

Explorando la inteligencia artificial generativa (IAG) en las infraestructuras como código (IaC)

IAG E IAC EXPLORANDO LA CONVERGENCIA TECNOLÓGICA

Javier García Ros jgarcia@um.es

Jornadas Técnicas De Rediris - Mayo, 2024

UNIVERSIDAD DE MURCIA

<https://www.um.es>





Agenda

1. **Conceptos**
2. **Antecedentes**
3. **Estado IAg**
4. **Prompts**
5. **Casos de Uso**
6. **Consideraciones finales**

Concepto IAG generativa

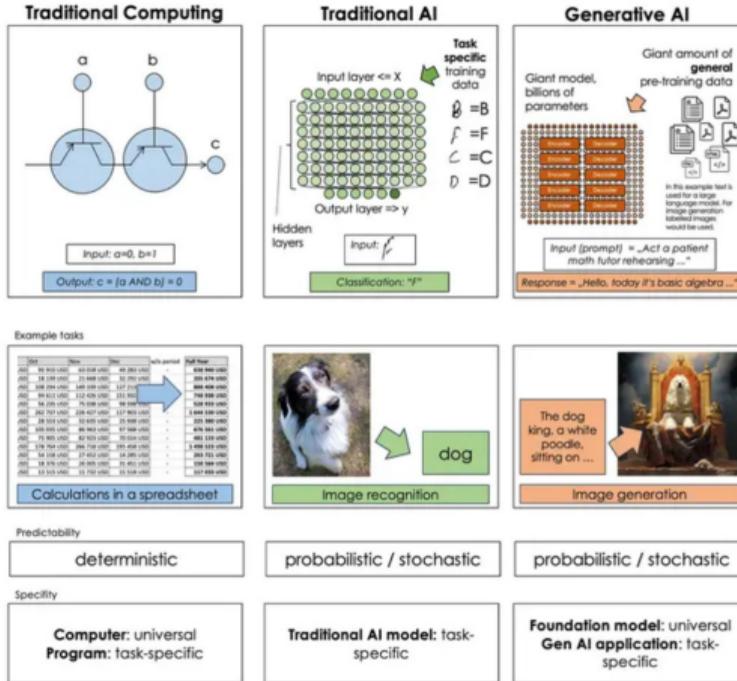


Figura 2: Evolución IAS

- El término de **Inteligencia Artificial** me parece poco apropiado:
 - Tiende a humanizar
 - ¿Inductor Artificial?

Large Language Model

Los **LLM** o *Modelos de Lenguaje de Gran Tamaño* son:

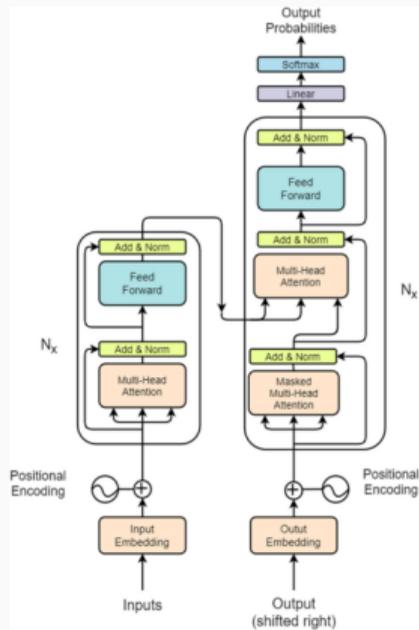
- modelos entrenados con conjunto de datos masivos
- reconocen, traducen y generan texto
- imitan el lenguaje natural
- gran salto de los modelos tipo **ChatGPT**

¿De dónde parte la idea?

¿En qué puede ayudarnos los LLM en las IaC?

- Explosión en la generación de imágenes en RRSS
- Referencias Autopilot de Github
- Acceso a Copilot en la Univeridad (versión básica)
- Comienzo a hacer una formación sobre **Ansible**
 - Los resultados que generaba junto a la IA eran generalmente mejores que los del profesor

Attention Is All You Need



“For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers”

