

Python + IA

Crea magia con pocas
líneas de código



¡Hola!

Somos **Daniel Delgado** y **Juan Carlos Sedano**.

Líderes de la comunidad **Pythonistas GDL**.

Más información en: pythonistas-gdl.org



¿Qué es la IA hoy?

- Mucho de lo que llamamos "IA" hoy en día son **LLMs** (Large Language Models).
- Los LLMs son solo una parte de un tema mucho más grande.
- La "magia" de la IA moderna es una combinación de modelos, datos y técnicas.

¿Qué es un LLM?

- LLM = Large Language Model
- Son modelos entrenados con grandes cantidades de texto.
- Pueden generar texto, responder preguntas, traducir, resumir, etc.

Tokens, contexto y embeddings

- **Token:** Fragmento de texto (palabra, parte de palabra o símbolo).
- **Contexto:** Cantidad de texto que el modelo puede "recordar" en una sola interacción.
- **Embedding:** Representación numérica de palabras o frases.

¿Cómo se entrenan los LLMs?

- Se alimentan con grandes cantidades de texto (libros, artículos, internet).
- Aprenden patrones, relaciones y significado de las palabras.
- El entrenamiento requiere mucho poder de cómputo y datos.

¿Por qué Python?

- Python es el lenguaje más usado en ciencia de datos e IA.
- Tiene muchas librerías:
 - **NumPy, pandas, scikit-learn, TensorFlow, PyTorch, LangChain, etc.**
- Comunidad activa y muchos recursos gratuitos.

Ejemplo: Agente IA con Python y LangChain

demo.pythonistas-gdl.org

¿Qué son las "tools" en un agente IA?

- Las "tools" (herramientas) permiten que el modelo haga más que solo conversar.
- Son funciones de Python que el agente puede usar para:
 - Consultar información real (por ejemplo, citas en la base de datos).
 - Crear, actualizar o eliminar datos.
 - Calcular fechas, validar reglas, etc.
- Así, el modelo puede interactuar con el mundo real y no solo "imaginar" respuestas.

Ejemplo de tools en nuestro agente

```
@tool
def list_appointments() -> str:
    """Lista todas las citas con su fecha, hora, duración, título y lugar."""
    # ...consulta la base de datos...
```

```
@tool
def create_appointment(date, time, duration, title, place) -> str:
    """Crea una cita SÓLO si no se empalma con otra en la MISMA FECHA."""
    # ...verifica empalmes y guarda la cita...
```

- Cada función decorada con `@tool` es una capacidad nueva para el agente.

Más herramientas útiles

```
@tool
def get_today() -> str:
    """Devuelve la fecha actual en formato YYYY-MM-DD."""
    # ...usa datetime...

@tool
def get_relative_date(days_from_today: int) -> str:
    """Devuelve la fecha sumando días a hoy."""
    # ...usa datetime...

@tool
def get_next_weekday(weekday: str) -> str:
    """Devuelve la fecha del próximo lunes, martes, etc."""
    # ...usa lógica de fechas...
```

- Así, el agente puede entender frases como “el próximo viernes” o “en dos semanas”.

¿Por qué es importante esto?

- El modelo por sí solo no sabe tus citas ni puede acceder a información actual.
- Con tools, el agente puede:
 - Consultar y modificar datos reales.
 - Evitar errores como empalmes de citas.
 - Entender fechas relativas y reglas personalizadas.

Problemas de los LLMs

- **Biases (sesgos):** Los modelos pueden repetir prejuicios presentes en los datos.
- **Alucinaciones:** A veces inventan información.
- **Prompt injection:** Manipulación de las instrucciones para obtener resultados no deseados.

Conclusiones

- La IA no es magia, pero con creatividad y buenas prácticas, puede parecerlo
- Python es una herramienta poderosa para crear aplicaciones de IA.
- Unete a Pythonistas GDL, futuros eventos en pythonistas-gdl.org

¿Preguntas?

¡Gracias por tu atención!

¡Te invitamos a JConf!

- Evento: **JConf Méx**
- Fecha: **7 de Junio**
- Usa el código: **pythonistas-gdl** para tu registro
- Más info: jconf.mx