Patrones de Diseño utilizados

1. Patrón Singleton implementado en el proyecto

- Propósito: garantizar que toda la aplicación web use una única conexión compartida hacia PostgreSQL.
- Ubicación: `database.py` en la clase `PostgresConnectionSingleton`.

Cómo funciona:
1. La primera vez que se pide una conexión, el `Singleton` invoca `ensure_database_exists()` para asegurarse de que la base exista.
2. Luego abre una conexión con `psycopg2.connect(...)` y la guarda en `_connection`.
3. Las peticiones posteriores reutilizan la misma conexión
4. Se expone mediante `get_connection()` para mantener el código del resto de funciones idéntico.

```python
class PostgresConnectionSingleton:
    _connection: ClassVar[PGConnection | None] = None

    @classmethod
    def get_connection(cls) -> PGConnection:
        if cls._connection is None or cls._connection.closed:
            ensure_database_exists()
            cls._connection = psycopg2.connect(POSTGRES_DSN)
            cls._connection.autocommit = True
        return cls._connection
```

- Beneficios:
  - Evita aperturas/cierres repetidos de conexiones.
  - Centraliza la configuración (DSN, `autocommit`).
  - Facilita el testing porque se puede hacer `PostgresConnectionSingleton.reset()` para forzar una reconexión limpia.

2. Escenario propuesto para patrón Strategy

- Escenario: módulo que sugiere recetas según necesidades calóricas individuales

Idea del patrón:
  - Definir una interfaz `CaloricPlanStrategy` con un método `select(recipes: list[Recipe]) -> list[Recipe]`.
  - Implementar estrategias concretas (`MantainWeight`, `WeightLoss`, `ExtremeWeightLoss`).

- En la vista (`app.py`) elegir dinámicamente la estrategia a partir del objetivo calórico solicitado por el usuario, mediante botones en la UI.
- Ventaja: separa las reglas de negocio, filtrado/cálculo de calorías en objetos intercambiables, evitando condicionales extensos en la ruta Flask y permitiendo añadir nuevas metas calóricas fácilmente.
- Estado actual: no está implementado en código, se documenta como ampliación para cumplir con el requisito de describir un segundo patrón.