



Bienvenidos a **Swiftle**



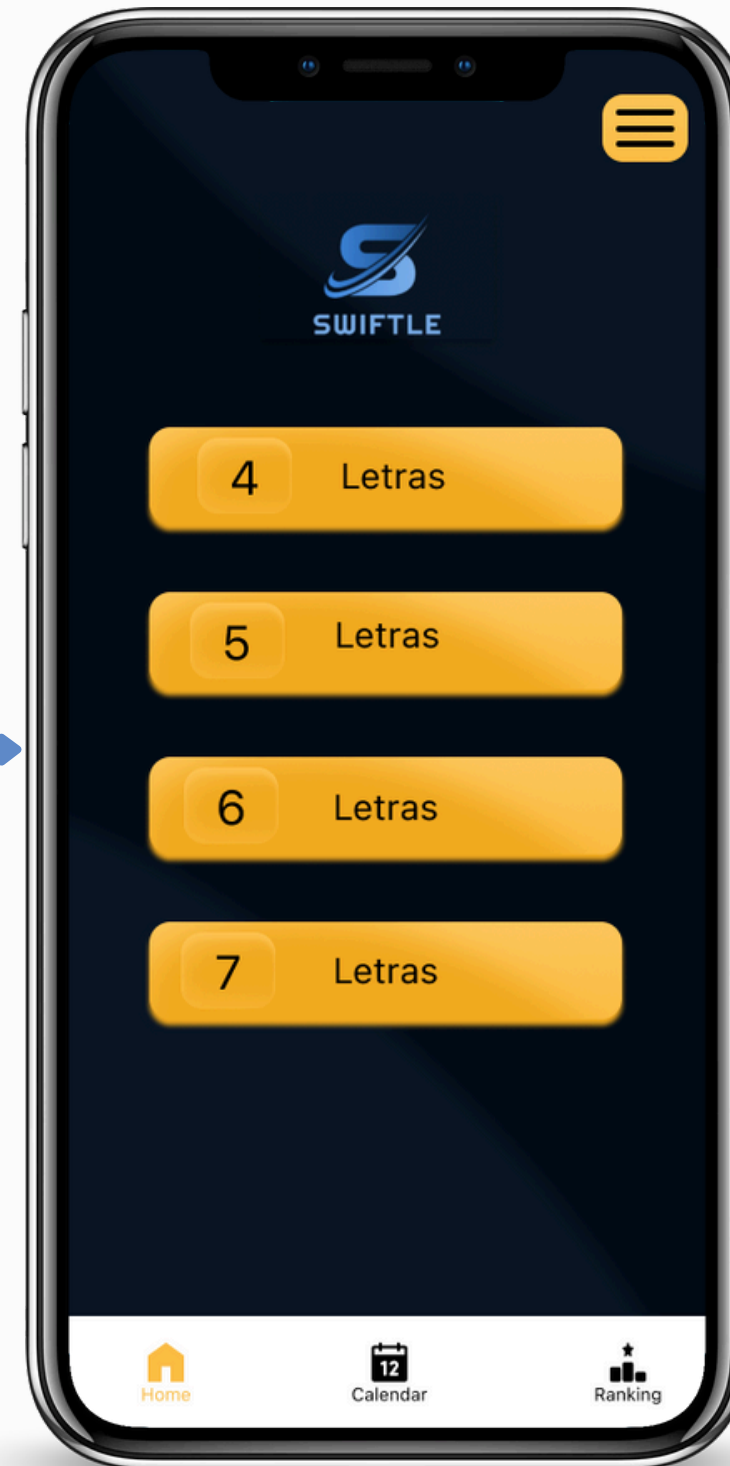


ORIGEN

Swiftle es una aplicación de entretenimiento diseñada para los amantes del lenguaje de programación Swift, basado en el famoso juego Wordle. Esta aplicación pondrá a prueba nuestros conocimientos de este lenguaje.



