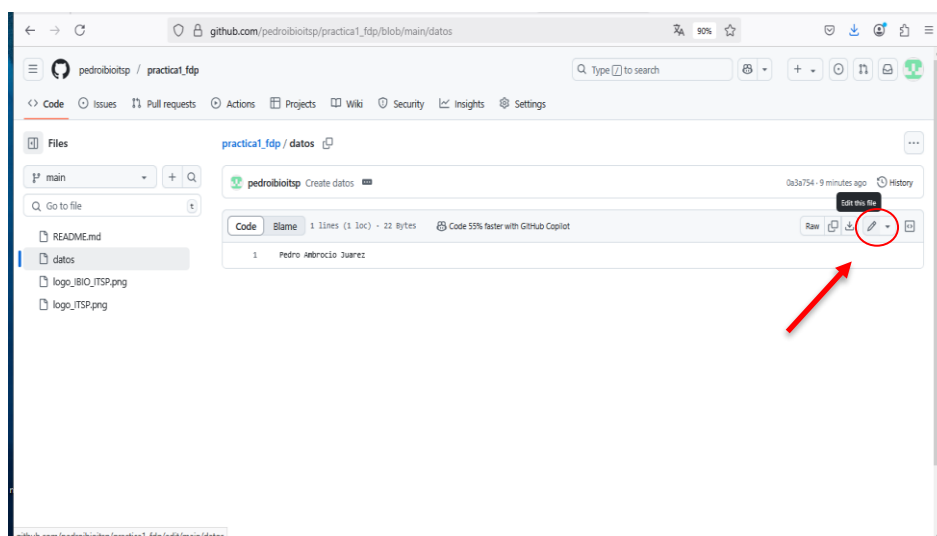
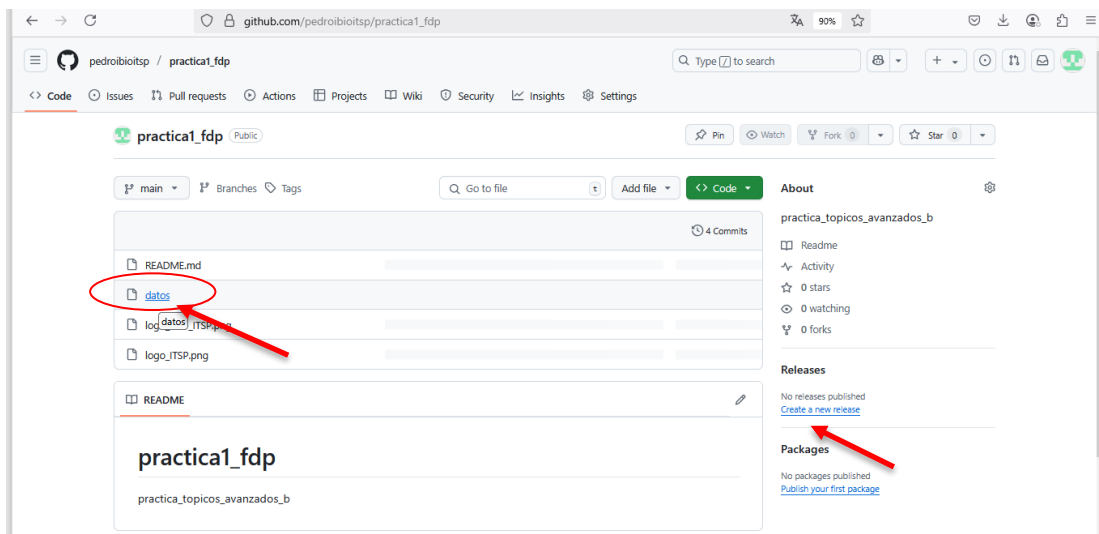
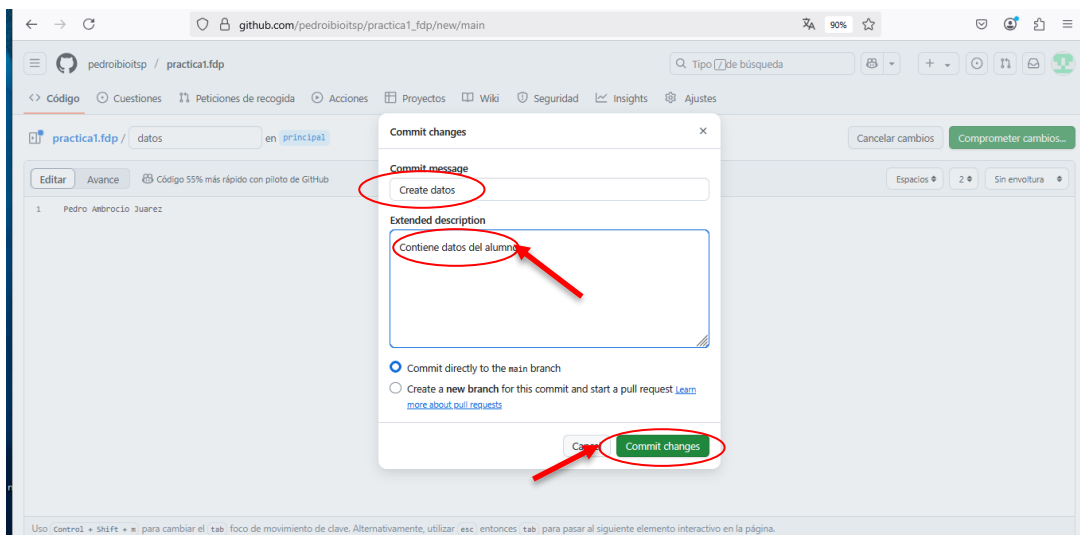


