

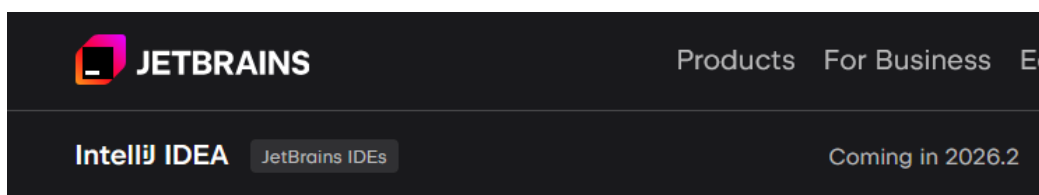
## Práctica de IDEs: Instalación e uso básico

Esta práctica será avaliada individualmente observando o traballo no teu ordenador.

### SECCIÓN 1: INTELLIJ IDEA

**Descarga** o IDE (Integrated Development Environment) chamada IntelliJ IDEA dende a súa páxina web, elixindo o teu sistema operativo (Windows, macOS, Linux):

- <https://www.jetbrains.com/idea/download>



[Windows](#) [macOS](#) [Linux](#)

## IntelliJ IDEA

The Leading IDE for Professional  
Development in Java and Kotlin

[Download](#) ▾

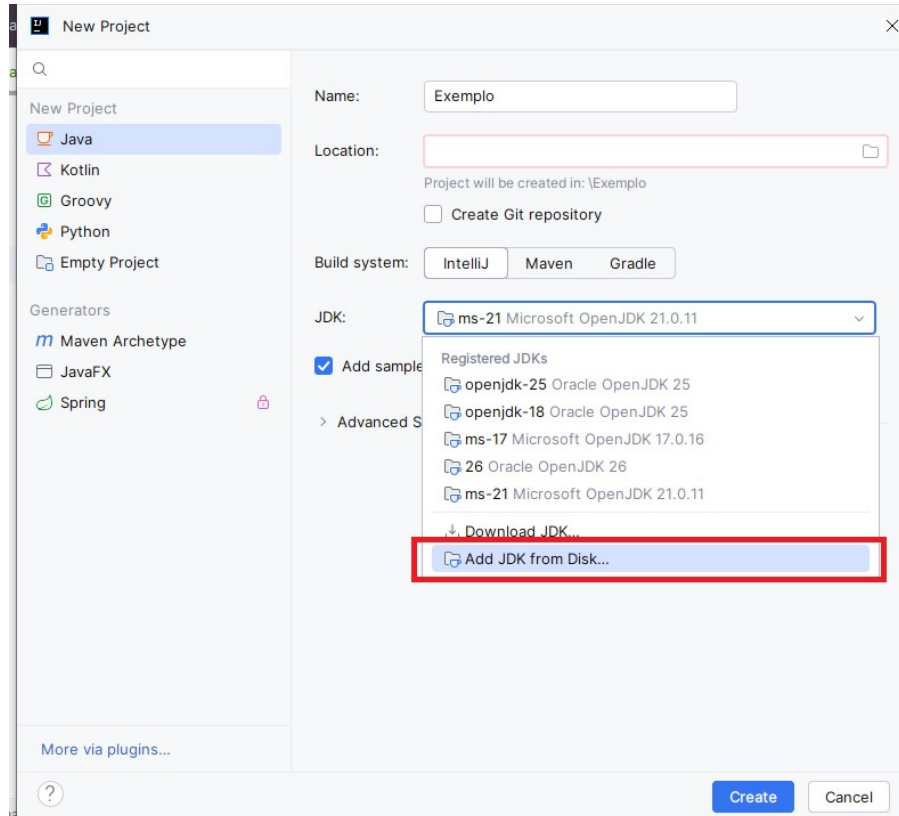
Free – with a 30-day Ultimate subscription trial included

Cando o teñas descargado, **instálalo**.

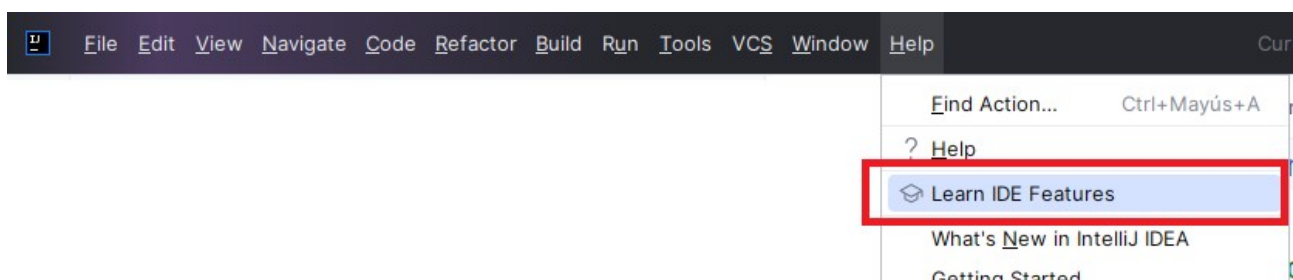
Ao abriro, atoparás as ventás iniciais que che permitirán seleccionar a opción de crear un novo proxecto.