Usabilidad sencilla





- Entorno didáctico
- Entretenido
- Intuitivo
- Globalizado
- Competitivo





APLICACIONES SIMILARES



- WORDLE
- SPEEDLE
- NERDLE
- SQUIRDLE
- MOVIEDLE



DIFERENCIAS

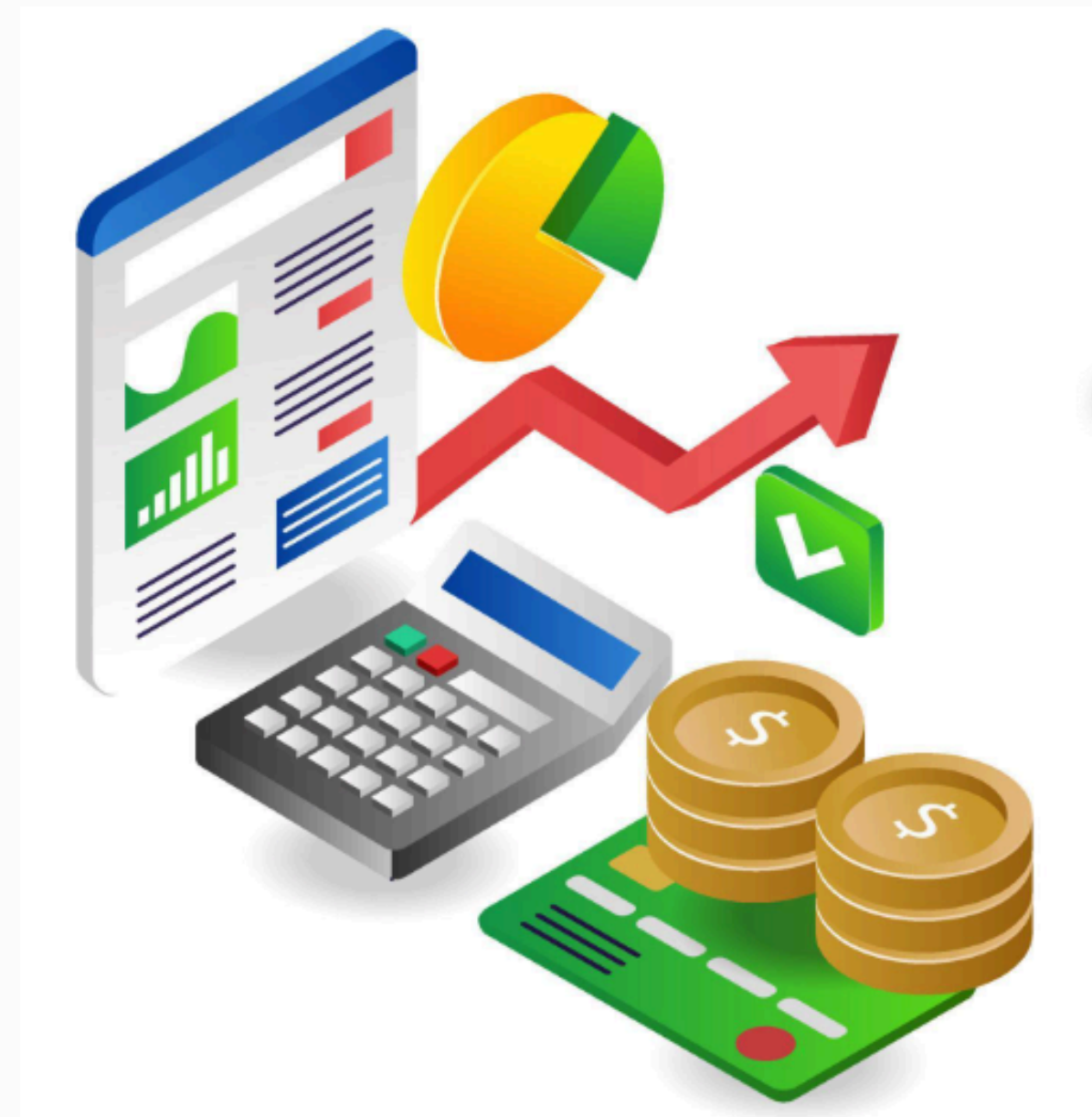
Swiftle está orientado a un público específico

Enfoque competitivo con sistemas de puntuaciones





ANÁLISIS FINANCIERO



INGRESOS



- Suscripción básica

Una versión gratuita que cubra las necesidades esenciales. Se ingresará por un sistema de anuncios.