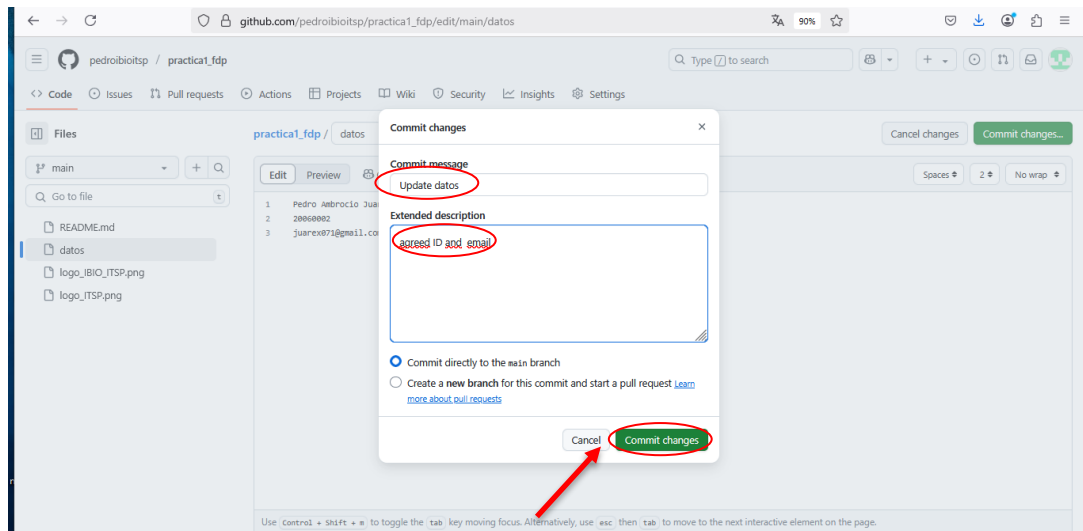
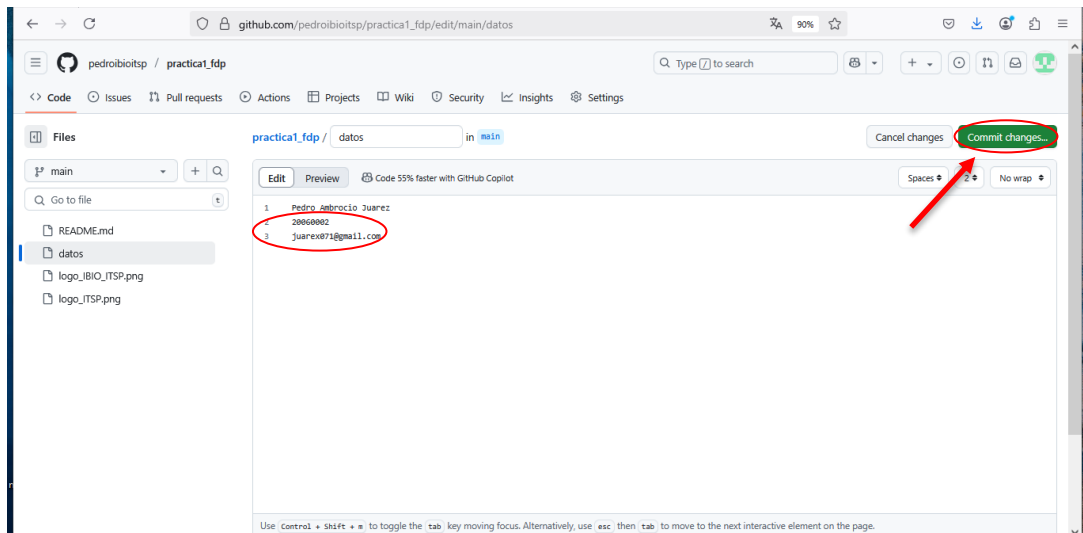


