# Stock

# Android Estudio V3.0.17/Flutter

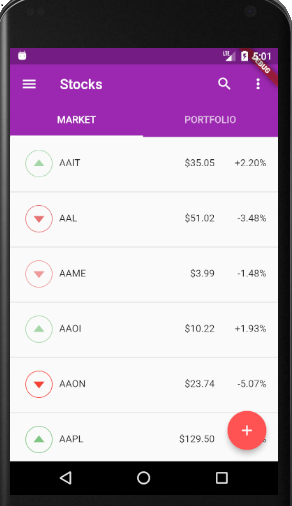
Objetivo: Revisar conceptos de una aplicación entera buscando instrucciones sueltas

F.Creación: 17-4-2018

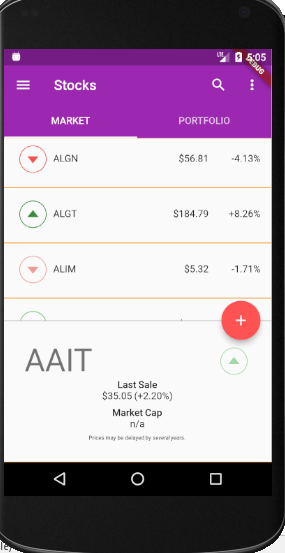
Programa: Viene con Flutter

Fuente: Para ejecutarlo se va a donde esta instalado Flutter, C:\Users\acer\flutter\examples\stocks. Al mover stock de lugar no me funciono

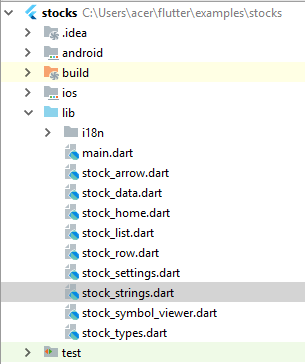
Presentación



Doble click

P

Librerias



## Análisis de main.dart

**library** stocks;  
  
**import 'dart:async'**;  
  
**import 'package:flutter/material.dart'**;  
**import 'package:flutter/rendering.dart' show** debugPaintSizeEnabled,  
 debugPaintBaselinesEnabled,  
 debugPaintLayerBordersEnabled,  
 debugPaintPointersEnabled,  
 debugRepaintRainbowEnabled;  
**import 'package:flutter\_localizations/flutter\_localizations.dart'**;  
  
**import 'stock\_data.dart'**;  
**import 'stock\_home.dart'**;  
**import 'stock\_settings.dart'**;  
**import 'stock\_strings.dart'**;  
**import 'stock\_symbol\_viewer.dart'**;  
**import 'stock\_types.dart'**;

Library🡪Declara una libreria

Show🡪 sirve para ponerle un prefijo a las librerias , para evitar colisiones.Naturalmente no pueden existir dos prefijos iguales. No se aprecia por que hay varias variables en el show, no son librerias de rendering.dart. tal vez sea casualidad pero, dentro de rendering aparece foundation.dart exportada con las mismas variables, pero los nombres no corresponden

export 'package:flutter/foundation.dart' show

VoidCallback,

ValueChanged,

ValueGetter,

ValueSetter,

DiagnosticLevel;

**class** \_StocksLocalizationsDelegate **extends** LocalizationsDelegate<StockStrings> {🡪

La **delegación** es un mecanismo, usado en la [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), por medio de la cual una clase delega en otra una determinada funcionalidad.

Se aplica como sustitución a la herencia. Como valor añadido, permite, combinado con la herencia múltiple de interfaces, sustituir la herencia múltiple de clases en los lenguajes donde no se permite esta última directamente. Además, los conflictos de nombres que se plantean en la herencia múltiple se resuelven manualmente con esta técnica.