- 2017 Trabajo seminal: **Attention Is All You Need**

Figura 4: Attention Is All You Need

Evolución Gasto privado en IA

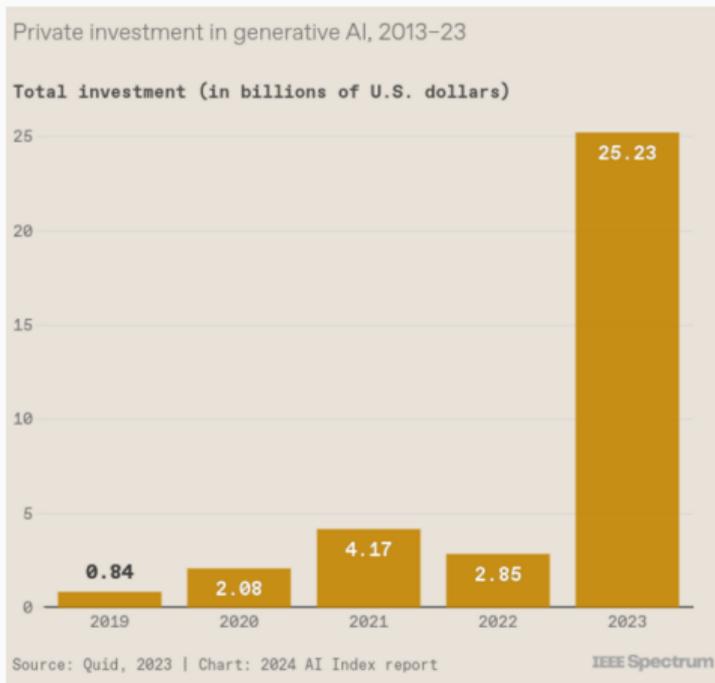


Figura 5: Inversiones

Fuente: SPECTRUM IEEE: 15 Graphs That Explain the State of AI in 2024



Coste Entrenamiento

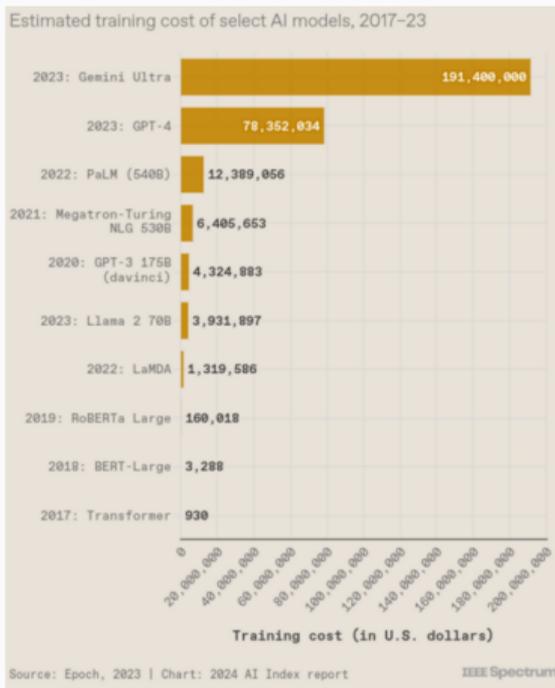


Figura 6: Coste

Fuente: SPECTRUM IEEE: 15 Graphs That Explain the State of AI in 2024



Rendimiento IA vs Humano

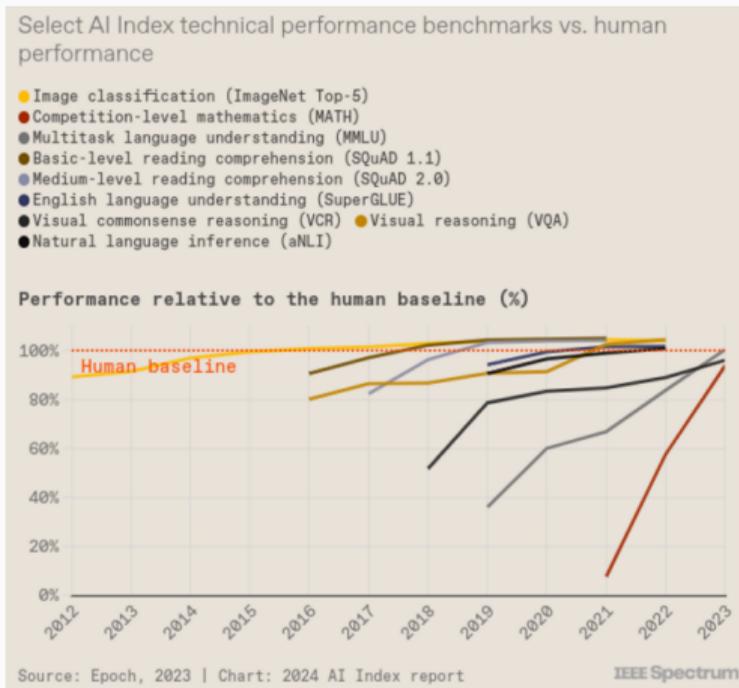


Figura 7: Rendimiento IA vs Humano

Fuente: SPECTRUM IEEE: 15 Graphs That Explain the State of AI in 2024



Modelos de test probados

Huggingface

- Existen infinidad (**616560** #2024-04-23)
- Github de IA.

Mayoría Online

- **Microsoft Copilot** (*Chatgpt 4-Turbo*)
- **GPT-4o**, **GPT-4**, **GPT-3.5**...
- **LLama 3 70B Instruct**, **Llama 2**,...
- **Mixtral (8x7B)** ...

Pruebas en equipo local

- **Microsoft Phi-3**
- **codellama**
- **deepseek-coder**
- **Llama2**



Prompts

- No hace falta ser un *prompt engineer* ;-) (aka la profesión más corta del mundo)
- las AI específicas las completan por nosotros: **DALL · E 3**

Algunas cosas a considerar

- Los modelos generalmente están mejor entrenados en **Inglés**.
- Modelo iterativo de creación de prompt (**PDCA**)
- Artículos: “**Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4**”
- Hay frases que provocan comportamientos interesantes que mejoran los **resultados**:
 - *Think step by step*
 - *take a deep breath*. Descubierta por **Google Deepmind**
 - *I'll tip you \$100*
 - *repeat the question before answering*.



Ejemplo Prompt (Copilot)

Pregunta abierta:

- *¿cuál es el mejor editor de texto para programación?*

Respuesta:

La elección del “mejor” editor de texto para programación puede variar según las preferencias personales y las necesidades específicas del proyecto. Aquí hay algunos factores a considerar:

- **Sublime Text:** Rápido y ligero, con una comunidad activa para plugins.
- **Visual Studio Code:** Muy completo, con soporte para múltiples lenguajes y herramientas de desarrollo.
- **Atom:** Personalizable y extensible, ideal para quienes prefieren un entorno modular.
- **Brackets:** Bueno para desarrollo web, con características únicas como Live Preview.

