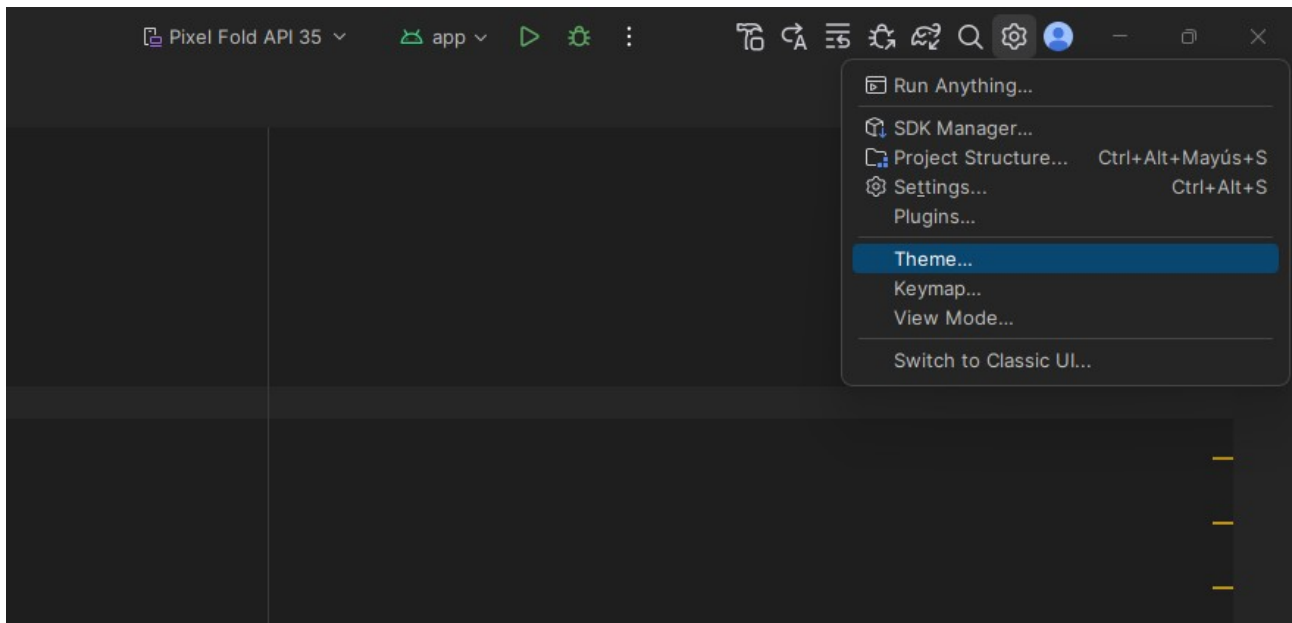


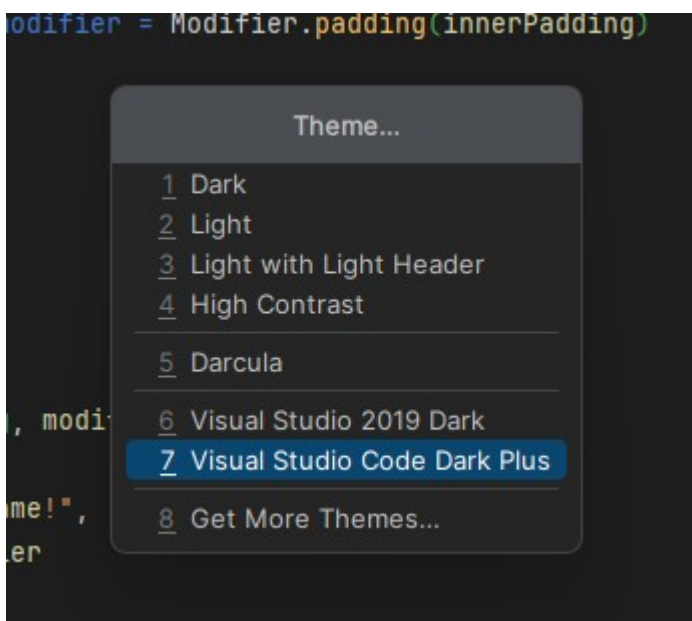
Retos iniciales Android Studio

1) Cambia el color de un tema

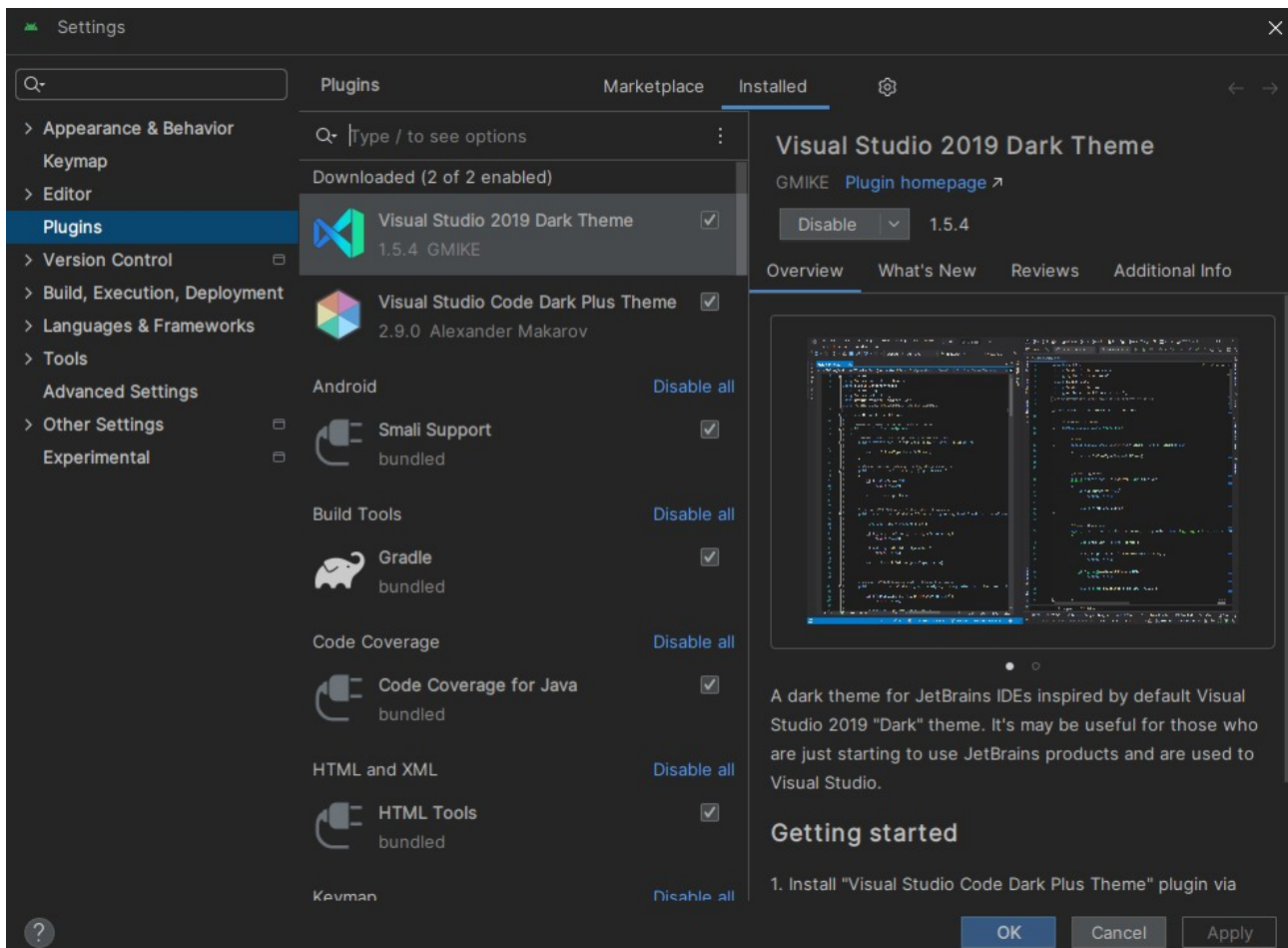
Para poder cambiar el color de un tema en Android Studio, debemos ir a la parte superior de la ventana y pulsar en “Ajustes” → “Temas”



Una vez pulsemos ahí, se nos abrirá una pequeña ventana para poder cambiarlo utilizando los que vienen instalados por defecto, o si lo preferimos, buscar más en marketplace.



