

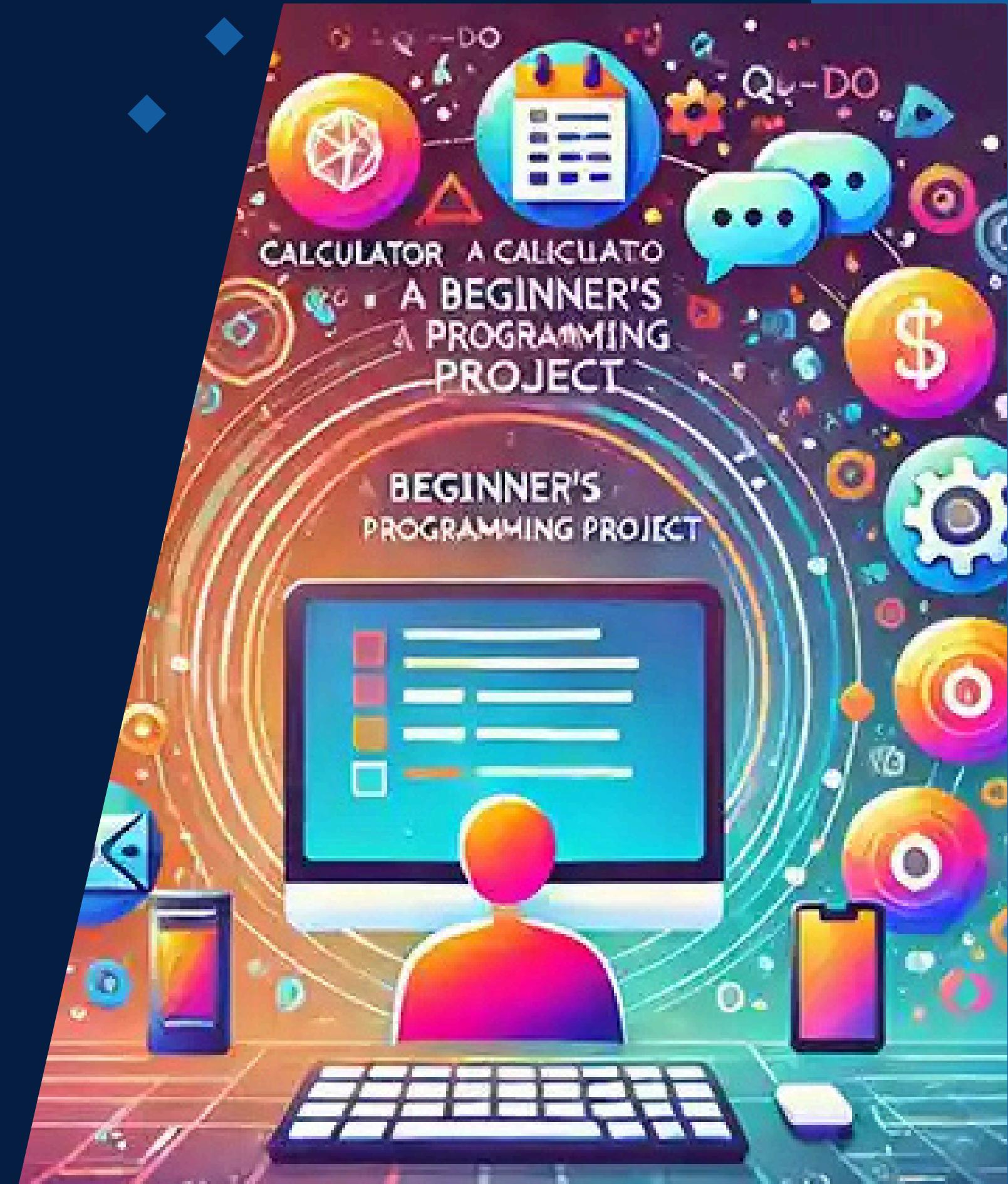
TUPAD



Programación 1
2C 2025

TUTORIAL PROYECTO INTEGRADOR

Por Martín Alejandro García



ETAPAS

01

Marco teórico

02

Caso práctico

03

Repositorio y Exposición

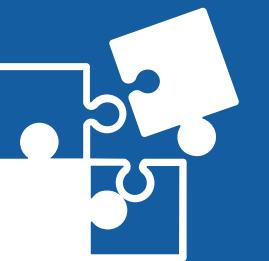




TEMAS PROPUESTOS

Los enunciados de los
distintos Trabajos
Integradores
propuestos serán
provistos por la cátedra

