

# **Activitat d'avaluació 1.4 - Crea una app amb connexió a backend**

**Postal Code Info**

CIFO L'Hospitalet  
Desenvolupament d'Aplicacions mòbils per iOS i Android amb Flutter

Alumne  
Carles Lázaro Costa

Tutor  
Eduard Carreras

11 d'octubre de 2024

# Índex

<b>Postal Code</b>	<b>3</b>
Instal·lació . . . . .	4
Estructura de l'aplicació . . . . .	4
Notes del desenvolupament . . . . .	6
Imatges . . . . .	9
Recursos . . . . .	26
Llibreries externes . . . . .	26

## Postal Code

Repositori del codi: [cifo\\_flutter\\_postal\\_code](#)

Repositori d'aplicacions: [cifo\\_flutter](#)

Activitat d'accés a una API externa utilitzant Flutter.

Hi ha dues pantalles, una primera per buscar la localitat d'un codi postal en un país determinat, i una segona per buscar a partir d'un nom les localitats que contenen el nom introduït amb els seus codis postals, juntament amb un selector de país i regió.

## Instal·lació

1. S'ha d'haver instal·lat el [Flutter SDK](#).

2. Clonar el repositori:

```
git clone https://github.com/Carleslc/cifo_flutter_postal_code.git  
# GitHub CLI: gh repo clone Carleslc/cifo_flutter_postal_code
```

```
cd cifo_flutter_postal_code
```

3. Instal·lar les dependències:

```
flutter pub get
```

4. Executar l'aplicació amb `flutter run` o desde l'IDE.

## Estructura de l'aplicació

```
lib  
  main.dart  
  models  
    data_model.dart  
    iso_country.dart  
    postal_places.dart  
  providers  
    country_provider.dart  
  screens  
    place_by_code_page.dart  
    place_by_code_results.dart  
    place_screen.dart  
    places_by_name_page.dart  
    places_by_name_results.dart  
  services  
    country_service.dart  
    zippopotam_service.dart  
  styles  
    app_styles.dart  
    color_styles.dart  
    text_styles.dart  
  utils  
    error.dart  
    logger.dart  
    strings.dart  
  widgets  
    map_button.dart
```

```

    place_info.dart
    places_list.dart
    postal_place_info.dart
    searching_indicator.dart
    selectors
      country_selector.dart
      selector.dart
      subdivision_selector.dart
    text_fields
      place_name_text_field.dart
      post_code_text_field.dart
      text_label_value.dart
  assets
    iso-3166-ca.json

```

L'inici de l'aplicació és a `main.dart`.

A la carpeta **models** es troben les classes de domini i serialització, com `PostalPlacesByCode`, `PostalPlacesByName`, `Country` i `Subdivision`, entre d'altres.

Les pantalles de l'aplicació es troben a la carpeta **screens**. La pantalla principal és `place_screen.dart`, que té dues pàgines `PlaceByCodePage` (`place_by_code_page.dart`) i `PlacesByNamePage` (`places_by_name_page.dart`). La búsqueda i els resultats es mostren a les pantalles `PlaceByCodeResults` (`place_by_code_results.dart`) o `PlacesByNameResults` (`places_by_name_results.dart`), respectivament.

A **providers** es guarda l'estat global de l'aplicació del país seleccionat per l'usuari, a `country_provider.dart`.

A la carpeta **services** hi ha el servei `CountryService`, que llegeix el fitxer `assets/iso-3166-ca.json` que conté els noms i codis dels països i les seves regions, i el servei `ZippopotamService` que accedeix a la API externa per obtenir codis postals i localitats, i conté informació sobre els països suportats per la API.

A la carpeta **styles** es troben els estils de l'aplicació, que s'accedeixen mitjançant la classe `AppStyles` definida a `app_styles.dart`. Les altres classes d'estils són per organitzar millor el codi i definir el tema de colors i textos.

A **widgets** hi ha els widgets propis que no es corresponen amb una pantalla determinada, que ajuden a mantenir el codi organitzat.

`PlacesList` mostra la llista de llocs, amb la capçalera `PostalPlaceInfo` que mostra informació compartida per les localitats, i cada localitat és un `PlaceInfo` que mostra la informació de la localitat individual.

A `widgets/text_fields` hi ha els widgets per l'entrada de text per part de l'usuari, per al codi postal `PostCodeTextField` i per al nom `PlaceNameTextField`.