- Suscripción medium

Eliminará el sistema de anuncios y rejugabilidad. Tendría un coste de 7.99€



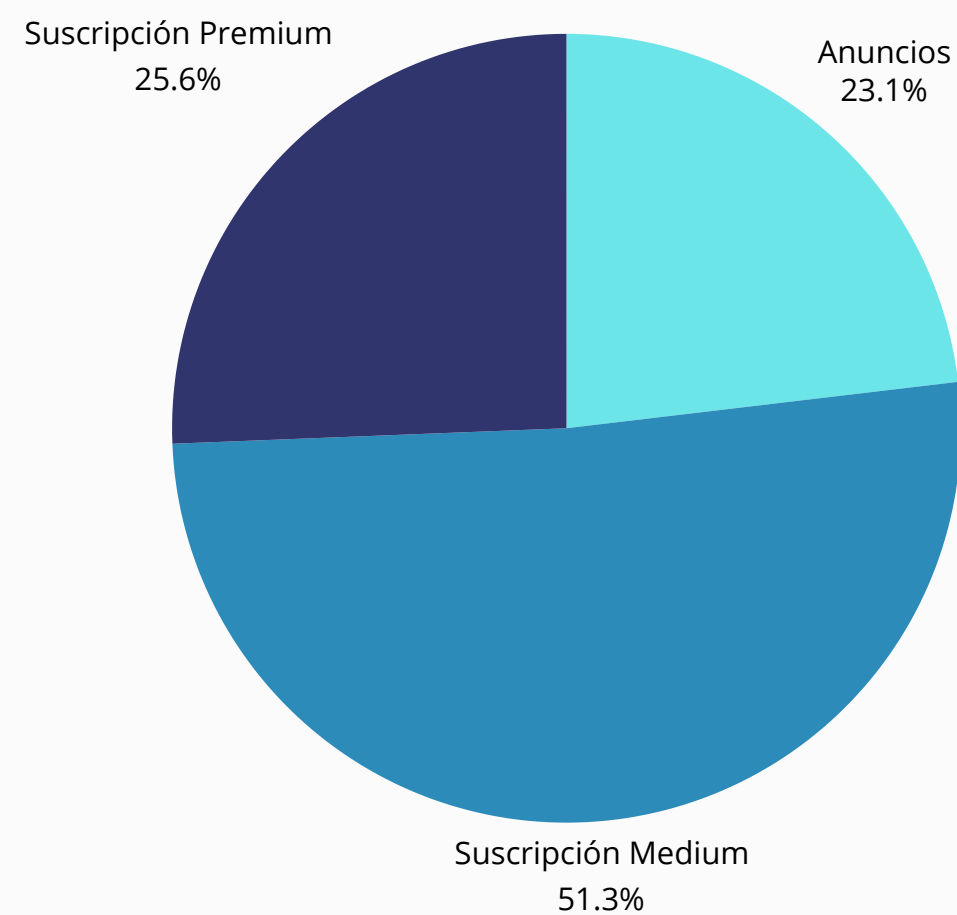
- Suscripción premium

Se podrá jugar un mayor número de palabras con un coste de 9.99€

