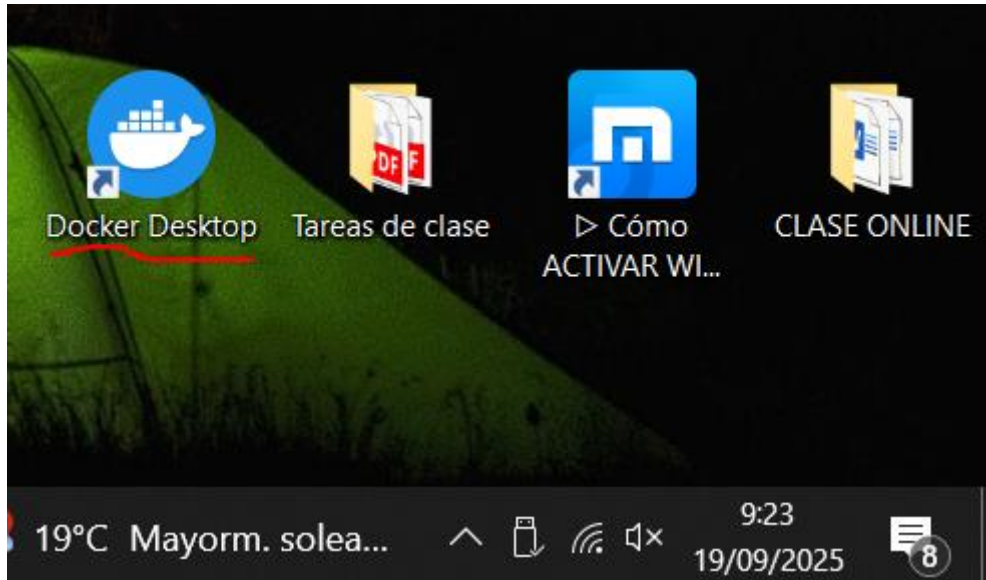


Instalación

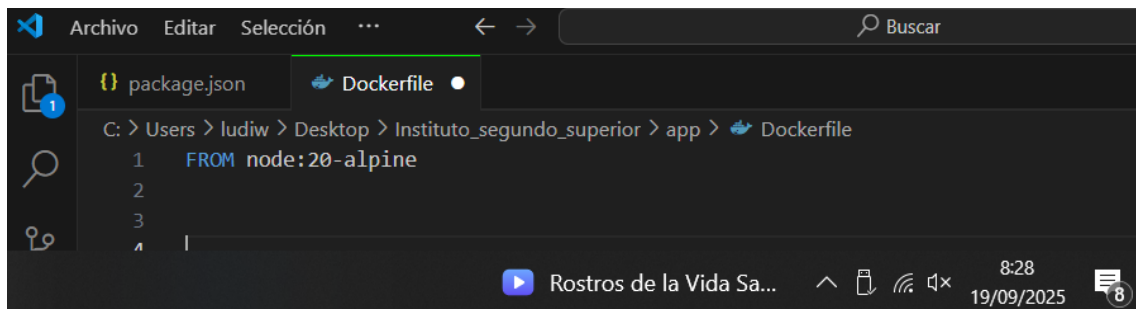
Para poder usar Dockerfile (si no usáis Linux) vais a tener que descargaros e instalaros Docker Desktop. Su descarga e instalación es fácil no, lo siguiente, os tenéis que descargar el instalador de la siguiente página:

Ya lo tenemos instalado

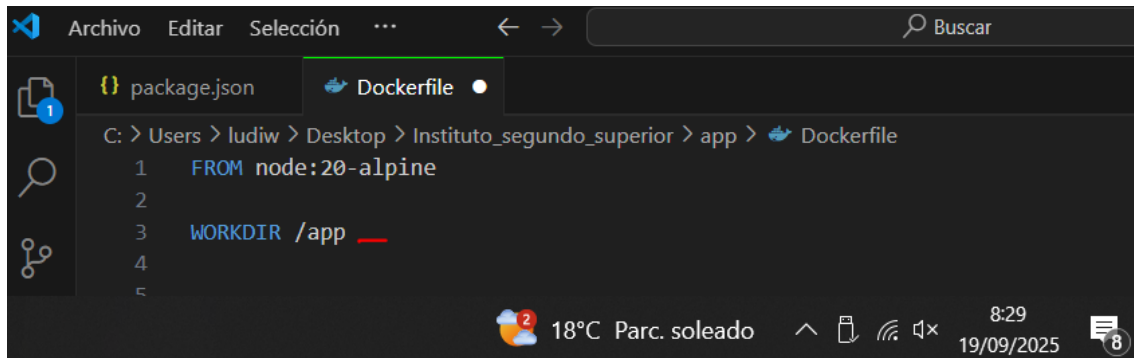


Creación del Dockerfile

Como ya os he comentado en el apartado anterior la creación empieza con crear un fichero llamado Dockerfile a secas, sin .txt ni .json ni nada. Una vez hecho eso vamos a rellenarlo siguiendo los siguientes pasos: 1. Vamos a definir la imagen base añadiendo la siguiente línea:



Ahora, vamos a definir el directorio de trabajo usando la instrucción WORKDIR. Esto especificará en qué directorio se van a ejecutar los comandos y dónde se van a copiar los ficheros dentro del contenedor:

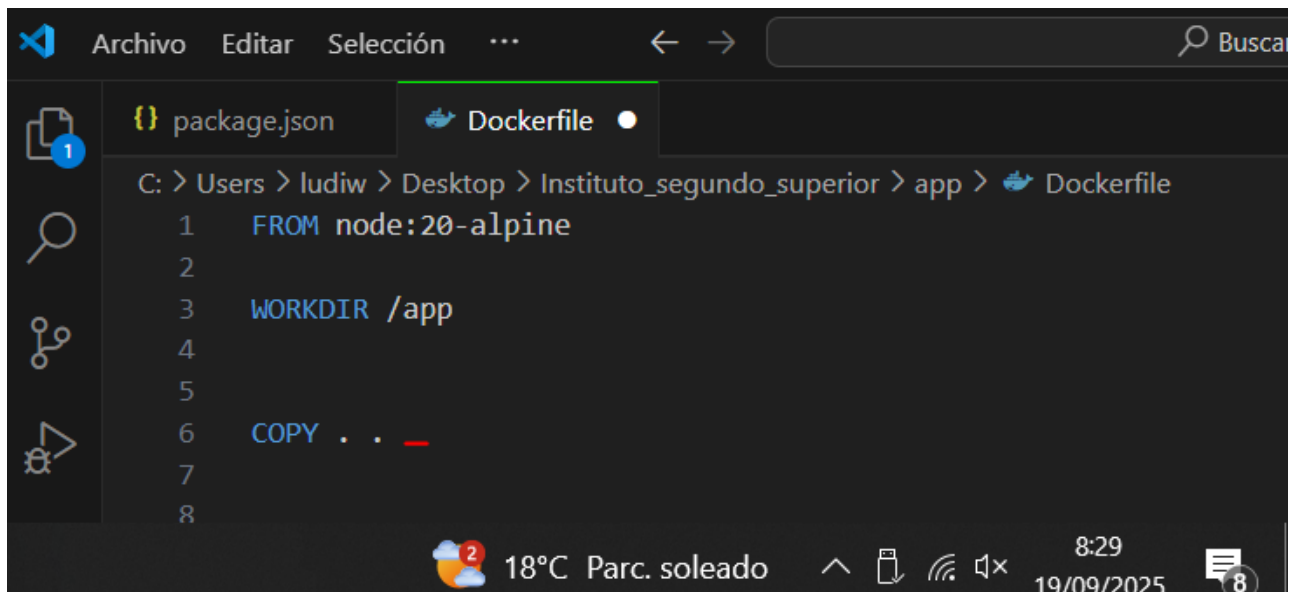


The screenshot shows the Visual Studio Code editor with a file named 'Dockerfile' open. The file path is 'C: > Users > ludiw > Desktop > Instituto_segundo_superior > app > Dockerfile'. The content of the Dockerfile is as follows:

```
1 FROM node:20-alpine
2
3 WORKDIR /app
4
5
```

The status bar at the bottom indicates a temperature of 18°C, 'Parc. soleado', and the date '19/09/2025' at '8:29'.