**Crea** un novo proxecto Java na ubicación do disco que ti queiras, e chámalo “**Exemplo**”.

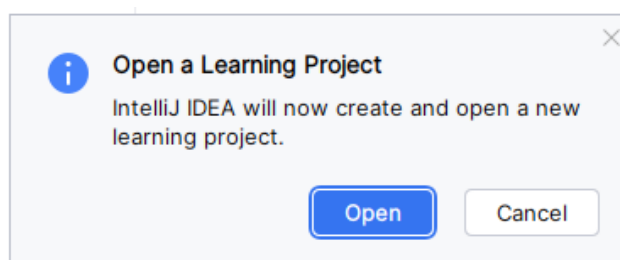
Na opción de “JDK”, selecciona “Add JDK from Disk...” e busca o JDK que instalaches na primeira actividade.



A continuación, preme nos tres guíons do menú superior e accede a Help > Learn IDE features:

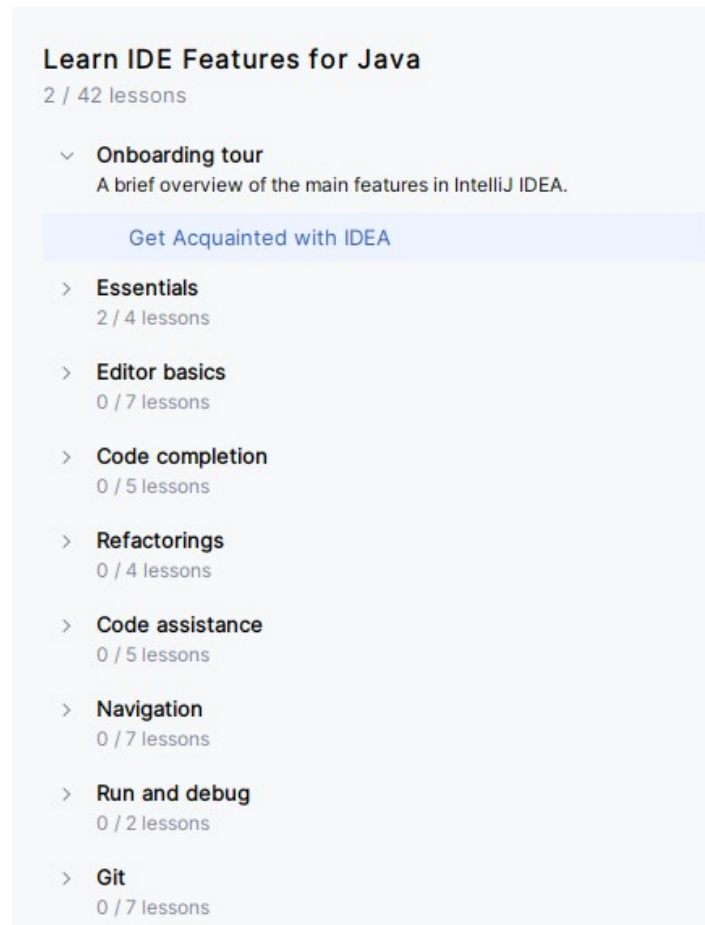


Cando comeces as leccións, IntelliJ IDEA creará un novo proxecto de aprendizaxe:



**Completa** as primeiras **35** leccións que están preparadas.

Opcionalmente, completa tamén as 7 últimas sobre Git (veremos Git máis adiante).



**Learn IDE Features for Java**  
2 / 42 lessons


- ∨ **Onboarding tour**  
A brief overview of the main features in IntelliJ IDEA.
  - [Get Acquainted with IDEA](#)
- > **Essentials**  
2 / 4 lessons
- > **Editor basics**  
0 / 7 lessons
- > **Code completion**  
0 / 5 lessons
- > **Refactorings**  
0 / 4 lessons
- > **Code assistance**  
0 / 5 lessons
- > **Navigation**  
0 / 7 lessons
- > **Run and debug**  
0 / 2 lessons
- > **Git**  
0 / 7 lessons

## SECCIÓN 2: VISUAL STUDIO CODE

Visita a páxina de VSCode e navega á sección de descargas dende o botón superior.