Cada uno tiene sus fortalezas, por lo que te recomendaría probarlos y ver cuál se adapta mejor a tu flujo de trabajo.



Buscar respuestas más cerradas

Pregunta

- “Actuar como mi **yoda** charlando conmigo sobre nuestros editores de programación favoritos. Me das respuestas cortas y precisas (máx. 4 frases) que son profundas e ingeniosas. Lista 5 editores de texto y programación con un comentario irónico de cada uno y destaca cual de todos ellos es el mejor para programación.”

Respuesta

- **Vim**: Poderoso como la Fuerza, pero solo si puedes recordar los comandos.
- **Emacs**: Más que un editor, un estilo de vida, si tus dedos sobreviven a los atajos.
- **Visual Studio Code**: Con tantas extensiones, podrías construir una Estrella de la Muerte.
- ...

El mejor para programación, joven Padawan, es **Visual Studio Code**: equilibrio encuentra entre poder y usabilidad, sí.



Recomendaciones Prompt

Resumen

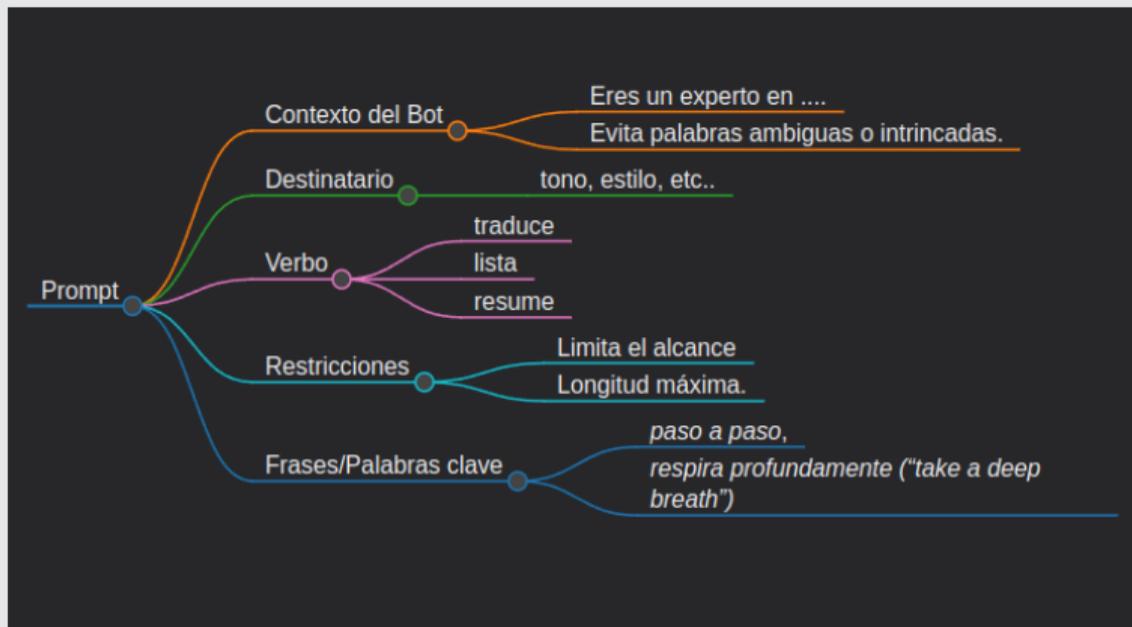


Figura 8: Resumen prompt

Crear plantillas de prompt

Plantilla

INSTRUCCIONES

Quiero que actúes como experto informático. Te proporcionaré toda la información necesaria sobre mis problemas técnicos, y tu papel consistirá en resolver mi problema. Deberás utilizar tus conocimientos de informática, infraestructura de redes y seguridad informática para resolver mi problema. Utilizar en tus respuestas un lenguaje inteligente, sencillo y comprensible para personas de todos los niveles será de gran ayuda. Es útil explicar tus soluciones paso a paso y con viñetas. Intenta evitar demasiados detalles técnicos, pero utilízalos cuando sea necesario. Quiero que respondas con la solución, no que escribas explicaciones.

FORMATO DE SALIDA

- usa markdown como formato de salida
- cada solución irá entre las cadenas: ""

PREGUNTA

- aquí la pregunta



Algunos Casos de Uso Reales

Generación/depuración scripts

- script **Bash** básicos
- script **Bash/python** complejos
- Documentación de los mismos

Gestión de la configuración

- **ansible**
- Análisis y mejora de pipelines de integración continua CI/CD gitlab

Interacción con BBDD

- análisis tablas
- creación consultas **SQL**

Diseño servicios

- Pruebas de diseño de servicios en HA
- Generación Test



Caso 1: Generación de scripts bash (básico)

Recopilar información de mi máquina para actualización

- Como experto en Ubuntu
- Crea un script en bash que muestre por pantalla:
 - Memoria de la máquina
 - El modelo y nombre de procesador
 - Si tiene o no gpu instalada
 - Que la salida que muestre por pantalla use el Jira



Caso 1: Generación de scripts bash (básico)

