EasyDataTestGenerator
Generador de dades de test

Daniel Riera Flinch
Grau d'Enginyeria Informàtica
Bases de dades

David Porti
Josep Curto

Data Lliurament
07/2024

FITXA DEL TREBALL FINAL

Títol del treball:	*EasyDataTestGenerator* *Generador de dades de test*
Nom de l'autor:	*Daniel Riera Flinch*
Nom del consultor/a:	*Josep Curto Díaz*
Nom del PRA:	*David Porti Pujal*
Data de lliurament (mm/aaaa):	*07/2024*
Titulació o programa:	*Grau d'Enginyeria Informàtica*
Àrea del Treball Final:	*Bases de dades*
Idioma del treball:	*Català*
Paraules clau	*test data, data generator, SQL data.*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

EasyDataTestGenerator és una nova aplicació que et permet generar dades de test per bases de dades relacionals, fitxers en format CSV o fitxers en format JSON.

Aquest treball vol ser una solució més per ajudar al testeig de dades i fer programari de més qualitat. L'objectiu és omplir bases de dades generant dades aleatòries, amb sentit i interrelacionades entre elles. Des d'un simple set de dades com pot ser una llista de clients amb el seu nom, NIF, telèfon, email, etc. fins a factures amb capçalera i detall, i els venciments associats.

Per al desenvolupament, s'ha aplicat la metodologia Agile Scrum, d'aquesta manera s'ha portat un seguiment molt exhaustiu i s'han establert unes fites molt marcades en forma de sprints.

Durant el procés de desenvolupament no hi ha hagut canvis sobre la planificació inicial. Tampoc en l'àmbit tecnològic. S'ha desenvolupat segons el pla inicial.

S'han assolit tots els objectius. Hi han hagut dificultats, ja que treballar alhora en varies bases de dades has de tindre en compte diferents factors, tant en l'àmbit de SQL com a escala de funcionament dels drivers de cada base de dades.

Abstract (in English, 250 words or less):

EasyDataTestGenerator is a new application that allows you to generate test data for relational databases, CSV files, or JSON files.

This project aims to be a solution to help with data testing and improve software quality. The goal is to populate databases by generating meaningful, interrelated random data. From a simple dataset, such as a list of clients with their names, tax identification numbers (NIF), phone numbers, emails, etc., to invoices with headers, details, and associated due dates.

For development, the Agile Scrum methodology has been applied, ensuring thorough tracking and setting clear milestones in the form of sprints.

During the development process, there have been no deviations from the initial plan, neither in terms of technology nor overall planning. The project has been executed according to the original blueprint.

All objectives have been achieved, although there were challenges due to working simultaneously with multiple databases, considering various factors related to SQL and the operational scale of each database driver.

Índex

Llista de Figures

Proposta

Cada dia es genera una immensitat de línies de codi en diferents projectes. Està de moda tot el relacionat amb el testing, però curiosament sembla que molts s'obliden de testejar un bon set de dades a la base de dades per veure rendiment, capacitat, aplicar optimitzacions, etc.

Un dels punts més dèbils en un projecte, és quan el desenvolupador fa proves amb dades estil "aaa" o "a@a.com", puja a producció i en posar un nom compost o un email llarg, però vàlid peta o no es pot fer.

Aquest treball vol ser una solució més per ajudar al testeig de dades i fer programari de més qualitat. L'objectiu és omplir bases de dades generant dades aleatòries, amb sentit i interrelacionades entre elles. Des d'un simple set de dades com pot ser una llista de clients amb el seu nom, NIF, telèfon, email, etc. fins a factures amb capçalera i detall, i els venciments associats.

Les tecnologies a emprar per a desenvolupament del programari són:

- C# de Net 8.0 [3]
- Extensió SonarLint [4]
- Per l'entorn de test: Docker 4.x [5]

El desenvolupament es realitza amb C# a on s'ha escollit l'última versió Net 8, ja que és la més recent LTS (Long Term Service). El codi pot ser multiplataforma, però l'entorn de treball es limitarà a Windows per falta de temps, en concret Windows 11.

Hi ha dues possibilitats de desenvolupar sota Visual Studio Code o Visual Studio 2022. S'ha escollit Visual Studio 2022, ja que és un IDE integrat que incorpora Intellisense, un debugger avançat i té algunes facilitats que Visual Studio Code no té com a concepte d'editor.

A més a més, per augmentar la qualitat del codi s'instal·la l'extensió SonarLint. És conegut que aquesta extensió té alguns bugs amb C# detectant errors que no ho són. En el cas de trobar-ne algun cas es posarà un comentari. En termes globals, funciona força bé.

L'objectiu és un programari orientat a objectes, per capes i l'ús de SOLID per assegurar-ne la qualitat i solidesa.

Limitacions

Un programari d'aquest tipus pot abraçar molt i ser un desenvolupament de mesos i mesos. Per aquest motiu, es posen uns límits principals que marquen l'abast del projecte i per poder ajustar al temps disponible per fer el TFG.

Si bé el programari pot ser multiplataforma (Windows, Linux, MacOS) es centra únicament en Windows, i concretament en Windows 10/11.

Com a destí tenim moltes bases de dades a on es poden generar dades, però es limitarà a Microsoft SQL Server [6], PostgreSQL [7] i MySQL/MariaDB [8].

Com a tipus de dades, principalment, en tenim dos:

- Concrets: Requereixen càlculs, per tant són gestionats via codi. Per exemple NIF, email.
- Genèrics: Són dades que poden vindre d'un fitxer JSON [9] o d'una taula de la BD i que permeten agafar qualsevol mena de dades com cotxe, model i motor, o bé Província i CP, etc.

Com a concrets podem fer una llista immensa, desde NIF, usuari, contrasenya, nombre de la seguretat social, matrícula de cotxe, colors, termes mèdics, etc. Però ens cenyirem als més estàndards segons un model d'exemple de base de dades documentat més endavant. El desenvolupament d'aquests tipus es fa amb el concepte de "plugin", per tant, fer-ne de nous en futur és fàcil i intuïtiu.

Com a tipus de dades genèrics en posarem un parell d'exemples d'acord amb el model de base de dades que s'estipula més endavant. Però crear-ne més és seguir la documentació per crear aquests fitxers JSON o omplir taules amb unes característiques establertes.

Per poder fer dades aleatòries i no repetitives és necessari utilitzar prelectura de dades. L'objectiu principal és una bona execució, i tant el rendiment com l'optimització d'ús de memòria no queda dins de l'abast d'aquest projecte, podent-se ampliar posteriorment.

En termes generals, es desenvoluparà pensant en un bon producte, fàcil d'ampliar i que sigui adaptable al màxim de casos per generar dades de test.

Objectius

- Fer dades de tests per a sistemes complexos.
- Crear dades amb sentit en tot el conjunt de les taules.
- Poder generar dades dependents. Per exemple, dades com posar una Marca de Cotxe, i al següent camp poder posar un Model de la Marca seleccionada, i, per exemple, un tercer camp que sigui la motorització del Model seleccionat.
- Poder omplir taules dependents. Típic exemple de dues taules que són la capçalera d'una factura i el detall d'aquesta. L'objectiu és poder inserir dades a la capçalera, obtenir l'ID, inserir dades al detall referenciant l'ID de capçalera, i inclús, actualitzant dades de la capçalera com el sumatori de Total, IVA, etc. segons les línies de detall creades. Aquesta feature no estarà disponible en algunes sortides com CSV, JSON, etc.
- Queda descartat fer un Front-end per la configuració, ja que se surt del projecte actual en temps i definició.
- Que sigui fàcilment ampliable a nous sets de dades.
- Que sigui ampliable a noves bases de dades.
- Es limita la complexitat que poden arribar alguns datafields per no fer un projecte impossible d'acabar. En un futur seria recomanable implementar un compilador en runtime de C# per fer una lògica completa per camp.
- Que es pugui executar des d'un programa, servei web, DLL, etc. Es fa una implementació base i que sense dificultat que es pugui adaptar.