2) Captura de los plugins



Settings

Q-

Appearance & Behavior

Keymap

Editor

Plugins

Version Control

Build, Execution, Deployment

Languages & Frameworks

Tools

Advanced Settings

Other Settings

Experimental

Plugins

Q- Type / to see options

[T]

Toml

bundled

✓

YAML

bundled

✓

Local AI/ML Tools

Disable all

Machine Learning Code Completion

bundled

✓

Turbo Complete

bundled

✓

Plugin Development

Disable all

Plugin DevKit

bundled

✓

Test Tools

Disable all

JUnit

bundled

✓

TestNG

bundled

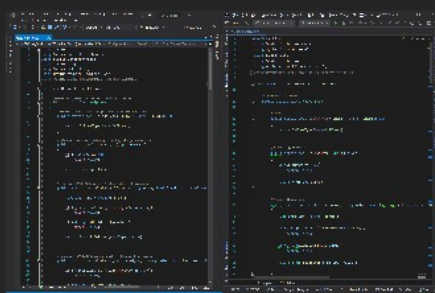
✓

Visual Studio 2019 Dark Theme

GMIKE [Plugin homepage](#)

Disable 1.5.4

Overview What's New Reviews Additional Info

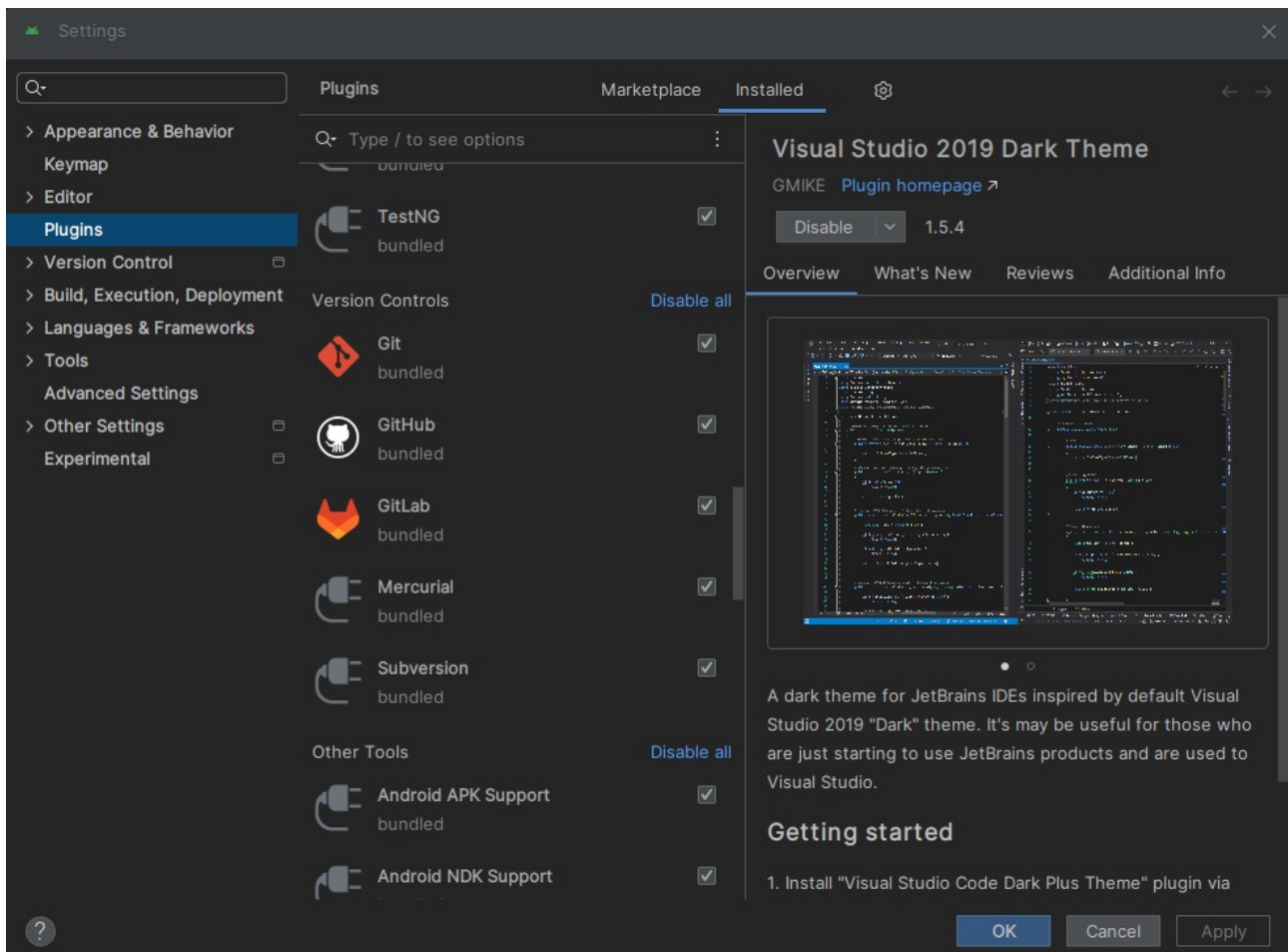


A dark theme for JetBrains IDEs inspired by default Visual Studio 2019 "Dark" theme. It's may be useful for those who are just starting to use JetBrains products and are used to Visual Studio.

Getting started

1. Install "Visual Studio Code Dark Plus Theme" plugin via

OK Cancel Apply



Settings

Q-

Plugins

Marketplace

Installed

> Appearance & Behavior

Keymap

> Editor

Plugins

> Version Control

> Build, Execution, Deployment

> Languages & Frameworks

> Tools

Advanced Settings

> Other Settings

Experimental

Q- Type / to see options

Android SDK Upgrade Assistant

bundled

✓

App Links Assistant

bundled

✓

ClangConfig

bundled

✓

Clangd Support

bundled

✓

ClangFormat

bundled

✓

Copyright

bundled

✓

Device Streaming

bundled

✓

EditorConfig

bundled

✓

Firebase Services

bundled

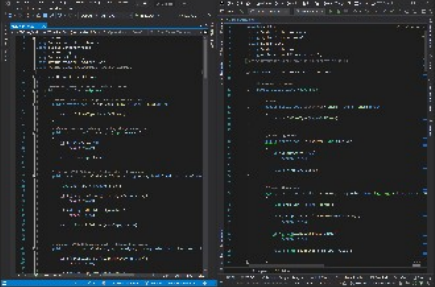
✓

Visual Studio 2019 Dark Theme

GMIKE [Plugin homepage](#)

Disable 1.5.4

Overview What's New Reviews Additional Info



A dark theme for JetBrains IDEs inspired by default Visual Studio 2019 "Dark" theme. It's may be useful for those who are just starting to use JetBrains products and are used to Visual Studio.

Getting started

1. Install "Visual Studio Code Dark Plus Theme" plugin via

OK

Cancel

Apply

Settings

Q-

Plugins

Marketplace

Installed

<

>

> Appearance & Behavior

Keymap

> Editor

Plugins

> Version Control

> Build, Execution, Deployment

> Languages & Frameworks


> Tools

Advanced Settings

> Other Settings


Experimental

Q- Type / to see options

 Gemini


bundled

✓

 Git for App Insights


bundled

✓

 Google Cloud Tools For Android St...


bundled

✓

 IntelliLang


bundled

✓

 Java Bytecode Decompiler


bundled

✓

 Java Internationalization


bundled

✓

 Java Stream Debugger


bundled

✓

 Jetpack Compose

bundled

✓

 Task Management

bundled

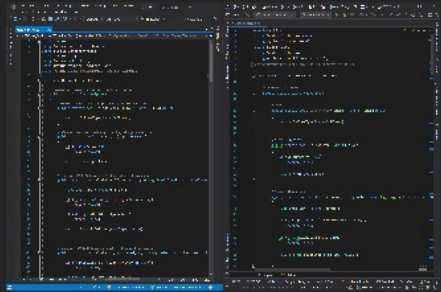
✓

Visual Studio 2019 Dark Theme

GMIKE [Plugin homepage](#)

Disable 1.5.4

Overview What's New Reviews Additional Info



A dark theme for JetBrains IDEs inspired by default Visual Studio 2019 "Dark" theme. It's may be useful for those who are just starting to use JetBrains products and are used to Visual Studio.