Copiamos todos los ficheros de nuestro proyecto dentro de la máquina (contenedor) usando el comando COPY:

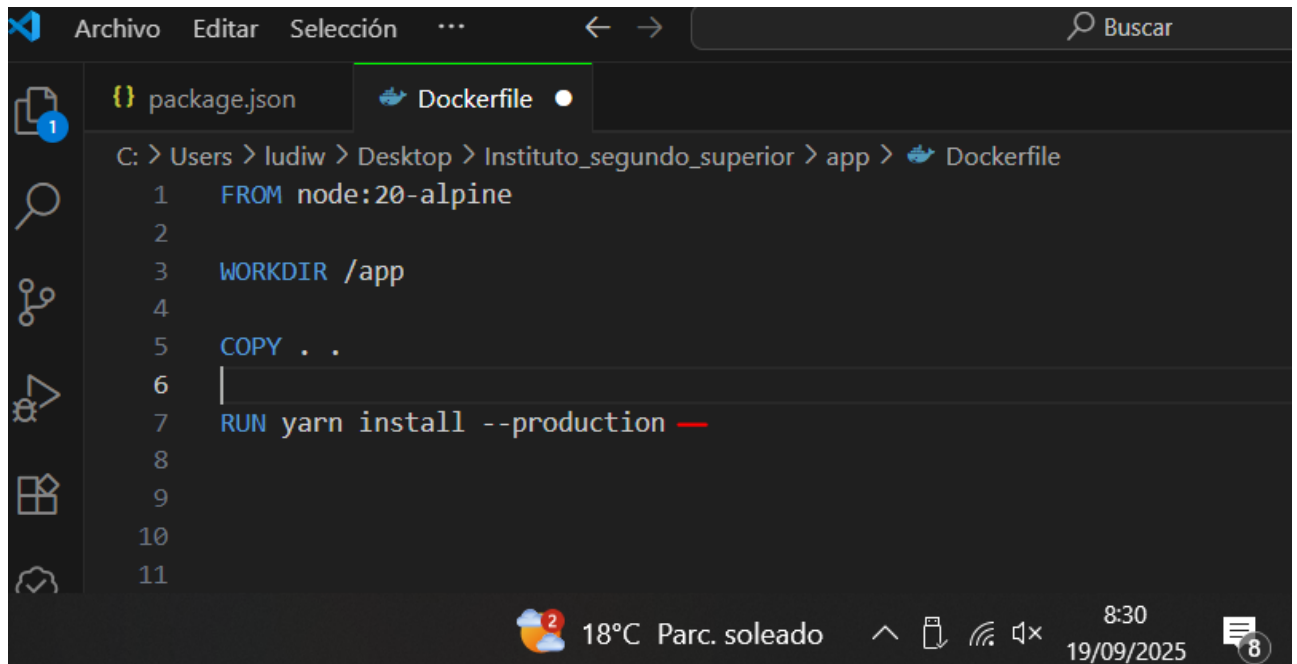


The screenshot shows the Visual Studio Code editor with the same 'Dockerfile' open. The content of the Dockerfile is now:

```
1 FROM node:20-alpine
2
3 WORKDIR /app
4
5
6 COPY . .
7
8
```

The status bar at the bottom indicates a temperature of 18°C, 'Parc. soleado', and the date '19/09/2025' at '8:29'.

