POC: GEOMETRIA APLICADA A AGENDES

Definició

La prova de concepte intenta demostrar l'ús de la geometria per trobar forats buits en diferents agendes el més pròxims possibles en el temps.

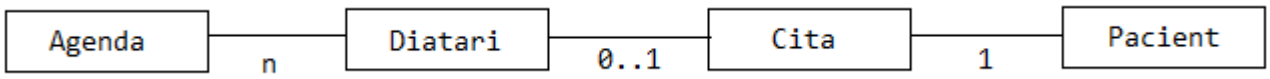
Com a exemple, imaginem que hem de donar cites a un pacient d'un hospital per fer un Preoperatori (proves que s'ha de realitzar abans d'una intervenció quirúrgica). Suposem, que un preoperatori consta d'una analítica, un electrocardiograma i una visita a l'anestesista. Per facilitar-li la feina al pacient, mirarem de donar-li les tres cites, el mateix dia i amb el temps d'estada al Hospital el més reduït possible..

Antecedents

Quan s'obre una agenda assistencial en un hospital, per generar hores disponibles (o forats) és juga amb quatre variables: AGENDA, Dia, Hora, i Temps estimat d'atenció. A cada forat li direm **Diatari** i quan s'assigna a un pacient, **Cita**.

POC

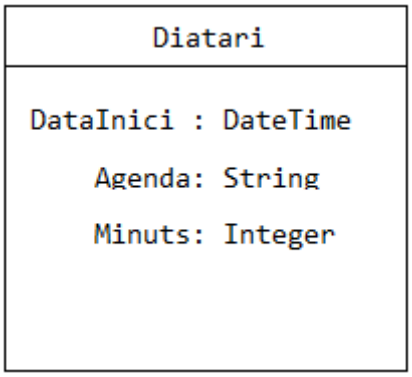
Mirarem de trobar els tres dietaris més propers entre ells d'un mateix dia per poder donar les tres Cites al Pacient.



"Una Agenda pot tenir n Diataris, un Diatari pot tenir una Cita i una Cita té un Pacient"

Com que no es tracta de fer un sistema d'agendes, sinó de fer un càlcul sobre els seus Diataris per poder assignar Cites, farem un model el més simplista possible. La resposta que esperem és: Tres Diataris el mateix dia (si és possible) el més pròxims possibles entre ells.

El model només tindrà una classe amb la definició inicial següent:



Exemple:

```
{ DataInici=2021/10/08 10:00, Agenda="LABO", Minuts=10}
{ DataInici=2021/11/27 10:00, Agenda="ELEC", Minuts=30}
{ DataInici=2021/12/31 15:30, Agenda="ANES", Minuts=20}
...
```

Ara imaginem que tenim n agendes d'analítiques, n d'electrocardiogrames, i n de anestèsia, en diferents dies i diferents horaris, i em de citar pacients per fer

preoperatoris. A cada pacient li hem de trobar cites en dietaris pròxims (el més pròxims possible), per tal de fer-li més agradable l'haver d'anar a l'hospital. Resumint, evitar que hi hagi d'anar tres dies diferents. Aparentment el problema sembla senzill, però en còmput no ho és tant.

Com comencem a resoldre? La primera pregunta que ens hem de fer és si l'ordre és important. Si l'ordre és important, hem de començar per establir aquest. L'ordre de recerca pot venir donat, per exemple, primer cal fer la Analítica, després l'Electrocardiograma i finalment anar a visitar al anestesista. O no, no ens ve donat un ordre preestablert i una de les coses que haurem de calcular es l'ordre pel qual comencem (el primer dietari que trobem d'una de les tres Agendes). Un cop trobat el primer Diatari, hem d'anar pel segon, que imaginem que trobem pel mateix dia, i després pel tercer, que el trobem per dos dies després. Que fem? Tornem a començar? Agafem un altre inici segons el criteri que haguem triat, cerquem el Diatari , i no el trobem pel mateix dia, tronem a començar i així contínuament fins arribar a un resultat (o no). Ara imaginem que aquestes recerques són concurrents des de diferents llocs de treball on és cerquen Preoperatoris, la "cosa" es complica. També hem de pensar en fer càlculs amb dates i hores en temps d'execució, ja sigui aprofitant el motor de la base de dades o de la tecnologia de la nostre aplicació amb les dades carregades en memòria. Provem-ho!! I si afegim criteris d'intervals de temps mínims i màxims entre Diataris, la complexitat temporal comença a ser una mica elevada.