Getting started

1. Install "Visual Studio Code Dark Plus Theme" plugin via

OK

Cancel

Apply

3) Captura, descripción y enlace del plugin asignado (Rainbow Brackets).

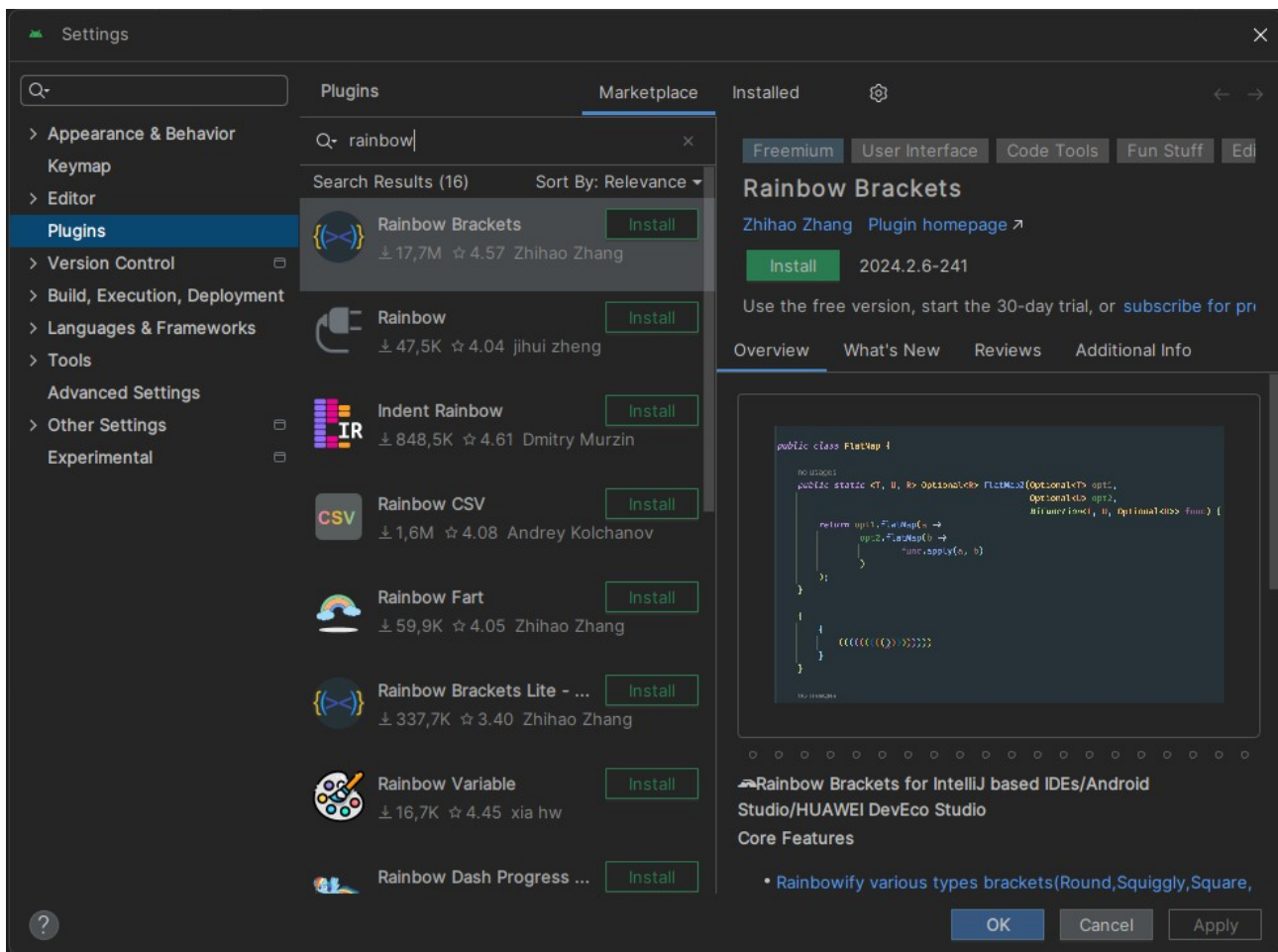


Figura 1: Captura del plugin

Rainbow Brackets es un plugin que permite colorear las diferentes etiquetas o llaves para poder tener una mayor distinción entre las etiquetas padre y sus hijas.

```

val dataContext = DataManager.getInstance().getDataContext(parent)
description?.let { errorContext.description = description }
errorContext.message = event.message
event.throwable?.let { throwable ->
    PluginUtil.getInstance().findPluginId(throwable)?.let { pluginId ->
        PluginManagerCore.getPlugin(pluginId)?.let { ideaPluginDescriptor ->
            if (!ideaPluginDescriptor.isBundled) {
                errorContext.pluginName = ideaPluginDescriptor.name
                errorContext.pluginVersion = ideaPluginDescriptor.version
            }
        }
    }
}

(event.data as? LogMessage)?.let { errorContext.attachments = it.includedAttachments }
val project = CommonDataKeys.PROJECT.getData(dataContext)
val reportValues = getKeyValuePairPairs(
    errorContext,

```

Same color

Rainbow indent guide lines

Figura 2: Ejemplo 1

```

class Doctors:
    def __init__(self):
        self.role = 'Doctor'
        self.name = 'Dany'

    def show(self):
        try:
            if self.role == 'Doctor':
                if self.name == 'Dany':
                    print('Hello Doctor')
                else:
                    print('Hello Guest')
            elif self.role == 'Guest':
                print('Hi Guest')
            else:
                raise UnknownError('unknown user...')

        except Exception as inst:
            try:
                print('error')
            except Exception as inst:
                print("??")
            finally:
                print("??")
        finally:
            print('done')

```

Nested Python keywords

Rainbow Indent guide lines

Figura 3: Ejemplo 2


```
MainActivity.kt x
1 package com.example.test
2
3 > import ...
15
16 class MainActivity : ComponentActivity() {
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         enableEdgeToEdge()
20         setContent {
21             TestTheme {
22                 Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
23                     Greeting(
24                         name = "Android",
25                         modifier = Modifier.padding(innerPadding)
26                     )
27                 }
28             }
29         }
30     }
31 }
32
33 @Composable
34 fun Greeting(name: String, modifier: Modifier = Modifier) {
35     Text(
36         text = "Hello $name!",
37         modifier = modifier
38     )
39 }
40
41 @Preview(showBackground = true)
42 @Composable
43 fun GreetingPreview() {
44     TestTheme {
45         Greeting(name = "Android")
46     }
47 }
```