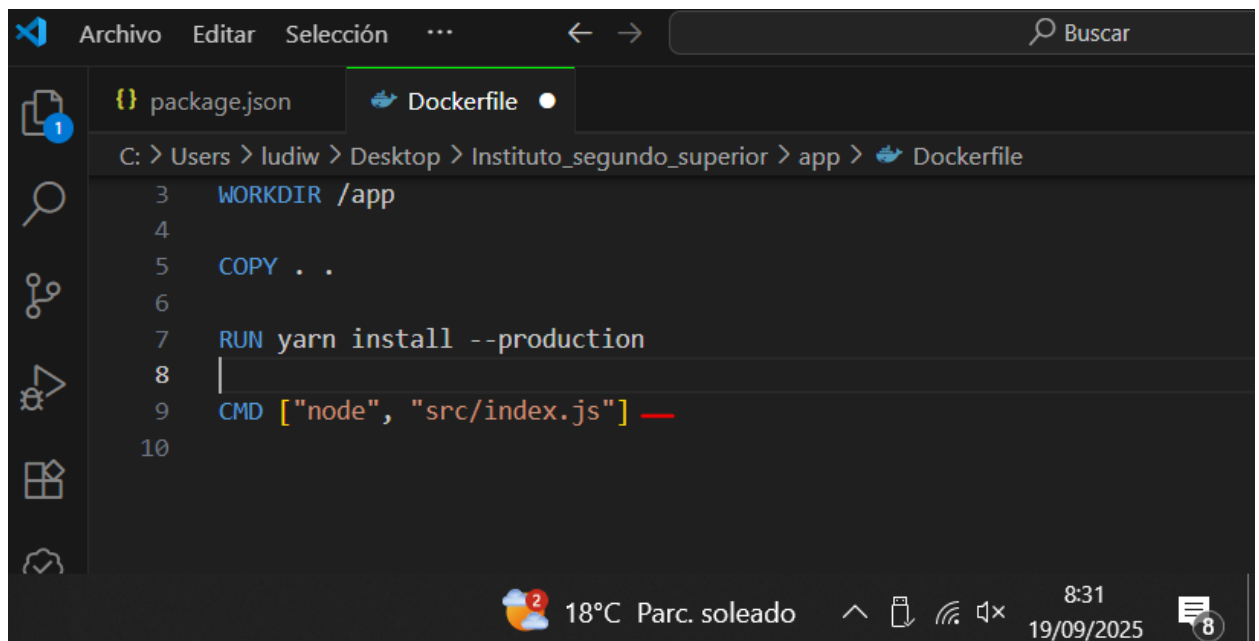
Instalamos todas las dependencias de la aplicación usando el comando yarn y su gestor de paquetes. Para hacer esto vamos a usar el comando RUN:



```
1 FROM node:20-alpine
2
3 WORKDIR /app
4
5 COPY . .
6
7 RUN yarn install --production --
```

The screenshot shows the Visual Studio Code editor with a Dockerfile open. The file path is C:\Users\ludiw\Desktop\Instituto_segundo_superior>app>Dockerfile. The Dockerfile contains the following instructions: FROM node:20-alpine, WORKDIR /app, COPY . ., and RUN yarn install --production --. The RUN instruction is currently on a new line with a red cursor at the end.

Para acabar vamos a especificar el comando por defecto que va a ejecutar nuestro contenedor usando la instrucción CMD:



```
3 WORKDIR /app
4
5 COPY . .
6
7 RUN yarn install --production
8
9 CMD ["node", "src/index.js"] --
```

The screenshot shows the Visual Studio Code editor with the same Dockerfile. The CMD instruction has been added to the file, resulting in: CMD ["node", "src/index.js"] --. The RUN instruction is now on line 7 and the CMD instruction is on line 9.