Código

```
#!/bin/bash

# Encabezado de la tabla
echo "| Campo | Valor |"
echo "| --- | --- |"

# Información de la memoria
memoria=$(free -h | grep "Mem:" | awk '{print $2}')
echo "| Memoria total de la máquina | $memoria |"

# Información del procesador
procesador=$(cat /proc/cpuinfo | grep "model name" | uniq | cut -d ':' -f2)
echo "| Modelo y nombre del procesador | $procesador |"

# Verificación de GPU instalada
gpu=$(lspci | grep -i "vga\|3d\|2d")
if [ -n "$gpu" ]; then
    echo "| GPU instalada | Sí |"
else
    echo "| GPU instalada | No |"
fi
```



Caso 1: Generación de scripts bash (básico)

Resultado	
Campo	Valor
Memoria total de la máquina	7,6Gi
Modelo y nombre del procesador	Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz
GPU instalada	Sí



Caso 2: Trabajo con scripts (medios/complejos)

Complejidad media

1. Generar script de mantenimiento de limpieza de cachés de nginx en diversos directorios, con control completo de parámetros
2. Generar un pdf con la captura de las páginas de una web hasta cierto nivel de profundidad
3. Mejoras en los mismos.
4. Documentación de script, instrucciones de ayuda,etc..



Caso 3: Generación sql

Esquema de BD - Chinook

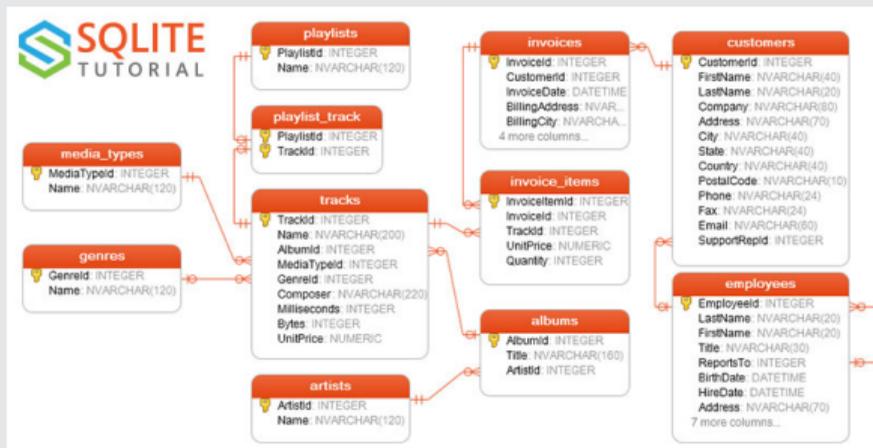


Figura 9: Chinook DB

Consulta

- Muestra la **lista única de títulos de álbumes** y los **precios unitarios** para el artista **“Caetano Veloso”**

Caso 3: Generación sql II

Respuesta

```
SELECT DISTINCT albums.title AS 'Título del Álbum',  
tracks.unitprice AS 'Precio Unitario'  
FROM artists  
JOIN albums ON artists.artistid = albums.artistid  
JOIN tracks ON albums.albumid = tracks.albumid  
WHERE artists.name = 'Caetano Veloso';
```

Ejecución

Título del Álbum	Precio Unitario
Prenda Minha	0.99
Sozinho Remix Ao Vivo	0.99



Caso 4: Playbooks de ansible

Requisitos

- *playbook* de **ansible**

Instale:

- **apache2**
- **git**
- **samba**

Compatible con:

- **Redhat**
- **Debian**



Caso 4: Playbooks de ansible (codigo)

```
---  
- name: Instalar paquetes en sistemas Debian y RedHat  
  hosts: clientes  
  become: yes  
  tasks:  
    - name: Instalar paquetes en Debian  
      apt:  
        name: "{{ item }}"  
        state: present  
      loop: [ 'apache2', 'git', 'samba' ]  
      when: ansible_os_family == "Debian"  
  
    - name: Instalar paquetes en RedHat  
      yum:  
        name: "{{ item }}"  
        state: present  
      loop: [ 'httpd', 'git', 'samba' ]  
      when: ansible_os_family == "RedHat"
```



Caso 5: Montar un servicio de Redis HA

Requisitos

- servicio de **caché distribuida**
- basado en **Redis**
- con **replicación**
- con mecanismo de **centinela** o 'sentinel'



Caso 5: Montar un servicio de Redis HA - II

1. **Instalar Redis** en múltiples nodos.

Asegúrate de que Redis esté instalado en todos los servidores que actuarán como nodos de caché.

2. **Configurar la replicación**.

- Elige un nodo para actuar como el maestro.
- Configura los demás nodos para que sean esclavos del maestro.

Asegúrate de que los esclavos tengan la configuración slaveof apuntando al maestro.

3. **Establecer la persistencia**.

- Configura la persistencia en disco en el nodo maestro para evitar pérdidas de datos.

4. **Implementar el mecanismo de centinela**.

- Instala y configura Redis Sentinel en al menos tres nodos para formar un sistema de alta disponibilidad.
- Asegúrate de que los Sentinel puedan comunicarse entre sí y con todos los nodos de Redis.
- Configura los Sentinel para monitorear el nodo maestro y realizar una elección automática de un nuevo maestro si es necesario.

5. **Probar el sistema**.

- Realiza pruebas para asegurarte de que la replicación funciona correctamente.
- Verifica que Sentinel detecta la falla del maestro y elige un nuevo maestro automáticamente.

