UNIVERSIDAD DE OVIEDO
ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN
(EPI)

Copilot inteligente para consultas LINQ/SQL

Autor:
Puga Lojo, Francisco Gabriel

Tutor (Mecalux):
Moldón Redondo, Daniel

Tutor (EPI):
Costa Cortez, Nahuel Alejandro

Memoria entregada en cumplimiento con los requisitos indicados por la asignatura Prácticas de Empresa del grado Ingeniería Informática en Tecnologías de la Información

28 de mayo de 2024

UNIVERSIDAD DE OVIEDO

Resumen

Escuela Politécnica de Ingeniería de Gijón (EPI)

Ingeniería Informática en Tecnologías de la Información

Copilot inteligente para consultas LINQ/SQL

por Puga Lojo, Francisco Gabriel

El manejo de bases de datos es fundamental para la gestión de información en las empresas hoy en día. En un mundo donde la información es clave, poder procesar datos importantes es vital para mantener y hacer crecer una empresa.

Sin embargo, trabajar con bases de datos puede ser complicado y requiere una formación específica que muchas personas fuera del ámbito informatico no tienen ni tiempo para aprender.

Mecalux es una de las compañías líderes en tecnología intralogística a nivel mundial. Es puntera en automatización de almacenes y desarrollo de software. En estas prácticas, he contribuido a desarrollar un asistente que ayuda a generar código LINQ SQL y que también explica las consultas generadas.

El objetivo de este asistente es facilitar el trabajo de los empleados, permitiéndoles crear y entender consultas SQL sin necesidad de conocimientos técnicos profundos, y simplificando las tareas para quienes sí los tienen. Esto agiliza los procesos internos y mejora la eficiencia en la toma de decisiones, lo cual es crucial en el entorno empresarial actual.

Índice general

Resumen I

1. Introducción: *El alumno y la empresa* 1

2. Data Analytics: *Innovación y solución de problemas* 2

2.1. Contexto 2

2.2. Metodología de Trabajo 2

2.2.1. Dinámica de trabajo durante las prácticas 3

3. MSSCopilot: *Generación de Código Automática* 4

3.1. Objetivo de las prácticas 4

3.2. Cosas 4

Lista de Abreviaturas

MSS	**M**ecalux **S**oftware **S**olutions
IT	**I**nformation **T**echnlogies
I+D+I	**I**nvestigación, **D**esarrollo e **I**nnovación

Capítulo 1

Introducción

El alumno y la empresa

Mecalux es una empresa reconocida internacionalmente en el sector de soluciones de almacenamiento. Desde su fundación en 1966, ha desarrollado un amplio portafolio que abarca una variedad de sistemas de almacenamiento, como estanterías, almacenes automatizados y soluciones de software para la gestión de almacenes.

Con una presencia global, Mecalux opera en numerosos países, gestionando un gran volumen de operaciones y personal especializado. Las prácticas fueron realizadas de manera presencial en las **oficinas de MSS en Gijón.** [1]

NOTA

Aunque las prácticas fueron presenciales, Mecalux tiene una metodología de teletrabajo muy arraigada de la cual muchos empleados de IT siguen beneficiándose.

Es notable cómo esta flexibilidad está bien integrada en su rutina diaria, y parte de la comunicación con el equipo ha sido de de esta manera.

Durante estas prácticas de empresa, tuve la oportunidad de formar parte del equipo de Data Analytics en la división de MSS. En el próximo capítulo se detallarán las actividades llevadas a cabo por este equipo, junto con sus metas y objetivos.

[1] Mecalux Software Solutions es la división de Mecalux dedicada enteramente al desarrollo de software para almacenes y logística.

Capítulo 2

Data Analytics

Innovación y solución de problemas

2.1. Contexto

He tenido la fortuna de trabajar dentro del departamento de Data Analytics, el cual se encarga de recopilar, limpiar e interpretar conjuntos de datos para responder a preguntas o resolver problemas. Es un equipo muy multidisciplinar, y me sorprendió positivamente que, aunque la mayoría de sus miembros no provienen de estudios de Informática (sino de áreas como Física, Matemáticas, etc.), todos poseen una gran capacidad para reflexionar y resolver problemas de manera lógica, metódica y, sobre todo, en equipo.