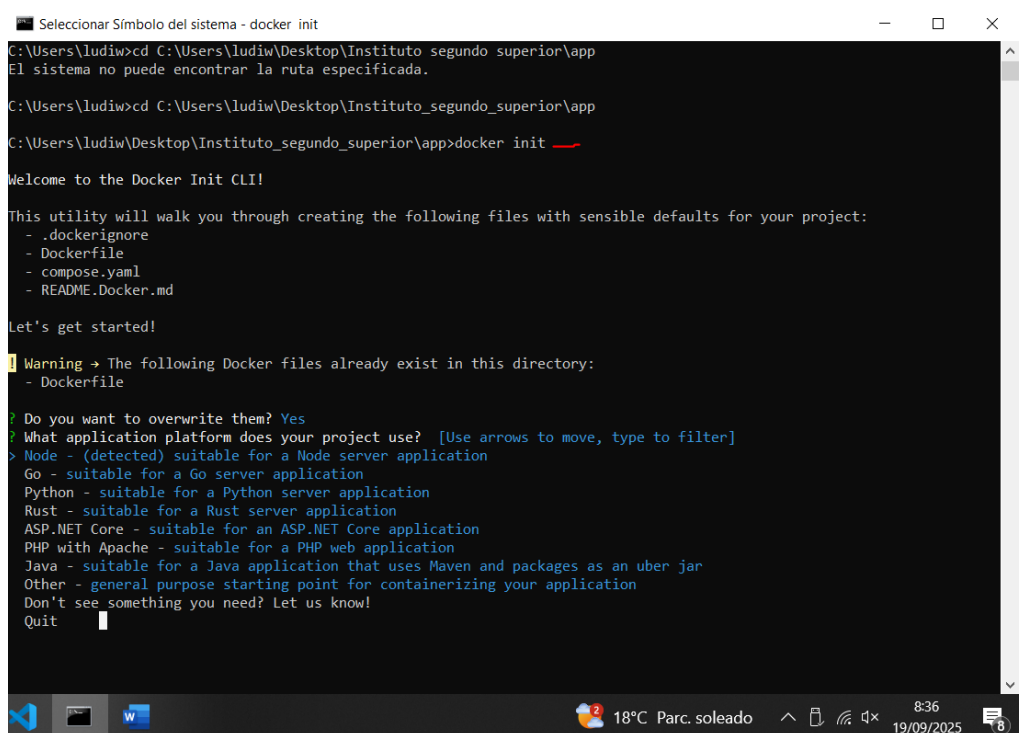
Lanzar el contenedor

Para ello vamos a usar el siguiente comando dentro de la carpeta donde está el Dockerfile:

Nos metemos dentro de la carpeta y cogemos la ruta y luego hacemos un cd con la ruta que hemos cogido y ya estaríamos dentro de la carpeta

```
C:\Users\ludiw>cd C:\Users\ludiw\Desktop\Instituto_segundo_superior\app
C:\Users\ludiw\Desktop\Instituto_segundo_superior\app> |
```

Aquí ponemos el comando Docker init que sirve para inicializar un proyecto en Docker desde la carpeta, también detecta el tipo de proyecto



```
Selecciónar Símbolo del sistema - docker init
C:\Users\ludiw>cd C:\Users\ludiw\Desktop\Instituto_segundo_superior\app
El sistema no puede encontrar la ruta especificada.

C:\Users\ludiw>cd C:\Users\ludiw\Desktop\Instituto_segundo_superior\app
C:\Users\ludiw\Desktop\Instituto_segundo_superior\app>docker init

Welcome to the Docker Init CLI!

This utility will walk you through creating the following files with sensible defaults for your project:
- .dockerignore
- Dockerfile
- compose.yaml
- README.Docker.md

Let's get started!

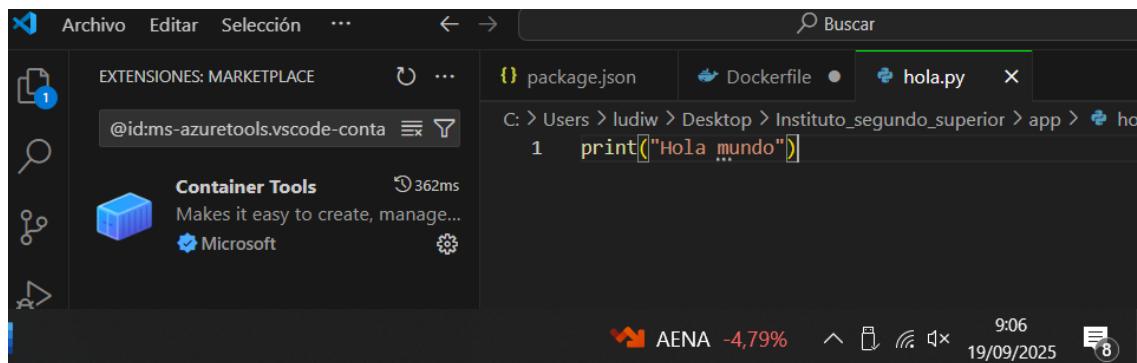
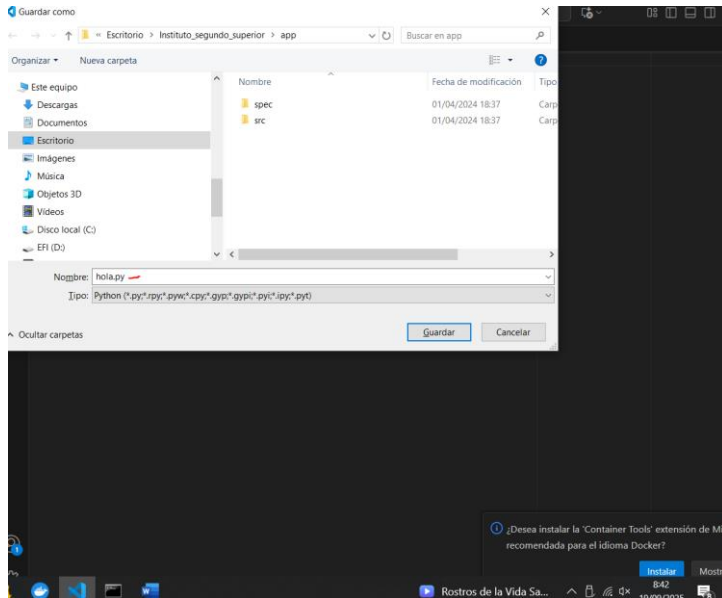
Warning → The following Docker files already exist in this directory:
- Dockerfile

? Do you want to overwrite them? Yes
? What application platform does your project use? [Use arrows to move, type to filter]
> Node - (detected) suitable for a Node server application
  Go - suitable for a Go server application
  Python - suitable for a Python server application
  Rust - suitable for a Rust server application
  ASP.NET Core - suitable for an ASP.NET Core application
  PHP with Apache - suitable for a PHP web application
  Java - suitable for a Java application that uses Maven and packages as an uber jar
  Other - general purpose starting point for containerizing your application
  Don't see something you need? Let us know!
  Quit
```