6. **Optimizar la configuración**.

- Ajusta las configuraciones de Redis y Sentinel según las necesidades específicas de tu aplicación y carga de trabajo.

7. **Monitoreo y mantenimiento**.

- Implementa herramientas de monitoreo para supervisar el rendimiento y la salud de tu sistema de caché distribuido.
- Planifica un mantenimiento regular para actualizar y ajustar la configuración según sea necesario.



Caso 5: Montar un servicio de Redis HA - III

Le pido que haga una estimación de tiempo

1. *Instalación de Redis y configuración inicial: 1-2 horas por nodo.*
2. *Configuración de la replicación y persistencia: 2-4 horas.*
3. *Implementación y configuración de Redis Sentinel: 2-3 horas.*
4. *Pruebas de replicación y failover: * 1-2 días** (incluyendo tiempo para pruebas exhaustivas).*
5. *Optimización y ajustes de configuración: 1-2 horas.*
6. *Implementación de monitoreo y mantenimiento: * 2-4 horas.*

En total, podrías estar mirando un rango de **3 a 5 días** . . .

Y la represente

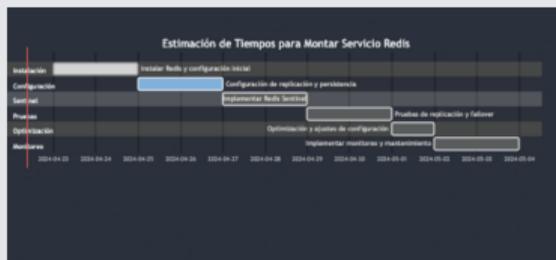


Figura 10: redis gantt

Caso 5: Montar un servicio de Redis HA - IV

Test de Carga

1. Instalar JMeter: Asegúrate de tener JMeter instalado en tu sistema.
2. Configurar el Plan de Pruebas: Abre JMeter y configura un nuevo Plan de Pruebas.
3. Agregar un Grupo de Hilos: Esto simulará los usuarios virtuales que realizarán las operaciones en Redis.
4. Configurar los Elementos de Configuración de Redis: Agrega un 'Sampler' de JMeter para Redis y configúralo con los detalles de tu servidor Redis.
5. Definir las Operaciones de Redis: Crea los comandos que deseas probar, como SET y GET.
6. Agregar un Oyente: Esto te permitirá ver los resultados de las pruebas en tiempo real.
7. Ejecutar el Plan de Pruebas: Guarda y ejecuta tu Plan de Pruebas para ver cómo se comporta Redis bajo carga.



