



DelphiDay  
italian conference

# LLM per potenziare le nostre App

RAG, Text-to-SQL e strutturazione dati descrittivi



# LUCA MINUTI



[lucominuti.it](https://lucominuti.it)



[luca.minuti@gmail.com](mailto:luca.minuti@gmail.com)



[dev.to/lminuti](https://dev.to/lminuti)



[github.com/lminuti](https://github.com/lminuti)



[www.linkedin.com/in/lucominuti](https://www.linkedin.com/in/lucominuti)



## DelphiDay

italian conference

19-20 Giugno 2025  
Piacenza



wintech  
italia



# OPEN-SOURCE PROJECTS

[github.com/Iminuti](https://github.com/Iminuti)

## WiRL

[github.com/delphi-blocks/WiRL](https://github.com/delphi-blocks/WiRL)

## Delphi SAML

[github.com/EtheaDev/Delphi-SAML](https://github.com/EtheaDev/Delphi-SAML)

## OpenSSL

[github.com/Iminuti/Delphi-OpenSSL](https://github.com/Iminuti/Delphi-OpenSSL)



19-20 Giugno 2024  
Piacenza





# AGENDA

---

1. Cenni storici
2. Cosa sono gli LLM
3. SDK per Delphi
4. Creare un chat bot
5. Modi alternativi per usare gli LLM



# INTRODUZIONE

- Il termine “Intelligenza Artificiale” è stato coniato da **John McCarthy** (creatore di LISP) nel 1956
- Ma già nel 1950 **Alan Turing** nell’articolo “Computing machinery and intelligence” proponeva il famoso **test di Turing** per valutare l’intelligenza di una macchina



# INTRODUZIONE

- Anni 50-60: entusiasmo ma poco uso pratico
- Anni 70-80: “Inverno delle AI”, sistemi esperti
- Anni ‘90: Deep Blue, anti-spam, dati anziché regole
- Anni 2000-2010: Motori di ricerca, raccomandazioni
- Anni 2010-2020: traduzioni, riconoscimento facciale, assistenti vocali, guida autonoma



# INTRODUZIONE

- 2020-Oggi:
  - Modelli generativi (come GPT, DALL·E, Midjourney, Claude): capaci di scrivere testi, generare immagini, video, codice
  - ChatGPT porta l'AI conversazionale al grande pubblico
  - Integrazione in lavoro, creatività, sanità, educazione, marketing, automazione, supporto medico, assistenza clienti, progettazione, ricerca scientifica



# TRANSFORMER

- Paper "Attention Is All You Need": Introduce l'architettura Transformer (Google Brain 2017)
- Meccanismo di Self-Attention: Ogni token può "guardare" tutti gli altri simultaneamente
- Parallelizzazione: Training molto più veloce
- Encoder-Decoder: Struttura modulare e flessibile





# TOKEN

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🍌🍌🍌🍌

Sequences of characters commonly found next to each other may be grouped together:  
1234567890

Text

Token IDs

<https://platform.openai.com/tokenizer>



# ADDESTRAMENTO

- **Pre-training:** Fase iniziale in cui il modello viene "allenato" su grandi quantità di testo o dati
- **Fine-tuning:** il modello viene "affinato" su compiti specifici con dataset annotati da esseri umani (es. domande/risposte, classificazioni, dialoghi, ecc.).
- **Reinforcement Learning (RLHF):** addestrato usando un meccanismo di rinforzo guidato da valutazioni umane



# GLOSSARIO

- **Token:** Unità base di input/output per un modello di linguaggio. Può essere una parola, una sillaba o anche solo una lettera/punteggiatura.
- **Parametri:** I "pesi" interni del modello (milioni o miliardi)
- **Embedding:** Rappresentazione numerica (vettore) di parole, frasi o immagini
- **Multimodalità:** Capacità di elaborare più tipi di dati