### 3. Creación y modificación de archivos:

- Se creó un archivo llamado Datos y se agregó el nombre del estudiante en la primera línea.
- Se realizó un commit con una breve descripción de los cambios realizados.
- Posteriormente, se modificó el archivo Datos para incluir el número de cuenta y el correo electrónico, registrando estos cambios con otro commit.

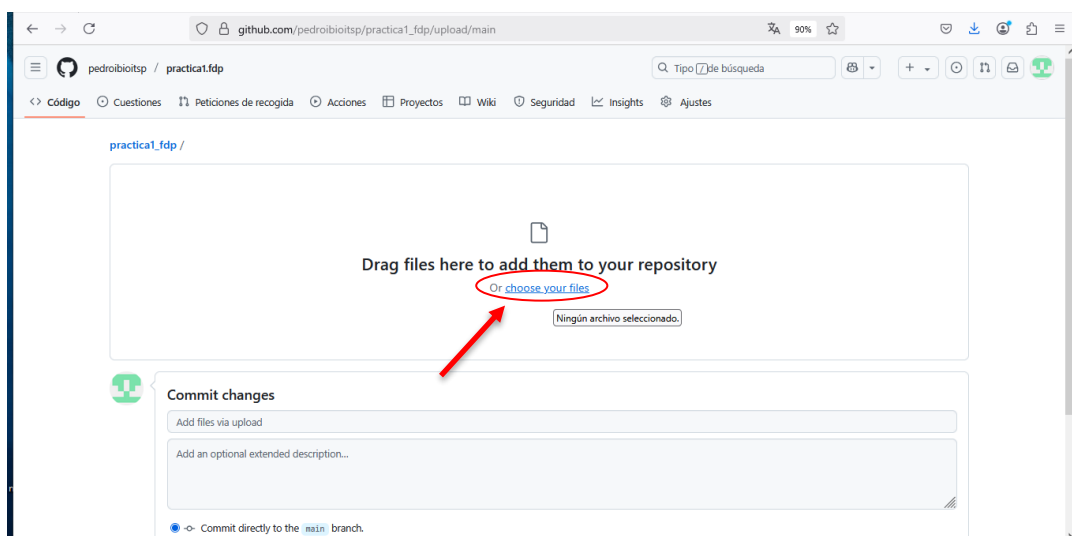
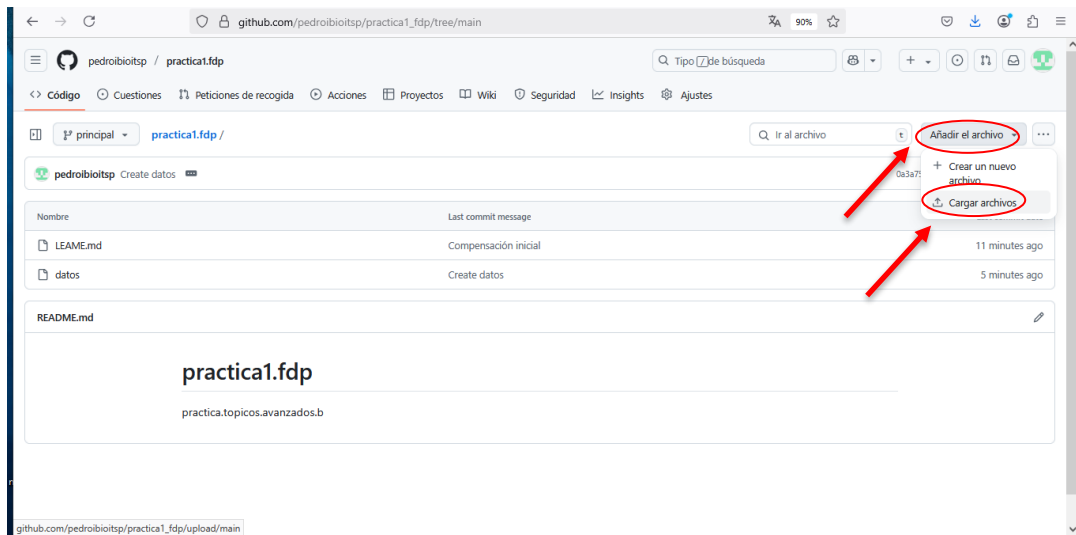


