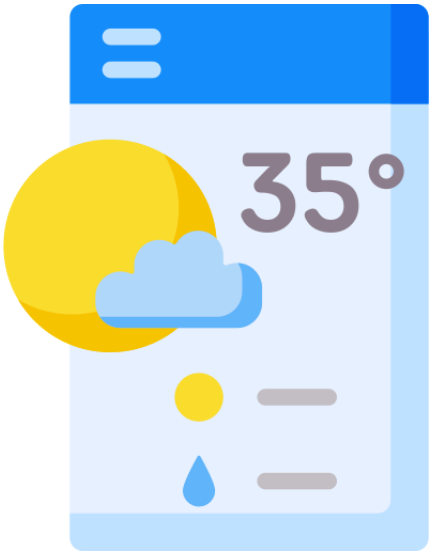
TP Météo

Flutter



Objectifs

- Affichage conditionnel,
- Emploi de formulaire,
- Programmation asynchrone (syntaxe *async / await, Future*),
- Emploi du widget *FutureBuilder*,
- Interaction avec une API REST,
- Utilisation de variables d'environnement,
- Géolocalisation,
- Emploi d'une carte.

Initialisation

- Créer un nouveau projet Flutter vierge nommé ***meteo***.

```
flutter create -e meteo
```

APIs

- L'application va devoir communiquer avec **2 API web** :
  - ***API Ninja City*** (https://api-ninjas.com/) pour obtenir les coordonnées GPS d'une ville.
  - ***API Open Weather*** (https://openweathermap.org/) pour obtenir des données météorologiques associées à un point GPS.
- Créez un compte gratuit sur ***OpenWeather*** https://openweathermap.org/ et obtenez une clé d'API.
  La clé d'API est à renseigner en tant que query string nommée ***appid***
  (ex: ?appid=xyz).

  Consultez la documentation de l'***API OpenWeather***.
- Créez un compte gratuit sur ***API Ninjas*** https://api-ninjas.com/ afin de disposer d'une clé d'API.
  La clé d'API est à renseigner dans le header ***X-Api-Key*** des requêtes HTTP.

  Consultez la documentation du End Point *"city"* (https://api-ninjas.com/api/city).
- Il est recommandé de tester les APIs au préalable avec un outil tel que ***Postman*** (https://www.postman.com/) ou ***curl*** afin de consulter la structure des réponses fournies au format JSON et des codes de statut HTTP retournés par l'API.
- Installez le package ***flutter_dotenv*** (https://pub.dev/packages/flutter_dotenv) afin d'utiliser un fichier ***.env*** contenant la valeur de chaque clé d'API (attention, ne pas inclure le fichier ***.env*** dans votre dépôt Git, cf. ***.gitignore***).

```
CITY_API_KEY='ABC1111111'

METEO_API_KEY='DEF2222222'
```

- Installez le package ***dio*** (https://pub.dev/packages/dio) pour interagir avec les API web.

  Consultez la documentation du package et le ***Cookbook*** de Flutter (https://docs.flutter.dev/cookbook#networking) pour apprendre à gérer les réponses communiquées par une API web au format JSON.

- Programmez une première ***méthode asynchrone*** (cf. https://dart.dev/codelabs/async-await) permettant d'effectuer une requête HTTP sur ***City API*** afin d'obtenir les coordonnées GPS d'une ville de votre choix, en renseignant son nom en paramètre (ex: Paris).

- Programmez une seconde ***méthode asynchrone*** (cf. https://dart.dev/codelabs/async-await) permettant d'effectuer une requête HTTP sur l'***API OpenWeather*** afin d'obtenir des informations météorologiques d'un lieu de votre choix (ex: Paris) en renseignant ses coordonnées GPS (latitude / longitude).

UI

- Mettez en place **un formulaire permettant à l'utilisateur d'obtenir les informations météorologiques d'une ville après avoir saisi son nom**.

- Gérez les éventuelles erreurs de saisie en notifiant l'utilisateur (par exemple, à l'aide d'un widget *Snackbar ou AlertDialog*).

- En arrière-plan, le programme doit effectuer **2 opérations successives** :

  - **obtention des coordonnées GPS** de la ville saisie,

  - **obtention des données météorologiques à partir des coordonnées GPS** obtenues.

- Mettez en place un widget de type ***FutureBuilder*** afin d'afficher dynamiquement le résultat des opérations asynchrones retournant des valeurs sous forme de ***Future***.

- Affichez le nom de la ville, son pays et la température en **degrés celsius**.

Map

- Installez le package ***flutter_map*** (https://pub.dev/packages/flutter_map) afin d'**afficher dynamiquement une carte centrée sur la ville saisie dans le formulaire, par le biais de ses coordonnées GPS**.

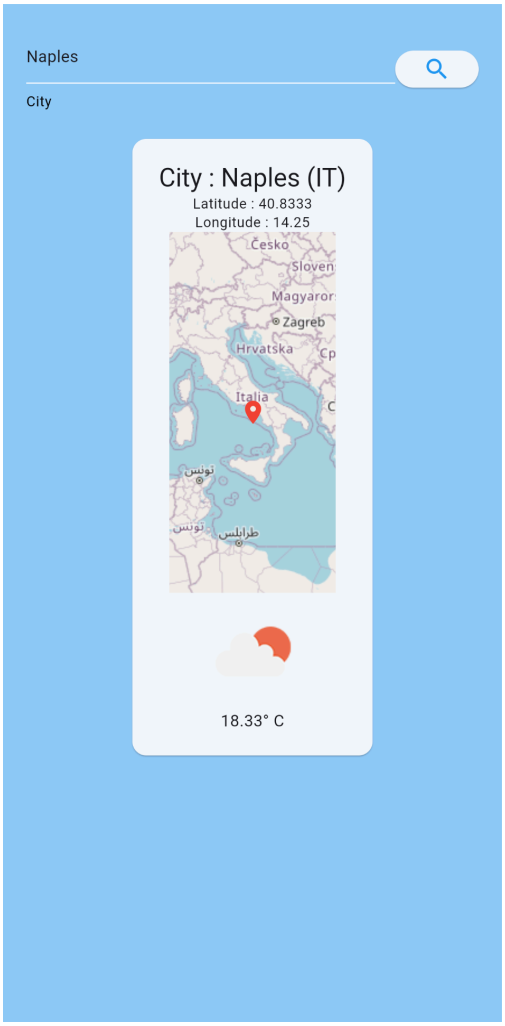
Géolocalisation

- **Proposez à l'utilisateur d'effectuer une recherche météorologique en employant ses coordonnées GPS** actuelles plutôt qu'en tapant le nom d'une ville.
- **Gérez les demandes de permissions du device mobile** afin d'accéder à la géolocalisation.
- Employez le package https://pub.dev/packages/geolocator

App Icon & Splash Screen

- Installez le package ***flutter_native_splash*** (https://pub.dev/packages/flutter_native_splash) et générez un splash screen pour l'application à l'aide des images fournies (crédits : FlatIcon.com).
- Installez le package ***flutter_launcher_icons*** https://pub.dev/packages/flutter_launcher_icons et générez une icône pour l'application à l'aide des images fournies (crédits : Flaticon.com).

Meteo App



Meteo App