ESTRUCTURA SUGERIDA



1. Título del tema
2. Marco teórico (con fuentes)
3. Objetivo del trabajo
4. Diseño del caso práctico
5. Metodología utilizada
6. Resultados obtenidos
7. Conclusiones
8. Repositorio GitHub (con README y código completo)



¿CÓMO SE INVESTIGA?

1. Buscar fuentes confiables

Documentación oficial

Libros/e-books/papers técnicos

Artículos especializados

2. Tomar notas y resumir conceptos clave.

3. Armar una explicación ordenada con ejemplos.

4. Citar correctamente (mínimo APA básico).

EJEMPLO

Marco Teórico – Módulos y Paquetes en Python

- En Python, un *módulo* es simplemente un archivo con extensión .py que contiene definiciones y funciones que se pueden reutilizar.
- Un *paquete* es una carpeta que contiene módulos y un archivo especial `__init__.py` que indica que esa carpeta puede ser tratada como un paquete.
- Permiten dividir el código en partes reutilizables y organizadas.

Fuente: Python Software Foundation. (2024). Python Docs.
<https://docs.python.org/3/tutorial/modules.html>

DISEÑO DEL CASO PRÁCTICO

Analizar enunciado

1

Diseñar lógica general

2

Construir la base del proyecto

3

Organizar el proyecto

4

Definir estructuras de datos adecuadas

5

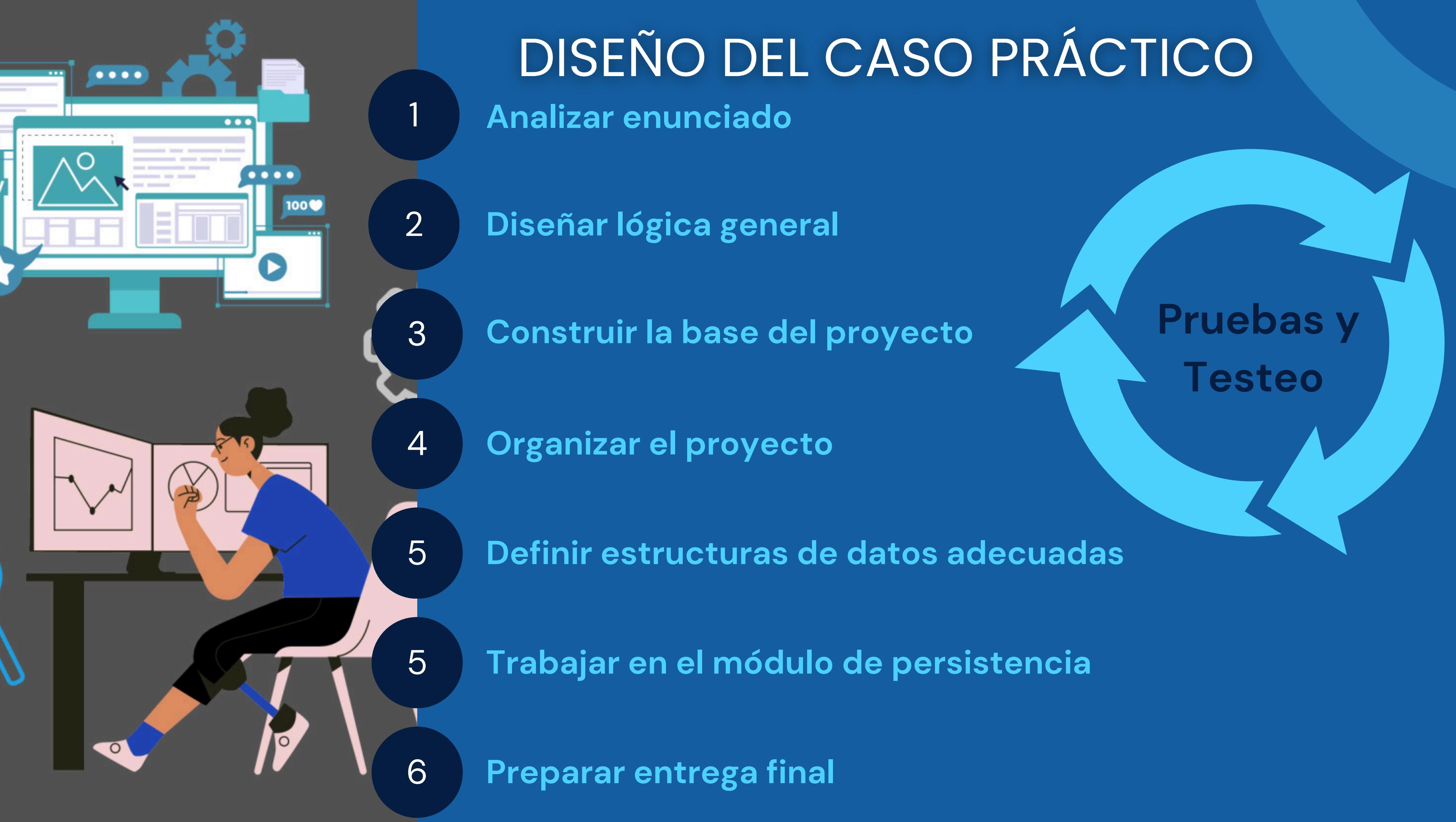
Trabajar en el módulo de persistencia

5

Preparar entrega final

6

Pruebas y
Testeo



MODULARIZACIÓN

Bloque Main

Búsquedas

Ordenamiento

Datos

Estadísticas

Filtros

Validaciones



RESULTADOS Y EVALUACION

- O1** Qué cosas funcionaron bien en el proyecto.
- O2** Qué fue difícil o que presentó un desafío mayor
- O3** Si hubo errores o decisiones que cambiaron durante el desarrollo.