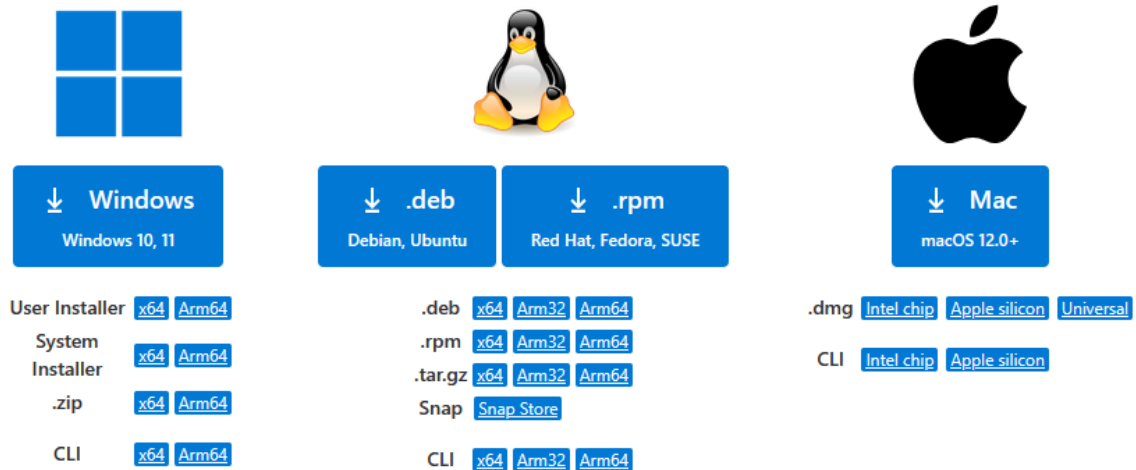
- <https://code.visualstudio.com/>

**Descarga** VSCode para o teu sistema operativo e **instálalo**.

Features ▾ Docs ▾ Release Notes Blog Learn Events  Resources ▾

# Download Visual Studio Code

Free and built on open source. AI agents, Git, debugging, and extensions included.



The screenshot shows the download page for Visual Studio Code. It features three main columns for different operating systems: Windows, Linux, and Mac. Each column has a logo at the top (Windows logo, Tux penguin, and Apple logo) and a blue button with a download icon and the OS name. Below the buttons are lists of available installation methods and their supported architectures.

OS	Installation Method	Supported Architectures
Windows	User Installer	x64, Arm64
	System Installer	x64, Arm64
	.zip	x64, Arm64
	CLI	x64, Arm64
	Windows 10, 11	
Linux	.deb	x64, Arm32, Arm64
	.rpm	x64, Arm32, Arm64
	.tar.gz	x64, Arm32, Arm64
	Snap	Snap Store
	CLI	x64, Arm32, Arm64
	Debian, Ubuntu	
	Red Hat, Fedora, SUSE	
Mac	.dmg	Intel chip, Apple silicon, Universal
	CLI	Intel chip, Apple silicon
	macOS 12.0+	

Cando se complete a instalación, VSCode vaise abrir automaticamente.

Este IDE é moi popular e emprégase para traballar en moitas linguaxes. Agora, en lugar de abrir unha carpeta e crear un proxecto (como é habitual), vamos a traballar cun só ficheiro e a aprender sobre o *Marketplace* de extensións.

**Visita** a seguinte páxina:

- <https://beeceptor.com/docs/concepts/openapi-sample-spec/>

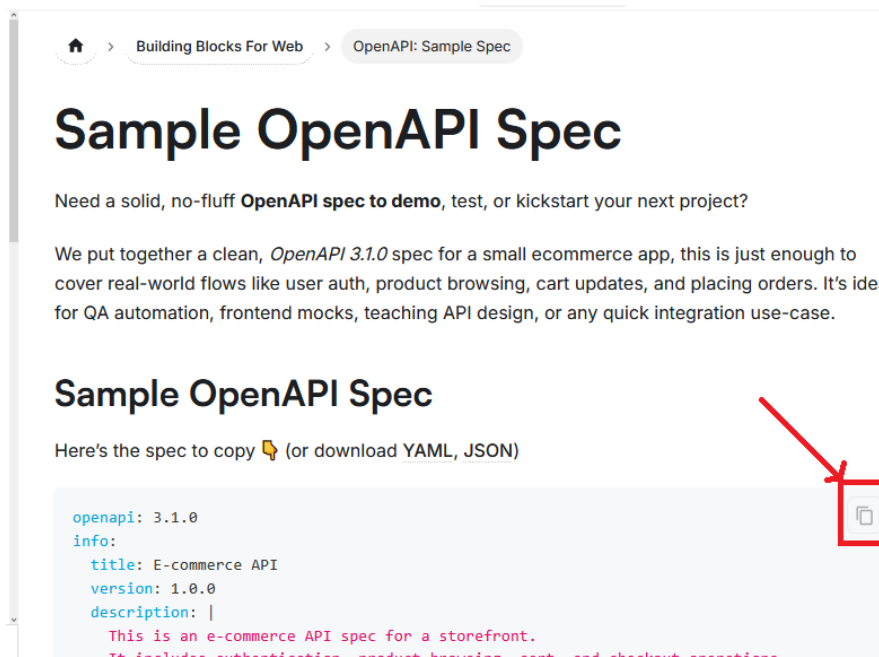
Nela atoparás un código de exemplo.