Además de su labor como analistas de datos, este equipo también cumple la función de resolver inconvenientes que puedan surgir en otros departamentos y se encarga de investigar y poner a prueba nuevas soluciones antes de su implementación en el entorno de producción. Se podría decir que realizan funciones similares a las de I+D+I, ya que investigan, desarrollan y prueban nuevas soluciones para mejorar los procesos y resolver problemas dentro de la empresa.

Durante mis prácticas, mi labor ha sido una combinación de investigación y desarrollo de software, lo cual ha sido posible gracias a la naturaleza y dinámica de este equipo.

2.2. Metodología de Trabajo

Como se mencionó en la introducción, parte del equipo realiza teletrabajo, algunos de manera ocasional y otros de forma regular. De hecho, hay un integrante que ni siquiera reside en Asturias. El equipo realiza reuniones diarias (dailys), a las cuales tuve el placer de asistir en la última etapa de mis prácticas. En estas reuniones, los miembros explican el progreso y las tareas realizadas el día anterior. Estas reuniones se llevan a cabo en salas especialmente equipadas en el edificio, las cuales cuentan con cámaras y proyectores para que los empleados que están teletrabajando puedan participar activamente en las reuniones.

2.2.1. Dinámica de trabajo durante las prácticas

El departamento estaba enfocado en sus proyectos, por lo que mi equipo de trabajo habitual se redujo a otro alumno de prácticas de la carrera de Ingeniería Informática del Software, y nuestro tutor, perteneciente a Data Analytics, que se encargaba de que fuésemos por el buen camino y con el que hacíamos también nuestras propias reuniones diarias para ver el progreso de nuestro trabajo.

Mi compañero y yo teníamos horarios diferentes, por lo que coincidíamos en las oficinas durante un tiempo limitado, lo que conllevó a la necesidad de coordinarnos lo mejor posible y así poder tener los objetivos claros y aprovechar al máximo el tiempo que teníamos juntos para desarrollar, estructurar nuestras tareas y debatir los problemas a los que nos enfrentaríamos.

Al entrar antes a la oficina, me encargaba de hablar con el tutor para ver si había nuevas opciones que explorar o investigar, por donde seguir si habíamos conseguido los resultados esperados, o en caso contrario comunicar los inconvenientes, y coordinar la planificación para el día.

De esta manera, cuando mi compañero llegaba, era todo más fácil para los tres, el tutor no necesitaba volver a explicarlo todo y yo podía comunicarle de forma efectiva que teníamos que hacer, las novedades, y como nos podríamos distribuir el trabajo.

Por otra parte, cuando yo me marchaba mi compañero seguía desarrollando y trabajando en el proyecto, por lo que me comunicaba sus avances por la plataforma de Teams para que cuando llegase al día siguiente supiese de manera más rápida que era lo que se había hecho.

Nuestra comunicación fue en todo momento muy buena, nos sentábamos en mesas próximas por lo que nos acercábamos a hablar sobre como realizar el trabajo frecuentemente, y cuando no queríamos molestar utilizábamos el chat de Teams.

Cabe resaltar que aunque no trabajase con todos los integrantes del departamento de Data Analytics directamente al no pertenecer al proyecto en sí, muchos de ellos me ayudaron a lo largo de las prácticas y hubo una comunicación cercana y efectiva independientemente de ello.

Capítulo 3

MSSCopilot

Generación de Código Automática

3.1. Objetivo de las prácticas

El objetivo principal de las prácticas fue desarrollar una herramienta que permitiera a los usuarios generar código de manera automática. Esta herramienta, denominada MSSCopilot, debía ser capaz de generar código en C# / Linq a partir de la base de código fuente existente en la compañía.

EJEMPLO

Un empleado le dice al MSSCopilot que desea obtener una lista con todos los Warehouses, el Copilot respondería con un código como el siguiente:

```
1     using (var context = new MyContext())
2     {
3         var result = context.Warehouse.Select(x => x);
4     }
```

Durante el desarrollo del programa se elaboraron más funcionalidades de las previstas inicialmente como es la explicación de código, y otras más que se explicarán con mayor profundidad posteriormente.