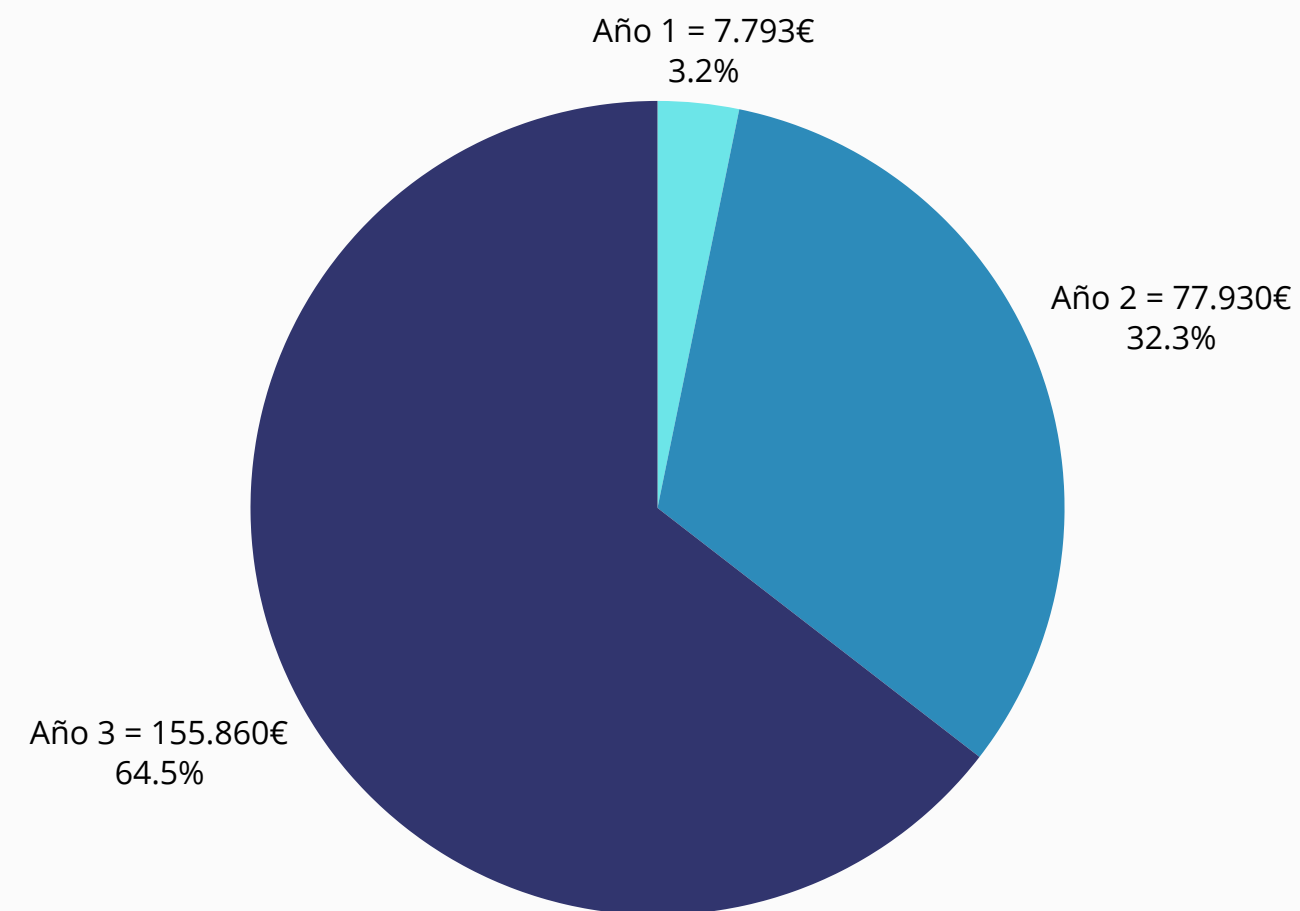


Estimación de ingresos

Porcentajes suscripción



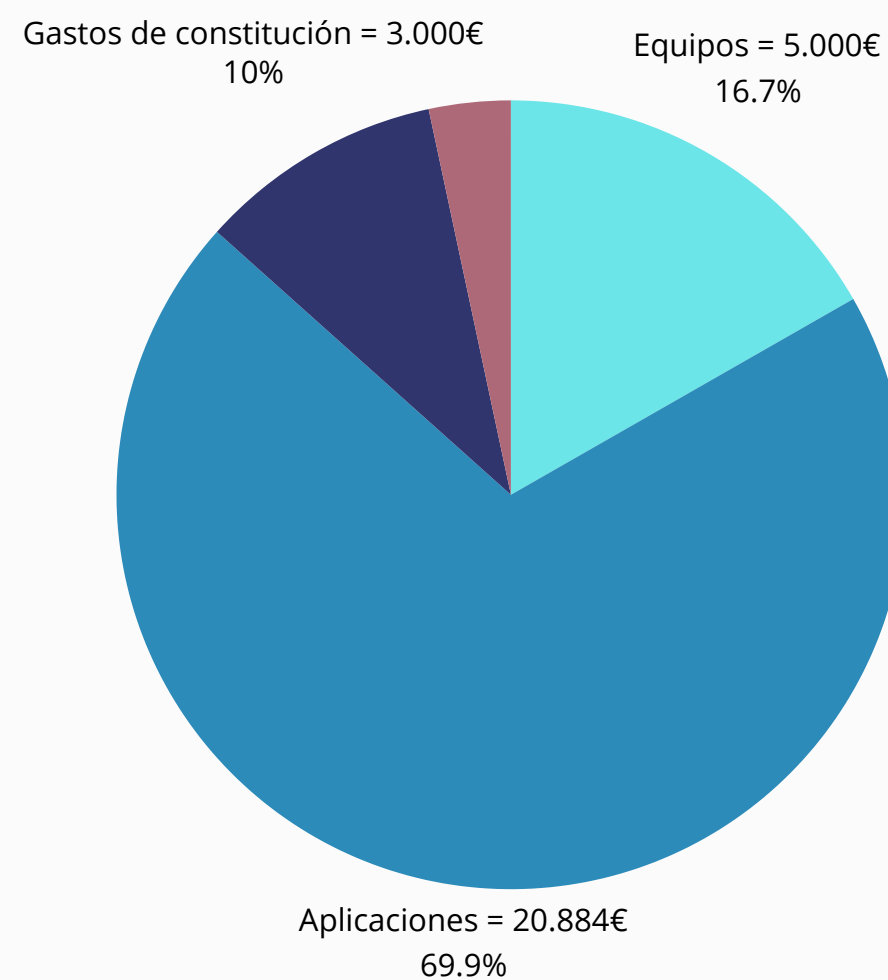
Previsión de 3 años



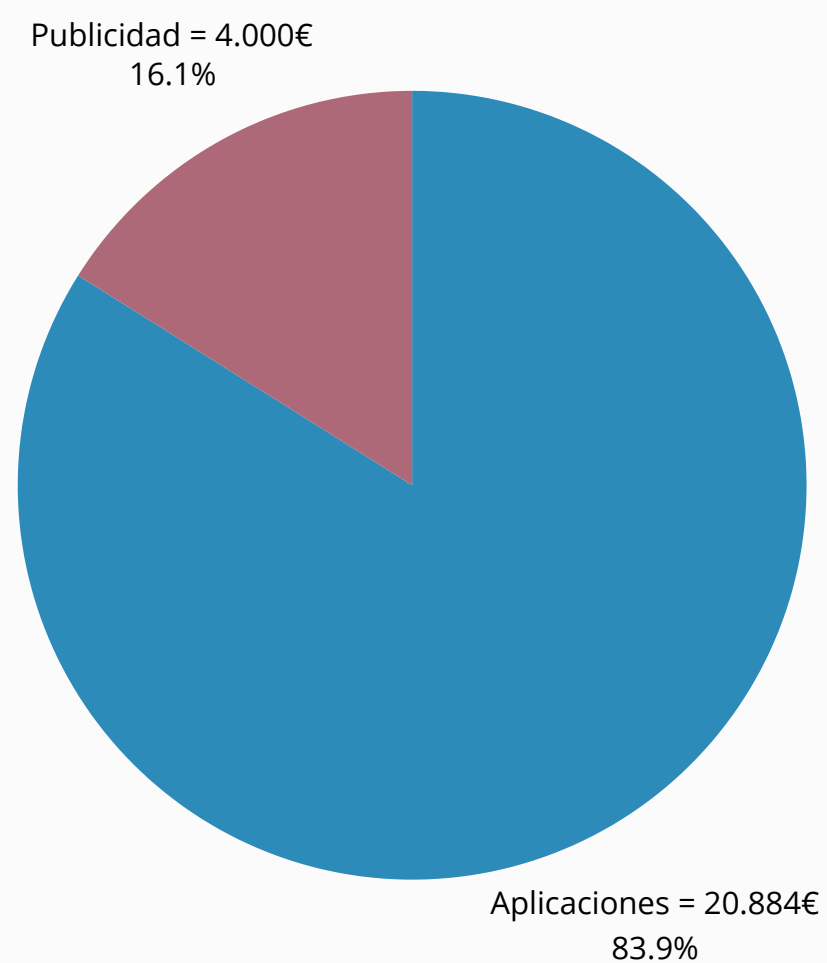