A widgets/selectors hi ha els selectors de país `CountrySelector` i regió `SubdivisionSelector`, que utilitzen el selector genèric `Selector<T>` de `selector.dart`.

Hi han altres widgets d'utilitat com `TextLabelValue` o `SearchingIndicator`.

A `utils` hi ha mètodes auxiliars per tractar strings `strings.dart` i per mostrar missatges d'error `error.dart` o missatges de debug `logger.dart`.

## Notes del desenvolupament

Per començar l'aplicació he llegit el pdf [Activitat d'avaluació 1.4 - Crea una app amb connexió a backend.pdf](#) per tenir una idea general de com funciona la API de [Zippopotam](#) i les tasques a desenvolupar.

Després he investigat la pàgina de la API i he provat els exemples `https://api.zippopotam.us/es/08900` i `https://api.zippopotam.us/es/ct/hospi`, entre d'altres, per veure quin és el format que retornen les dues crides, una per buscar la localitat d'un codi postal i l'altre per buscar les diferents localitats amb el seus codis postals a partir d'un nom.

Les classes d'estils `AppStyles`, `ColorStyles` i `TextStyles` s'han copiat de l'anterior aplicació [1.3](#) i s'han modificat per adaptar el tema dels widgets utilitzats. La paleta de colors s'ha definit mitjançant `ColorScheme.fromSeed` i `Colors.blueGrey`.

He definit els models del fitxer `models/postal_places.dart`, primerament utilitzant la pàgina [quicktype](#) i després modificant el resultat perquè utilitzi herència, ja que en les dues crides els atributs dels llocs són els mateixos, però estan disposats amb una estructura diferent. He creat la classe base `PostalPlaces` amb els atributs comuns a les dues crides (`country`, `countryAbbreviation`, `places`) i les classes `PostalPlacesByCode` i `PostalPlacesByName` que afegeixen els atributs diferents per cada cas d'ús. La llista de llocs utilitza una estructura similar, amb `Place` com a classe base pels atributs comuns a un lloc (`name`, `latitude`, `longitude`) i les classes `PlaceByCode` i `PlaceByName` per afegir els atributs diferents de cada crida.

Després he creat el servei `ZippopotamService` per afegir els dos tipus de cerca, per codi postal i per nom, utilitzant els models definits prèviament.

He investigat com funcionen els paràmetres dels codis de país i regió, que utilitzen l'estàndard [ISO-3166](#) ([ISO-3166-1-alpha2](#) pel codi de país de dos dígitos com es per Espanya, i la segona part del codi [ISO-3166-2](#) pel codi de la regió, com ct per Catalunya). A l'apartat [Countries Supported](#) de la pàgina de [Zippopotam](#) es mostren quins codis de país es poden utilitzar, però no hi ha gaire informació sobre les regions. Més informació a [geonames.org](#). Per generar els codis de països amb les seves regions (subdivisions) he creat un script amb Python a `scripts/iso3166.py`, que utilitzant la llibreria [pycountry](#) genera un fitxer amb els països i les seves regions, que he afegit a `assets/iso-3166-ca.json` i definit a `pubspec.yaml` perquè estigui disponible a l'aplicació. Aquest fitxer es llegeix mitjançant el servei `CountryService` i els països es filtren per guardar només els suportats mitjançant el mètode `loadSupportedCountries()` definit a `ZippopotamService`, que es crida a l'inici de l'aplicació a `main.dart` perquè estiguin disponibles pels selectors de país i de regió. D'aquesta manera quan

es renderitza la pantalla principal ja apareix el selector, amb el país per defecte seleccionat i el nom traduït al català.

S'han afegit les dues pantalles de búsqueda mitjançant un `PageView` dins de la pantalla principal `PlaceScreen`.

La primera pàgina és `PlaceByCodePage` (`screens/place_by_code_page.dart`) i conté un selector de país mitjançant el widget `CountrySelector`, un `text field` per introduir el codi postal amb el widget `PostCodeTextField` (`widgets/text_fields/post_code_text_field.dart`), i un botó `ElevatedButton` per buscar resultats. També funciona mitjançant l'acció `onSubmit` del teclat a l'introduir el codi postal.

