LA GENERACIÓ AUTOMÀTICA DE PERSISTÈNCIA I ELS ORM

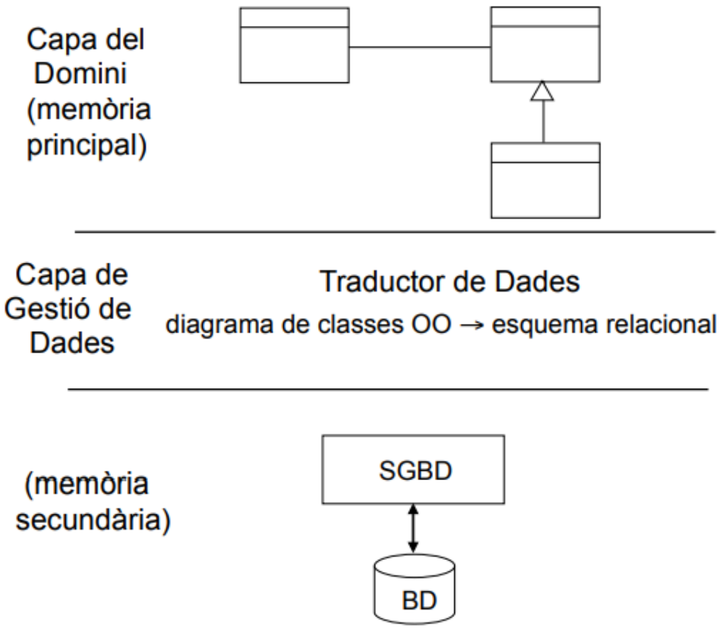
Realitzat per Adrián Patiño, Carlota Corcuera, Nadia Khier i Ariadna Gamez

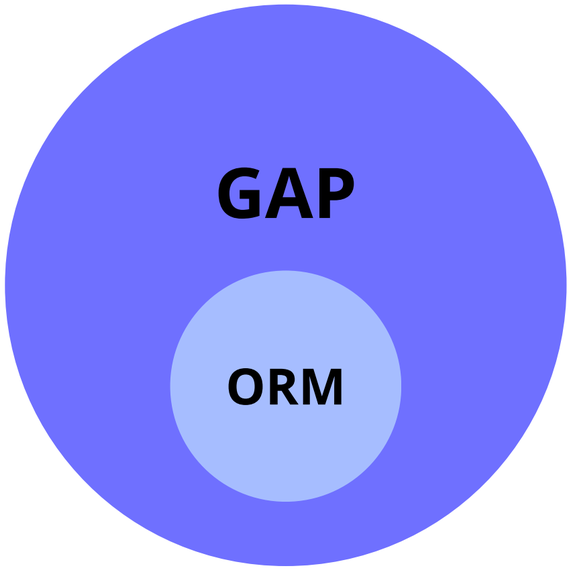
ÍNDEX

01. Què és GAP?

- Capacitat d'eines o frameworks per generar automàticament el codi necessari per connectar les aplicacions amb bases de dades.

- Inclou funcionalitats per a operacions CRUD sense necessitat d'implementació manual.





02. Requisits de les eines GAP

Compatibilitat amb diversos SGBD

Generació de codi optimitzat

Gestió de transaccions

Escalabilitat i adaptabilitat

Suport per a herència i polimorfisme

Abstracció del SQL

03. Importància i utilitat de GAP

Eficiència en el desenvolupament

Automatitza tasques tècniques repetitives, permetent als desenvolupadors centrar-se en la lògica de negoci.

Reducció d'errors

Minimitza els errors en escriure codi SQL i garanteix coherència entre el model de dades i les taules.

Productivitat

Allibera els desenvolupadors de part del treball, estalviant temps i esforç.

Escalabilitat i mantenibilitat

Facilita canvis en el disseny de dades i l'adaptació automàtica de les estructures.

Adaptabilitat

S'adapta als canvis en l'esquema de dades i requisits.

Abstracció de l'emmagatzematge

Els desenvolupadors no han de preocupar-se per les diferències entre sistemes de bases de dades.

03. Importància i utilitat de GAP

Creació automàtica de taules

Generen automàticament les estructures de taules a la base de dades a partir dels models definits al codi.

Gestió automàtica de consultes

Automatitza operacions CRUD sense necessitat d'escriure SQL manualment, facilitant la interacció amb les dades.