COSTES

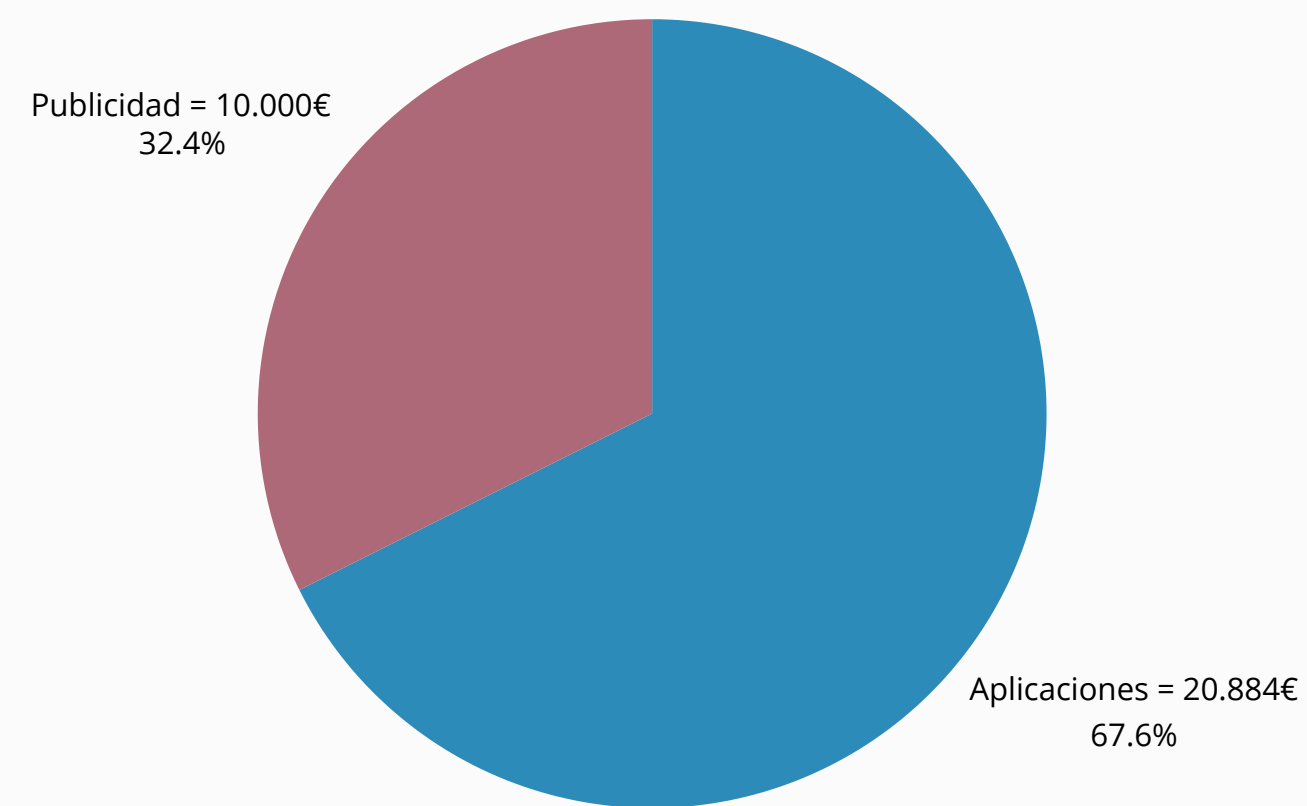
AÑO 1



AÑO 2



AÑO 3





BALANCES

Ingresos

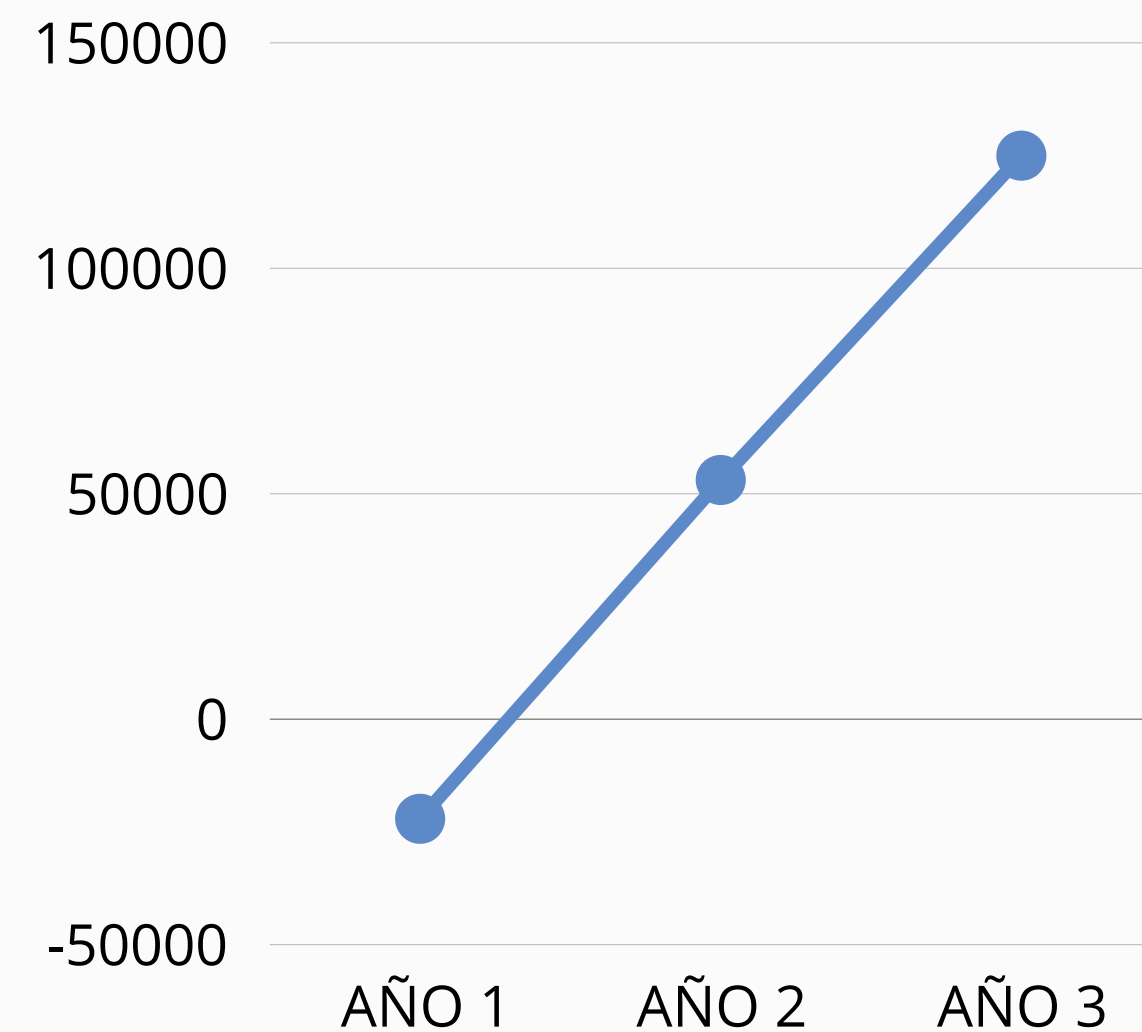
- Año 1 = 7.793€
- Año 2 = 77.930€
- Año 3 = 155.860€