El estado del proyecto al comenzar las prácticas estaba en una etapa inicial, lo que permitió un entendimiento más profundo del mismo y facilitó la integración de nuevas ideas y funcionalidades desde el principio.

3.2. Cosas

Definir las tecnologías utilizadas en la realización del MSSCopilot resulta complicado porque como se ha comentado en la sección 2.1, aunque se trate de desarrollar un producto de software, ha sido un proyecto en el que, por la naturaleza de la IA, que sigue siendo un sector de tecnologías emergentes, las directrices a seguir no son tan claras. Debido a esto, gran parte del proyecto ha consistido en un proceso de investigación en el que se han utilizado tecnologías y técnicas que, con el tiempo, se han descartado en favor de otras opciones mejores a medida que se identificaban.

Los LLM son un tipo de modelo de inteligencia artificial diseñado para comprender y generar texto, como es el famoso GPT, para este proyecto se ha utilizado Ollama.

En la descripción de las prácticas una de las tareas fundamentales era diseñar y entrenar el sistema a partir de la base de código fuente existente en la compañía. Para entrenar un LLM con información nueva hay dos opciones:

- Ajustar el modelo preentrenado utilizando un cojunto de datos más pequeño y específico, proceso conocido como **Fine-tuning.**
- Convertir palabras, frases o textos completos en vectores numéricos. Estos **Embeddings** capturan el significado y las relaciones semánticas del texto. Se pueden convertir las preguntas del usuario en embeddings y compararlos con los embeddings de la información que tenemos almacenada en una BDD, y si ambos tienen x similitud darle esa información al LLM en tiempo real para que responda con nuestra información.

En resumen, en la primera opción ponemos al modelo a estudiar los conocimientos nuevos generando así un nuevo modelo, y en la segunda opción es como si el modelo para cada pregunta, consultase en un diccionario/libro con respuestas, buscando la que más se parezca a la pregunta del usuario.

Inicialmente se optó por utilizar Embeddings, usando SQLite para la BDD y C# por compatibilidad con toda la infraestructura de la empresa, permitiendo el poder integrar a futuro esta utilidad de asistente de código a todos los servicios que ofrecen.

La base de datos contenía previamente querys correctamente escritas en C# junto con su respectiva descripción, y el embedding asociado, esto se descartó posteriormente ya que no era viable para la empresa ponerse a generar querys con descripciones para cada casuística, y en versiones posteriores se optaría por pasarle el contenido de las tablas.

Dicha comparación que se realiza entre Embeddings de la pregunta del usuario y la BDD se tuvo que implementar en código, se implementó tanto la similitud coseno como la euclídea:

Para calcular la similitud coseno entre dos vectores de embeddings se han de realizar dos pasos:

Sean **A** y **B** los vectores con embeddings cuya similitud se desea calcular. El primer paso consiste en obtener el valor del coseno del ángulo que forman ambos vectores.

Sea θ el ángulo que forman los dos vectores. Su coseno se puede calcular de la siguiente manera:

$$cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Dado que se da que $cos(\theta) \in [-1, 1]$, el segundo paso ha de ser que el resultado pase a encontrarse en el rango $[0, 1]$, con el fin de simplificar el cambio de un algoritmo de similitud a otro. De esta manera, el valor de la similitud entre los vectores **A** y **B** sería el siguiente:

$$S_C(A, B) = cos(\theta) \cdot 0.5 + 0.5$$

Sean **A** y **B** los vectores con embeddings cuya distancia se desea calcular. Calcular la distancia entre ambos consiste en calcular la norma euclídea de su diferencia:

$$d(A, B) = |A - B\| = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + ... + (A_n - B_n)^2}$$

En este caso, no es necesario modificar el rango, ya que se da que $d(A, B) \in [0, \infty)$. No obstante, es necesario realizar un segundo paso. El valor que se ha obtenido no es la similitud entre los vectores, sino su distancia. Por tanto, realizar una comparación con el valor obtenido directamente resultaría en resultados incorrectos, ya que a mayor valor de $d(A, B)$, más distintos son los vectores.

Por tanto, para calcular la similitud entre los vectores en base a la distancia euclídea, es necesario multiplicar el resultado por -1. Así, la similitud entre los vectores se calcularía de la siguiente manera:

$$S_E(A, B) = -d(A, B)$$