Contenido

[Angular + Jest 2](#_Toc101128136)

[Eliminar las dependencias de Jasmine y Karma 2](#_Toc101128137)

[Incorporar Jest 4](#_Toc101128138)

[Carpeta Coverage 5](#_Toc101128139)

[Correr test unitarios 6](#_Toc101128140)

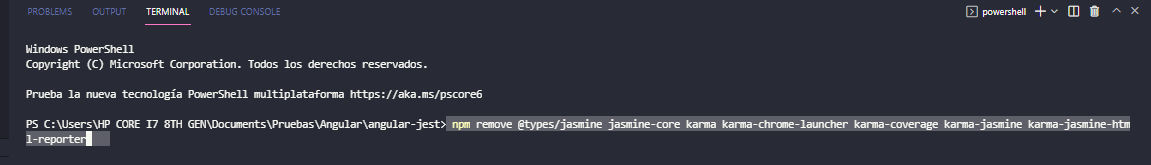
[Ejemplo de estructura de un fichero de prueba 8](#_Toc101128141)

[Configurar fichero test 8](#_Toc101128142)

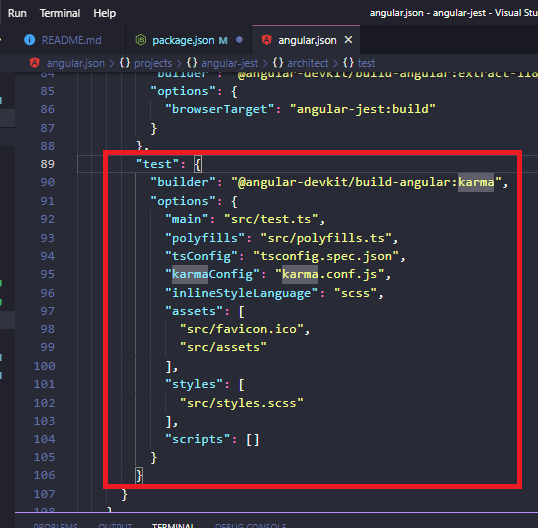
# Angular + Jest

## Eliminar las dependencias de Jasmine y Karma

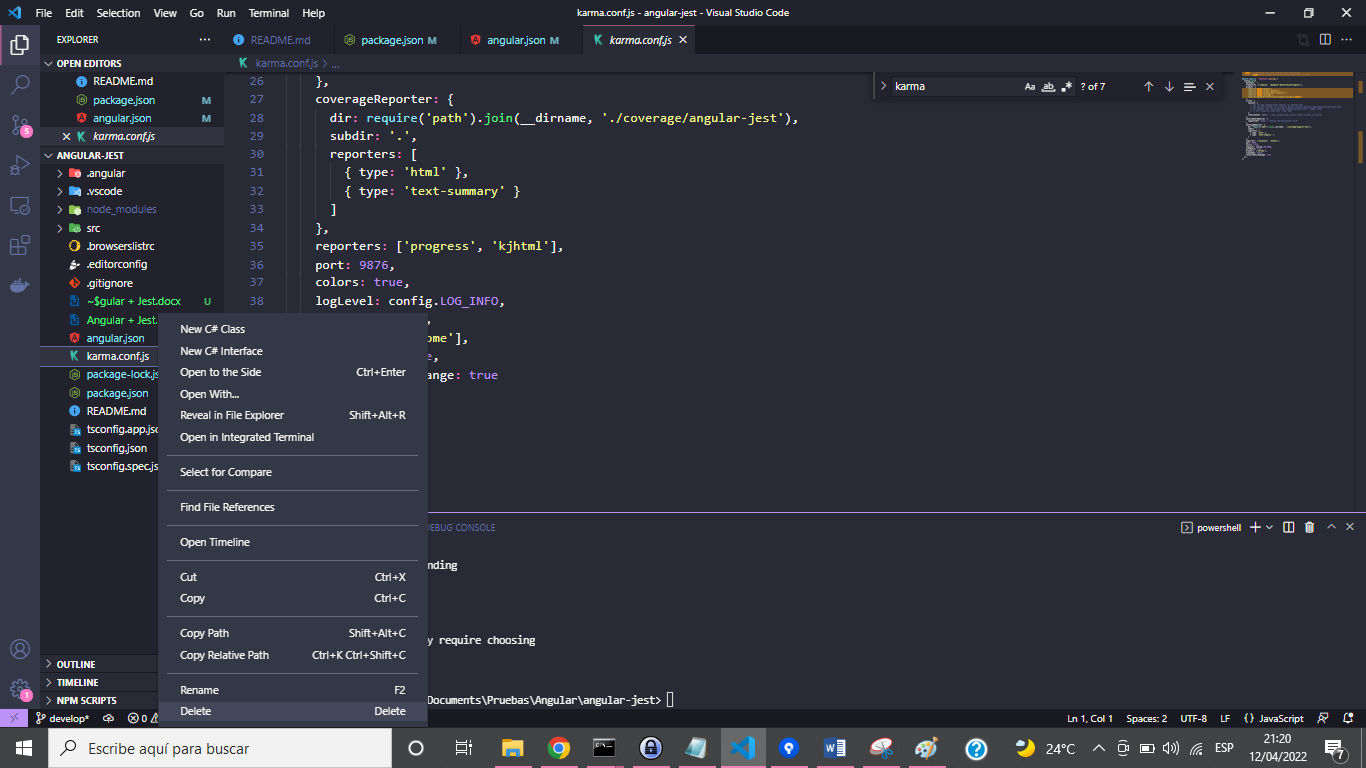
1. Ubicarse en el archivo package.json
2. Buscar todas las dependencias relacionadas con Jasmine o Karma.
3. Abrir una nueva terminal y digitar el siguiente comando.



