

**Convocatoria para la provisión, mediante procedimiento de ingreso, por el sistema de oposición, de 6 plazas del puesto de trabajo de gestor P.B. (Servicio Informático), nivel B, al servicio de la Universidad Pública de Navarra**

(Aprobada mediante Resolución 2412/2023, de 15 de noviembre del gerente de la Universidad Pública de Navarra)

**SEGUNDA PRUEBA  
TRADUCCIÓN DE CASTELLANO A INGLÉS Y  
VICEVERSA**

**Tiempo de realización: 90 minutos**

10 de junio de 2024

**NO PASE A LA HOJA SIGUIENTE  
MIENTRAS NO SE LE INDIQUE QUE PUEDE COMENZAR**



## BLOQUE I TRADUCCIÓN DEL SIGUIENTE TEXTO DE INGLÉS A CASTELLANO (5 puntos)

### Introducing ChatGPT<sup>1</sup>

We have trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer follow-up questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human Artificial Intelligence trainers provided conversations in which they played both sides—the user and an Artificial Intelligence assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect data, we took conversations that Artificial Intelligence trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had Artificial Intelligence trainers rank them. Using these reward models, we can fine-tune the model using optimization methods. We performed several iterations of this process.

Some limitations of ChatGPT:

- ChatGPT sometimes writes plausible-sounding but incorrect or nonsensical answers. Fixing this issue is challenging, as: (1) during Reinforcement Learning (RL) training, there's currently no source of truth; (2) training the model to be more cautious causes it to decline questions that it can answer correctly; and (3) supervised training misleads the model because the ideal answer depends on what the model knows, rather than what the human demonstrator knows.
- ChatGPT is sensitive to tweaks to the input phrasing or attempting the same prompt multiple times. For example, given one phrasing of a question, the model can claim to not know the answer, but given a slight rephrase, can answer correctly.
- Ideally, the model would ask clarifying questions when the user provided an ambiguous query. Instead, our current models usually guess what the user intended.
- While we've made efforts to make the model refuse inappropriate requests, it will sometimes respond to harmful instructions or exhibit biased behavior. We're enthusiastic to collect user feedback to aid our ongoing work to improve this system.

---

<sup>1</sup> Texto adaptado de <https://openai.com/index/chatgpt/>

## BLOQUE II TRADUCCIÓN DEL SIGUIENTE TEXTO DE CASTELLANO A INGLÉS (5 puntos)

### Programación en Android: 5 lenguajes más usados <sup>2</sup>

La programación en Android ha experimentado un crecimiento significativo en la última década, convirtiéndose en un campo esencial para desarrolladores de todo el mundo. Para crear aplicaciones funcionales en el sistema operativo Android, los desarrolladores recurren a diversos lenguajes de programación. A continuación, se indican los cinco lenguajes de programación más utilizados en Android:

- Java ha sido el lenguaje principal de programación en Android desde su inicio. Las principales razones son su portabilidad y capacidad para ejecutarse en diferentes plataformas. La plataforma Android Studio, el entorno de desarrollo oficial, está optimizada para trabajar con Java, lo que facilita la creación de aplicaciones robustas y escalables.
- Kotlin ha ganado rápidamente popularidad siendo una alternativa moderna a Java. Kotlin fue anunciado oficialmente por Google en 2017. Kotlin ofrece ventajas significativas, como una sintaxis más concisa y la eliminación de numerosas limitaciones de Java. Su interoperabilidad con Java facilita la migración gradual de proyectos existentes.
- Cuando se trata de desarrollo de juegos y aplicaciones que requieren un rendimiento excepcional, muchos desarrolladores recurren a C++. La herramienta Android NDK permite la integración de código nativo en aplicaciones Android, y C++ se convierte en una elección acertada para aquellos que buscan un rendimiento óptimo y una gestión eficiente de recursos.
- React Native, basado en JavaScript, permite a los desarrolladores construir aplicaciones para iOS y Android utilizando un único código base. Aunque no es un lenguaje nativo de Android, su popularidad ha crecido gracias a la eficiencia en el desarrollo multiplataforma. Los desarrolladores familiarizados con JavaScript pueden aprovechar sus habilidades para crear aplicaciones Android de manera rápida y efectiva.
- Flutter es un marco de desarrollo de código abierto respaldado por Google. Aunque no es un lenguaje tradicional de este sistema operativo, Flutter ha ganado terreno en la programación en Android gracias a su capacidad para construir interfaces de usuario atractivas y fluidas de manera rápida. Dart ofrece un rendimiento eficiente y una experiencia de desarrollo ágil.

La elección del lenguaje dependerá de factores como la complejidad del proyecto, las preferencias del desarrollador y los requisitos de rendimiento.

---

<sup>2</sup> Texto adaptado de <https://keepcoding.io/blog/programacion-en-android/>