Este código está en formato YAML, que emprega claves e valores separados por dous puntos (:) con distintos niveis de indentación para expresar xerarquía.

O contido do ficheiro segue unha especificación chamada OpenAPI.

🧠 Para que se usa OpenAPI?

**Copia** dito código:



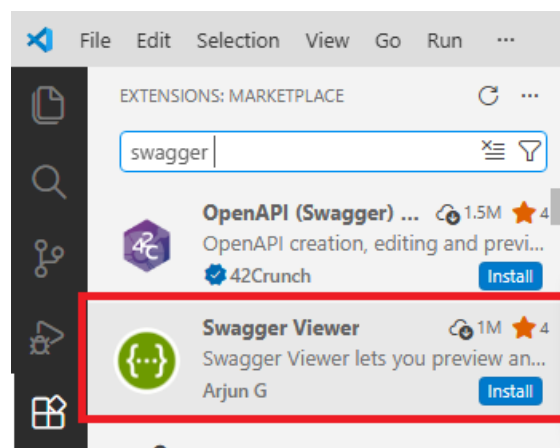
E, agora, en VSCode, **crea** un novo ficheiro de texto na ubicación de disco que ti queiras.

**Pega** o contido e gárdao.

Este código serve para definir un API REST, no noso caso, un API REST de exemplo que non está implementado realmente.

**Navega** en VSCode na pestaña da esquerda á Extensións, e busca “Swagger”.

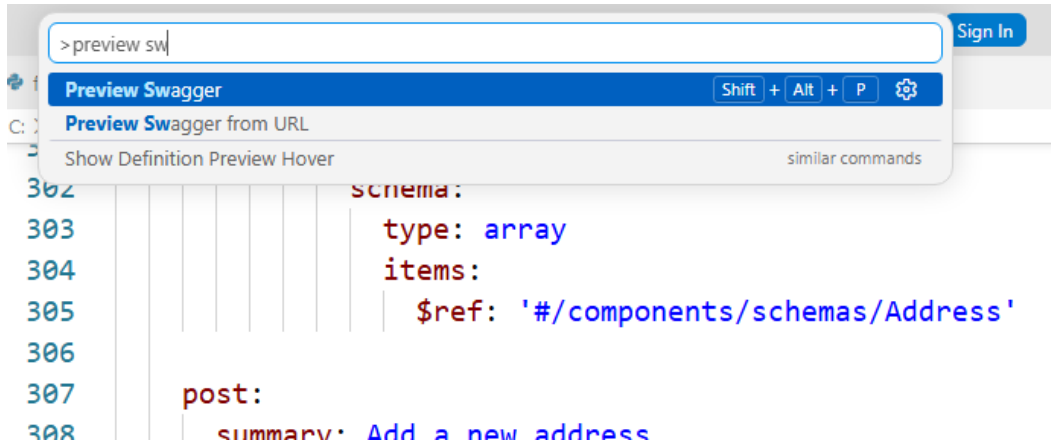
**Instala** “Swagger Viewer” como se ve na imaxe:



Cando estea instalado, **desmarca** a opción “Auto-Update” de Swagger Viewer.