Costes

- Año 1 = 29.884€
- Año 2 = 24.884€
- Año 3 = 30.884€

Balance

- Año 1 = **-22.091€**
- Año 2 = **53.046€**
- Año 3 = **124.976€**





PLAN DE INVERSIONES Y CRECIMIENTO

FASE 1 - Desarrollo

Valoración = 50k

Subvenciones,
créditos o
inversores externos

FASE 2 - Crecimiento

Valoración = 100k

Participaciones

FASE 3 - Asentamiento

Valoración = 250k

Explotación por
parte de los socios

Venta a inversores
privados



ANÁLISIS TÉCNICO





ARQUITECTURA MVVM

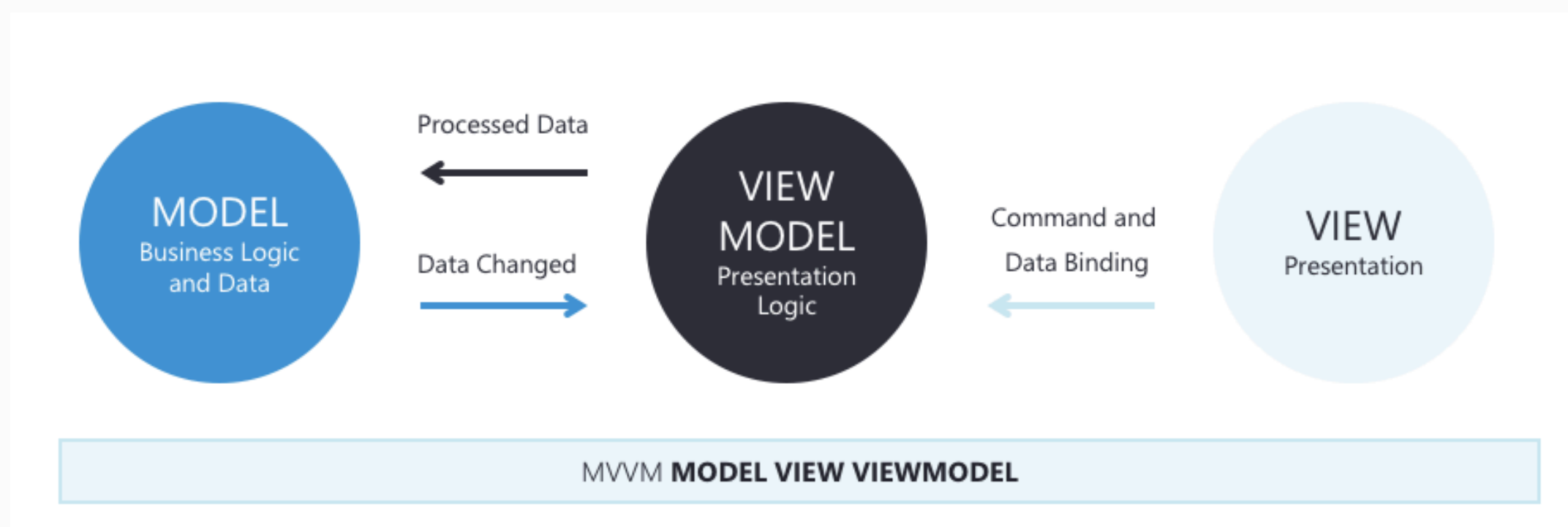
1. Separación de Problemas: Al dividir el código en Model, View y ViewModel, se consigue una clara separación de problemas, lo que facilita la mantenibilidad y escalabilidad de la aplicación.