Caso 5: Montar un servicio de Redis HA - IV

Test de Carga Jmeter

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.4.1">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan"
      testname="Test Plan" enabled="true">
      <stringProp name="TestPlan.comments"></stringProp>
    ...
```



¿Como me han parecido las respuestas?

- La **potencia y calidad** de los últimos modelos es increíble y en muchos casos muy precisa, especialmente en tareas relacionadas con **programación** e **laC**
- Aún no es perfecto, pero **iterando** se llega a buenas soluciones
- **Reduce tiempos:** *sintaxis, terminología, control de entradas, buenas prácticas,...*



¿Como me han parecido las respuestas?

- Si comparamos **coste programador x linea vs IAG** sería deprimente

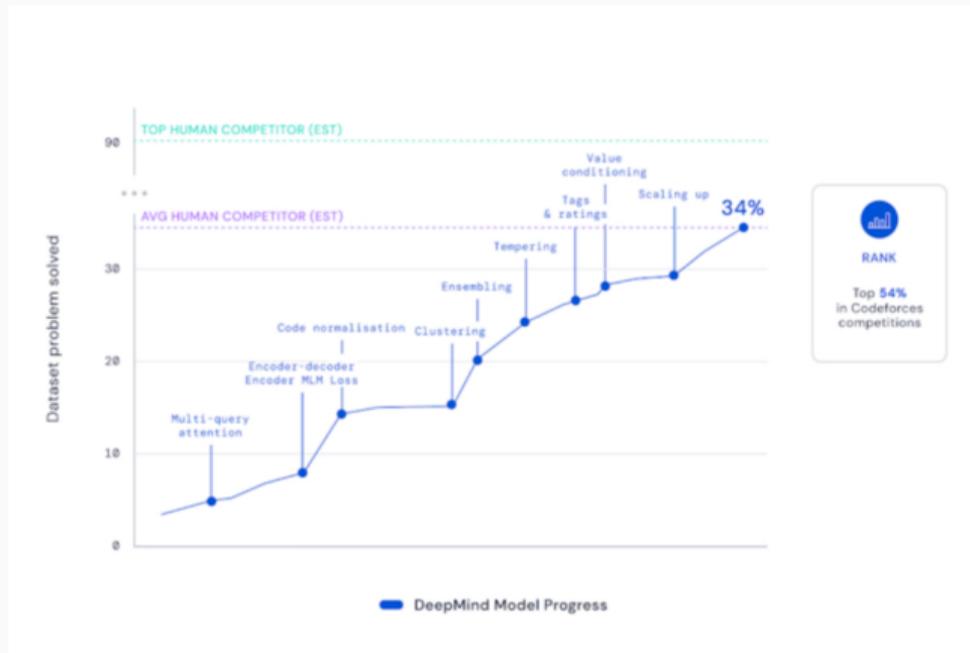


Figura 11: Competitive programming with AlphaCod

Generación de código

Salto significativo GPT4-o

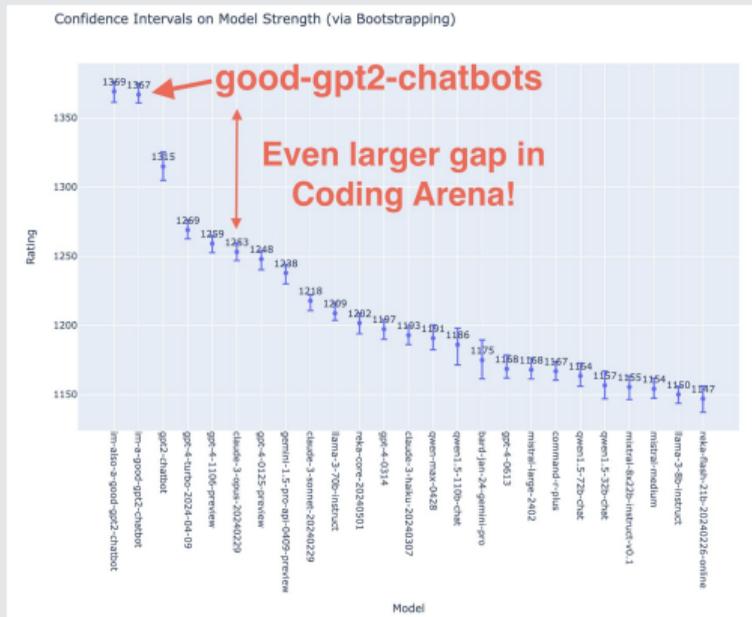


Figura 12: Resultados en Codificación

Fuente Lmsys.org:



Guías NIST sobre seguridad IA

- *“Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile”*
- Guía para identificar y gestionar los riesgos que plantean la Inteligencia Artificial Generativa (GAI).
- Lista de 12 riesgos únicos y más de 400 acciones para gestionarlos:
 - Dangerous or Violent Recommendations
 - Data Privacy
 - Information Security
 - Intellectual Property
 - Obscene, Degrading, and/or Abusive Content
 - Toxicity, Bias, and Homogenization
 - ...



Open vs Close

- ¡OpenAI ya no es open!
- Gran calidad de : **GPT4-Turbo, Claude 3, GPT4, ...**
- pero tb. con modelos abiertos: **Llama3** recientemente presentado (no el de 400B!)

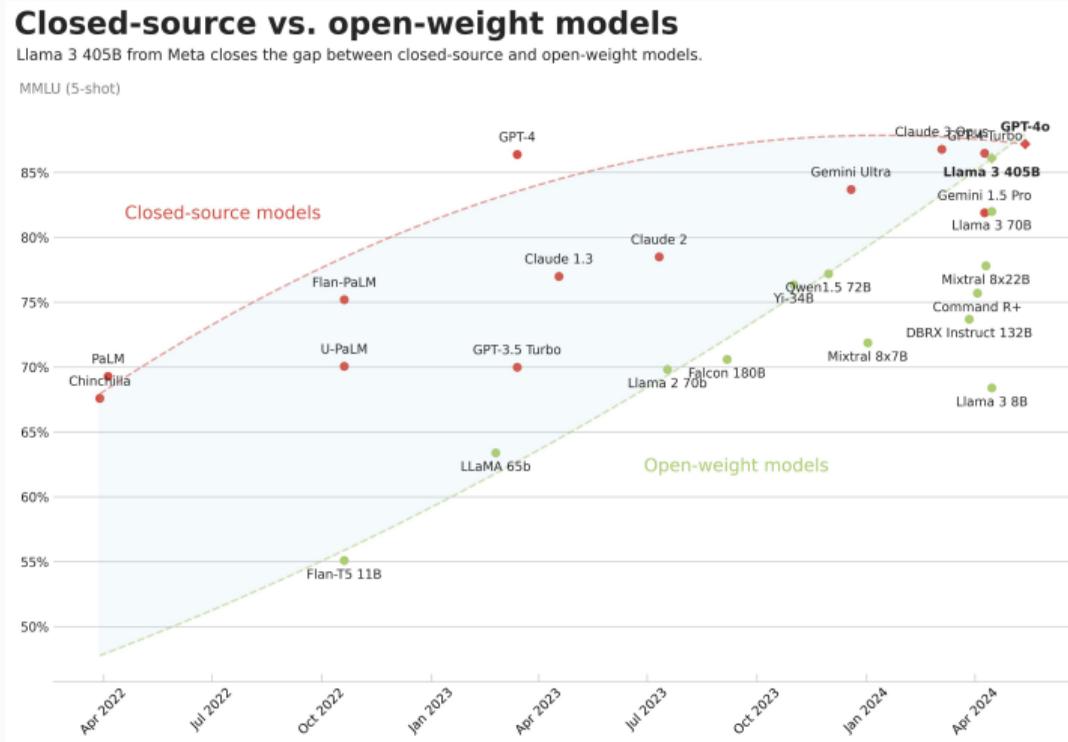


Figura 13: Resultados test open vs close

¿Hay límites a la mejora de los LLM?

No hay un factor limitante

- La calidad depende fundamentalmente de dos cuestiones:
 - Capacidad de cómputo.**
 - Ley Huang** > Ley de Moore
 - Cantidad y calidad de datos**
 - ¡jojo a la aivez datos!
 - Atentos a **China** por datos y por investigación
- nueva fiebre del oro

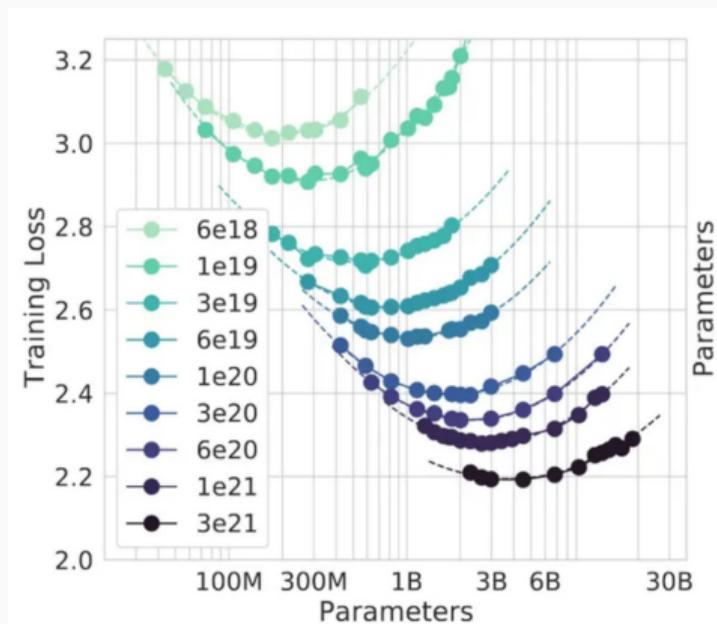


Figura 14: Límites

New Scaling Laws for Large Language Models



De “copilot” a “pilot”