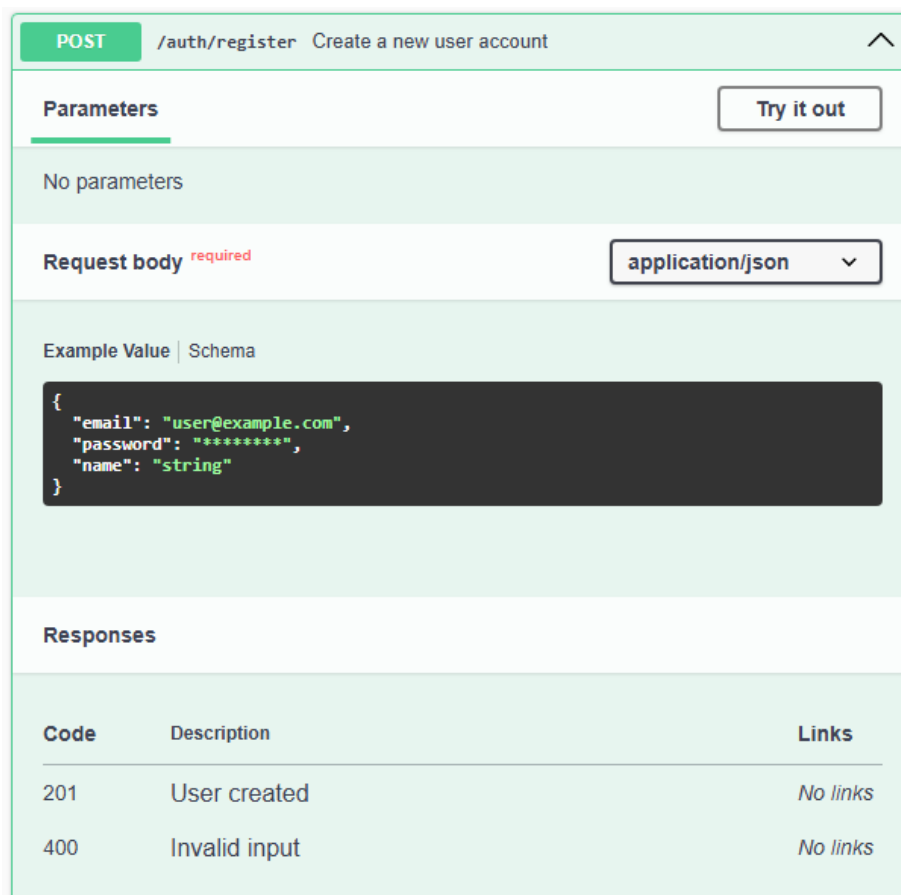
Despois de instalalo, terás a posibilidade de “visualizar” de xeito alternativo o contido do ficheiro OpenAPI que creaches anteriormente.

Co ficheiro aberto, **pulsa** F1 e teclea “preview swagger”. **Executa** ese comando:



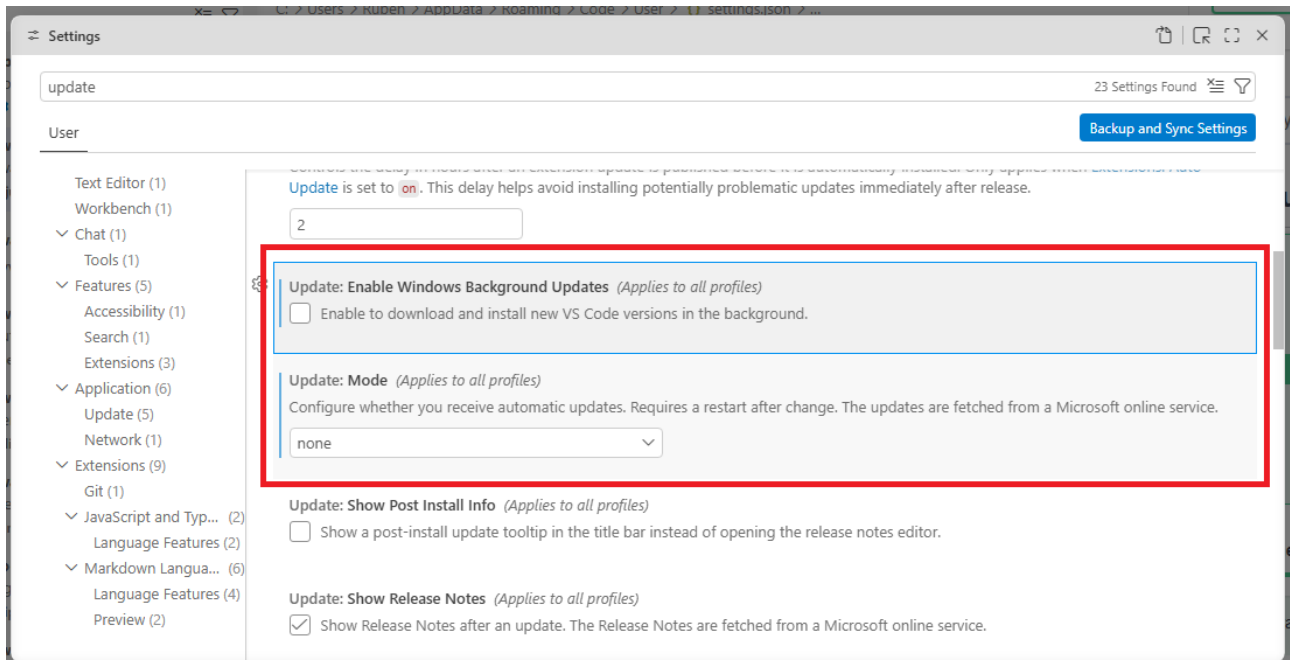
! É posible que salte unha alerta de Firewall se estás en Windows. Isto débese a que a extensión permite, na “previsualización Swagger”, enviar peticións HTTP dacordo coa especificación. Non te preocupes.