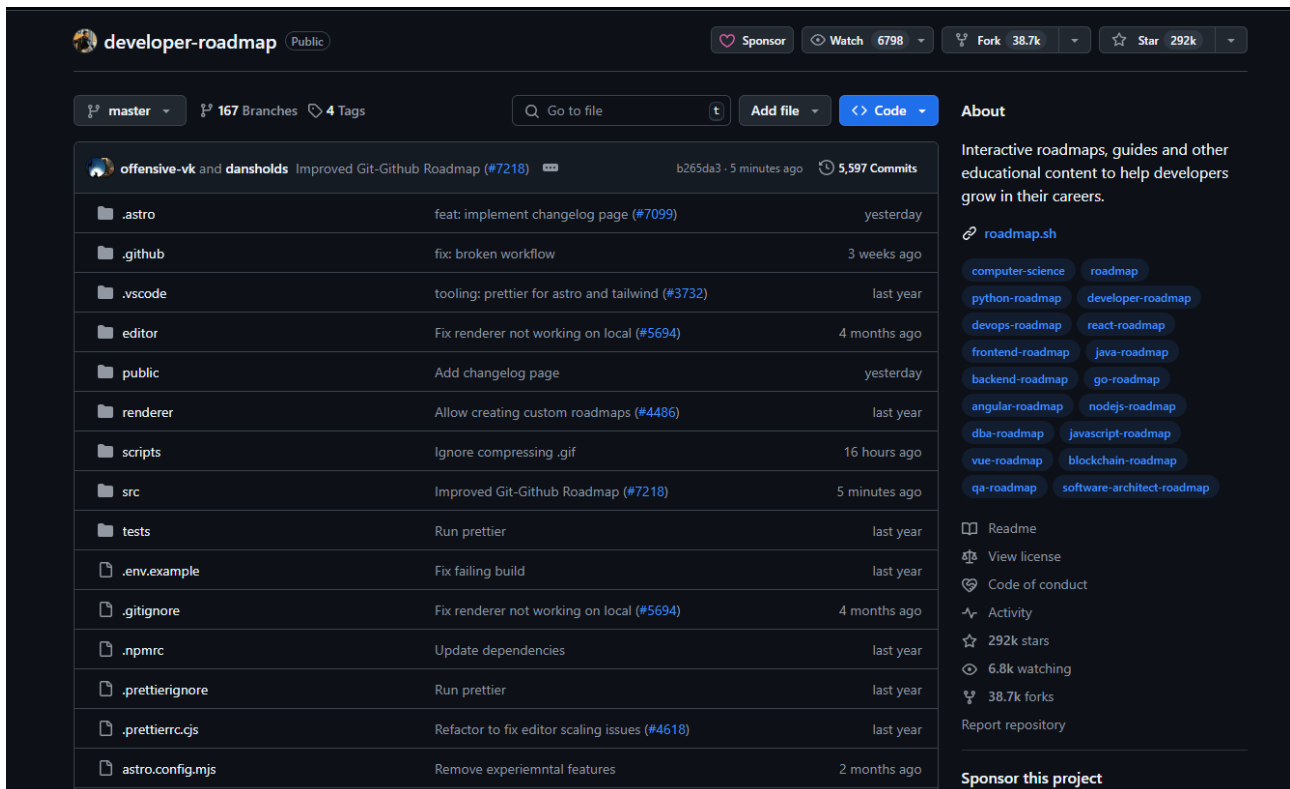
Figura 4: Ejemplo 3

Enlace: <https://plugins.jetbrains.com/plugin/10080-rainbow-brackets>

4) Repositorios de Android

1. Repositorio developer-roadmap

Este repositorio hecho con TypeScript, nos proporciona varias guías, roadmaps y otros elementos educativos para ayudar a que los desarrolladores sigan creciendo.



The screenshot shows the GitHub interface for the `developer-roadmap` repository. At the top, it displays the repository name, a 'Public' badge, and statistics: 6798 watches, 38.7k forks, and 292k stars. Below this, there's a search bar and buttons for 'Add file' and 'Code'. The main content area shows a list of files and their commit history. The right sidebar contains an 'About' section with a description, a list of related roadmaps, and a 'Sponsor this project' button.

File	Commit Message	Time Ago
<code>.astro</code>	feat: implement changelog page (#7099)	yesterday
<code>.github</code>	fix: broken workflow	3 weeks ago
<code>.vscode</code>	tooling: prettier for astro and tailwind (#3732)	last year
<code>editor</code>	Fix renderer not working on local (#5694)	4 months ago
<code>public</code>	Add changelog page	yesterday
<code>renderer</code>	Allow creating custom roadmaps (#4486)	last year
<code>scripts</code>	Ignore compressing .gif	16 hours ago
<code>src</code>	Improved Git-Github Roadmap (#7218)	5 minutes ago
<code>tests</code>	Run prettier	last year
<code>.env.example</code>	Fix failing build	last year
<code>.gitignore</code>	Fix renderer not working on local (#5694)	4 months ago
<code>.npmrc</code>	Update dependencies	last year
<code>.prettiignore</code>	Run prettier	last year
<code>.prettierrc.cjs</code>	Refactor to fix editor scaling issues (#4618)	last year
<code>astro.config.mjs</code>	Remove experiemntal features	2 months ago

About
Interactive roadmaps, guides and other educational content to help developers grow in their careers.

[roadmap.sh](#)

- computer-science roadmap
- python-roadmap
- devops-roadmap
- frontend-roadmap
- backend-roadmap
- angular-roadmap
- dba-roadmap
- vue-roadmap
- qa-roadmap
- developer-roadmap
- react-roadmap
- java-roadmap
- go-roadmap
- nodejs-roadmap
- javascript-roadmap
- blockchain-roadmap
- software-architect-roadmap