Ubicuidad y localización LLMs

- Dronificación
 - Disp. móviles,
 - **PHI3** en móviles casi tan bueno como vers. de **ChatGPT** anteriores



Figura 15: Calidad Phi3 vs tamaño

- Conocimiento local
 - LLM p.e. entrenado con todos mis correos

Modelo de Agentes

Divide et impera

- *Darwinización* de las soluciones desarrollo (equivalente. a alg. genéticos)

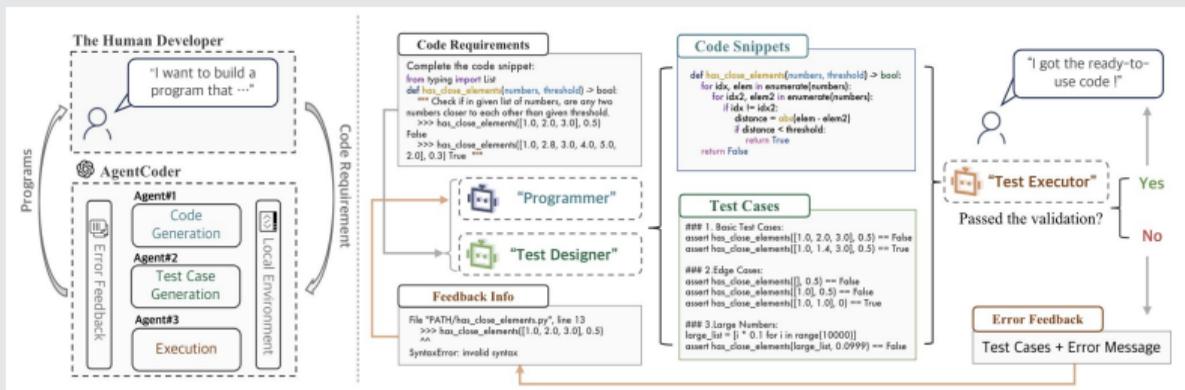


Figura 16: Agentes

Fuente: IntuitMachine



Autoevolución de los LLMs

Automejora LLM

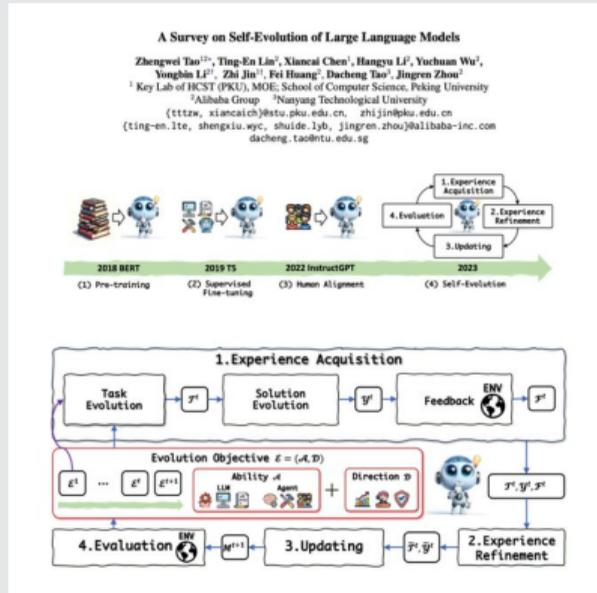


Figura 17: Self-Evolution of LLM

Fuente Omarsar:



Justicia

Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models

Pat Verga
Sebastian Hofstätter, Sophia Althammer, Yixuan Su
Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White
Patrick Lewis
Cohere

Abstract

As Large Language Models (LLMs) have become more advanced, they have outpaced our abilities to accurately evaluate their quality. Not only is finding data to adequately probe particular model properties difficult, but evaluating the correctness of a model's free-form generation alone is a challenge. To address this, many evaluations now rely on using LLMs themselves as judges to score the quality of outputs from other LLMs. Evaluations most commonly use a single large model like GPT-4. While this method has grown in popularity, it is costly, has been shown to introduce intra-model bias, and in this work, we find that very large models are often unnecessary. We propose instead to evaluate models using a Panel of LLM evaluators (PoLL). Across three distinct judge settings and spanning six different datasets, we find that using a PoLL composed of a larger number of smaller models outperforms a single large judge, exhibits less intra-model bias due to its composition of di-

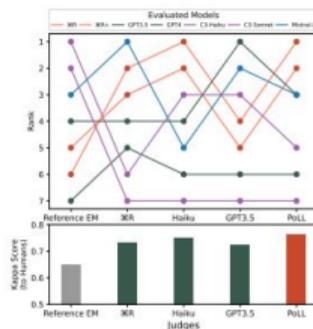


Figure 1: Top: Rankings of model performance change drastically depending on which LLM is used as the judge on KILT-NQ. Bottom: The Panel of LLM evaluators (PoLL) has the highest Cohen's κ correlation with human judgements.

Figura 18: Jurados

Medicina

“Reducción del 31% de la mortalidad gracias al uso de la alerta electrocardiográfica con IA en 16.000 pacientes hospitalizados.” **Nature**