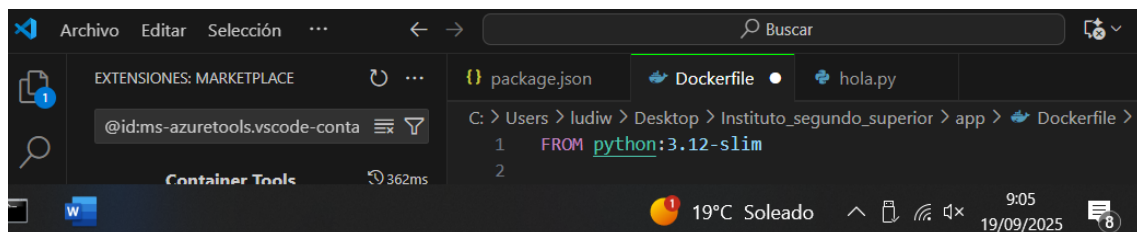
Ejercicio

- Crea una aplicación de Python simple que imprima en pantalla “Hola Mundo” en un fichero llamado hola.py. Después escribe un Dockerfile que cumpla los siguientes requisitos:

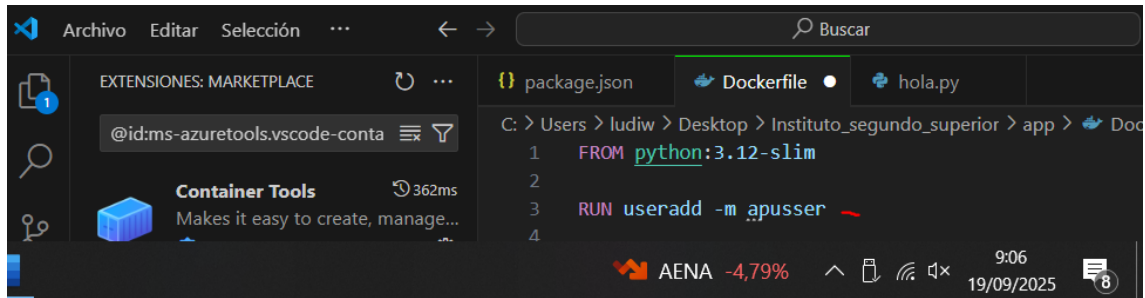
Creamos el fichero hola.py



- El contenedor debe usar una imagen base ligera, como python:3.X-slim o python:3.X-alpine.



- Se debe crear un usuario no root para ejecutar el script.