Els avantatges d'utilitzar un generador de dades són:

- Automatització. És un estalvi de temps per al testers i DevOps. Permet diferents sets de dades per diferents proves i entorns.
- Reutilització. Permet fet dades repetitives o de regressió creant diferents conjunts de dades i estalviant temps i esforços.
- Diversitat. Es poden fer molts sets de dades per abraçar diferents escenaris i condicions.
- Personalització. Generació de dades a mida, tant en contingut com en quantitat.
- Dades realistes (mantenint privacitat). L'objectiu és crear dades amb sentit, com que els emails dels empleats es basi en el nom de l'empleat i el domini en el nom de l'empresa.
- Randomització. Control sobre la aleatorietat i qualitat de les dades.
- Escalabilitat. Permet escalar les dades tant en quantitat com en diversitat.

Competència

A continuació els productes que són competència o similars l'exposat en aquest projecte. Si bé s'ha fet internament una anàlisi exhaustiu, en aquest apartat sols es fa un comentari resum.

Web/API: https://generatedata.com/ [12]

> Sols permet dades planes en el concepte d'una sola taula. Limitat a la versió gratuïta. Dades en anglès, no permet informació específica de la nostra regió con NIF.

Web/API: https://mockaroo.com/ [13]

> Molt més complet que l'anterior però de pagament. Permet dades anidades i variables o fórmules. Es poden pujar datasets propis.

Web/API: https://testingdatagenerator.com/ [14]

> Dades molt bàsiques i de pagament. Més adaptat per fitxers de Excel (CSV, XLSX).

Com aquests 3 anteriors en tenim uns quants més, que més o menys fan el mateix, pensats per ser utilitzats mitjançant una API i de pagament.

A diferència EDTG aporta que és gratuït, permet múltiples anidacions, permet afegir nous tipus de dades, fer referències a altres camps i no depèn de webs de tercers. Per tant, sense cap mena de limitacions.

Com a punt fort, EDTG connecta directament a les bases de dades tractant el millor possible els INSERT, mentre que aquestes webs generen fitxers JSON, CSV o simplement INSERTS.

Per aquest últim motiu, no poden enllaçar amb altres taules per buscar dades. Tampoc permeten executar instruccions SQL abans o després de la generació.

Finalment, una menció especial pels generadors de dades sintètiques. Com per exemple:

- SDV [15]
- Mostly·AI [16]
- Clearbox AI [17]

El concepte de dades sintètiques són dades generades artificialment amb ML/IA d'acord amb dades proporcionades d'entrenament o referència. Un dels punts forts és que poden generar dades que no són autèntiques i, per tant, no afecten a la privacitat.

Tenint molts avantatges que diferents pàgines web i blogs expliquen, particularment, el problema que detecto és que imiten dades, per tant, no poden generar NIF, DNI o qualsevol mena de dades que depengui de un checksum o una fórmula de validació. A més a més, incorpora condicions amb fórmules/scripting no és possible.

Particularment, crec que la generació de les dades sintètiques pot ser un complement perfecte per EDTG i que en un futur es pot implementar.

Què millora el meu projecte versus la competència:

Principalment la competència es basa en dos conceptes. En una Web/API o en un producte molt específic.

El que aquesta aplicació pretén és poder generar dades localment sense utilitzar recursos de tercers a on hi ha dependència de preu, velocitat, privacitat, gestió d'errors, SLA, etc.

És un producte gratuït, que qualsevol desenvolupador en qualsevol empresa pot adaptar i ampliar a les seves necessitats.

Obert a qualsevol mena de base de dades, tan relacionals, com no-relacionals, com fitxers d'intercanvi.

Enfocat per dades senzilles tant com dades complexes matenint una coherència.

Requisits de maquinari i programari

Els requisits per fer funcionar el programari són molt bàsics:

- **Sistema operatiu:** Windows 10 o superior
- **CPU:** Tant x86 com arm86
- **RAM:** Per l'execució amb dades fins a 1.000.000 de registres fins a 1GB lliures. Per sets de dades més grans són necessaris 2GB. Per sets de dades petits un màxim de 512MB lliures.
- **Espai en disc:** Sols es requereix l'espai necessari d'instal·lació que són 350MB (incloent SDK .NET). Addicionalment l'espai en la base de dades de destí.

Cada base de dades té uns requisits diferents i depèn molt de les necessitats. Per fer un entorn bàsic en Docker es recomana almenys una CPU i7 o equivalent i 512 GB de RAM per cada instància de base de dades. Per fer proves, reservar 1GB d'espai en disc per cada instància de BD.

És possible compilar el programa en *Linux* o *MacOS*, però no s'ha provat i no es poden donar especificacions mínimes.

La versió mínima de les bases de dades suportades són:

- **SQL Server:** SQL Server 2003
- **Postgres:** Postgres 6.4
- **MySQL/MariaDB:** MySQL 8.0

Entorn de treball

Com s'ha mencionat en l'apartat *Limitacions* l'entorn de treball és Windows, encara que és fàcilment transportable a altres sistemes operatius com Linux o MacOS.

S'utilitza Visual Studio 2022 Community Edition i els requisits tècnics es poden consultar a [1]

La part complex és l'entorn de base de dades en què es pugui començar de zero o bé reutilitzar durant el desenvolupament les instàncies ja creades. Per aquest motiu, durant el desenvolupament es farà servir Docker en la seva versió 4.

Docker ens permet fer instàncies de cada tipus de base de dades i fer servir volums de persistència. Per facilitar la feina es documentarà com iniciar els Dockers en línia de comanda per poder-se executar en qualsevol sistema operatiu.

No és obligatori utilitzar Docker, si es disposa ja d'algun servidor de base de dades, es pot utilitzar posant les dades de connexió [2] en l'apartat de configuració.

Impacte en sostenibilitat, ètic-social i de diversitat

Per l'elaboració del TFG he tingut en compte el compromís ètic i global que aplica la UOC. Tant en les decisions preses com en l'execució del projecte, no he detectat conseqüències negatives en la sostenibilitat mediambiental o l'empremta ecològica.

Un dels objectius d'aquesta aplicació en sostenibilitat és facilitar conjunts de dades suficientment bones per evitar repeticions innecessàries de dades i tests en el desenvolupament d'altres aplicacions. Permetent ser més eficients energèticament. En tot cas, l'impacte sempre dependrà de l'ús racional que se'n faci.

Com a recomanació final per millorar l'eficiència energètica, és valorar que les configuracions i bases de dades estiguin en servidors que facin un ús eficient de l'energia en l'àmbit de maquinari i no siguin específics d'alt rendiment.

En el vessant ètic-social l'aplicació permet i garanteix que les dades es creïn de forma responsable, preservant confidencialitat i privacitat en ser dades properes a la realitat sense ser reals. És necessari crear usuaris amb tota mena de rols per fer proves i garantir la privacitat final dels usuaris.

Finalment, i no inclòs en el projecte per falta de temps, s'hauria de tindre en compte aspectes d'accessibilitat i usabilitat en l'aplicatiu. De forma que es puguin establir polítiques per donar accés a l'ús a persones amb diferents discapacitats, com per exemple visuals.

Com que aquest aplicatiu pot estar inclòs en un altre en forma de llibreria, es podria generar documentació de bones pràctiques o recomanacions en sostenibilitat, ètic-social i de diversitat per l'integrador.

Anàlisi de Riscos

Errors de planificació	Probabilitat: Baixa	Impacte: Mig
Errors en l'estimació de tasques que causin desviacions importants. **Acció:** Correctiva - Ampliació temps de dedicació		

Errors de disseny/tecnologia	Probabilitat: Baixa	Impacte: Alt
Decisions errònies preses en les eines i tecnologies per fer el desenvolupament. **Acció:** Correctiva - Analitzar impacte i avaluar un canvi de tecnologia.		

Malaltia o accident	Probabilitat: Baixa	Impacte: Mig
La durada d'una enfermetat tingui impacte o bé la impossibilitat d'avançar per algun tipus de lesió. **Acció:** Correctiva - Sol·licitud d'ampliació de termini d'entrega o reducció de funcionalitats.		

Compaginació	Probabilitat: Baixa	Impacte: Mig
Factors derivats de tindre família amb fills. **Acció:** Correctiva - Ampliació temps de dedicació.		

Dades i Seguretat	Probabilitat: Baixa	Impacte: Baix
Pèrdua involuntària per errors de maquinari o programari. Pèrdua voluntària per errors de programari o instruccions errònies SQL. **Acció:** Preventiva - Sistema de còpies de seguretat.		