DELPHI E LLM

1



# INTEGRAZIONE CON DELPHI

- Tutti i principali vendor (OpenAI, Anthropic, Google, Mistral, ...) distribuiscono un SDK per diversi linguaggi
- Ma... sono tutte API ReST!
- Sì trovano diverse librerie sia su GitHub che su GetIt





# DelphiOpenAI

---

```
var Completions := OpenAI.Completion.Create(  
  procedure(Params: TCompletionParams)  
  begin  
    Params.Prompt(MemoPrompt.Text);  
    Params.MaxTokens(2048);  
  end);  
try  
  for var Choice in Completions.Choices do  
    MemoChat.Lines.Add(Choice.Index.ToString + ' ' + Choice.Text);  
finally  
  Completions.Free;  
end;
```

demo time





# CHAT TO SQL

# 2



# CHAT TO SQL

- Gli LLM sono in grado di generare codice a partire da un prompt
- Anche SQL...
- E se gli diamo in pasto lo schema di una base dati?



# TEMPLATE CHAT TO SQL

---

Sei un **esperto di SQL**. In particolare del dialetto SQL di **FirebirdSQL 3.0**. Devi rispondere alla seguente domanda con una query SQL (la risposta deve contenere **solo la SELECT** e non deve essere formattata in markdown, solo una semplice query). Dopo la domanda troverai le istruzioni DDL del database da usare per scrivere la query. La domanda è:

%QUESTION%

Segue lo schema del database:

%SCHEMA%



demo time





BLOB TO TAB

3



# BLOB TO TABLE

- Gli LLM sono bravi a trovare pattern all'interno del testo, schematizzare dei dati, fare riassunti, ecc.
- Perché non usare le loro abilità per trasformare testo non strutturato in dati tabulari?



# TEMPLATE BLOB TO TABLE

---

I dati che seguono sono diverse righe di descrizioni di movimenti bancari.  
**Cerca di trovare i dati comuni e crea un array json** con un elemento per ogni movimento. Ogni riga dell'array json deve avere gli stessi campi che deciderai in base ai dati presenti all'intero della descrizione dei movimenti.  
Se vedi delle date formattale come dd/mm/yyyy.  
E' importante che nella risposta ci sia solo il json senza formattazione markdown, senza in introduzione testuale ecc.

