

## 1. Escritura de código

- Usa nombres descriptivos para variables, funciones y clases.
- Ejemplo: `get_average_speed()` es mejor que `gas()`.
- Evita duplicar código. Usa funciones o clases reutilizables.
- Mantén las funciones cortas. Una función debe hacer una sola cosa.
- Usa comentarios útiles y concisos, pero no comentes lo obvio.
- Aplica la regla KISS: Keep It Simple, Stupid. No compliques innecesariamente.
- No repitas el código (DRY): Don't Repeat Yourself.

## 2. Legibilidad

- Usa indentación consistente (por ejemplo, 4 espacios en Python).
- Agrupa el código lógicamente con espacios entre secciones.
- Evita nombres de variables como `x`, `y`, `z` salvo en bucles simples.
- Elimina código muerto o no usado.

## 3. Pruebas y control de errores

- Maneja errores con `try/except` (en Python) o estructuras similares.
- Escribe pruebas automatizadas si el proyecto es mediano o grande.
- Valida entradas de usuario o datos externos siempre.
- No ignores los errores silenciosamente, infórmanos correctamente.

## 4. Organización del proyecto

- Organiza tu código en módulos o paquetes.
- Ejemplo: `utils/`, `models/`, `controllers/`.
- Usa un archivo `README.md` para explicar cómo usar o instalar el proyecto.
- Incluye un `.gitignore` para evitar subir archivos innecesarios.
- Versiona tu código con Git y escribe mensajes de commit claros.

## 5. Buen uso de herramientas

- Formatea tu código automáticamente (ejemplo: `black` en Python, `Prettier` en JS).
- Haz revisiones de código (code review) si trabajas en equipo.
- Usa linters (como `flake8`, `pylint`, `eslint`) para detectar errores antes de tiempo.
- Documenta tu código (docstrings en funciones, por ejemplo).

## 6. Mentalidad del desarrollador

- Aprende y mejora constantemente: frameworks, patrones, testing, seguridad.
- Lee código de otros para aprender buenas (y malas) prácticas.
- Haz refactorización frecuente: mejora el código sin cambiar lo que hace.
- Ten en cuenta la escalabilidad y mantenimiento desde el inicio.

python

```
def calcular_promedio(lista):  
    """  
    Calcula el promedio de una lista de números.  
    """  
    if not lista:  
        return 0  
    return sum(lista) / len(lista)
```