

# Tecnologias Usadas

## 1 Introdução

Neste documento estão expostas as tecnologias que iremos utilizar no desenvolvimento do nosso projeto relativo ao *Jogo da Sueca*, bem como a justificação para a escolha de cada uma delas, tendo em conta as necessidades do projeto.

## 2 Back-end: Python

A linguagem Python foi escolhida para o desenvolvimento do **back-end** devido à sua forte ligação com a área da Inteligência Artificial. Sendo esta considerada a linguagem mais adequada para este tipo de desenvolvimento.

Python destaca-se também pela sua sintaxe simples e legibilidade, o que torna o processo de desenvolvimento mais rápido e o código mais simples de compreender, facilitando o trabalho em equipa.

Por fim, é importante também referir que o desenvolvimento da *back-end* está a ser realizado no **Visual Studio Code (VS Code)**.

## 3 Back-end (Card-Identification): YOLO e OpenCV

Escolhemos um sistema híbrido que combina OpenCV com YOLO para identificar cartas, pois, após uma análise das características de cada tecnologia, concluímos que esta solução oferece o melhor equilíbrio entre rapidez, precisão e robustez.

O OpenCV destaca-se pelo pré-processamento eficiente das imagens, como correção de perspetivas e segmentação, preparando o ambiente para deteções mais fiáveis.

Já o YOLO é um modelo avançado de deteção de objetos em tempo real, capaz de identificar cartas mesmo em condições complexas, com elevada confiança e velocidade.

A combinação permite que o OpenCV otimize a imagem antes da análise do YOLO, que faz a identificação precisa, resultando num sistema confiável e eficaz para o reconhecimento automático de cartas (<https://tinyurl.com/4jejyk2u>).

## 4 Front-end (Mockup): Figma

O **Figma** foi a ferramenta escolhida para a fase de planeamento do design da aplicação, permitindo criar um *mockup* visual e interativo da interface.

A escolha desta ferramenta deve-se essencialmente à sua facilidade de utilização e ao suporte para colaboração em tempo real, o que otimiza o trabalho em equipa e a comunicação durante o desenvolvimento do design.

Assim, o Figma ajuda-nos a definir o aspeto visual e a estrutura de navegação da aplicação antes da implementação, garantindo uma melhor coerência no design final.

## 5 Front-end: Kotlin

Inicialmente, considerámos utilizar o **React Native** para o desenvolvimento do front-end do nosso Jogo da Sueca; no entanto, após uma análise mais aprofundada, decidimos optar por **Kotlin** como ferramenta principal para o *front-end*.

Esta decisão foi motivada por vários fatores. Em primeiro lugar, Kotlin é a linguagem oficial para o desenvolvimento Android, oferecendo uma integração nativa e um desempenho superior face a soluções híbridas.

Além disso, verificámos que a maioria das aplicações de jogo da sueca disponíveis na Play Store são desenvolvidas em Kotlin, o que demonstra a sua adequação e eficiência para aplicações semelhantes à nossa.

Kotlin apresenta também uma sintaxe moderna e segura com total compatibilidade com o ecossistema Java, facilitando o desenvolvimento.

Assim, a utilização de Kotlin assegura um desenvolvimento eficiente e uma aplicação estável, alinhada com as práticas atuais do mercado.

## 6 Conclusão

A combinação destas tecnologias — Python, YOLO e OpenCV no *back-end*, Figma para a realização do mockup e planeamento do design e Kotlin no *front-end* — proporciona um equilíbrio entre eficiência, escalabilidade e uma experiência de utilizador moderna e intuitiva.