Sponsor this project

Figura 5: Repositorio de developer-roadmap

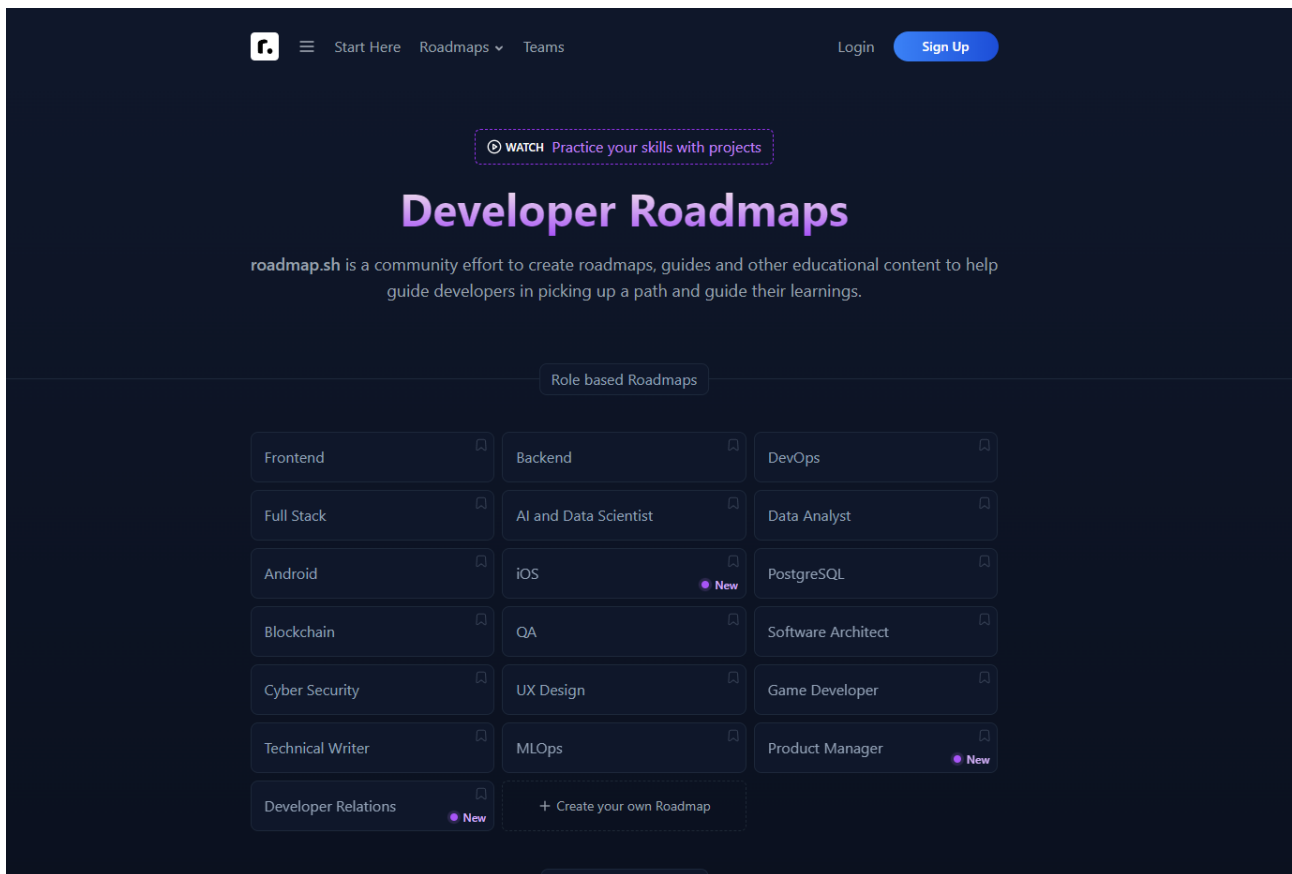


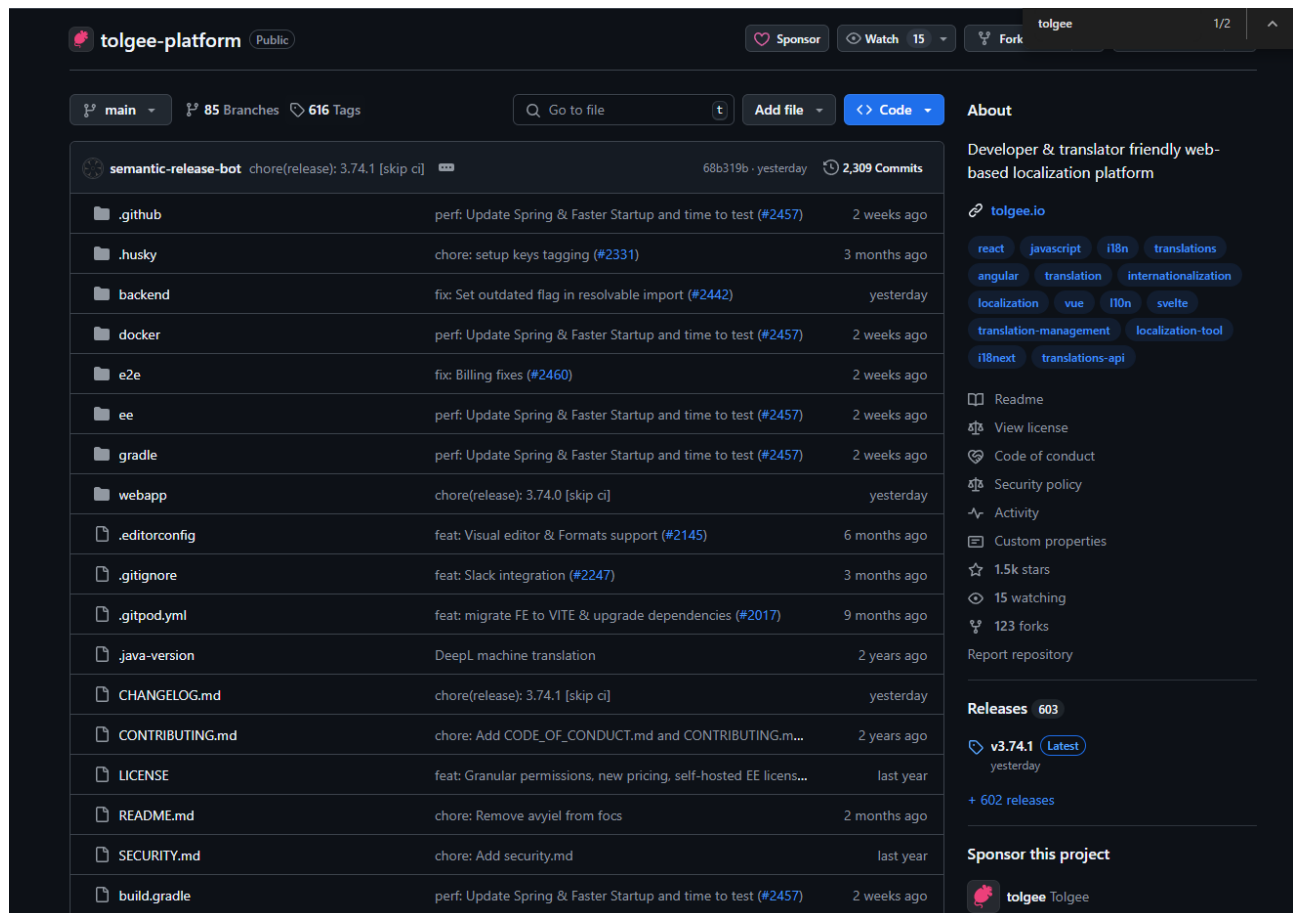
Figura 6: Página web de roadmap

Enlace al repositorio: <https://github.com/kamranahmedse/developer-roadmap>

Enlace a la web: <https://roadmap.sh/>

2. Repositorio de Tolgee

Tolgee es una app creada con Kotlin, JavaScript, React entre otras, para facilitar la traducción de aplicaciones con herramientas como edición en contexto, traducción automática o memoria de traducción.



The screenshot shows the GitHub repository for 'tolgee-platform'. The repository is public and has 85 branches and 616 tags. The main branch is selected. The repository is managed by 'semantic-release-bot' and has 2,309 commits. The repository is described as a 'Developer & translator friendly web-based localization platform'. The repository includes a README, a license, a code of conduct, a security policy, and an activity feed. The repository also has 1.5k stars, 15 watchers, and 123 forks. The repository is categorized under 'translation-management' and 'localization-tool'. The repository is also tagged with 'react', 'javascript', 'i18n', 'translations', 'angular', 'translation', 'internationalization', 'localization', 'vue', 'i10n', 'svelte', 'translation-management', 'localization-tool', 'i18next', and 'translations-api'. The repository has 603 releases, with the latest release being 'v3.74.1' from yesterday. The repository is sponsored by 'tolgee Tolgee'.

File	Commit Message	Commit Date
.github	perf: Update Spring & Faster Startup and time to test (#2457)	2 weeks ago
.husky	chore: setup keys tagging (#2331)	3 months ago
backend	fix: Set outdated flag in resolvable import (#2442)	yesterday
docker	perf: Update Spring & Faster Startup and time to test (#2457)	2 weeks ago
e2e	fix: Billing fixes (#2460)	2 weeks ago
ee	perf: Update Spring & Faster Startup and time to test (#2457)	2 weeks ago
gradle	perf: Update Spring & Faster Startup and time to test (#2457)	2 weeks ago
webapp	chore(release): 3.74.0 [skip ci]	yesterday
.editorconfig	feat: Visual editor & Formats support (#2145)	6 months ago
.gitignore	feat: Slack integration (#2247)	3 months ago
.gitpod.yml	feat: migrate FE to VITE & upgrade dependencies (#2017)	9 months ago
.java-version	DeepL machine translation	2 years ago
CHANGELOG.md	chore(release): 3.74.1 [skip ci]	yesterday
CONTRIBUTING.md	chore: Add CODE_OF_CONDUCT.md and CONTRIBUTING.m...	2 years ago
LICENSE	feat: Granular permissions, new pricing, self-hosted EE licens...	last year
README.md	chore: Remove avyiel from focs	2 months ago
SECURITY.md	chore: Add security.md	last year
build.gradle	perf: Update Spring & Faster Startup and time to test (#2457)	2 weeks ago