🧠 De feito, a maioría de extensións e funcionalidades están deshabilitadas cando abres unha carpeta ou ficheiro en “modo restrinxido”. Que é o modo restrinxido? Por que existe? O hai noutros IDEs coma IntelliJ IDEA? Se tes curiosidade, pregúntalle á IA.



Para rematar, polo de agora, con **VSCode**, abre a File > Preferences > Settings.

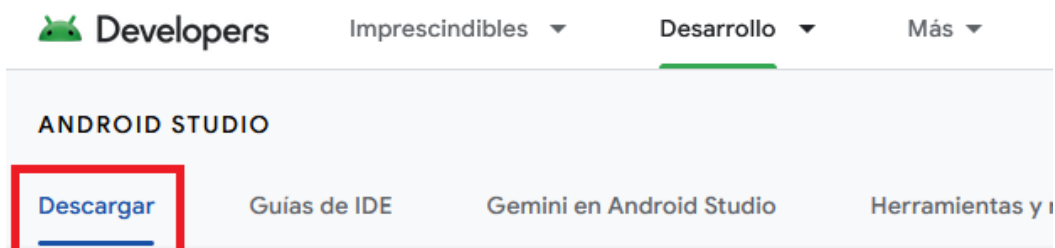
En Settings, busca “update” e **deshabilita** estas dúas opcións da actualización automática.



### SECCIÓN 3: ANDROID STUDIO

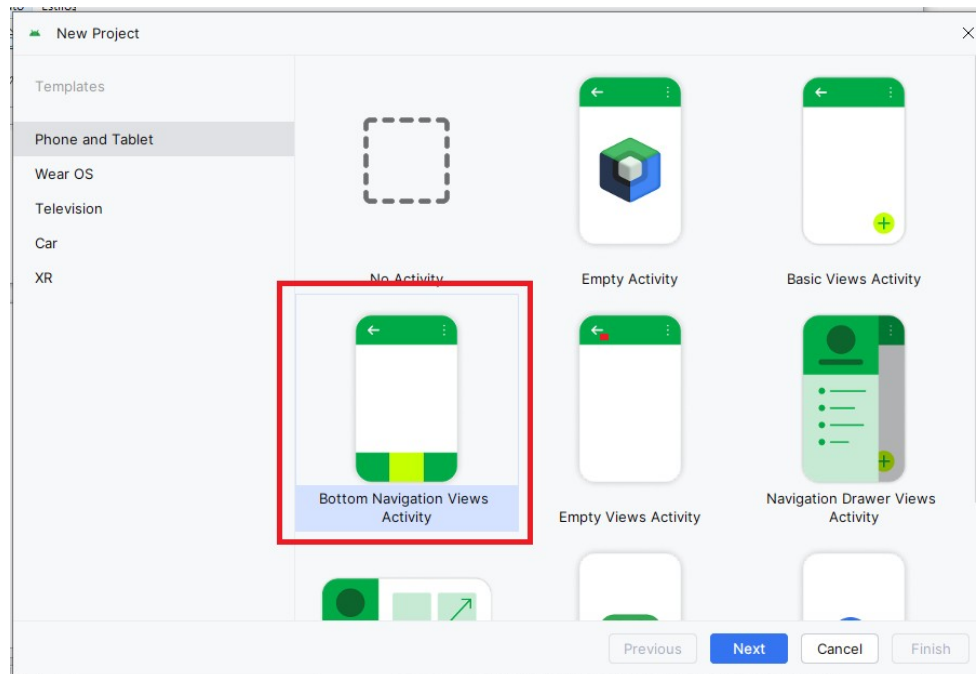
Navega á páxina de Android Studio e **descárgao**:

- <https://developer.android.com/studio>



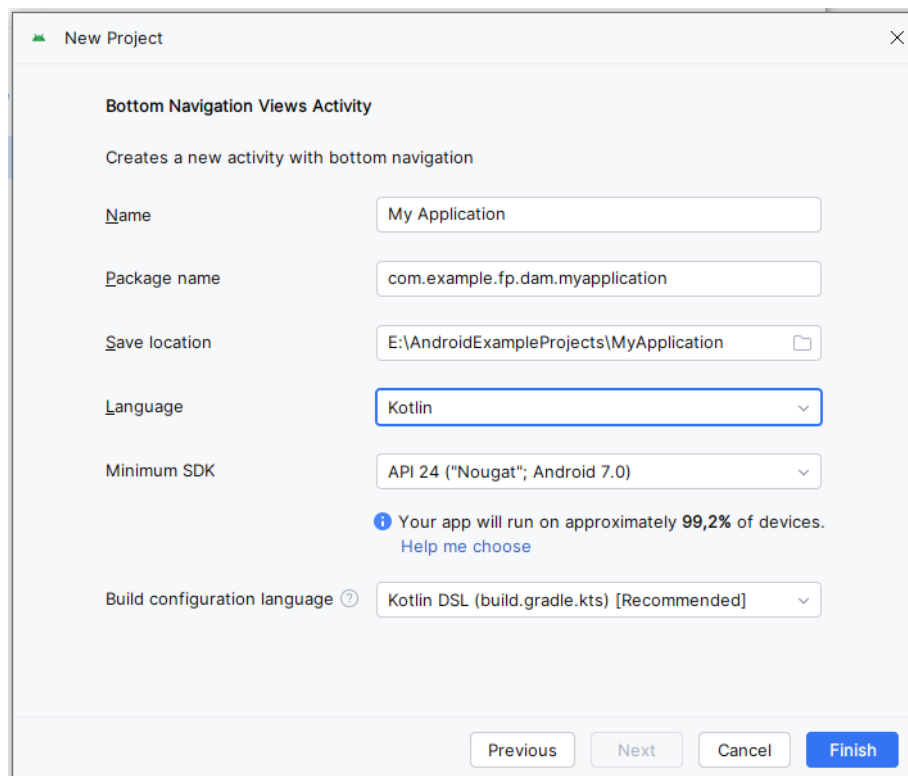
Despois, **instala** Android Studio.

Tras a instalación, ábreo e **crea** un novo proxecto. Escolle o tipo “Bottom Navigation Views Activity”



**Chama** o novo proxecto “My Application” e configúrao como se mostra a continuación.

🧠 Que é a opción de Minimum SDK? Se tes curiosidade, preme no botón “Help me choose”

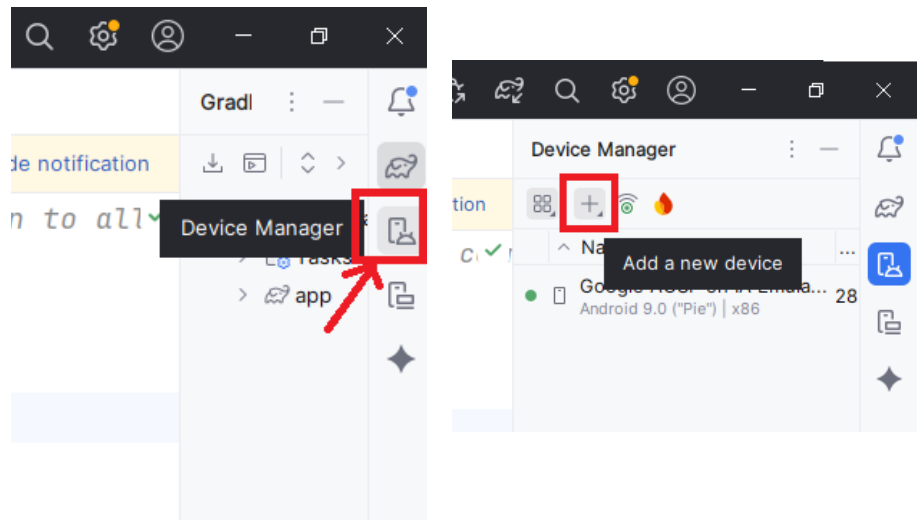


Cando estea creado, verás á esquerda a xerarquía de arquivos do proxecto. En Android, a programación do comportamento das aplicacións faise en linguaxe Kotlin<sup>1</sup> (ou Java) e os ficheiros que definen as interfaces visuales están en XML.

Hai moito que saber sobre desenvolvemento Android 😊

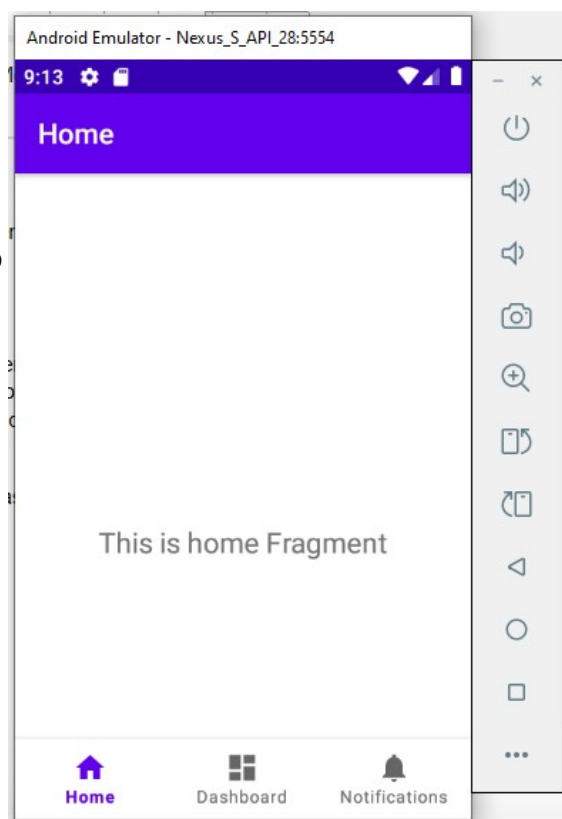
De momento, **abre** a pestaña “Device Manager” no panel dereito.