Seguono i movimenti:  
%MOVIMENTI%

demo time







# TOOLS

# 4

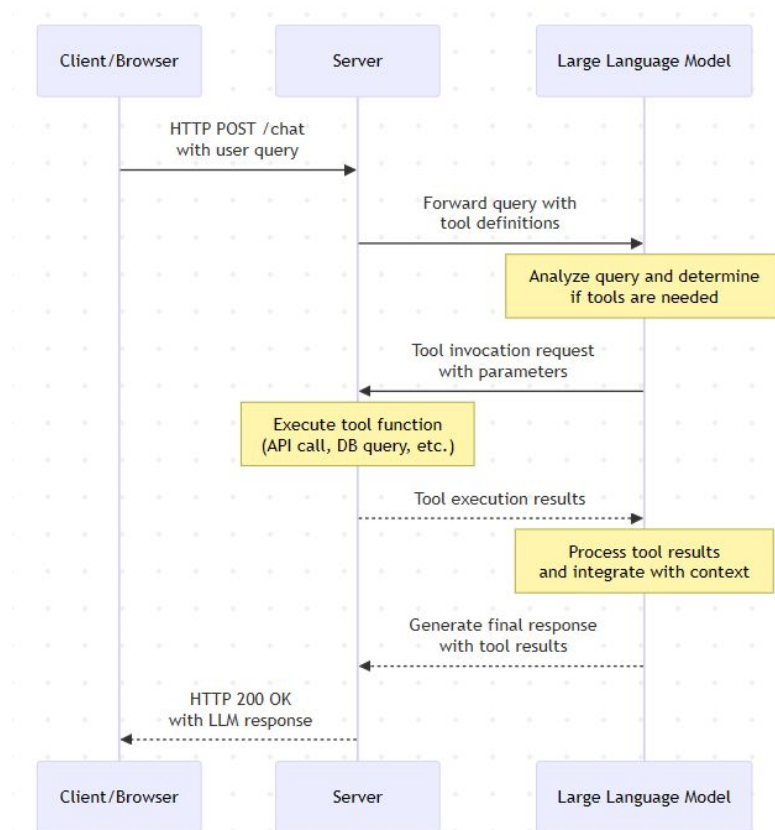


# TOOLS

- La maggior parte dei modelli permette di “allegare alla richiesta” una serie di tool
- Se il modello ritiene necessario l’uso dei tool per fornire una risposta migliore, invece di dare direttamente la risposta richiederà l’uso di un tool
- Solo dopo aver ottenuto i dati necessari produrrà la risposta



# TOOLS





# TOOLS

---

```
TMyTools = class (TForm)
public

    function EventStartDate (const EventName: string): string;

    function SendMail (const TargetEmail, Subject, Body: string): string;
end;
```



# TOOLS

---

```
TMyTools = class(TForm)
public
  [ChatTool('Return the start date of an event in the ISO format yyyy-mm-dd' )]
  function EventStartDate (
    [ChatTool('The name of the event (without spaces or special characters)' )]
    const EventName: string
  ): string;

  [ChatTool('Send an e-mail to a specific email address' )]
  function SendMail (
    [ChatTool('The email address' )]
    const TargetEmail: string;
    [ChatTool('The subject of the email' )]
    const Subject: string;
    [ChatTool('The body of the email (max 200 characters)' )]
    const Body: string
  ): string;
end;
```



# TOOLS

---

```
tools: [{  
  "type": "function",  
  "function": {  
    "name": "EventStartDate",  
    "description": "Return the start date of an event in the ISO format yyyy-mm-dd" ,  
    "parameters": {  
      "type": "object",  
      "properties": {  
        "EventName": {  
          "type": "string",  
          "description": "The name of the event..."  
        }  
      },  
      "required": [ "EventName" ]  
    }  
  },  
  ...  
}]
```

