CryptoTracker

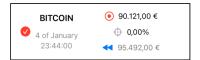
Autor

Nombre: Juan Calzada Bernal
Repositorio: <u>GitHub repo</u>

Funcionalidades destacadas

Alertas

- Se pueden configurar alertas para monitorear el precio de una criptomoneda.
- Hay dos tipos de alertas: Precio y Volatilidad.
 - o Precio se activará cuando el precio de la criptomoneda alcance un valor determinado.
 - Volatilidad se activará cuando el precio de la criptomoneda cambie en un porcentaje determinado.
- La alerta, una vez creada, mostrará
 - o precio actual de la criptomoneda 🔾
 - o target (precio o volatilidad) ⊚
 - o precio cuando se estableció la alerta 🥨
- La alerta tiene 3 posibles estados:
 - o Objetivo alcanzado 🔵
 - ∘ Expirada ●
 - o Activa 🔵



9 Jan 2025 19:29

EXPIRATION DATE

Expiration date

Enable notifications

Notificaciones

- Opción para la recepción de notificaciones al configurar una alerta.
- Se recibirá una notificación cuando la alerta expire.

API Key

- Es recomendable configurar una API Key para evitar el límite de peticiones a la API pública de CoinGecko.
- La API Key se guarda de forma segura en Apple Keychain.
- Capacidad para comprobar la validez de la API Key en el apartado de ajustes de la aplicación.

Gráficos

- Visualizaciones de gráficos de la evolución del precio de cualquier criptomoneda.
- Hay que especificar el rango de tiempo para la visualización del gráfico.
- Solo se pueden visualizar rangos de tiempo pasados.

Preferencias

- Opción para cambiar la moneda de visualización de los precios.
- Opción para activar las notificaciones.



Aspectos importantes del desarrollo

MVVM

- Se ha utilizado el patrón de diseño MVVM para la arquitectura de la aplicación.
 - o Model: Encargado de la definición y la persistencia de los datos.
 - View : Encargado de la presentación de los datos.
 - o ViewModel: Encargado de la lógica de negocio y la comunicación entre Model y View.

Servicios

CryptoRetrieveService

• Encargado de la obtención de datos de la API de CoinGecko.

NotificationService

• Encargado de la gestión de notificaciones.

Persistencia

- SwiftData para la persistencia de las alertas y las criptomonedas guardadas.
- Apple Keychain para la persistencia de la API Key.
- UserDefaults para la persistencia de las preferencias de usuario.

Dependencias

• KeyChainAccess para una API simplificada de Apple Keychain.

Version Control

- Se ha utilizado Git para el control de versiones.
- Se han creado diferentes ramas para el desarrollo de las funcionalidades v1, v2, v3, v4.