Despois, **clica** en “Add a new device”



No asistente que sae, segue os pasos necesarios para crear un novo dispositivo virtual (Android Virtual Device)

E, a continuación, **executa** a aplicación no emulador clicando no botón “Play” verde da barra superior.



1 Podes facer un interesante tutorial introdutorio a Kotlin na páxina oficial: <https://kotlinlang.org/docs/kotlin-tour-welcome.html>

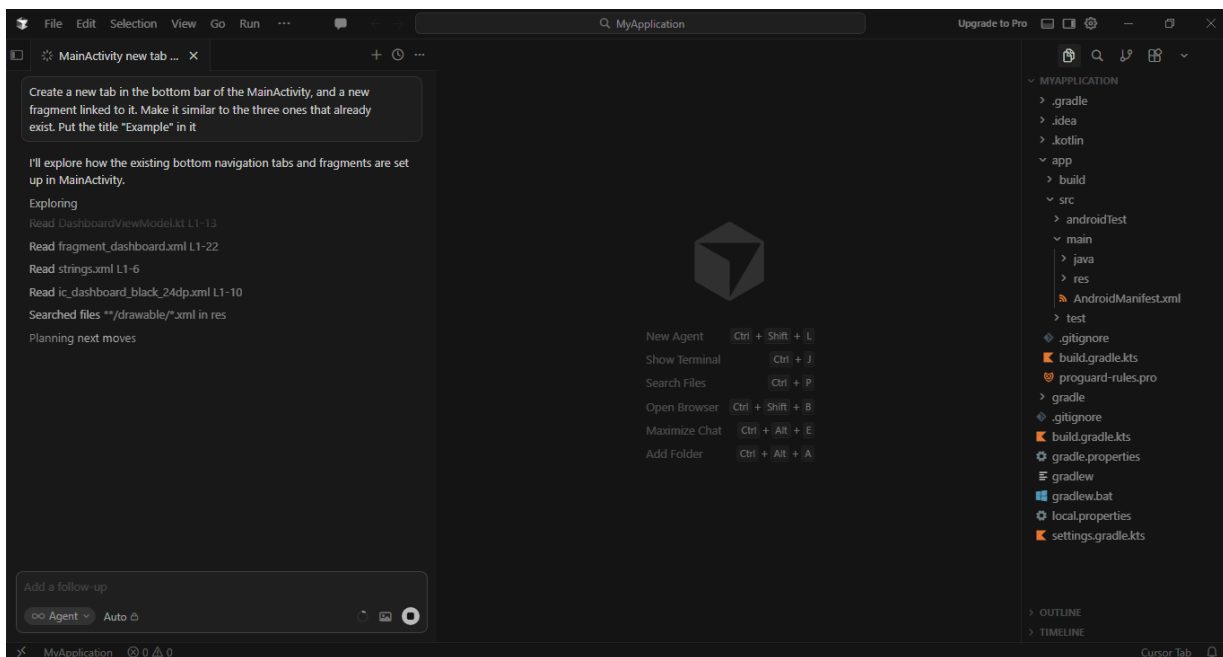
## SECCIÓN 4: CURSOR

**Descarga** o IDE Cursor desde a súa páxina web:

- <https://cursor.com/>

**Instálao** e ábreo. Despois, abre proxecto “My Application” que creaches na sección anterior.

Á esquerda atoparás a ventana de chat coa IA. Pídelle que cree unha nova pestaña na aplicación, que agora mesmo ten 3 pestañas:



Levou a cabo o traballo correctamente? Como verificas se o que fixo funciona? Como comprobas que foi o que fixo?

**Define** uns obxectivos que queres conseguir para transformar “My Application”, e **traballa** co chat de Cursor para completar esas funcionalidades que ti consideres oportunas.

Na avaliación deste apartado só se vai ter en conta:

- Os teus *prompts*
  - Deben ser acotados (non pedir cosas xerais)
  - Debes traballar complementariamente con outra IA de apoio que che dea explicacións sobre como funciona o código que propón o axente. Proba a pestaña “Ask” en lugar de “Agent”
  - Debes revisar e comprender o que propón, sen aceptalo indiscriminadamente
  - Deben estar orientados a obxectivos plantexados por ti previamente. O que che da a IA “non está ben porque sí”. É unha ferramenta de apoio
- O rexistro de que validaches (e como o fixeches) o código proposto polo axente IA

Non se avaliará a aplicación resultante que consigas.