2. Reutilización de Código: La separación de responsabilidades permite una mayor reutilización del código, ya que los Models y ViewModels pueden ser fácilmente probados y reutilizados en diferentes partes de la aplicación o en diferentes aplicaciones.

3. Pruebas Unitarias: La lógica de negocios y la lógica de UI están contenidas en el ViewModel, facilitando la realización de pruebas unitarias en el código. Esto aumenta la calidad y fiabilidad del software.

4. Desarrollo Paralelo: Los desarrolladores de software y los diseñadores de interfaz de usuario pueden trabajar de manera paralela en el Model y la View sin interferencias, gracias a la independencia de los componentes.

5. Flexibilidad y Extensibilidad: La arquitectura de MVVM se puede adaptar fácilmente a cambios en los requisitos del negocio o en la interfaz de usuario, proporcionando una mayor flexibilidad y extensibilidad.





TECNOLOGÍAS UTILIZADAS

Lenguaje Swift y SwiftUI

- Su sintaxis declarativa permite describir las interfaces de usuario de forma más concisa y legible.
- Su sistema de previsualización permite ver inmediatamente los cambios sin necesidad de compilación.
- La integración con Swift permite utilizar las últimas incorporaciones del lenguaje, como la API FormatStyle, directamente en las zonas de texto.
- La división de las vistas en componentes reutilizables optimiza la mantenibilidad de la aplicación.
- El apoyo a la internacionalización y accesibilidad facilita el uso de la app por parte de todos sin tener que escribir código explícito, a diferencia de UIKit.



TECNOLOGÍAS UTILIZADAS

Vapor



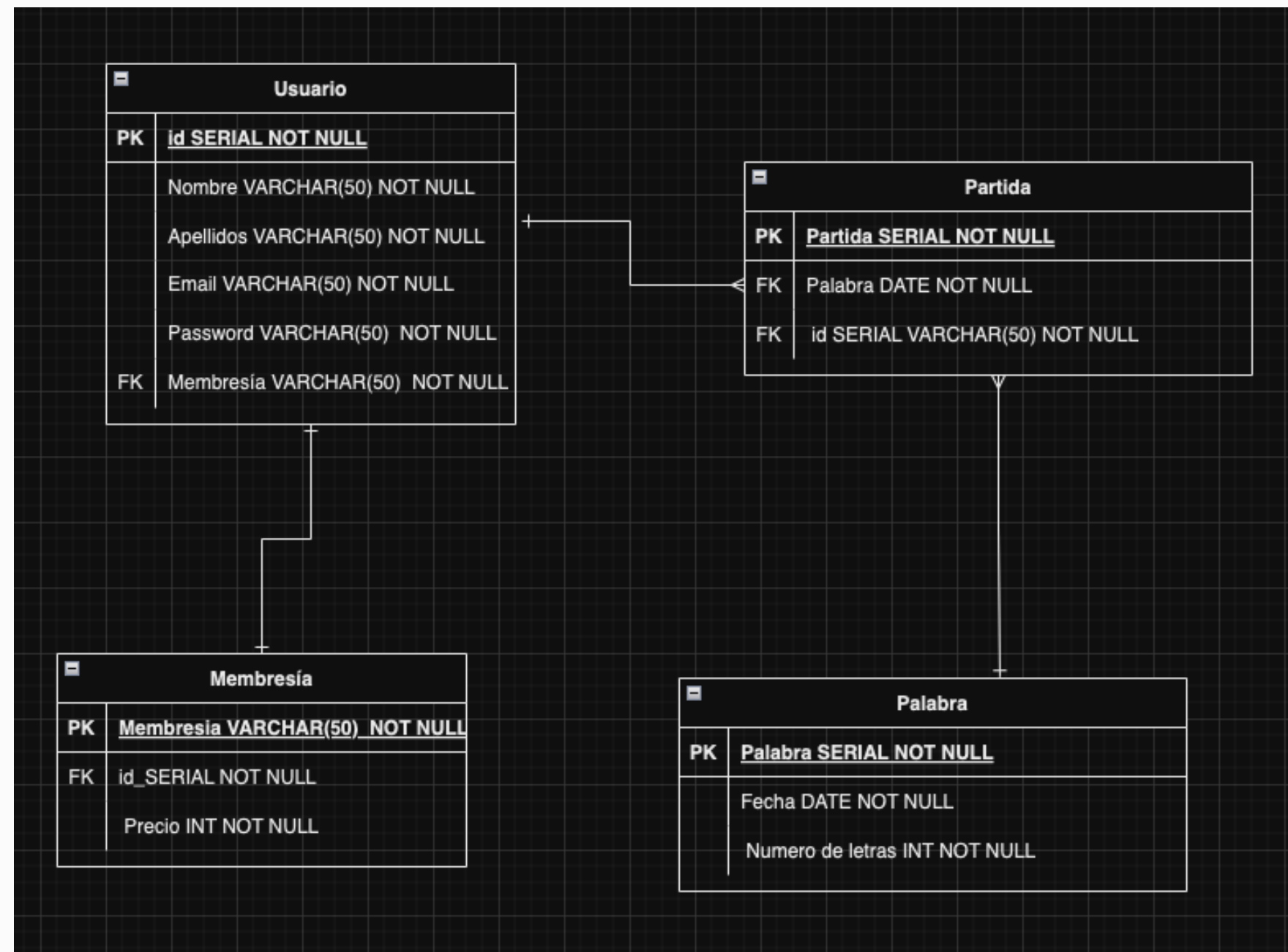
Vapor permite a los desarrolladores construir aplicaciones y servicios backend en Swift, un lenguaje que normalmente se asocia con el desarrollo de aplicaciones iOS. Vapor aprovecha Swift para ofrecer un desarrollo seguro, y rápido en el lado del servidor.