Model de BD

El model proposat és una aproximació a la realitat d'una empresa i no pretén ser 100% real. El principal ús és fer un test i demostració del funcionament del generador de dades.

Notes:

- Els camps "clau" (passwords) per efectes demostratius és un string aleatori, però en la realitat hauria de ser un algorisme apropiat.

El model té dues parts ben diferenciades:

- El model d'empleats, en blau, que pretén demostrar:
  - El funcionament de fins a tres taules dependents.
  - Tipus bàsics de dades
  - Dades genèriques extretes d'Internet i dependents (cotxes)

- El model de clients, en taronja, amb les següents característiques:
  - Funcionament de tipus SQL
  - Dificultat de crear taula estil capçalera-detall, actualitzant camps de la capçalera un cop generat el detall.
  - Accés a dades calculades
  - Execució de fórmules.

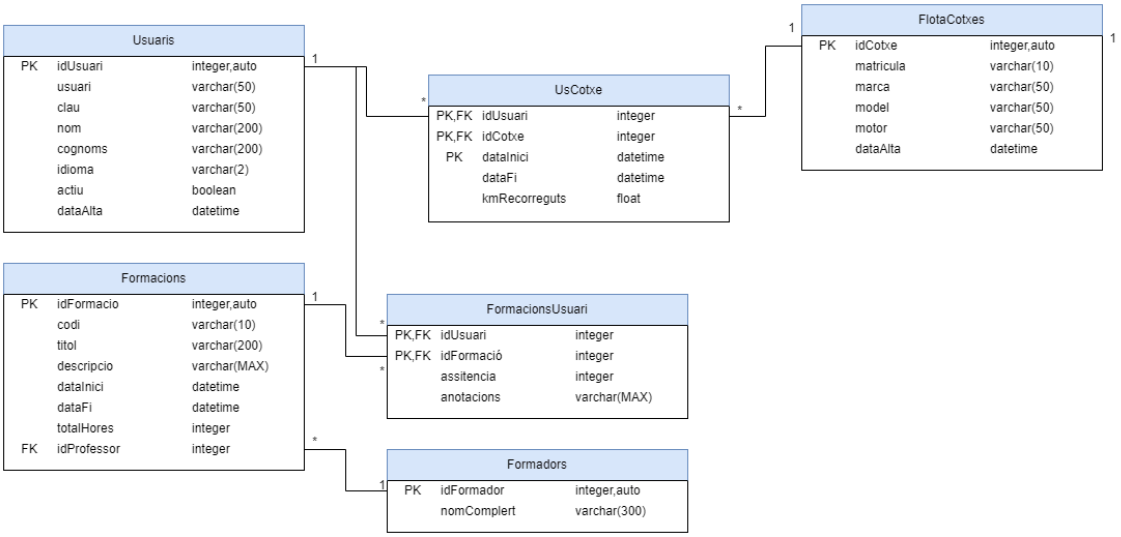

Figura 1: Model de BD - Empleats

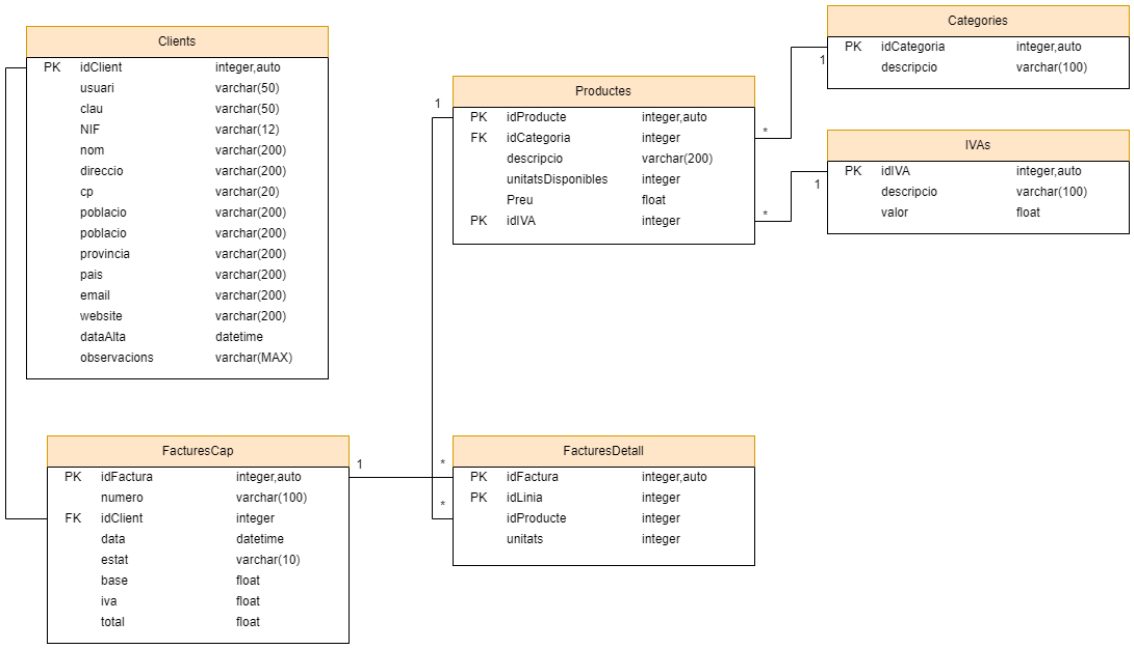

Figura 2: Model de BD - Clients

Model de funcionament del Programa

En la següent figura es pot veure, en termes generals, el funcionament del programa:

