

# Mòdul Professional 8

Programació multimèdia i  
dispositius mòbils

Marta Millan Lombraña  
[marta.millan@iesjoandaustria.org](mailto:marta.millan@iesjoandaustria.org)

# POKEDEX

▶ En aquesta pràctica crearem una pokedex

▶ Treballarem:

- Connexions HTTPS
- API
- JSON
- TRY & CATH



TEORIA

# Teoria. Threads. Async

- Android obliga a realitzar les connexions a través de la xarxa fora del thread principal de la nostra App. Amb això volen evitar que la interfície d'usuari es quedi bloquejada (sense refrescar-se i sense atendre a esdeveniments d'usuari) per culpa de una connexió lenta o un servidor extern no accessible.
- Hi ha diferents mètodes de tractar els Threads en Android però l'opció més bàsica que ens permetrà realitzar tasques en segon pla i després actualitzar el nostre Thread principal s'anomena `AsyncTask`.
- `AsyncTask` és una classe que permet executar un nou Thread des del Thread principal que gestiona la nostra UI.

Documentació oficial: <https://developer.android.com/reference/android/os/AsyncTask>

# Teoria. Threads. Async

- Android ens ha facilitat molt la comunicació entre els UI Thread i l'AsyncTask mitjançant la implementació dels següents mètodes.
  - `doInBackground(Params...)` — En aquest mètode escriurem el codi que s'ha d'executar en l'AsyncTask. Rebrà com a paràmetres d'entrada el que nosaltres definim (modificant Params pel tipus de variable).
  - `onProgressUpdate(Progress...)` — Aquest mètode serveix per actualitzar la UI abans de que s'acabi de finalitzar el codi de l'AsyncTask i té utilitat en threads que realitzen accions molt llargues (no serà el nostre cas).
  - `onPostExecute(Result)` — Aquest mètode es defineix en la classe AsyncTask però realment s'executarà en el thread principal, això ens permetrà des d'aquest mètode especificar els canvis a realitzar en la nostra UI.

# Teoria. Threads. Async

- Haurem de modificar l'argument Result pel tipus d'argument que retornarà la funció doInBackground.
- No podem instanciar directament la classe AsyncTask i haurem d'estendre aquesta classe amb la nostra pròpia classe.
  - `public MyAsyncTask extends AsyncTask<Params, Progress, Result>` — hem d'utilitzar els tipus definits en els anteriors mètodes per a Params, Progress i Result (en cas de no utilitzar tots els mètodes utilitzarem Void).
- Per facilitar-nos la modificació de la nostra UI des del AsyncTask ens ajudarà definir aquesta nova classe com a classe interna dins de la nostra Activity.

# Teoria. Connexions HTTP

Una funcionalitat força habitual que hauran de tenir les nostres Apps és la d'haver de realitzar connexions a través de la xarxa amb altres equips/servidors. En aquest cas ens serà de molta utilitat conèixer com es realitzen les connexions HTTP mitjançant Android.

Els passos per poder realitzar una connexió HTTP són:

1. Demanar permisos des del Manifest per utilitzar Internet des de la nostra App.
2. Implementar un nou thread on realitzar la connexió, mitjançant AsyncTask.
3. Utilitzar els blocs de Try & Catch en els punts on poden haver-hi errors.
4. Aconseguir una URL amb l'adreça a la qual volem connectar-nos. Habitualment agafarem l'adreça amb un String (ex.: `http://www.google.es`) i utilitzarem el constructor de la classe URL amb aquest String per aconseguir la URL.
5. Obrir una connexió HTTP amb aquesta URL. Això ho farem utilitzant el mètode `openConnection()` sobre l'objecte URL anterior.
6. Un cop oberta la connexió HTTP obtindrem les dades amb la informació mitjançant un `InputStream`. I a partir d'aquí programareu la lectura d'aquest `InputStream`.

# Teoria. Connexions API

Com a desenvolupadors d'Apps és molt probable que ens veiem obligats a fer ús serveis desenvolupats per altres empreses/desenvolupadors.

Aquests serveis oferiran una API en la que ens donaran informació del tipus de serveis que puc demanar i com ho haig de demanar.

Les API que permeten transferència de informació utilitzaran habitualment algun dels formats d'intercanvi d'informació més habituals (xml, json , csv, ...). Vosaltres ja heu treballat amb xml en el curs passat i per ampliar els vostres coneixements explicarem com treballar amb el format JSON.

El tipus d'API que utilitzarem serà basat en servidors web i utilitzarem connexions HTTP i expressions URL.



# Teoria. JSON

JSON (JavaScript Object Notation) és un format d'intercanvi de dades entre computadors, independent del tipus de llenguatge de programació que utilitzem, i fàcil d'entendre també pels humans.

JSON utilitza una sintaxis de tipus Key/Value (com també usen altres elements d'Android) per guardar les dades, utilitza “,” per separar diferents atributs, utilitza “{ }” per separar objectes, i utilitza “[ ]” per generar Arrays d'objectes.

Dins d'Android utilitzem les llibreries de Java per treballar amb JSON i principalment utilitzarem les següents classes.

- JSONObject — Qualsevol String que segueixi la sintaxi de JSON podrà ser interpretat com a JSONObject.
- JSONArray — Aquest tipus d'objecte ens permetrà accedir als diferents elements d'un array.

# Teoria. JSON

De moment per treballar amb APIs de 3rs utilitzarem JSON com a consumidors de dades, es a dir ens connectarem contra un servidor i realitzarem una petició que retornarà les dades en format JSON.

Per poder interpretar les dades rebudes (que nosaltres llegirem com un String) haurem de manipular aquest String per traslladar aquestes dades a objectes de Java. Alguns dels mètodes de la classe JSONObject que ens seran d'utilitat són:

- `getJSONObject(String s)` — Aquest mètode em retorna un JSONObject sobre un String amb la sintaxis correcta.
- `JSONArray(String key)` — Aquest mètode em retorna un JSONArray si li pasem la clau amb la que comença aquest Array dins del JSONObject.
- `getString(String key)` — Aquest mètode em retorna com a variable de tipus String el valor que trobem associat a la clau anomenada key del nostre JSONObject.
- `getInt(String key)` — Aquest mètode em retorna com a variable de tipus Int el valor que trobem associat a la clau anomenada key del nostre JSONObject.
- `getBoolean(String key)` — Aquest mètode em retorna com a variable de tipus Boolean el valor que trobem associat a la clau anomenada key del nostre JSONObject.



PAS A PAS

## Pràctica. API

<https://github.com/martamill3/Pokedex>

## Pràctica. API

<https://pokeapi.co/>

<https://pokeapi.co/docs/v2>

# Exercicis

## Exercicis

1. Fes que quan s'obri l'aplicació es mostri el primer pokemon de l'api
2. Fes que amb els botons puguis iterar els diferents resultats
3. Fes que quan cliquis al botó blau aparegui un alert i puguis seleccionar els diferents tipus de pokemon, a continuació fes que amb les fletxes els puguis iterar.

Documentació alerts: <https://developer.android.com/guide/topics/ui/dialogs?hl=es>

4. Implementa alguna funcionalitat extra