demo time





RAG

5



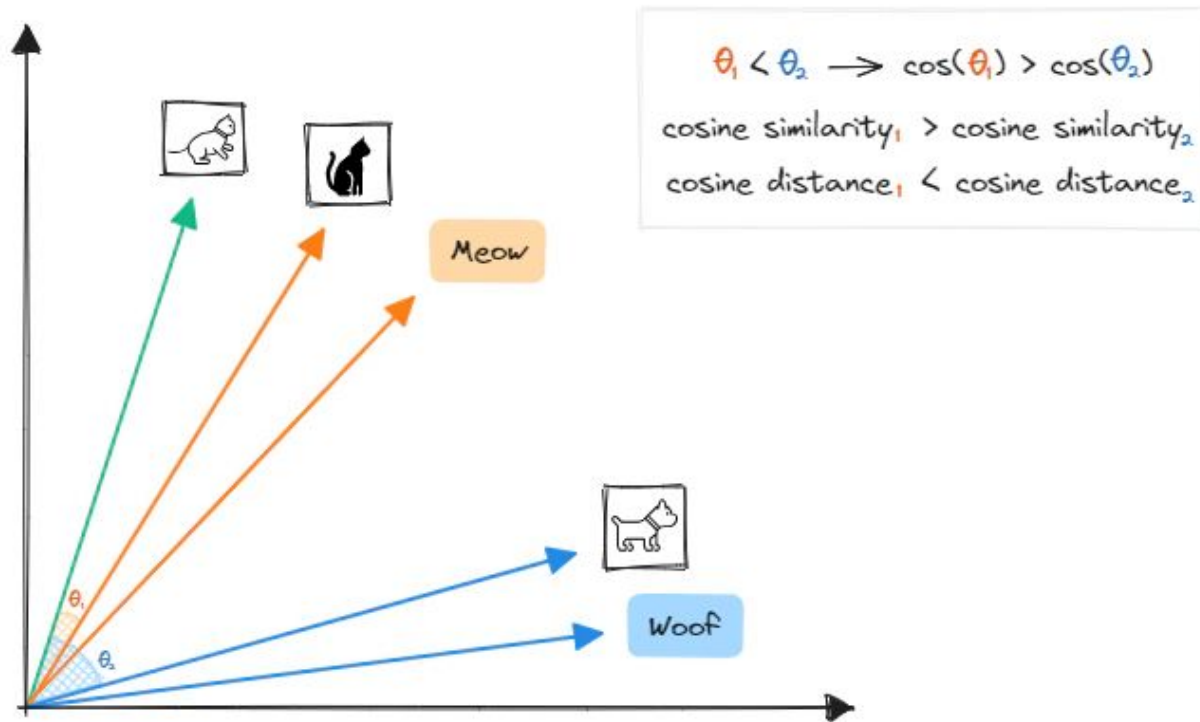


# RAG

- Allucinazioni e informazioni non pubbliche
- Inserendo nel contesto i dati che ci servono la risposta diventa molto più precisa
- Ma come fare se il contesto è molto grande?
- Possiamo dividere questa base di conoscenza in chunk e calcolare gli embedding in modo da aggiungere al contesto solo i chunk inerenti la domanda



# RAG



demo time





MCP

6

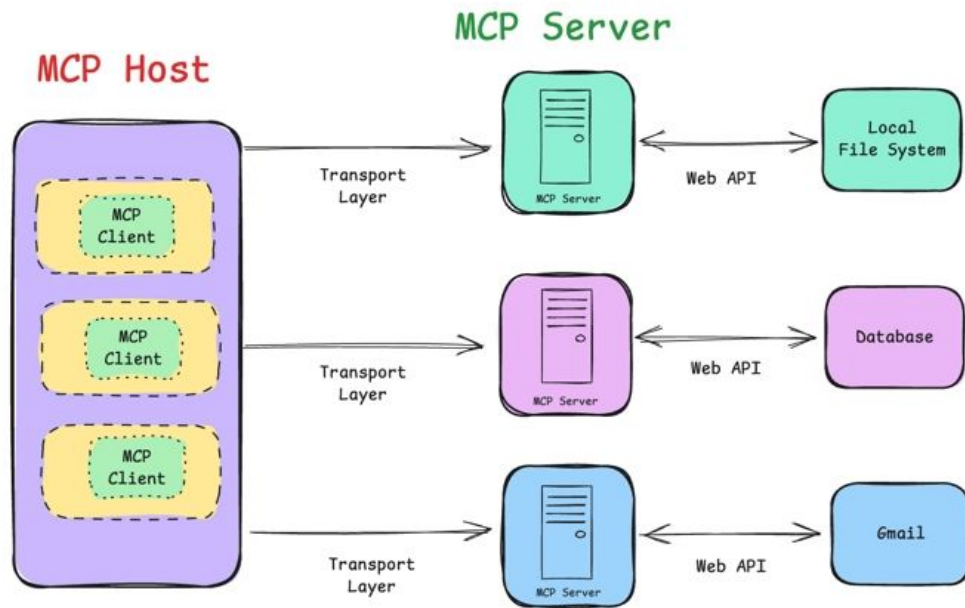


# COS'È MCP

- MCP sono un passo in più rispetto ai tools
- Permettono registrare delle API fornite da diversi vendor
- LLM potrà poi chiamarle autonomamente qualora ne avesse bisogno



# COS'È MCP



## MCP Host:

L'applicazione che fa uso dell'LLM

## MCP Client:

La parte dell'host che si occupa della comunicazione

## MCP Server:

Il servizio che esegue l'operazione

# Applicazioni AI-Driven con Delphi



Prossima data: **8 Luglio 2025 (mattina)**





THANK YOU