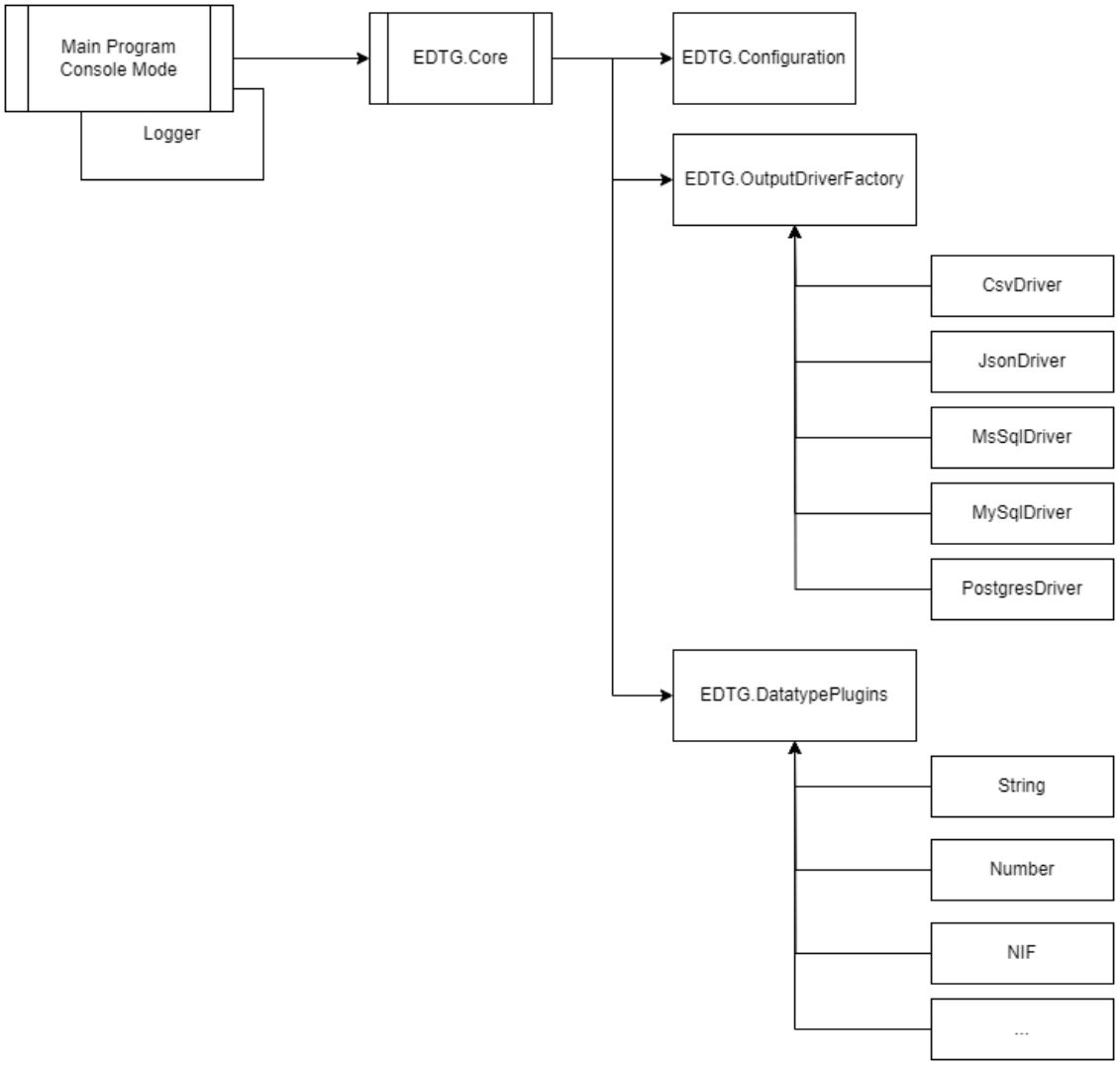

Figura 3: Model de funcionament del programa

- **Main Program**: En aquest projecte s'ha utilitzat un programa en consola a on es pot indicar per línia d'arguments la configuració que ha d'utilitzar. És simplement una mostra de com crear un Logger i fer servir EDTG. Podria ser una DLL, un Controller, etc.
- **EDTG.Core**: És el punt d'inici de la llibreria, i rebent com a arguments la configuració a aplicar, comença el parsing de la configuració, la definició de on aniran les dades i l'execució de les sentències.
- **EDTG.OutputDriverFactory**: És una factoria per crear una instància del destí de les dades. A través d'una interface el core sols necessita anar passant

instruccions que cada driver farà el que pertoca segons si es un Csv o un PostgreSQL. La implementació d'un nou driver es força senzilla.

- **EDTG.DatatypePlugins**: Cada tipus de dades es un plugin a on es defineix les seves propietats i fins a on pot arribar. Té dues missions: entendre els paràmetres i donar un valor d'acord amb els paràmetres. Per exemple, pots definir un String, que sigui aleatori, que tingui entre 1 i 5 caràcters i que el 50% pugui ser null.

Característiques suportades

Output drivers

- Base de dades: Microsoft SQL Server, MySQL i PostgreSQL
- Text:
  - Format de fitxer JSON
  - Format de fitxer CSV [3]

Datatype Plugins

- AUTO: Valor autonumèric suportat en cada base de dades. En el cas d'Oracle seria el fet de treballar amb un Sequence. Gestionat automàticament.

  Atributs: Cap

- NUMERIC: Valor numèric que pot ser un Smallint, Int, Float, Double, Decimal, etc. L'objectiu és poder donar un nombre.

  Atributs: Decimals, Value, Min, Max, %Blank.

- STRING: Valor de cadena.

  Atributs: Value, Values (llista), MinLength, MaxLength, Mask, Trim, Upper, Lower, %Blank, Formula.

- DATETIME: Treball amb valors data, hora o data hora.

  Atributs: Value, Values, Min, Max, Date (true/false), Time (true/false), %Blank, Formula.

- BOOLEAN: Valors true o false. Equivalent a bit en algunes bases de dades.

  Atributs: Value, %Blank, Formula.

- EMAIL: Generar un email

  Atributs: Value, %Blank, Username, Domain, Formula

- USERNAME: Generar un nom d'usuari per fer login.

  Atributs: Value, %Blank, MinLength, MaxLength

- TABLE: Relació amb un camp d'una taula de la BD

  Atributs: Value, %Blank.

Per veure tots els plugins i el seu funcionament exacte cal veure el manual en el Annex 2.

Alguns tipus com PASSWORD (generar una clau) s'han implementat amb el STRING utilitzant la creació d'una cadena random.

L'objectiu és que es puguin desenvolupar en codi tots els tipus necessaris per a una empresa o entorn de treball, com poden ser: NIF, matrícules, nº de Seguretat Social, Comptes Bancaris, Dades estadístiques, etc.

Exemple proposta de configuració

```
{
    "General": {
        "DemoMode": false
    },
    "Output": {
        "Driver": "mssql",
        "ConnectionString": "Server=localhost\\sqlexpress;Initial
Catalog=TestUOC;Integrated Security=True;User
Id=sa;Password={EDTG_PASSWORD}",
        "File": "",
        "StopOnSqlError": true,
        "StopAfterNumErrors": 10,
        "InitialStatments": [
            "CREATE TABLE ...",
            "INSERT INTO ..."
        ],
        "TruncateTables": false
    },
    "Tables": {
        "Customers": {
            "Truncate": true,
            "MinRows": 10,
            "MaxRows": 0,
            "Fields": {
                "Id": {
                    "Value": "Auto"
                },
                "Name": {
                    "Type": "String",
                    "Nullable": 30,
                    "MinLength": 0,
                    "MaxLength": 300
                }
            },
            "Tables": {
                "CustomerSales": {
                    "Truncate": true,
                    "MinRows": 10,
                    "MaxRows": 0,
                    "Fields": {
                        "Id": {
                            "Type": "Auto"
                        },
                        "IdCustomer": {
                            "Type":  "Number",
                            "Value": "{Parent.Id}"
```

```
                    },
                    "Sales": {
                        "Nullable": 30,
                        "Value": "Float",
                        "Args": {
                            "Decimals": 2,
                            "MinValue": 0.00,
                            "MaxValue": 9999.99
                        }
                    }
                }
            }
        }
    }
}
```

Metodologia i Planificació.

El ideal és aplicar una metodologia àgil com pot ser Scrum. Però el fet que sols hi ha un desenvolupador i dates tancades, he optat per utilitzar Jira [3], en un projecte Scrum però sense Sprints, aplicant sols dates d'inici i de fi.

La divisió per èpiques són els blocs principals de desenvolupament. Però moltes tasques incorporant canvis incrementals a tasques ja fetes prèviament. Per exemple, al crear un plugin nou és possible que s'hagi d'adaptar el sistema de configuració, o potser, complementar el format de SQLs que s'han de fer en el driver.

Planificació resum per mesos:

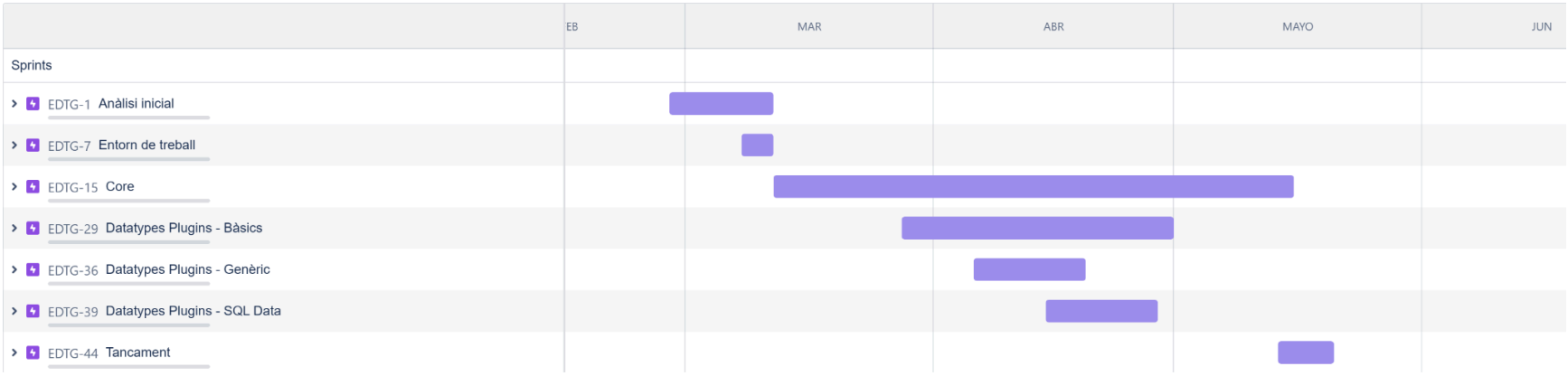

Figura 4: Planificació - Resum per mesos

Resum per setmanes:

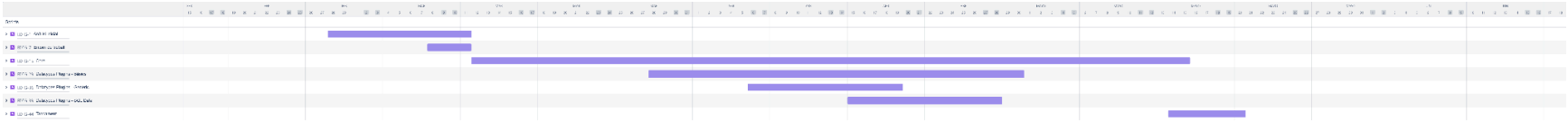

Figura 5: Planificació - Resum per setmanes

Detall:

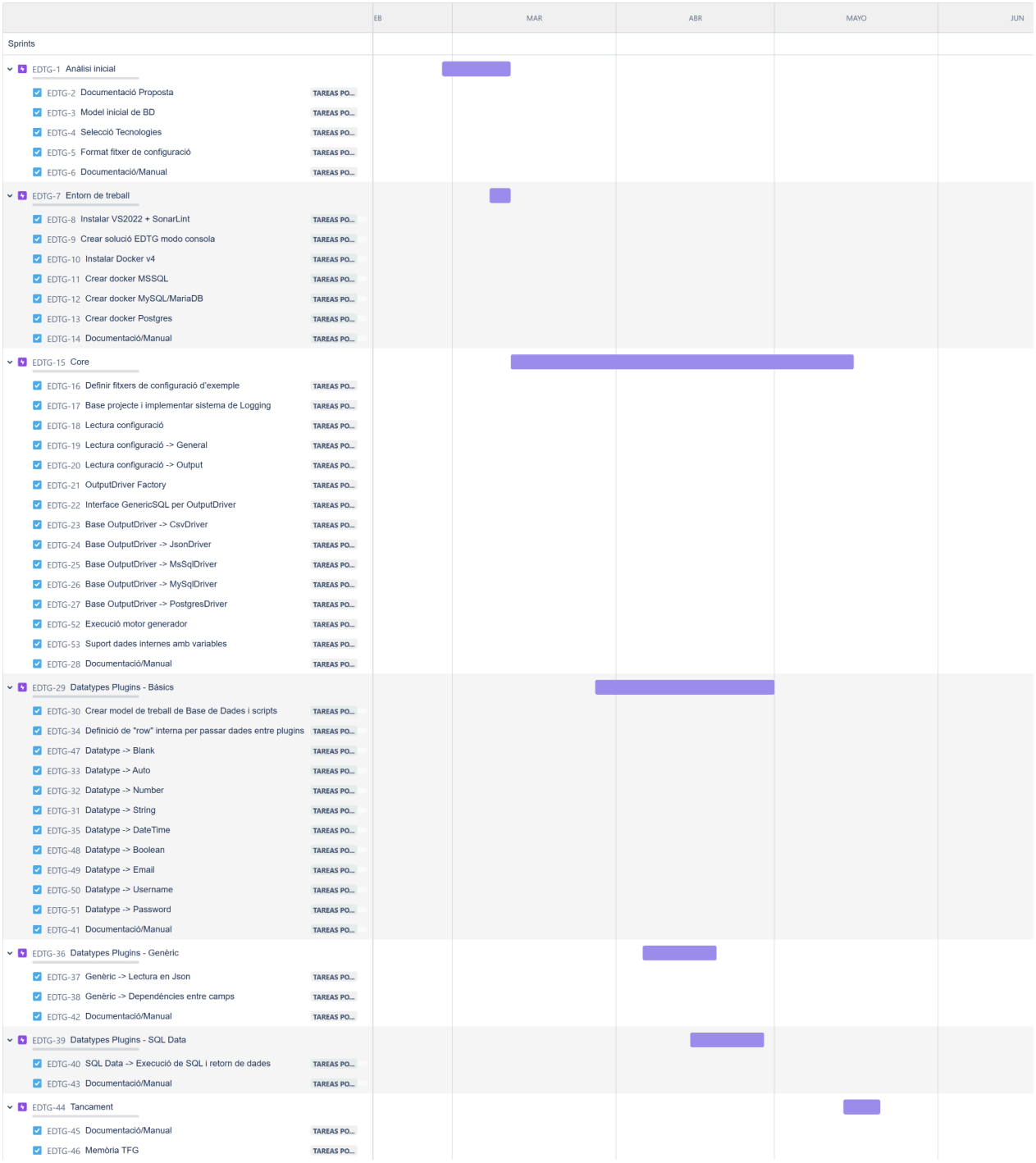

Figura 6: Planificació - Detall de tasques

Estimació d'hores:

Identificador	Èpica	Tasca	Temps (h)
EDTG-1	**Anàlisi inicial**		
EDTG-2		Documentació Proposta	10
EDTG-3		Model inicial de BD	5
EDTG-4		Selecció Tecnologies	5
EDTG-5		Format fitxer de configuració	18
EDTG-6		Documentació/Manual	30
EDTG-7	**Entorn de treball**		
EDTG-8		Instalar VS2022 + SonarLint	4
EDTG-9		Crear solució EDTG modo consola	2
EDTG-10		Instalar Docker v4	2
EDTG-11		Crear docker MSSQL	2
EDTG-12		Crear docker MySQL/MariaDB	2
EDTG-13		Crear docker Postgres	2
EDTG-14		Documentació/Manual	4
EDTG-15	**Core**		
EDTG-16		Definir fitxers de configuració d'exemple	8
EDTG-17		Base projecte i implementar sistema de Logging	16
EDTG-18		Lectura configuració	24
EDTG-19		Lectura configuració -> General	4
EDTG-20		Lectura configuració -> Output	10
EDTG-21		OutputDriver Factory	20
EDTG-22		Interface GenericSQL per OutputDriver	24
EDTG-23		Base OutputDriver -> CsvDriver	16
EDTG-24		Base OutputDriver -> JsonDriver	16
EDTG-25		Base OutputDriver -> MsSqlDriver	4
EDTG-26		Base OutputDriver -> MySqlDriver	4
EDTG-27		Base OutputDriver -> PostgresDriver	4
EDTG-52		Execució motor generador	80
EDTG-53		Suport dades internes amb variables	60
EDTG-28		Documentació/Manual	40
EDTG-29	**Datatypes Plugins - Bàsics**		
EDTG-30		Crear model de treball de Base de Dades i scripts	16

EDTG-34		Definició de "row" interna per passar dades entre plugins	24
EDTG-47		Datatype -> Blank	4
EDTG-33		Datatype -> Auto	8
EDTG-32		Datatype -> Number	8
EDTG-31		Datatype -> String	16
EDTG-35		Datatype -> DateTime	16
EDTG-48		Datatype -> Boolean	4
EDTG-49		Datatype -> Email	16
EDTG-50		Datatype -> Username	8
EDTG-51		Datatype -> Password	8
EDTG-41		Documentació/Manual	
EDTG-36	**Datatypes Plugins - Genèric**		
EDTG-37		Genèric -> Lectura en Json	18
EDTG-38		Genèric -> Dependències entre camps	18
EDTG-42		Documentació/Manual	6
EDTG-39	**Datatypes Plugins - SQL Data**		
EDTG-40		SQL Data -> Execució de SQL i retorn de dades	40
EDTG-43		Documentació/Manual	4
EDTG-44	**Tancament**		
EDTG-45		Documentació/Manual	8
EDTG-46		Memòria TFG	24
		Total:	**662**

Figura 7: Planificació - Estimació d'hores

Com que és un desenvolupament a on la majoria de les tasques tenen modificacions contínues al nucli, a continuació es determina que és el que espero entregar a cada PAC.

PAC2:

- Model de Base de Dades amb un script per cada tipus de BD utilitzat.
- Executable funcionant correctament amb tipus de dades bàsiques: BLANK, AUTO, INT, STRING, DATETIME, BOOLEAN.
- Capaç de generar dades bàsiques.