Figura 7: Repositorio de Tolgee

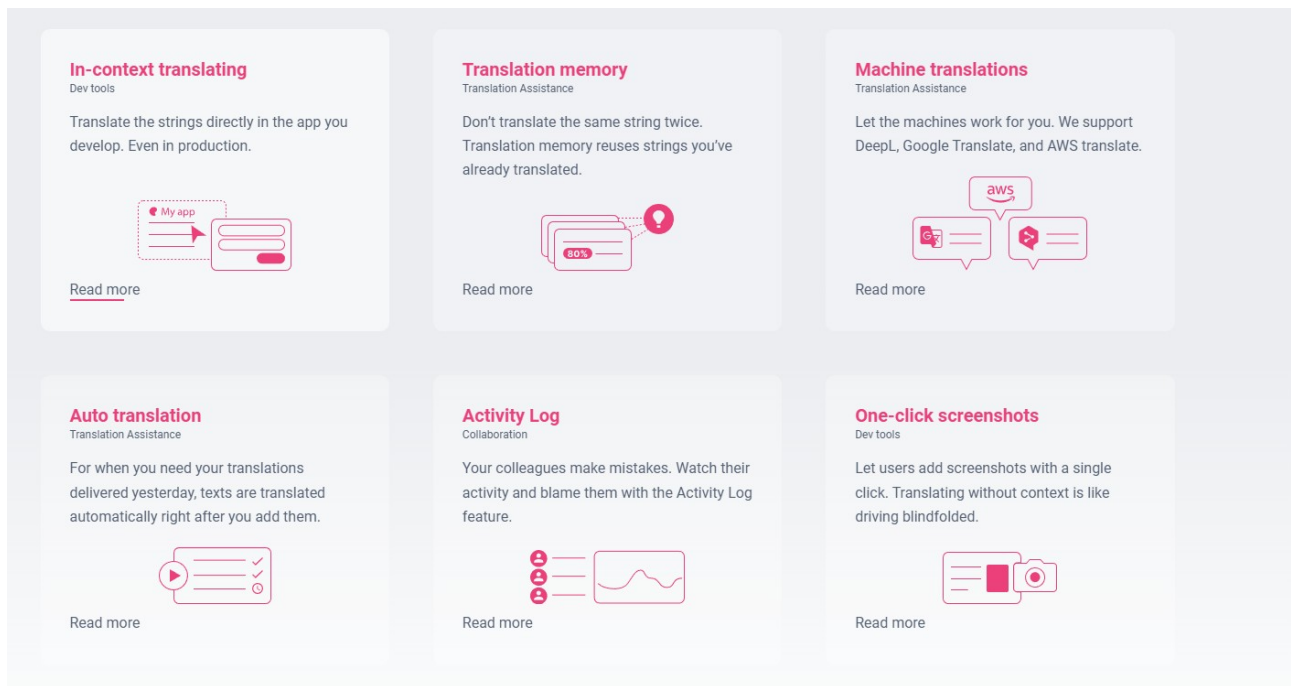


Figura 8: Diferentes herramientas que Tolgee proporciona

Enlace al repositorio: <https://github.com/tolgee/tolgee-platform>

Enlace a la web: <https://tolgee.io/>

5) Cómo se importa un proyecto de GitHub en Android Studio?

1. Primero debemos ir al repositorio del proyecto que deseamos descargar y pulsar en “<> Code” y luego en “Download ZIP”.

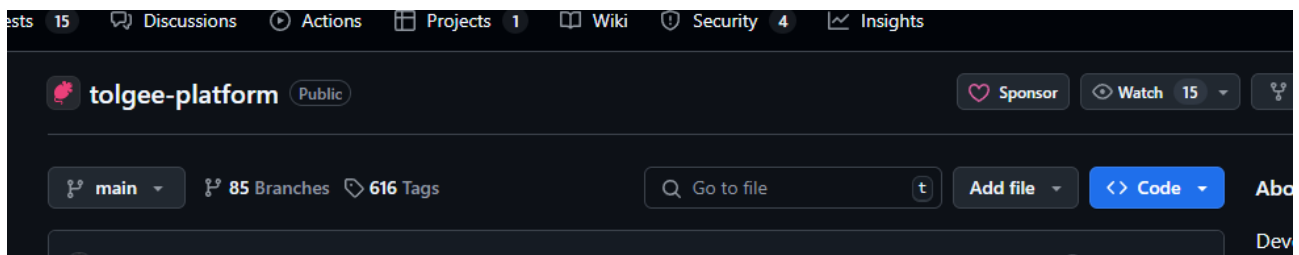


Figura 9: Boton <>Code

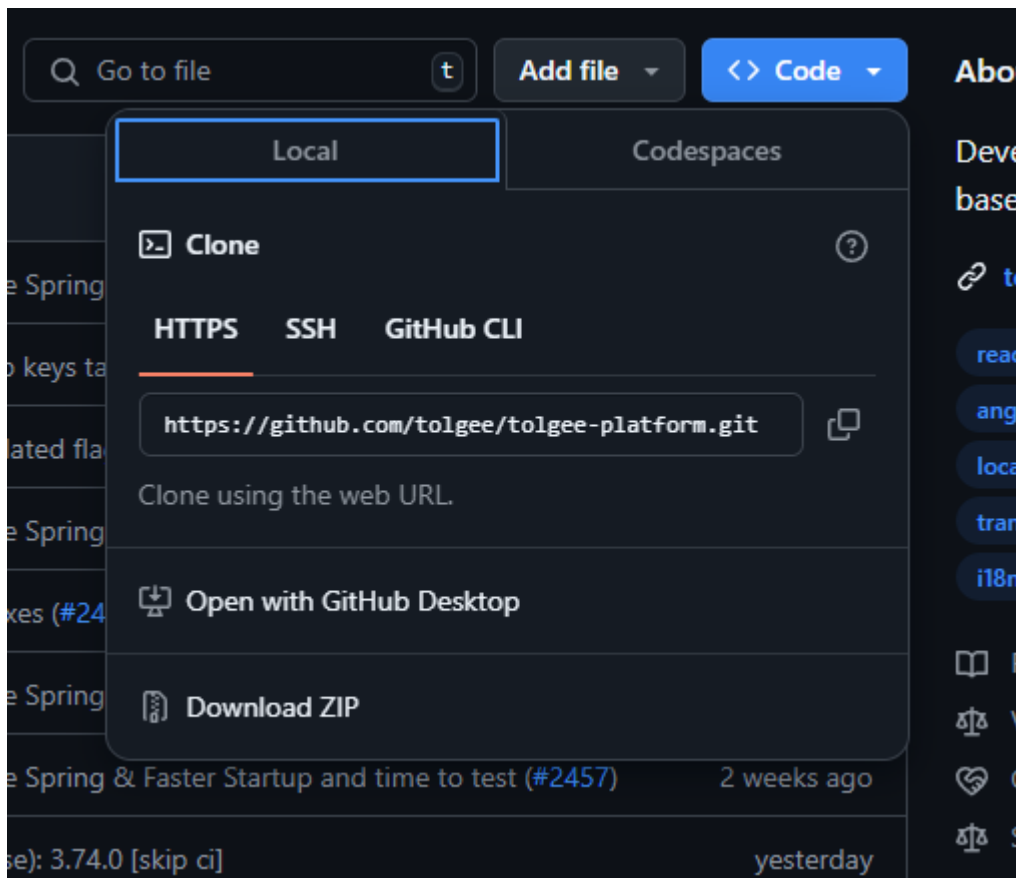


Figura 10: Download ZIP

2. El paso siguiente será abrir Android Studio. Una vez dentro pulsaremos en las tres barras de la esquina superior izquierda y le daremos en “File” → “Open”.