Mapeig Objecte-Relacional (ORM)

Els frameworks ORM permeten gestionar dades amb objectes, evitant escriure SQL i reduint errors.

Validació de dades

Asseguren que només es permetin dades vàlides a la base de dades, implementant restriccions definides al nivell del model.

Gestió de transaccions

Facilita l'execució segura de transaccions, garantint la coherència de les dades fins i tot en casos d'errors o interrupcions.

Migració de base de dades

Permet actualitzar esquemes de bases de dades fàcilment quan els models de l'aplicació canvien, reduint errors en modificacions estructurals.

04. Problemes que presenta

- Complexitat en models avançats
- Falta de personalització
- Corba d'aprenentatge
- Dependència d'eines específiques
- Sobrecàrrega en el rendiment

05. Exemples d'ús de GAP

- Sistemes de gestió empresarial:
  - ERP (Odoo)
  - CRM (Salesforce)

- Aplicacions Web i Mòbils
  - Django (Python)
  - Ruby on Rails (Active Record)

- Sistemes d'E-Commerce
  - Magento

06. Què és ORM?

Object Relation Mapping (ORM)
Una eina per connectar el món de la programació orientada a objectes amb les bases de dades relacionals.

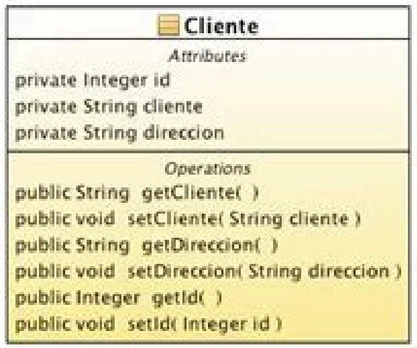
Mapejar objectes a taules de la BBDD

Generar consultes SQL

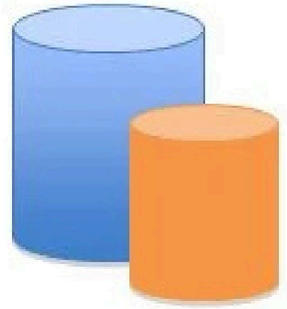
Gestionar operacions CRUD (Crear, Llegir, Actualitzar, Esborrar).

06. Què és ORM?

Modelo de Datos



Esquema E-R



ID	CLIENTE	DIRECCION
1	luis	zacatepec

07. Avantatges i desavantatges

- **Facilitat per treballar amb bases de dades** sense coneixements avançats de SQL.
- **Portabilitat:** el mateix codi funciona amb diferents sistemes de bases de dades.
- **Eficient en operacions CRUD** senzilles.

- **Més lent**: consultes subòptimes a la BBDD i dificultat per treballar amb gran volum de dades.
- **Poc eficient** en operacions que requereixin de molts joins.
- **Corba d'aprenentatge** alta.
- **Sobrecàrrega de memòria** (*eager loading*).

08. SQL vs ORM

SQL

```sql
SELECT name, email
FROM users
WHERE age > 18;

```

ORM

```python
from sqlalchemy import create_engine, Column, Integer, String, select
from sqlalchemy.orm import declarative_base, Session

# Configuración básica
Base = declarative_base()

class User(Base):
    __tablename__ = 'users'
    id = Column(Integer, primary_key=True)
    name = Column(String)
    email = Column(String)
    age = Column(Integer)

# Crear conexión a la base de datos
engine = create_engine('sqlite:///example.db')
session = Session(engine)

# Consulta usando ORM
query = session.query(User.name, User.email).filter(User.age > 18)
for name, email in query:
    print(f"Name: {name}, Email: {email}")

```

09. Referències

freeCodeCamp. (2022, 10 de febrer). Què és un ORM? El significat de les eines de mapatge objecte-relacional. Recuperat el 6 de desembre de 2024, de https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/

gvadmin. “Introduction to Persistence, ORM & Hibernate Framework.” GeoViz Inc., 24 Apr. 2014, geo-viz.com/blog/introduction-persistence-orm-hibernate-framework/. Accessed 6 Dec. 2024.

Mayol, E. (2024). Disseny de la Capa de Gestió de Dades: Estratègies de Gestió de Persistència de Dades [Diapositives de classe]. FIB - UPC.

“Migrating from Hand-Written Persistence Layer to ORM.” Stack Overflow, stackoverflow.com/questions/2867325/migrating-from-hand-written-persistence-layer-to-orm.

PREGUNTES?