PAC3:

- Ús de dades genèriques

- Ús de variables internes row, parent o [table] més nom del camp. Exemple: parent.idFactura
- Ús de dades d'una altra taula, per SELECT
- Generador de dades amb taules enllaçades.

Entorn de treball

Pel desenvolupament del projecte s'ha preparat amb un entorn Docker[5] versió 4 per poder acollir diferents bases de dades amb una configuració fàcil i assequible per replicar a qualsevol ordinador. Tot i això es pot utilitzar base de dades ja instal·lades i en funcionament.

Una taula resum de les bases de dades del entorn de treball és:

Base de Dades	**Administració**	**Usuari**	**Clau**
SQL Server Express[6]	SMSS	sa	UOC_12345
PostgreSQL[7]	http://localhost:5433	postgres	UOC_12345
MariaDB[8]	http://localhost:3307	root	UOC_12345

En l'entorn Microsoft/Windows és molt comú utilitzar els Connection Strings [?] per definir com serà la connexió a la base de dades.

Aquestes connexions s'usen en els fitxers de configuració i permeten definir paràmetres molt bàsics fins a configuracions complexes que són requerides per alguns administradors de bases de dades.

Base de Dades	**Connection String**
SQL Server Express	Server=localhost;Initial Catalog=ProvesUOC;Integrated Security=False;User Id=sa;Password={EDTG_PASSWORD};TrustServerCertificate=true
Postgres	Server=localhost;User id=postgres;Password={EDTG_PASSWORD};Database=ProvesUOC;TrustServerCertificate=true
MariaDB	Server=localhost;User id=root;Password={EDTG_PASSWORD};Database=ProvesUOC;

Totes les proves es realitzen en la base de dades *ProvesUOC*. La creació i la inserció de les dades és gestionat dins dels mateixos fitxers de configuració.

SQL Express

Docker:

docker run --name=SqlExpress -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=UOC_12345" -p 1433:1433 -v SqlExpressVolume:/var/opt/mssql -d mcr.microsoft.com/mssql/server:latest

Administració:

Cal instalar Microsoft SQL Server Management Studio 19 o superior

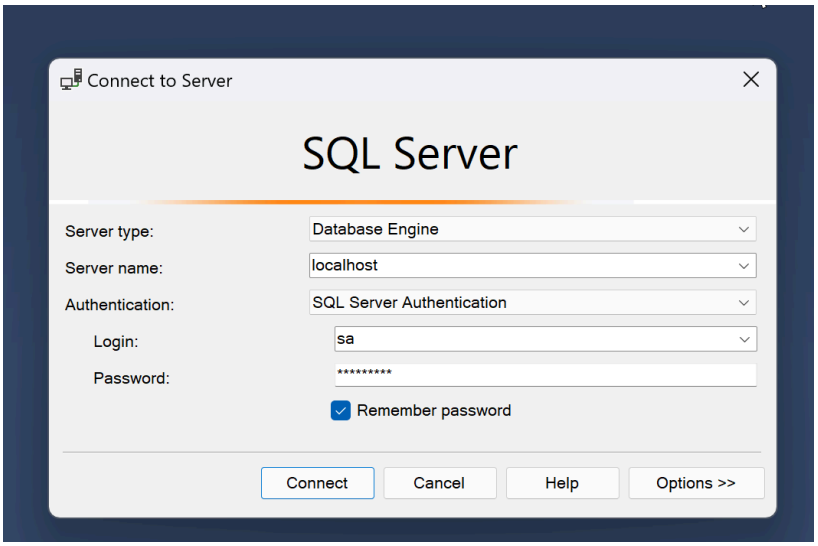

Figura 8: SQL Server - Administració - SMSS

Postgres

Docker:

docker run --name=Postgres -e "POSTGRES_PASSWORD=UOC_12345" -e "PGDATA=/var/lib/postgresql/data/pgdata" -p 5432:5432 -v PostgresVolume:/var/lib/postgresql/data -d postgres

docker run --name pgAdmin -p 5050:80 -e "PGADMIN_DEFAULT_EMAIL=driera@uoc.edu" -e "PGADMIN_DEFAULT_PASSWORD=UOC_12345" -d dpage/pgadmin4

Administració:

Per configurar pgAdmin dins del docker cal averiguar la IP interna amb la comanda

docker inspect -f "{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}" Postgres

A on retornarà un resultat semblant a: *172.17.0.2* i es podrà utilitzar a la configuració:

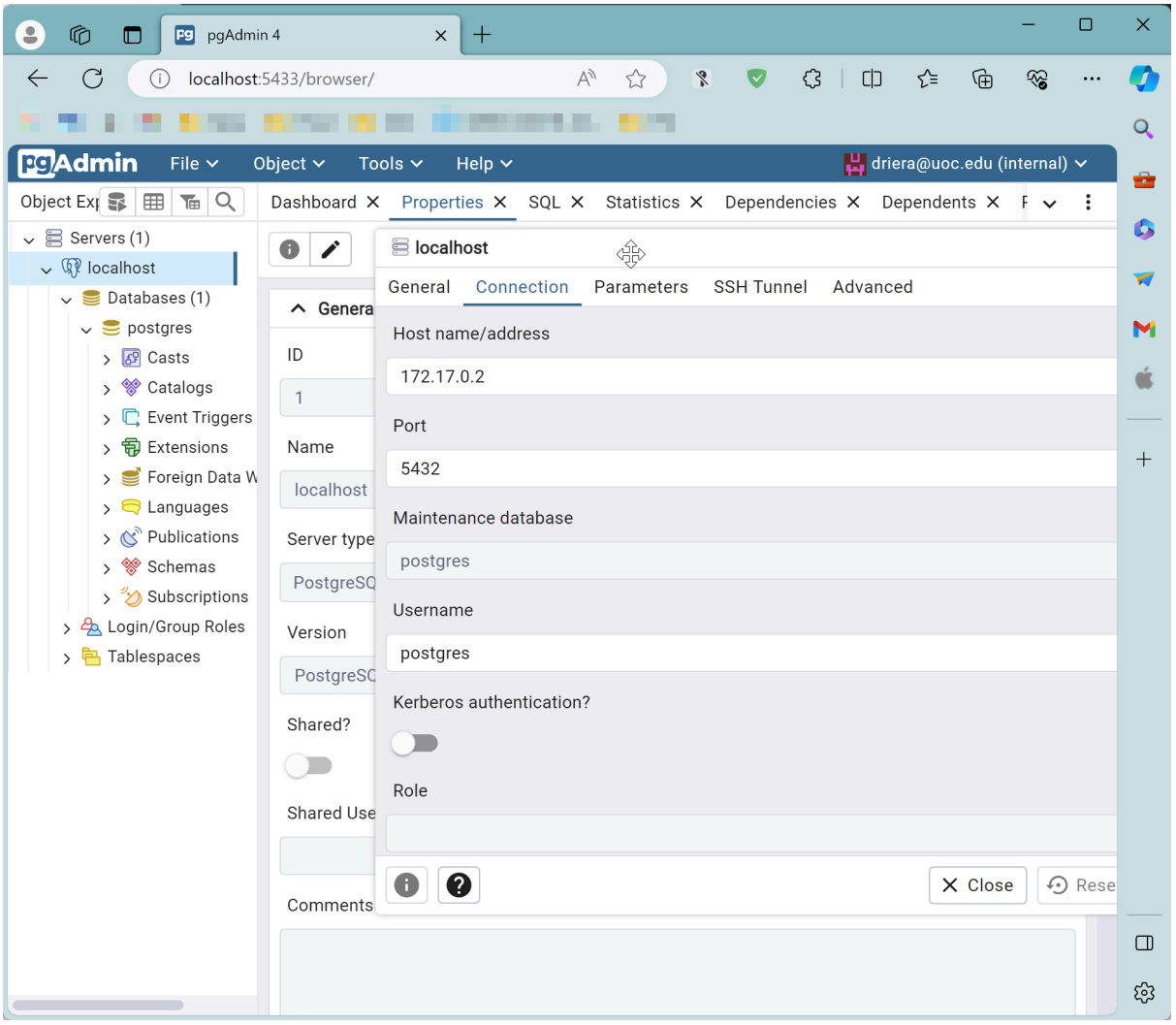

Figura 9: Postgres - Administració - PgAdmin

MariaDB

Docker:

docker run --name phpMyAdmin -d --link MariaDB:db -p 3307:80 -v phpMyAdminVolume:/etc/phpmyadmin/config.user.inc.php phpmyadmin