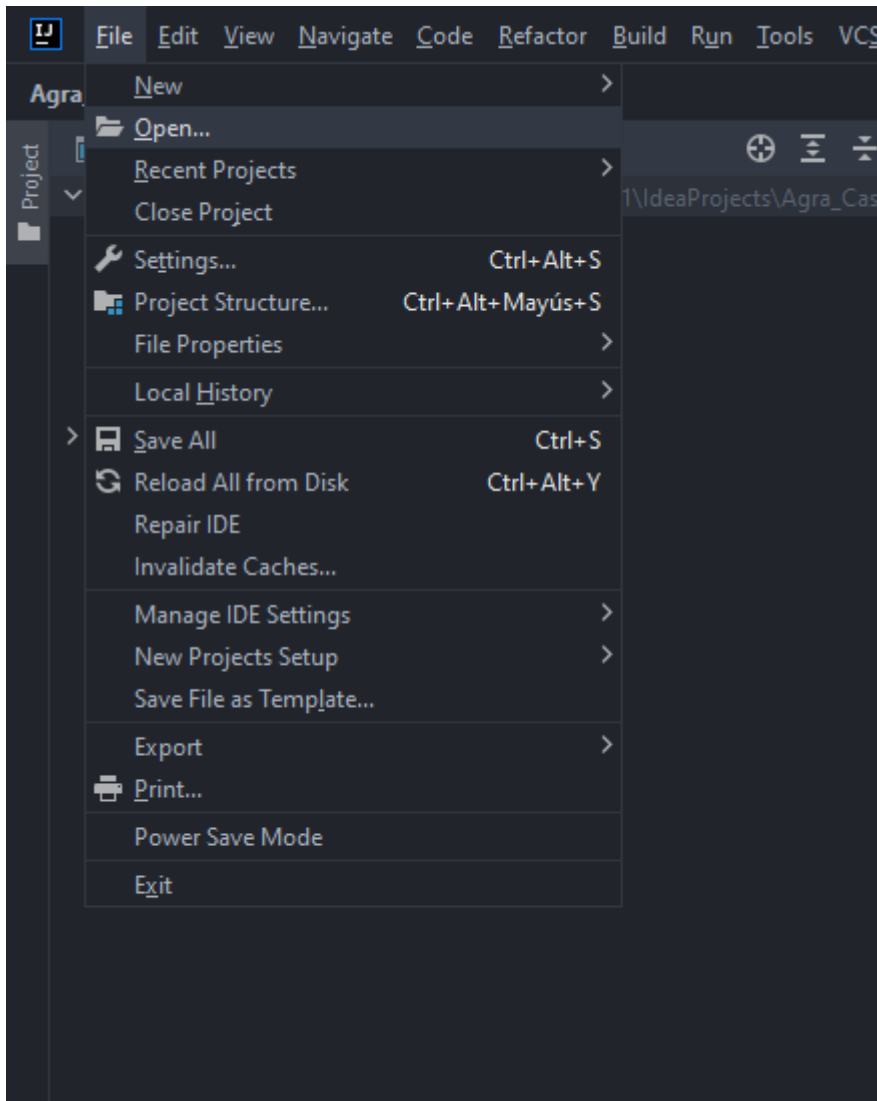


Figura 11: Open Project

3. Luego deberemos indicar la ruta donde se encuentra el proyecto y pulsaremos en “Ok”.

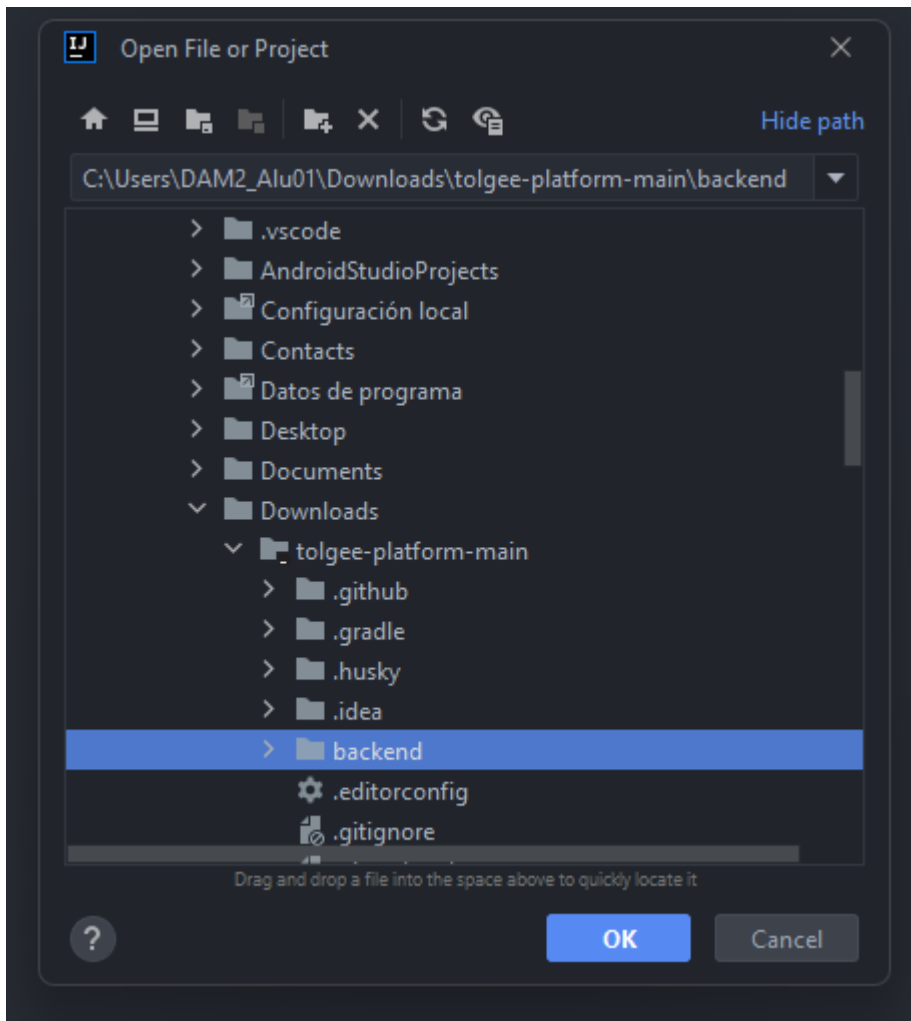


Figura 12: Ruta absoluta del proyecto

4. Por último nos saldrá un aviso de seguridad. Pulsamos en “Trust Project”.

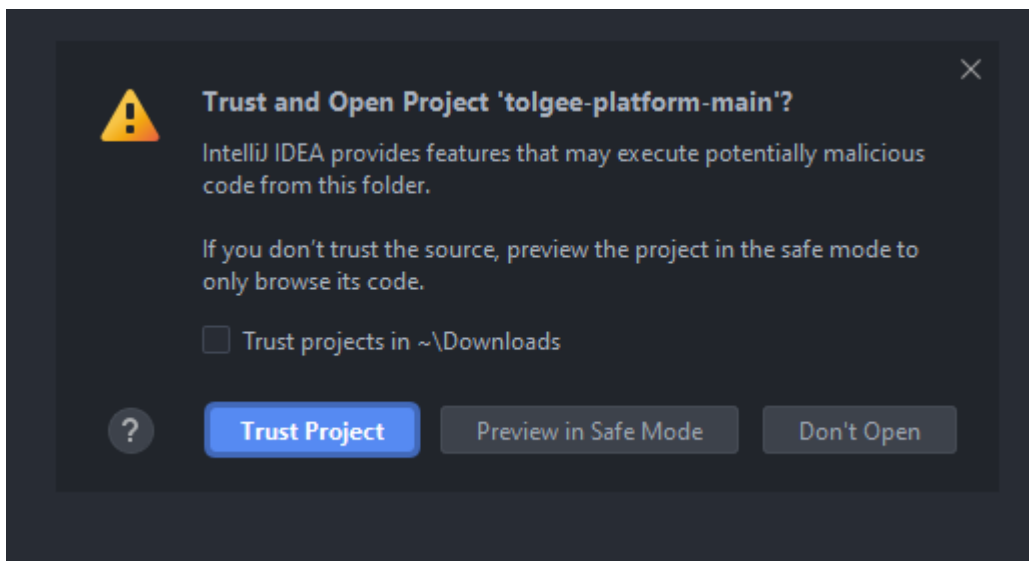


Figura 13: Trust Project

Finalmente tendremos el proyecto ya importado.