S'ha afegit validació del format del codi postal espanyol, per verificar que té 5 dígits numèrics i que està dins del rang que es mostra a l'apartat *Countries Supported* de la pàgina de *Zippopotam*. Es podria millorar ampliant aquesta comprovació per la resta de països, que actualment només verifica que no sigui buit i es limita la mida màxima a 10 caràcters. Pels països amb codi postal on tots els caràcters son numèrics s'utilitza el teclat numèric, per la resta el teclat de text d'adreça.

La pantalla de resultats és `PlaceByCodeResults` (`screens/place_by_code_results.dart`), que utilitza un `FutureBuilder` per fer la crida asíncrona `fetchPlacesByCode` utilitzant el servei `ZippopotamService` i gestionar la resposta. Els resultats es mostren amb el widget `PlacesList`, que mostra una capçalera amb el widget `PostalPlaceInfo` per la informació comuna a tots els llocs trobats, i després una llista `ListView` amb els llocs, on cada lloc utilitza el widget `PlaceInfo` per mostrar la seva informació com nom i regió. Generalment s'espera que només hi hagi un resultat, però aquest widget es pot reutilitzar per més resultats com en el següent cas.

La segona pàgina és `PlacesByNamePage` (`screens/places_by_name_page.dart`), que és similar a l'anterior però afegeix el selector de regió `SubdivisionSelector` i el `text field` permet introduir text fins a 100 caràcters utilitzant el widget `PlaceNameTextField` (`widgets/text_fields/place_name_text_field.dart`).

La pantalla de resultats és `PlacesByNameResults` (`places_by_name_results.dart`), que igual que a l'anterior pantalla utilitza un `FutureBuilder` per fer la crida asíncrona `fetchPlacesByName` utilitzant el servei `ZippopotamService` i gestionar la resposta. Es reutilitza el mateix widget `PlacesList` per mostrar els resultats, ja que requereix un objecte `PostalPlaces` que correspon a la classe base dels dos casos, i dins dels widgets `PostalPlaceInfo` i `PlaceInfo` es decideix quina informació s'ha de mostrar a la fila en funció del tipus específic de dades, `PostalPlacesByCode` / `PostalPlacesByName` i `PlaceByCode` / `PlaceByName`. En el cas d'aquesta segona pantalla que busca per nom, es mostra el nom de la localitat i el seu codi postal. Aquesta informació es pot copiar ja que s'utilitza un `SelectableText`.

També hi ha un botó `MapButton` (`widgets/map_button.dart`) a cada localitat dels resultats que obre l'aplicació de Google Maps amb la ubicació de la localitat utilitzant la llibreria `url_launcher`. El botó té un `Tooltip` que mostra les coordenades si es fa una pulsació llarga.

Per tal de que es comparteixi i es sincronitzi la selecció del país entre les dues pantalles s'utilitza el `provider` `CountryProvider`, inicialitzat a `main.dart` i utilitzat amb `Consumer<CountryProvider>`

a `PlaceByCodePage` i `PlacesByNamePage` amb `setCountry` i `setSubdivision` que criden a `notifyListeners`.

Les principals dificultats que he trobat han sigut modelar les dades de la API perquè utilitzin herència degut a que la majoria d'atributs es repeteixen en les dues crides de la API però tenen una estructura diferent, i modelar els països i les seves regions, utilitzant el fitxer generat `assets/iso-3166-ca.json` i les classes de models `iso_country.dart`, per fer els selectors de país i regió, que comparteixen l'estat entre pantalles mitjançant el provider `CountryProvider`. També he trigat temps en organitzar el codi, com per exemple utilitzant `Selector<T>` pels selectors i widgets d'utilitat com `TextLabelValue`, i els widgets de les dades `PostalPlaceInfo` i `PlaceInfo`.

Es podria seguir millorant el codi per exemple generalitzant les classes dels diferents `text_fields`, que són similars. Probablement també es podrien refactoritzar els fitxers `place_by_code_page.dart` i `place_by_name_page.dart` ja que la seva estructura és molt similar, al igual que les pàgines de resultats corresponents.

També es podria investigar com inicialitzar els països d'una manera diferent que a `main.dart`, ja que actualment no s'inicia l'aplicació fins que es carreguen els països suportats, i pot ser un procés una mica lent.


He utilitzat l'IDE *Visual Studio Code* durant tot el desenvolupament, utilitzant principalment un mòbil físic Pixel 8 amb Android 14 (API 35). També he provat l'aplicació amb un emulador amb Android 10 (API 29) i mitjançant l'IDE web [Project IDX](#).