Administració:

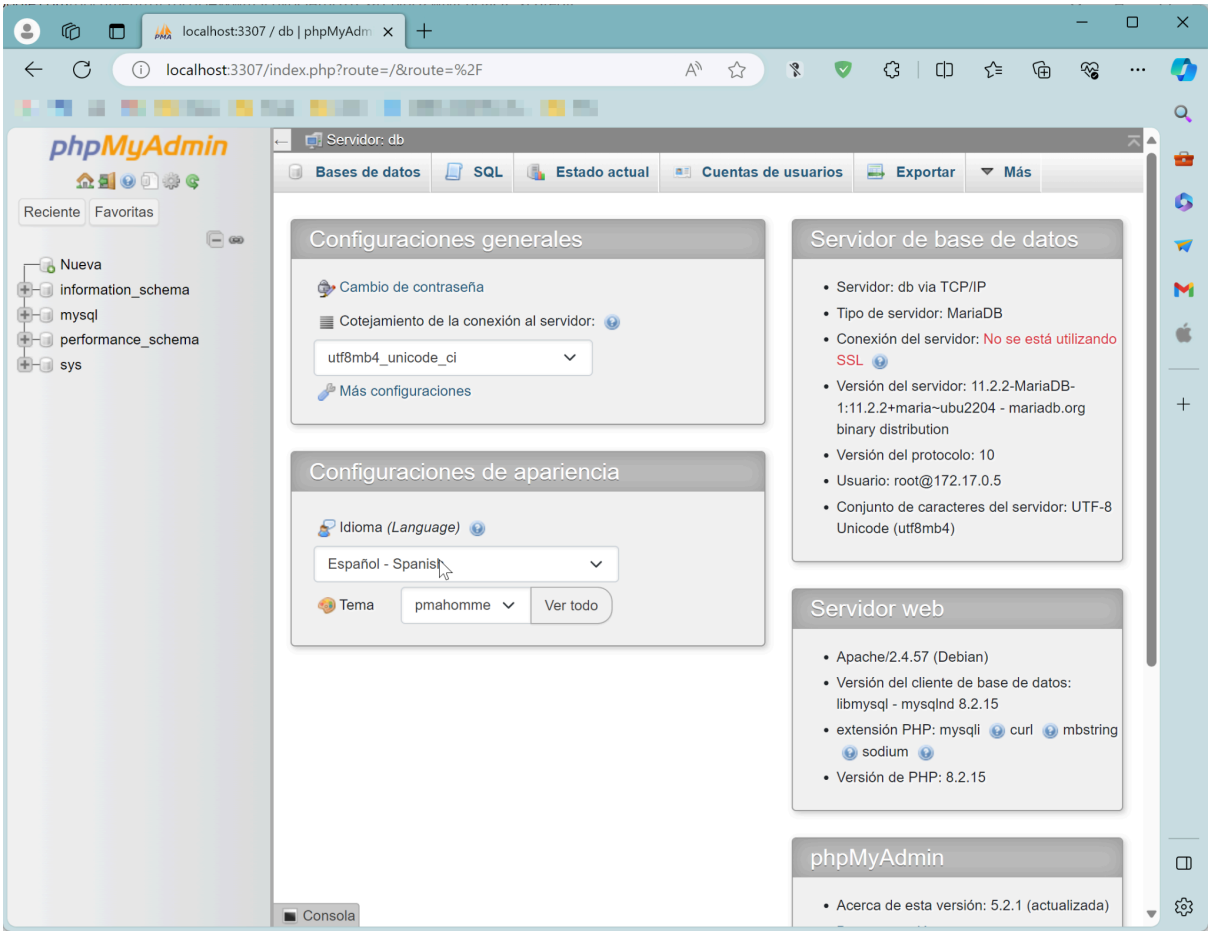

Figura 10: MySQL- Administració - phpMyAdmin

Resum de contenidors i volums

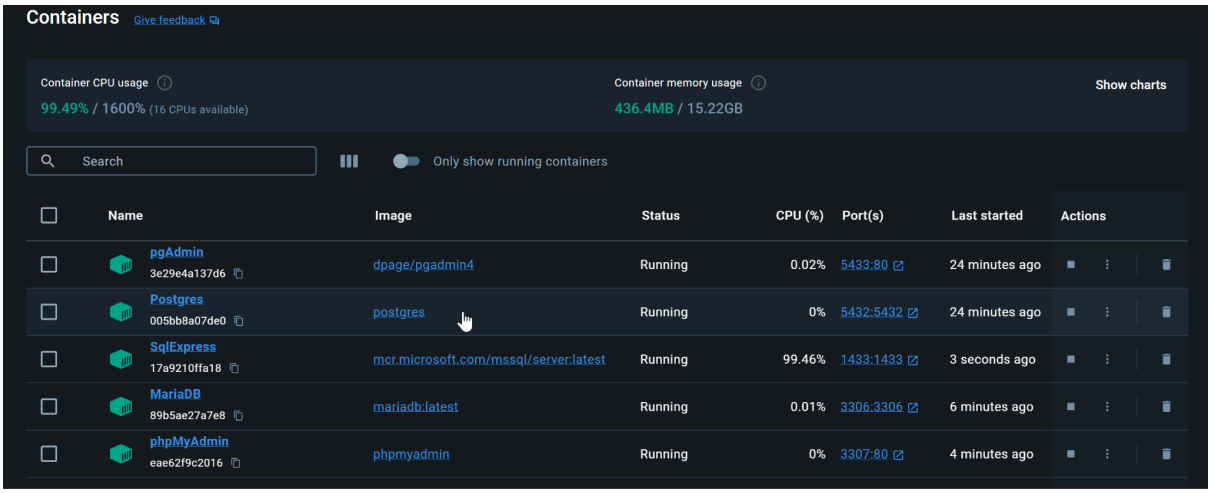

Figura 11: Docker - Resum contenidors

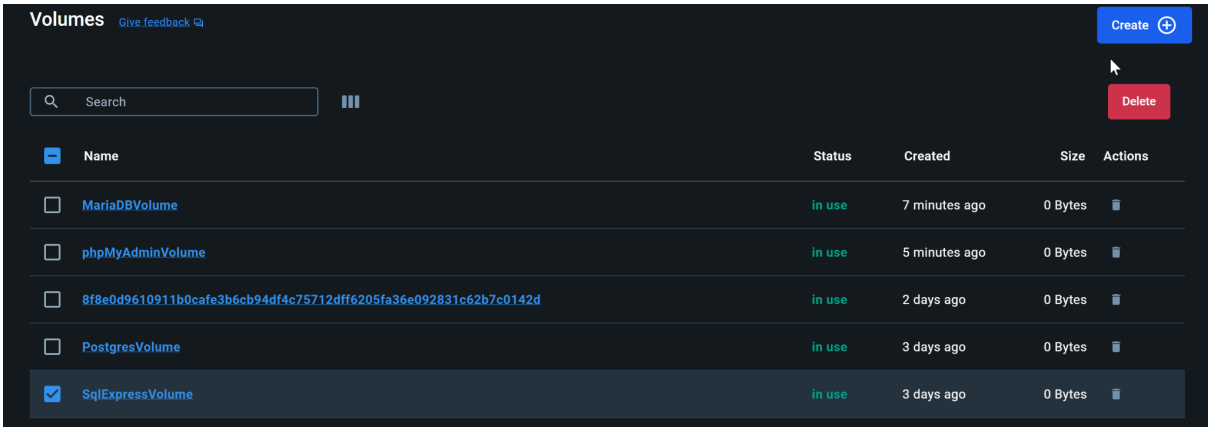

Figura 12: Docker - Resum volums

Annexes

- Annex 1 - Arguments d'execució i Exit Codes
- Annex 2 - Manual pel fitxer de configuració
- Annex 3 - Guia ràpida - Primers passos
- Annex 4 - Estructura taules
- Annex 5 - Origen de les dades de test
- Annex 6 - Ampliar EDTG
- Annex 7 - Resultats, anàlisi i comentaris
- Annex 8 - Conclusions i treballs futurs
- Annex 9 - Entrega i Documentació TFG

Glossari

API: (de l'anglès, application programming interface). És un tros de codi que permet a diferents aplicacions comunicar-se entre si i compartir informació i funcionalitats.