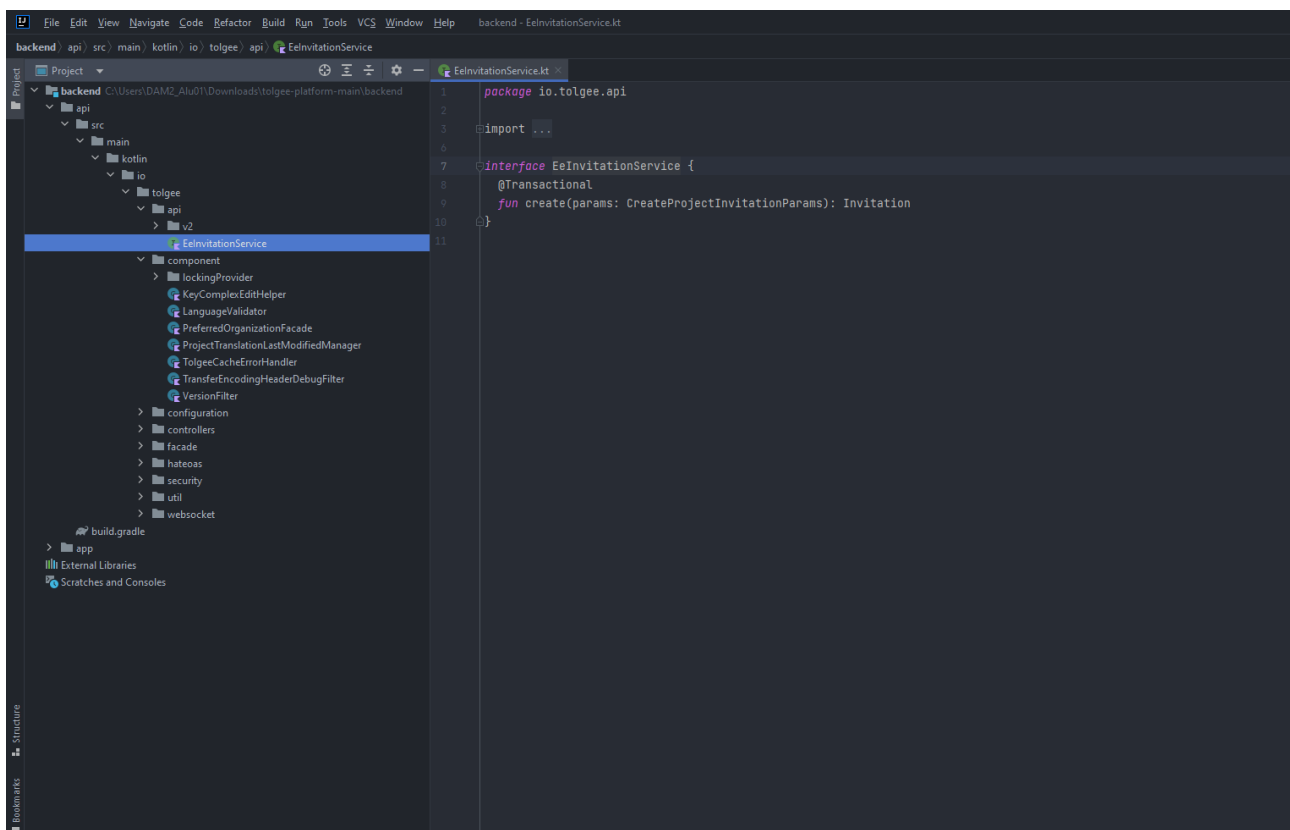


Figura 14: Proyecto importado

5.2) Readme del documento

```
<h1 align="center" style="border-bottom: none">
  <b>
    <a href="https://tolgee.io">Tolgee</a><br>
  </b>
  An open-source localization platform<br/> developers enjoy to work with
  <br>
</h1>

<div align="center">

  ![Logo] (https://user-images.githubusercontent.com/18496315/188628892-33fcc282-26f1-4035-8105-95952bd93de9.svg) (https://tolgee.io)

  An open-source alternative to Crowdin, Phrase or Lokalise

  ![example workflow] (https://github.com/tolgee/tolgee-platform/actions/workflows/test.yml/badge.svg)
  ![kotlin] (https://img.shields.io/github/languages/top/tolgee/tolgee-platform)
  ![docker] (https://img.shields.io/docker/v/tolgee/tolgee/latest?label=DockerHub) (https://hub.docker.com/repository/docker/tolgee/tolgee)
  ![github release] (https://img.shields.io/github/v/release/tolgee/tolgee-platform?label=GitHub%20Release) (https://github.com/tolgee/tolgee-platform/releases/latest)
  ![licence] (https://img.shields.io/badge/license-Apache%20%2F%20Tolgee%20EL-blue)
  ![github stars] (https://img.shields.io/github/stars/tolgee/tolgee-js?style=social&label=Tolgee%20JS) (https://github.com/tolgee/tolgee-js)
  ![github stars] (https://img.shields.io/github/stars/tolgee/tolgee-platform?style=social&label=Tolgee%20Platform) (https://github.com/tolgee/tolgee-platform)
  ![Github discussions] (https://img.shields.io/github/discussions/tolgee/tolgee-platform) (https://github.com/tolgee/tolgee-platform/discussions)
</div>

<div align="center">

  
  (https://www.facebook.com/Tolgee.i18n)
  
  (https://twitter.com/Tolgee_i18n)
  
  (https://www.linkedin.com/company/tolgee)

  **Become part of the family. Join [slack channel ] (https://join.slack.com/t/tolgeecomunity/shared_invite/zt-195isb5u8-_RcSRgVJfvgsPpOBIOk~IQ) **

</div>

  ![Tolgee] (https://user-images.githubusercontent.com/18496315/188632536-3547fd70-755c-4a32-9b1e-fb1afbfb84b33.png)
```

Quick links

- [Tolgee Website] (<https://tolgee.io>)
 - Product (Learn more about the great features)
 - [Dev tools] (<https://tolgee.io/features/dev-tools>)
 - [Translation assistance] (<https://tolgee.io/features/translation-assistance>)
 - [Collaboration] (<https://tolgee.io/features/collaboration>)
- Integrations (Learn how to integrate Tolgee with you favourite technology)
 - [React] (<https://tolgee.io/integrations/react>)
 - [Angular] (<https://tolgee.io/integrations/angular>)
 - [Vue] (<https://tolgee.io/integrations/vue>)
 - [Svelte] (<https://tolgee.io/integrations/svelte>)
 - [Next.js] (<https://tolgee.io/integrations/next>)
 - [More...] (<https://tolgee.io/integrations/all>)
- [Tolgee platform docs] (<https://tolgee.io/docs/platform>)
 - [Self-hosting] (https://tolgee.io/docs/platform/self_hosting/running_with_docker)
- [Development notes (How to develop Tolgee locally)] (<https://github.com/tolgee/tolgee-platform/wiki/Development>)

Why to use Tolgee?

Beacause it saves a lot of time you would spend on localization tasks without it. Because it enables you to provide perfectly translated software.

![Frame 47] (<https://user-images.githubusercontent.com/18496315/188637819-ac4eb02d-7859-4ca8-9807-27818a52782d.png>)

In context translating & One click screenshots

Add translations in the code, and translate them directly in the app with the Tolgee i18n tool. Clicking an element while holding the ALT/option key opens a dialog where you can simply modify your strings. No need to edit large .json/.po/.whatever file. In-context translating works great also in the production environment.

Once. That's how many times you have to click to take a screenshot from your application with highlighted phrases to translate. Just ALT + click a string and hit the camera button. Boom! Screenshot generated.

![Sep-06-2022 16-38-49] (<https://user-images.githubusercontent.com/18496315/188672133-064d2a26-e414-4f5e-ab43-549af8cb2145.gif>)

Translating on production

In-context translating also works in the production environment of your deployed app. Using the Tolgee Tools Chrome plugin, you can simply provide your API key and start translating. This enables anybody without developer knowledge to translate your app.

True integrations

Tolgee is not just another localization platform offering integrations that just sync your local data with the backend. Tolgee is truly integrated into your app via SDKs.

Machine translation

We support DeepL, Google Translate, and AWS Translate. Select which services you want to use in the settings section. The machine translation features make the whole localization process significantly faster. Translators can just use translation suggestions provided by third-party machine translation services.

Translation memory

Tolgee automatically makes suggestions from translations you already used in the project so you can translate similar phrases in a similar way.

Translation memory suggestions also show the similarity percentage, the key, and the original text of the translated string.

Auto translation

When enabled, Tolgee automatically translates new keys using translation memory or machine translation services. Your strings are translated immediately, right after creation. Select if you would like to use translation memory and/or which machine translation service you would like to use to automatically translate new keys.

Activity log

See who modified, reviewed or commented on the phrases in your project. Clearly.

Comment on translations

Something look off? Tell others what you would change. You can comment every translation on Tolgee platform.

Translation history

See the changes to specific translations of a specific key in a specific language. Something is wrong? You know where to point the finger!

For more detailed documentation about Tolgee, visit tolgee.io.

Quickstart 🚀

1. Sign up at [app.tolgee.io](https://app.tolgee.io/sign_up) or access your self-hosted instance
2. Create a project
3. Follow a guide in integration section of your project
4. Have fun!

![Integration guides](https://user-images.githubusercontent.com/18496315/188818166-d70d4676-7bd2-4328-91eb-720add935ab6.gif)

Contributors

Let us know what you think! #feedbackwanted ❤️

En este readme podemos ver con más detalle como funcionan sus herramientas y como empezar a utilizarla.