Es una buena costumbre usar la delegación en lugar de la herencia si no es necesario hacer conversiones hacia arriba. En general, es mejor cualquier tipo de composición, si es factible, en lugar de la herencia. Sin embargo solo es una buena elección si simplifica más de lo que complica.[Ver](https://es.wikipedia.org/wiki/Delegaci%C3%B3n_(inform%C3%A1tica))

El tipo del delegado es usado para identificar el objeto creado por un metodo individual de recarga del delegado [Ver](https://docs.flutter.io/flutter/widgets/LocalizationsDelegate-class.html)

Delega la clase StockStrings en \_StocksLocalizationsDelegate

Future<StockStrings> load(Locale locale) => StockStrings.*load*(locale);🡪 Un futuro implica asincronia, define la función load que esta en StockString

**static** Future<StockStrings> *load*(Locale locale) {  
 **return** initializeMessages(locale.toString())  
 .then((Object \_) {  
 **return new** StockStrings(locale);  
 });  
}

StockConfiguration **\_configuration** = **new** StockConfiguration(🡪

Esta en el archivo stock\_types

**enum** StockMode { **optimistic**, **pessimistic** }  
**enum** BackupMode { **enabled**, **disabled** }

configurando el tema

ThemeData **get theme** {  
 **switch** (**\_configuration**.**stockMode**) {  
 **case** StockMode.**optimistic**:  
 **return new** ThemeData(  
 brightness: Brightness.**light**,  
 primarySwatch: Colors.*purple* );  
 **case** StockMode.**pessimistic**:  
 **return new** ThemeData(  
 brightness: Brightness.**dark**,  
 accentColor: Colors.*redAccent* );  
 }  
 **assert**(**\_configuration**.**stockMode** != **null**);  
 **return null**;  
}

Route<**dynamic**> \_getRoute(RouteSettings settings) {🡪

Route: Una route es una abstracción para una pantalla ó página de una app y un Navigator es un widget que maneja Router, enruta mediante push y un pop

Esta función válida y devuelve una ruta válida o null. StockData es una clase que sera vista más tarde, cuya instancia es stock

Route<**dynamic**> \_getRoute(RouteSettings settings) {  
 *// Routes, by convention, are split on slashes, like filesystem paths.* **final** List<String> path = settings.**name**.split(**'/'**);  
 *// We only support paths that start with a slash, so bail if  
 // the first component is not empty:* **if** (path[0] != **''**)  
 **return null**;  
 *// If the path is "/stock:..." then show a stock page for the  
 // specified stock symbol.* **if** (path[1].startsWith(**'stock:'**)) {  
 *// We don't yet support subpages of a stock, so bail if there's  
 // any more path components.* **if** (path.**length** != 2)  
 **return null**;  
 *// Extract the symbol part of "stock:..." and return a route  
 // for that symbol.* **final** String symbol = path[1].substring(6);  
 **return new** MaterialPageRoute<**void**>(  
 settings: settings,  
 builder: (BuildContext context) => **new** StockSymbolPage(symbol: symbol, stocks: **stocks**),  
 );  
 }  
 *// The other paths we support are in the routes table.* **return null**;

localizationsDelegates: <LocalizationsDelegate<**dynamic**>>[  
 **new** \_StocksLocalizationsDelegate(),  
 GlobalMaterialLocalizations.*delegate*,  
 GlobalWidgetsLocalizations.*delegate*,🡪  
]

Delegados que producen [WidgetsLocalizations](https://docs.flutter.io/flutter/widgets/WidgetsLocalizations-class.html) y [MaterialLocalizations](https://docs.flutter.io/flutter/material/MaterialLocalizations-class.html) son incluidos automaticamente, apps pueden proporcionar sus propias versiones de estas localizaciones por crear implementaciones de [LocalizationsDelegate](https://docs.flutter.io/flutter/widgets/LocalizationsDelegate-class.html) [LocalizationsDelegate](https://docs.flutter.io/flutter/widgets/LocalizationsDelegate-class.html) cuyo método retorna versiones personalizadas de [WidgetsLocalizations](https://docs.flutter.io/flutter/widgets/WidgetsLocalizations-class.html) or [MaterialLocalizations](https://docs.flutter.io/flutter/material/MaterialLocalizations-class.html).