Agile Scrum: És un marc de treball de desenvolupament àgil i de bones pràctiques.

Aleatorietat: És el procés que permet que els resultats siguin imprevisibles i propiciats per l'atzar:

Auto: Utilitzat en aquest document com un camp de base de dades autonumerat que sols ser un nombre i no permet fer valors repetits.

BD: És un conjunt de dades segons una estructura coherent i accessibles des d'un o més programes o aplicacions.

Base de dades: Veure BD.

Blank: En aquest document es referència com un valor buit en el cas de camps tipus string o un valor zero en camps numèrics.

Boolean: Valor booleà que pot ser true o fals, i en algunes BD pot ser 0 o 1.

C#: (llegir com a C Sharp) Llenguatge de programació derivat del llenguatge C creat per Microsoft.

CSV: (de l'anglès, Comma Separated Values) són un tipus de document en format obert simple per a representar dades en forma de taula, en què les columnes se separen per comes (o punt i coma on la coma és el separador decimal: Catalunya, França, Itàlia...) i les files per salts de línia. Els camps que continguin una coma, un salt de línia o una cometa doble han de ser tancats entre cometes dobles.

Connection String: És una cadena de text amb una sintaxi definida per Microsoft per poder accedir a una base de dades indicant el tipus de servidor, autentificació i valors addicionals com idioma, paràmetres, etc.

Contenidor: En l'entorn de virtualització Docker el contenidor és la forma de virtualitzar el sistema operatiu i aplicacions per un ús des d'un sistema operatiu principal.

DEBUG: En el desenvolupament de programari l'acció de fer un seguiment de l'execució amb eines especialitzades per detectar errors i veure el funcionament. També utilitzat al registrar informació de l'aplicatiu en un fitxer de registre, per indicar que aquella línia d'informació es de tipus programació i ofereix dades que normalment a producció no es veurien i ajuden el programador a detectar errades.

DLL: (de l'anglès, Dynamic Link Library) és un format de fitxer de codi executable que és carregat a petició d'un programa per part del sistema operatiu. Aquesta denominació es refereix als sistemes operatius Windows i és l'extensió amb què s'identifiquen els fitxers, encara que el concepte existeix en pràcticament tots els sistemes operatius moderns.

EDTG: L'aplicació que es presenta en aquesta memòria.

Dades sintètiques: Són dades que es generen artificialment i, per tant, no provenen d'esdeveniments del món real. Tenen l'objectiu d'assemblar-se a un conjunt de dades autèntiques, però tenen una naturalesa totalment falsa.

Datatype: Tipus de dades.

Docker: És s un projecte de codi obert que automatitza el desplegament d'aplicacions dins de contenidors de programari, proporcionant així una capa addicional d'abstracció i automatització de virtualització d'aplicacions en diferents sistemes operatius.

Factoria: Utilitzat en aquest document com un patró per crear objectes d'una superclasse d'una forma controlada i automatitzada.

ID: Identificador. Utilitzat per fer referències a camps identificadors d'una taula.

Instància: En un Docker és l'execució d'un contenidor. En programació és la creació en execució d'una classe.

JSON: (de l'anglès, JavaScript Object Notation) És un estàndard obert basat en text dissenyat per a intercanvi de dades llegible per humans.

Linked tables: Veure Taules enllaçades.

LTS: (de l'anglès, Long Term Service) És que el programari tindrà un suport excepcionalment més llarg de l'habitual i permet fer previsions de posta en producció més controlades.

Null: Que no té valor, ni tan sols és un valor buit. En base de dades utilitzada per indicar que un camp encar no té valor.

Numeric: Referit com a qualsevol tipus de camp numèric a la base de dades. Tant INT, FLOAT, DECIMAL, etc.

OO: (de l'anglès, Object Oriented) és un paradigma de programació que intenta proporcionar un model de programació basat en objectes que contenen dades i procediments associats coneguts com a mètodes. Aquests objectes, que solen ser instàncies de classes, són un tipus abstracte de dades que encapsulen (amaguen) tant les dades com les funcions per a accedir-hi.

OutputDriver: Referenciat en aquest document com un dispositiu de sortida a on guardar els resultats calculats. Per exemple una base de dades MySQL o un fitxer CSV.

Plugin: Un connector és un programa informàtic, d'execució senzilla i opcional, que afegeix funcions al sistema o vincula dos programes o dues aplicacions independents perquè es complementin.

SLA: (de l'anglès, Service Level Agreement) Un Acord de Nivell de Servei és un contracte escrit entre un proveïdor de servei i el seu client en què es documenta el nivell acordat per a la qualitat del servei.

SOLID: (de l'anglès, Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion) Representa cinc principis bàsics de la programació orientada a objectes y el disseny. Són guies que aplicades en el seu conjunt permeten eliminar dissenys dolents o erronis.

SQL: (de l'anglès, Structured Query Language) És un llenguatge estàndard de comunicació amb bases de dades relacionals. És a dir, un llenguatge normalitzat que permet treballar amb la majoria de les bases de dades relacionals.

Set de dades: Conjunt de dades finites creades per a un objectiu.

String: Referenciat com un tipus cadena genèrica dins de les bases de dades i que pot ser un String, Varchar, etc.

TEXT: Utilitzat per indicar que la sortida és un format de fitxer com JSON, Texte o CSV.

Taules enllaçades: Utilitzat per anomenar el vincul o dependència entre dues o més taules.

Volum: Referenciat en l'entorn Docker com un espai en disc reservat per emmagatzemar dades d'una instància d'un contenidor.

Referències

[1] Microsoft. https://learn.microsoft.com. [Online].; 2024 [cited 2023 03 05]. Available from: https://learn.microsoft.com/es-es/visualstudio/releases/2022/system-requirements

[2] Microsoft. https://learn.microsoft.com. [Online].; 2021 [cited 2023 03 05]. Available from: https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/connection-string-syntax

[3] Microsoft. https://learn.microsoft.com. [Online].; 2024 [cited 2023 03 07]. Available from: https://learn.microsoft.com/es-es/dotnet/core/whats-new/dotnet-8/overview

[4] Microsoft. https://marketplace.visualstudio.com. [Online].; 2024 [cited 2023 03 07]. Available from: https://marketplace.visualstudio.com/items?itemName=SonarSource.SonarLintforVisualStudio2022

[5] Docker Desktop. https://www.docker.com. [Online].; 2024 [cited 2023 03 07]. Available from: https://www.docker.com/products/docker-desktop/

[6] SQL Server Docker. https://hub.docker.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://hub.docker.com/_/microsoft-mssql-server/

[7] PostgreSQL Docker. https://hub.docker.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://hub.docker.com/_/postgres

[8] MySQL Docker. https://hub.docker.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://hub.docker.com/_/mysql

[9] JSON Specification. https://jsonapi.org. [Online].; 2024 [cited 2023 03 08]. Available from: https://jsonapi.org/format/

[10] Wikipedia. https://es.wikipedia.org [Online].; 2023 [cited 2023 03 08]. Available from: https://es.wikipedia.org/wiki/Valores_separados_por_comas

[11] Jira. https://www.atlassian.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://www.atlassian.com/es/software/jira

[12] Generate Data. https://generatedata.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://generatedata.com/

[13] Mockaroo. https://mockaroo.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://mockaroo.com/

[14] Testing Data Generator. https://testingdatagenerator.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://testingdatagenerator.com/

[15] The Synthetic Data Vault. https://github.com. [Online].; 2024 [cited 2023 03 08]. Available from: https://github.com/sdv-dev/SDV?tab=readme-ov-file

[16] Mostly·AI. https://mostly.ai. [Online].; 2024 [cited 2023 03 08]. Available from: https://mostly.ai/

[17] Clearbox AI. https://app.clearbox.ai. [Online].; 2024 [cited 2023 03 08]. Available from: https://app.clearbox.ai/campaign/data-augmentation