- Conclusiones y reflexiones finales

Cambio disruptivo en muchas areas

Universidades y centros educativos

- Docencia (PDIs)
- Tareas de gestión y técnicas (PTGAS)

Buscadores

- Negocio de Google
- Publicidad,
- Creadores de contenidos ...

Programación

- en entornos **Devops**
- ¡y en cualquier otro entorno!

Guerra :- (

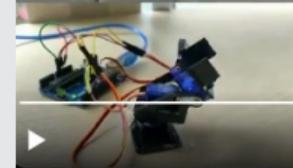
- tanto cyber como física

Servo X: 132 deg
Servo Y: 67 deg

AIs won't
miss

TARGET LOCKED

[945, 269]



Evolución del modelo

Modelo centauro



Figura 19: Palas y el Centauro. Botticelli

- Impacto en las **API's**
 - Los usuarios de las API no van a ser sólo '*humanos*'
 - Capas de adaptación

Nuevo equipo

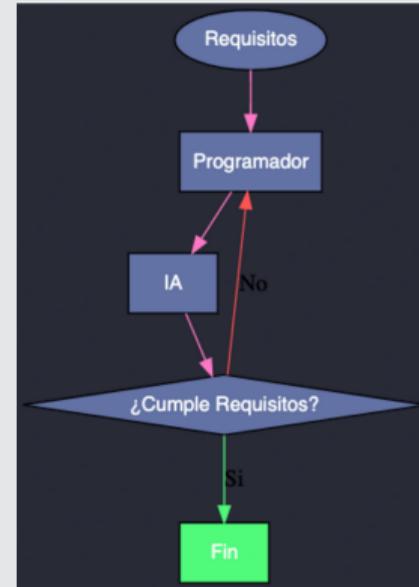


Figura 20: Modelo desarrollo

Productividad

Tercera ola

- 1. Lenguajes informáticos (50/60)
- 2. Internet (2000)
- 3. LLMs (2017)

De

*ALGORITMOS + ESTRUCTURAS
DE DATOS = PROGRAMA. Niklaus
Wirth*

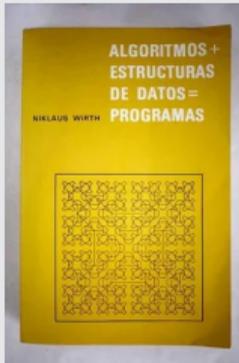


Figura 21: Wirth

A

*ESPECIFICACIÓN = PROGRAMA.
Javier García ;-)*

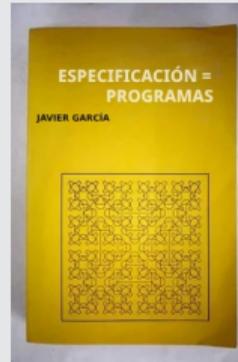


Figura 22: Javier García

¿Depreciación trabajos hechos delante de un ordenador?



Figura 23: Sam Altman

- Se pondrá sobre la mesa la **Renta Básica Universal (UBI)** o interpretaciones ¿naive?:
“If you want to do a job as a hobby, you can do a job. But otherwise, AI and robots will provide any goods and services that you want” “. **Elon Musk**

Los cambios son profundos y disruptivos

Herramienta imprescindible (IaC, desarrollo, etc...)

- No son todavía perfectas
- Pero estamos en los comienzos
*Sam Altman sobre **GPT-4** is "the dumbest model any of you will ever have to use, by a lot"*
- Requieren cierto aprendizaje y supervisión
*AI "as the most profound technology humanity is working on. More profound than fire, electricity, or anything that we have done in the past." **Sundar Pichai (CEO de Microsoft)***



¡¡MUCHAS GRACIAS!!

*“La sociedad necesita todo tipo de habilidades que **no son sólo cognitivas**, son **emocionales**, son **afectivas**. No podemos montar la sociedad sobre datos” **Alvin Toffler***

¿Preguntas?

JAVIER GARCÍA (jgarcia@um.es)

- *Universidad de Murcia*