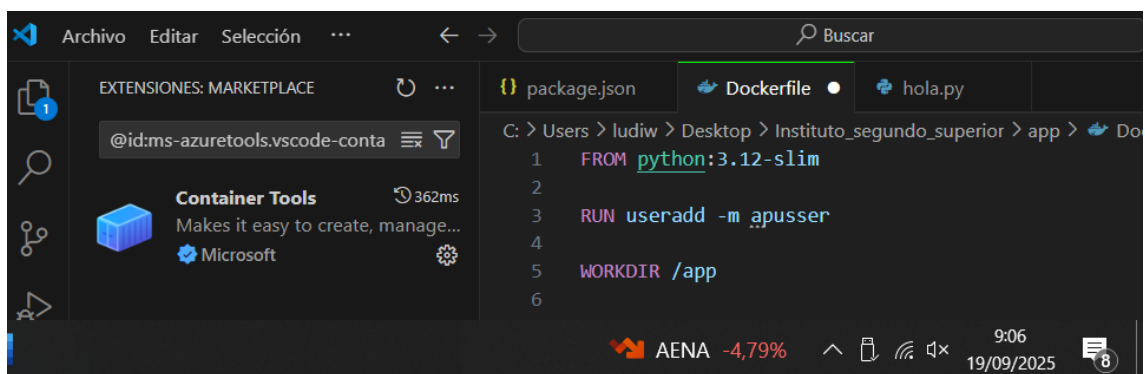
The screenshot shows the Visual Studio Code interface. The left sidebar displays the 'EXTENSIONES: MARKETPLACE' with the 'Container Tools' extension by Microsoft. The main editor area shows a file named 'Dockerfile' with the following content:

```
1 FROM python:3.12-slim
2
3 RUN useradd -m apusser
4
```

The status bar at the bottom indicates the file is located at 'C:\Users\ludiw\Desktop\Instituto_segundo_superior\app\Doc' and the system clock shows 9:06 on 19/09/2025.

- El Dockerfile debe estar limpio:

- Usar WORKDIR.

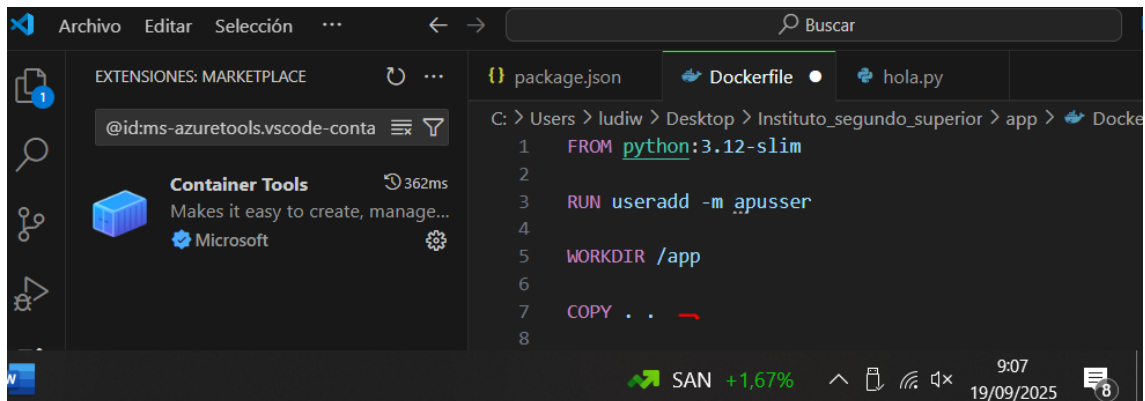


The screenshot shows the Visual Studio Code interface. The left sidebar displays the 'EXTENSIONES: MARKETPLACE' with the 'Container Tools' extension by Microsoft. The main editor area shows a file named 'Dockerfile' with the following content:

```
1 FROM python:3.12-slim
2
3 RUN useradd -m apusser
4
5 WORKDIR /app
6
```

The status bar at the bottom indicates the file is located at 'C:\Users\ludiw\Desktop\Instituto_segundo_superior\app\Doc' and the system clock shows 9:06 on 19/09/2025.

- Usar COPY de forma precisa.