## Stock\_arrow.dart

Se usa la función canvas

## Stock\_data.dart

**import 'package:http/http.dart' as** http;🡪permite usar un prefijo comodo

**final** math.Random \_rng = **new** math.Random();

**percentChange** = (\_rng.nextDouble() \* 20) - 10;🡪número aleatorio

http

**\_httpClient** = **new** http.Client();

**\_httpClient**.get(\_urlToFetch(**\_nextChunk**++)).then<Null>((http.Response response) {  
 **final** String json = response.**body**;  
 **if** (json == **null**) {  
 debugPrint(**'Failed to load stock data chunk** ${**\_nextChunk** - 1}**'**);  
 \_end();  
 **return**;  
 }  
 **const** JsonDecoder decoder = **const** JsonDecoder();  
 add(decoder.convert(json));  
 **if** (**\_nextChunk** < *\_kChunkCount*) {  
 \_fetchNextChunk();  
 } **else** {  
 \_end();  
 }  
});

}

**\_httpClient**?.close();  
**\_httpClient** = **null**;

String \_urlToFetch(int chunk) {  
 **return 'https://domokit.github.io/examples/stocks/data/stock\_data\_**$chunk**.json'**;  
}

## Stock\_home.dart

**final** GlobalKey<ScaffoldState> **\_scaffoldKey** = **new** GlobalKey<ScaffoldState>();🡪 globalkey se usa para guardar valores únicos con una llave. ScaffoldState es el estado Para un Scaffold

cuelga de un appbar

**new** PopupMenuButton<\_StockMenuItem>(  
 onSelected: (\_StockMenuItem value) { \_handleStockMenu(**context**, value); },  
 itemBuilder: (BuildContext context) => <PopupMenuItem<\_StockMenuItem>>[  
 **new** CheckedPopupMenuItem<\_StockMenuItem>(  
 value: \_StockMenuItem.**autorefresh**,  
 checked: **\_autorefresh**,  
 child: **const** Text(**'Autorefresh'**),  
 ),  
 **const** PopupMenuItem<\_StockMenuItem>(  
 value: \_StockMenuItem.**refresh**,  
 child: **const** Text(**'Refresh'**),  
 ),  
 **const** PopupMenuItem<\_StockMenuItem>(  
 value: \_StockMenuItem.**speedUp**,  
 child: **const** Text(**'Increase animation speed'**),  
 ),  
 **const** PopupMenuItem<\_StockMenuItem>(  
 value: \_StockMenuItem.**speedDown**,  
 child: **const** Text(**'Decrease animation speed'**),  
 ),  
 ],  
),

appBar: **\_isSearching** ? buildSearchBar() : buildAppBar(),🡪choro

## Stock\_list.dart

## Stock\_row.dart

GestureTapCallback \_getHandler(StockRowActionCallback callback) {  
 **return** callback == **null** ? **null** : () => callback(**stock**);🡪 Firma para cuando se hace un tap.  
}

## Stock\_setting.dart

**final** ValueChanged<StockConfiguration> **updater**;🡪Firma para callback que reportan un cambio aun valor subyacente

## Stock\_string.dart

**static** Future<StockStrings> *load*(Locale locale) {  
 **return** initializeMessages(locale.toString())  
 .then((Object \_) {  
 **return new** StockStrings(locale);  
 });  
}🡪 se hara un trabajo en 1.31 locale

## Stock\_symbol\_viewer.dart

**const** <TextSpan>[  
 **const** TextSpan(text: **'several'**, style: **const** TextStyle(fontStyle: FontStyle.**italic**)),  
 **const** TextSpan(text: **' years.'**),  
]🡪

Un [TextSpan](https://docs.flutter.io/flutter/painting/TextSpan-class.html) puede ser estilizado. Al igual que sus hijos

## Stock\_types.dart