## Postal Code Info

### Buscar per codi postal


Introdueix un codi postal a continuació i busca informació sobre la localitat a la que pertany.

 Espanya

Selecciona el país del codi postal


# Codi postal

Introdueix el codi postal0/5

 Buscar codi postal

123

Codi postal



Buscar per nom

## Postal Code Info

### Buscar per codi postal

Introdueix un codi postal a continuació i busca informació sobre la localitat a la que pertany.



Espanya



Selecciona el país del codi postal

# 08901

Introdueix el codi postal

5/5



Buscar codi postal (08901)

1

2

3

–

4

5

6

⌋

7

8

9

⌫

,

0

.

✓



Figura 2: postal\_code\_2.png



## Codi postal

País: **Espanya (ES)**

---

Codi postal: **08901**

**L Hospitalet De Llobregat**

Catalunya (CT)



## Postal Code Info

### Buscar per codi postal

Introdueix un codi postal a continuació i busca informació sobre la localitat a la que pertany.

 Andorra

Selecciona el país del codi postal

# AD500

Introdueix el codi postal

5/10

 Buscar codi postal (AD500)

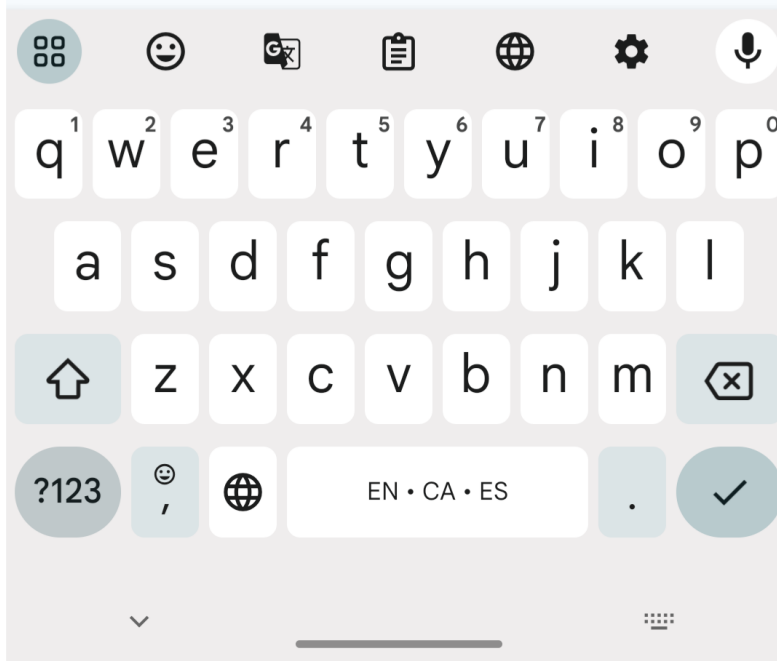


Figura 4: postal\_code\_4.png



Figura 5: postal\_code\_5.png

## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Espanya



Selecciona el país on buscar localitats



Catalunya



Selecciona la regió on buscar localitats



Nom

Introdueix el nom a buscar

0/100



Buscar localitats

123

Codi postal



Buscar per nom

Figura 6: postal\_code\_6.png

## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Espanya



Selecciona el país on buscar localitats



Catalunya



Selecciona la regió on buscar localitats



hospi



Introdueix el nom a buscar

5/100



Buscar localitats (hospi)

123

Codi postal




Buscar per nom

Figura 7: postal\_code\_7.png

← ES-CT  hospi

País: **Espanya (ES)**

Regió: *Catalunya (CT)*

 L Hospitalet De Llobregat

**L Hospitalet De Llobregat**

08900



**L Hospitalet De Llobregat**

08901



**L Hospitalet De Llobregat**

08902



**L Hospitalet De Llobregat**

08903



**L Hospitalet De Llobregat**

08904



**L Hospitalet De Llobregat**

08905



**L Hospitalet De Llobregat**

08906



**L Hospitalet De Llobregat**

08907



**L Hospitalet De Llobregat**

08908



Figura 8: postal\_code\_8.png



## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Espanya



Selecciona el país on buscar localitats



Catalunya



Selecciona la regió on buscar localitats



gran



Introdueix el nom a buscar

4/100



Buscar localitats (gran)