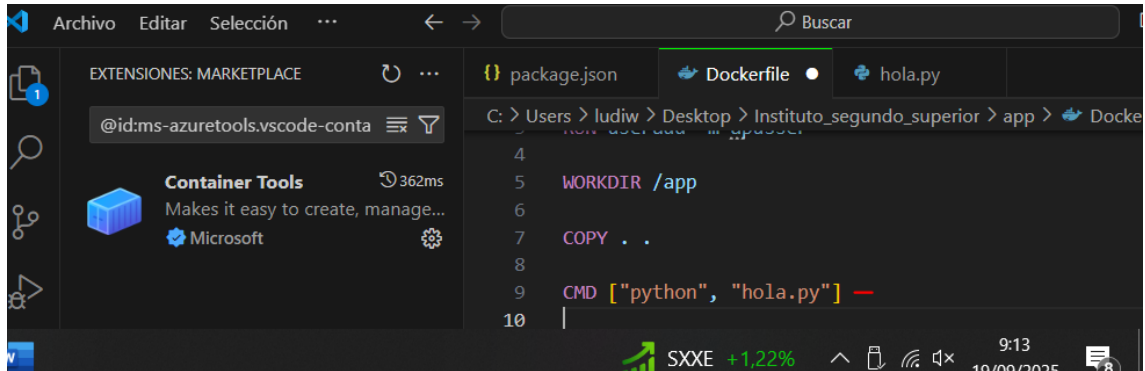
The screenshot shows the Visual Studio Code interface. The left sidebar displays the 'EXTENSIONES: MARKETPLACE' with the 'Container Tools' extension by Microsoft. The main editor area shows a file named 'Dockerfile' with the following content:

```
1 FROM python:3.12-slim
2
3 RUN useradd -m apusser
4
5 WORKDIR /app
6
7 COPY . .
8
```

The status bar at the bottom indicates the file is located at 'C:\Users\ludiw\Desktop\Instituto_segundo_superior\app\Doc' and the system clock shows 9:07 on 19/09/2025.

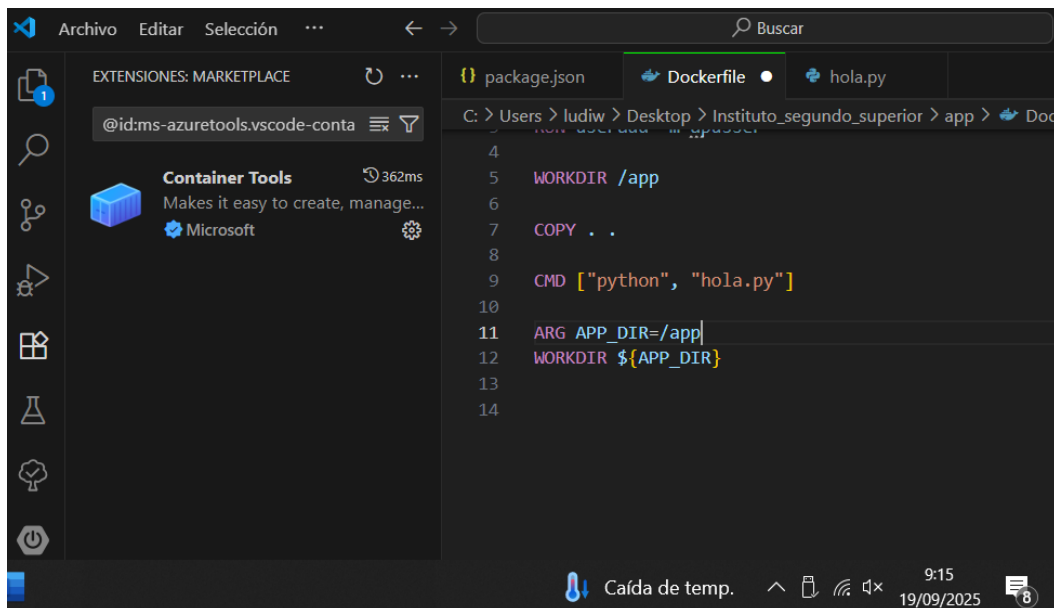
- Usar CMD.

- El script debe ejecutarse automáticamente al iniciar el contenedor.



- Se debe definir un argumento de build llamado APP_DIR para cambiar el directorio de trabajo fácilmente.

El ARG define el argumento con el valor por defecto y el WORKDIR establece el directorio de trabajo en el contenedor



- La imagen final no debe contener pip, build tools ni cache si no son necesarias.

Esto evita que pip deje archivos temporales en la imagen