Per aconseguir la fita, primer cal triar un motor de Base de dades que suporti geometria. En aquest cas he triat SQL Server 2019 per Linux corrent dins un contenidor Docker.

https://hub.docker.com/ /microsoft-mssql-server

Solució

Per dur a terme el POC, primer de tot cal imaginar-nos un espai 2D, representat per les coordenades X i Y. Partint del model inicial del Diatari , tenim DataInici i minuts. Amb Això podem calcular la data inici del diatari, la hora inici del diatari, la data final del diatari (és la mateixa que la d'inici) i la hora final del diatari. Per tant, separem dates i hores (molt important per indexar dates a les bases de dades). Tot seguit podem representar un diatari en un món 2D, posant per exemple les dates en les coordenada X i les hores en la coordenada Y.

POC: GEOMETRIA APLICADA A AGENDES

Si superposem el món 2D de cada Agenda un sobre l'altre, visualment veurem quins dies tenen diataris les tres Agendes i quines franges horàries utilitzen, de tal manera que podríem trobar tres diataris per un mateix dia que no es solapin. Ara bé, si tenim moltes agendes, i molts diataris en moltes dates, la cosa de manera visual es complica.

Tractant-se d'un model 2D, podem fer una transformació de les dates i les hores en punts geomètrics. Com? La veritat que podem fer-ho de diferents maneres. La que he triat jo és la següent:

Per representar una data en concret:

CoordenadaX = new DateTime(2021, 10, 10, 0, 0, 0).Ticks / 10000000

Per representar les hores:

1minut = 60

1hora = 60 * 60 = 3600

"08:30" = (8 * 1hora) + (30*1minut)

CoordenadaY = (8 * 1hora) + (30*1minut)

La hora final és fàcil de calcular a partir dels minuts del model inicial.

Desenvolupament

El desenvolupament està fet en Net Core 3.1.

El primer que faré és adaptar el model inicial a un model més pràctic per poder dur a terme la POC.

```
public class Diatari
{
    // Identificador entitat
    public Guid Id { get; set; }
    // Codi Agenda
    public string Agenda { get; set; }
    // Data
    public DateTime DataInici { get; set; }
    // Hora Inici
    public TimeSpan HoraInici { get; set; }
    // Hora Fi
    public TimeSpan HoraFi { get; set; }
    // Minuts
    public int Minuts { get; set; }
    // Punt Inicial
    public Point PuntInici { get; set; }
    // Punt Finalff
    public Point PuntFi { get; set; }

}
```

Tot seguit he creat la BD a partir del model, i he creat registres.

Tot això ho podem veure a la solució. Aquesta compte amb quatre projectes.

PWP.Domain: Aquí tenim definides les entitats del domini. En aquest cas i per fer-ho el més senzill possible només hi trobem la entitat Diatari.

PWP.Infra: Aquí tenim definit el context de dades que depèn de la infraestructura triada, en aquest cas SQL Server.

DbMigrations: Projecte per migrar el Domini i el Context de dades a la Infraestructura.

PWPConsole: Projecte per carregar la BD amb diataris, i testejar si la teoria presentada en el POC funciona.

Exemples

Dins el Projecte **PWPConsole** trobem tres mètodes per provar la idea que presenta la POC.

Test1_FirstPlanner

El primer Test intenta trobar tres diataris seguin l'odre Laboratori, Electrocardiograma, Anestesista, amb un interval mínim de 10 minuts entre ells, i ,menys de 30 minuts entre el Diatari de Laboratori i el de Electrocardiograma, i menys de 60 minuts entre el diatari de Electrocardiograma i el de Anestesia.

Test2_SecondPlanner_NotOrdened

El mateix que el primer test, sense tenir en compte l'ordre de les Agendes i amb intervals diferents.

Test3_ThirdPlanner_MoreThanOne

El mateix que el primer test, però tornant les primeres 20 opcions a triar que compleixen els criteris.

Us animo a fer més testos, i també a fer el mateix sense geometria. Si ho poseu en producció cal pensar en els indexes geomètrics de la BD.