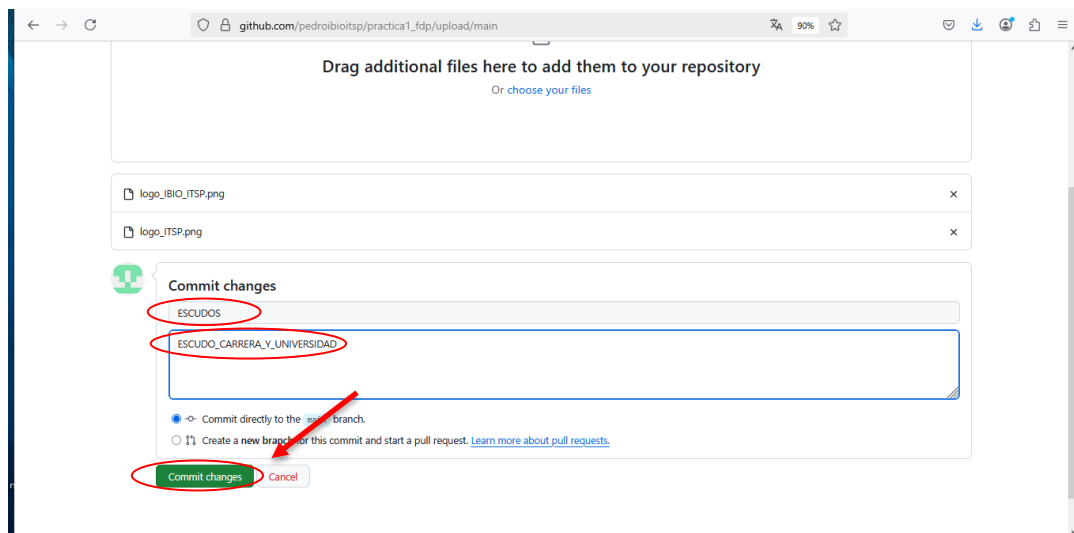
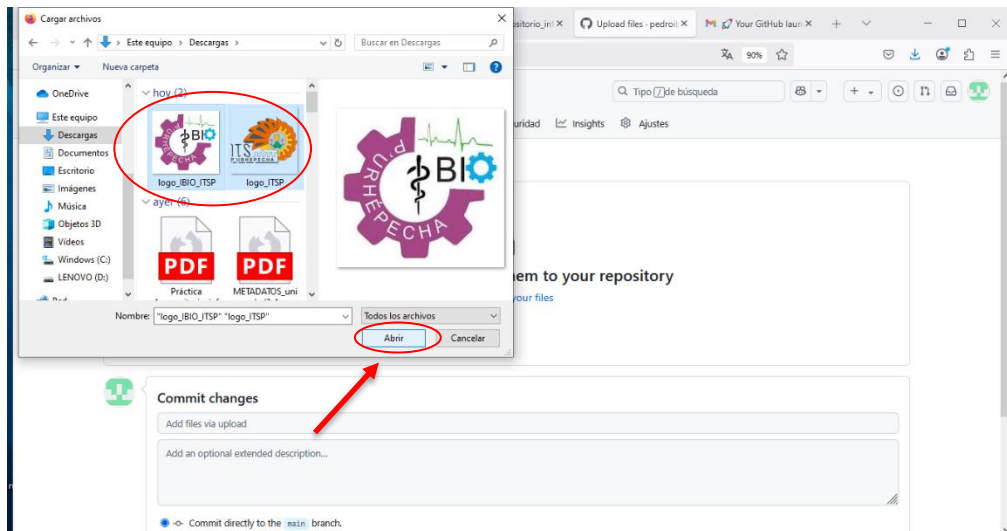




#### 4. Subida de imágenes:

- Se subieron dos imágenes locales (escudo de la facultad y de la universidad) al repositorio utilizando la opción "Upload files".
- Se explicaron los archivos agregados en el commit correspondiente.





[https://github.com/pedroibioitsp/practical1\\_fdpb](https://github.com/pedroibioitsp/practical1_fdpb)

---

---

# Capítulo 4

## Conclusiones

Esta práctica integró conocimientos teóricos y prácticos sobre control de versiones, destacando la importancia de GitHub como herramienta estándar en la industria. Los conceptos aprendidos sentaron las bases para futuros proyectos académicos y profesionales.



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®



Instituto Tecnológico Superior P'urhépecha  
DEPARTAMENTO DE INGENIERÍA BIOMÉDICA

# Bibliografía

<https://github.com/>