123

Codi postal



Buscar per nom

Figura 9: postal\_code\_9.png

← ES-CT  gran

País: **Espanya (ES)**

Regió: *Catalunya (CT)*

 Granera

**Granera**

08183



**Granollers**

08400



**Granollers**

08400



**Granollers**

08401



**Granollers**

08402



**Granollers**

08403



**Granollers**

08404



**La Granada**

08792



**Granollers De Rocacorba**

17153



Figura 10: postal\_code\_10.png

## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Espanya



Selecciona el país on buscar localitats



Andalusia



Selecciona la regió on buscar localitats

Andalusia

AN

Aragó

AR

Astúries

AS

Cantàbria

CB

Canàries

CN

Castella - la Manxa

CM

Castella i Lleó

CL

Catalunya

CT

Figura 11: postal\_code\_11.png

## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Espanya



Selecciona el país on buscar localitats



Andalusia



Selecciona la regió on buscar localitats



grana



Introdueix el nom a buscar

5/100



Buscar localitats (grana)

123

Codi postal



Buscar per nom

Figura 12: postal\_code\_12.png

← ES-AN 🇪🇸 grana

País: **Espanya (ES)**

Regió: *Andalusia (AN)*

📖 Valdelagrana

**Valdelagrana**

11500



**Granadilla**

14950



**Granada**

18001



**Granada**

18002



**Granada**

18003



**Granada**

18004



**Granada**

18005



**Granada**

18006



**Granada**

18007



Figura 13: postal\_code\_13.png









## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.

Selecciona el país on buscar localitats

	Eslovènia	SI
	Espanya	ES
	Estats Units	US
	Filipines	PH
	Finlàndia	FI
	França	FR
	Groenlàndia	GL
	Guadalupe	GP

123

Codi postal



Buscar per nom

Figura 14: postal\_code\_14.png

## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Estats Units



Selecciona el país on buscar localitats



California



Selecciona la regió on buscar localitats

Alabama

AL

Alaska

AK

Arizona

AZ

Arkansas

AR

California

CA

Colorado

CO

Connecticut

CT

Delaware

DE

Figura 15: postal\_code\_15.png

## Postal Code Info

### Buscar per nom

Introdueix un nom a continuació i busca informació sobre les localitats que tenen aquest nom.



Estats Units



Selecciona el país on buscar localitats



California



Selecciona la regió on buscar localitats



san fra



Introdueix el nom a buscar

7/100



Buscar localitats (san fra)

123

Codi postal



Buscar per nom

Figura 16: postal\_code\_16.png



← US-CA 🇺🇸 san fra

País: **Estats Units (US)**

Regió: *California (CA)*

📖 South San Francisco

**South San Francisco**

94080



**South San Francisco**

94083



**South San Francisco**

94099



**San Francisco**

94101



**San Francisco**

94102



**San Francisco**

94103



**San Francisco**

94104



**San Francisco**

94105



**San Francisco**

94106



Figura 17: postal\_code\_17.png

## Recursos

### Zippopotam API

Codis de país:

- [ISO-3166-1 \(iso.org\) \(search\)](#)
- [ISO-3166-1 \(Wikipedia\)](#)
- [Geonames: Country Codes](#)
- [Països suportats a Zippopotam](#)
- [ISO-3166 Download: World Countries, ip2location](#)

Codis de subdivisions i regions:

- [ISO 3166-2 \(Wikipedia\)](#)
- [ISO 3166-2 Spain \(iso.org\)](#)
- [Geonames: Subdivisions \(Spain\)](#)
- [Geonames: Subdivisions \(United States\)](#)
- [ISO-3166-2 Download: World Countries, ip2location](#)

Relacionats amb Flutter:

- [JSON Serialization](#)
- [TextField](#)
- [DropDownMenu](#)
- [PageView](#)
- [BottomNavigationBar](#)
- [ListView](#)
- [SelectableText](#)
- [GlobalKey](#)
- [FutureBuilder](#)
- [ThemeData \(ColorScheme, TextTheme\)](#)
- [ChangeNotifierProvider<T>](#)
- [Consumer<T>](#)

Altres:

- [Quicktype](#)
- [Font: Montserrat](#)
- [pycountry](#)

### Llibries externes

- [http](#)
- [url\\_launcher](#)
- [google\\_fonts](#)
- [provider](#)