Es una opción popular entre los desarrolladores de Swift que quieren mantener un único lenguaje en todo el desarrollo, desde el cliente hasta el servidor.

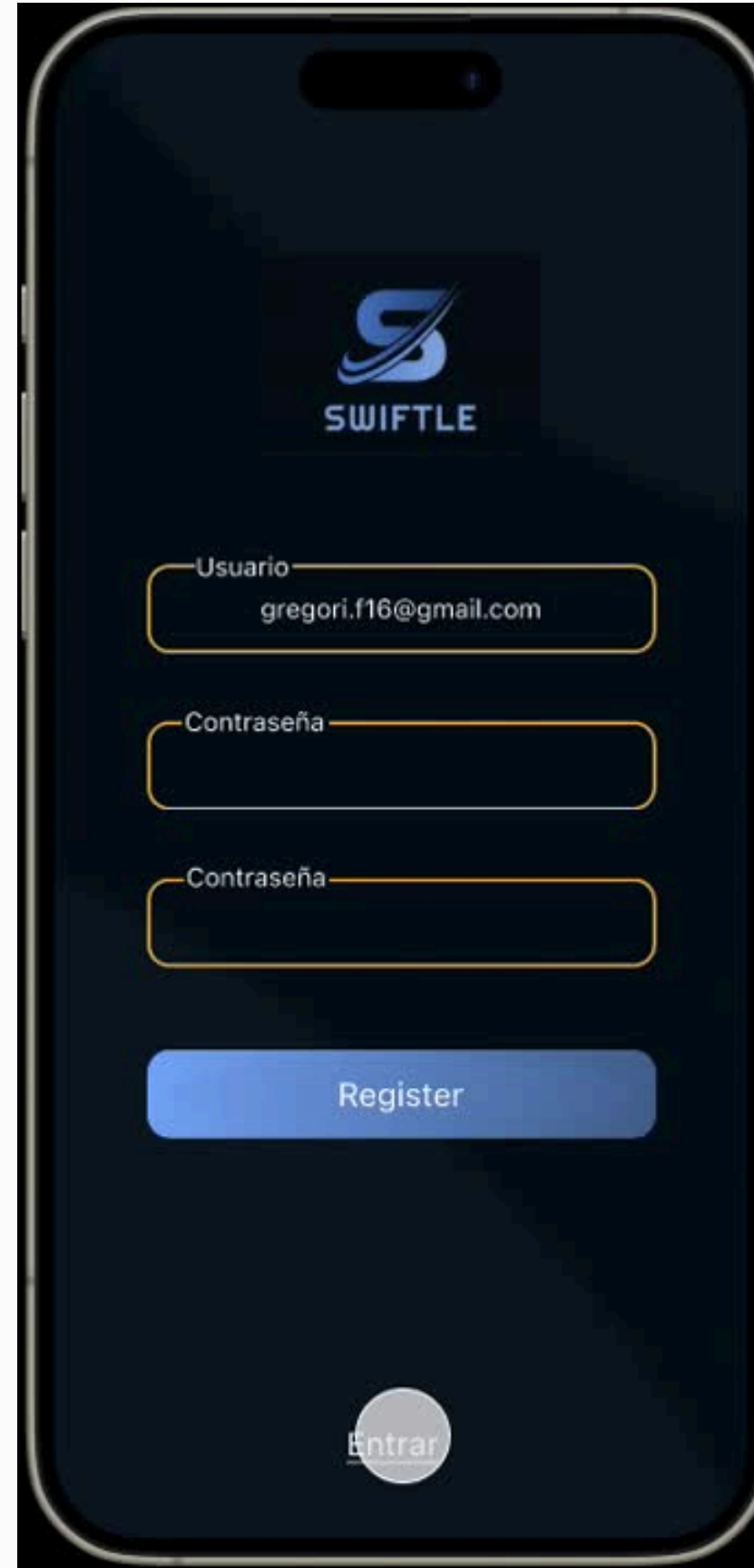


TECNOLOGÍAS UTILIZADAS

Modelo Entidad/Relación



DEMO



The image shows a mobile application interface for 'SWIFTLE'. At the top, there is a logo consisting of a stylized blue 'S' with a swoosh, and the word 'SWIFTLE' in white capital letters below it. Below the logo, there are three input fields with yellow borders. The first field is labeled 'Usuario' and contains the email address 'gregori.f16@gmail.com'. The second and third fields are both labeled 'Contraseña' and are currently empty. Below these fields is a blue button with the text 'Register' in white. At the bottom of the screen, there is a circular button with the text 'Entrar' in white.

SWIFTLE

Usuario
gregori.f16@gmail.com

Contraseña

Contraseña

Register

Entrar

SIMULADOR iOS





GRACIAS

Realizado por:

Gregori Farías

David Ortega Iglesias