- O4** Como fué la comunicación del equipo



CONCLUSIONES

- 01 Qué aprendieron
- 02 Por qué les parecen importantes estas herramientas.
- 03 Justificación del uso de las herramientas y ejemplos de uso
- 04 Cómo organizaron el trabajo entre los miembros del equipo.

DOCUMENTACIÓN

- “La documentación habla por tu código cuando vos no estás presente.”
- “Un buen README le ahorra tiempo a quien lee tu proyecto (incluso a uno mismo en el futuro!).”
- “Tu código puede estar bien hecho, pero si no se entiende, pierde valor.”

¿Qué incluir en el README?

- Descripción del proyecto
- Datos de la Universidad y la cátedra
- Integrantes
- Datos de profesores
- Estructura
- Instrucciones de ejecución
- Uso de librerías de terceros si corresponde
- Links a video y repositorio GitHub
- Ejemplos de entrada y salida



VIDEO

INTRODUCCIÓN Y
PRESENTACIÓN

EXPLICACIÓN

DEMOSTRACIÓN

REFLEXIÓN Y
CONCLUSIÓN

Especificaciones y
características
importantes

10 - 15 minutos

Mostrarse

Apoyo audiovisual

Fondo claro y letra
grande

ALGUNAS HERRAMIENTAS



ENTREGABLES

Repositorio GitHub

PDF CON MARCO TEÓRICO

VIDEO

CASO PRÁCTICO

README



ALGUNAS RECOMENDACIONES

- Tener constante comunicación de equipo
- No dejar el código para el final.
- Repartir tareas y cumplir plazos.
- Organizar bien el repositorio y gestión de ramas
- Practicar antes de grabar el video tutorial.



TUPAD



Programación 1
2C 2025

GRACIAS A TODOS/AS

Por leer **esta diapositiva** y ver el tutorial

Dudas y consultas en los foros del aula virtual y también en los encuentros sincrónicos semanales