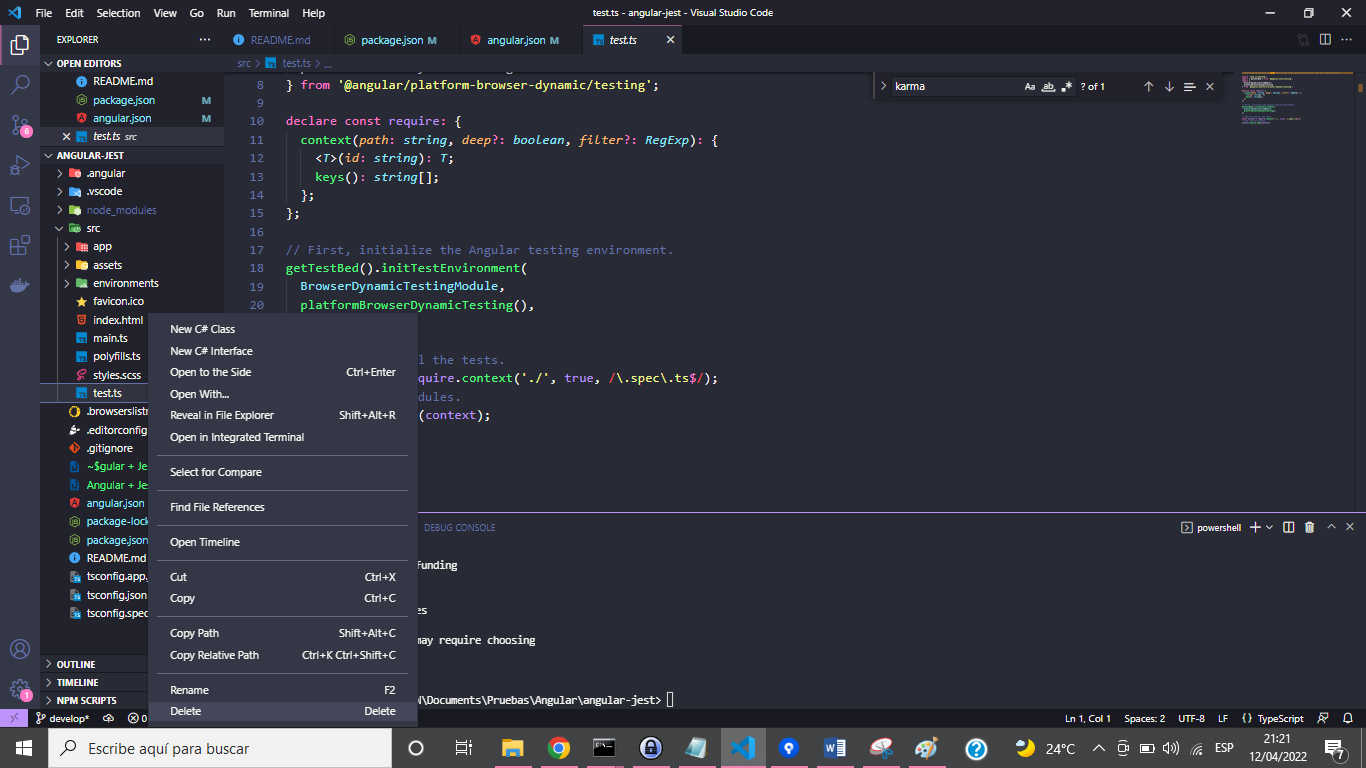
1. Eliminar en los scripts la línea que hace referencia para correr los test por defecto (ng test).
2. Ir al archivo angular.json
3. Eliminar todas las referencias test: Jasmine y Karma.



1. Eliminar el fichero karma.conf.js

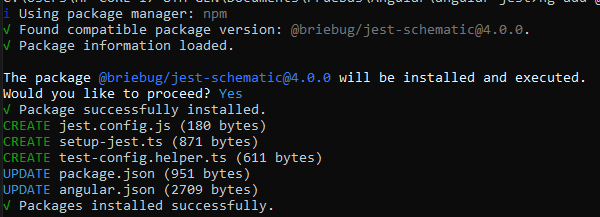


1. Eliminar el fichero src/test.ts



# Incorporar Jest

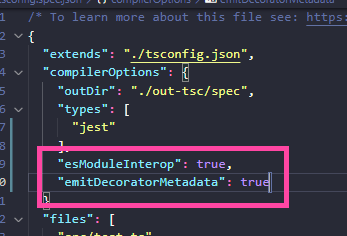
1. Dirigirse a <https://github.com/briebug/jest-schematic>
2. Copiar el comando:



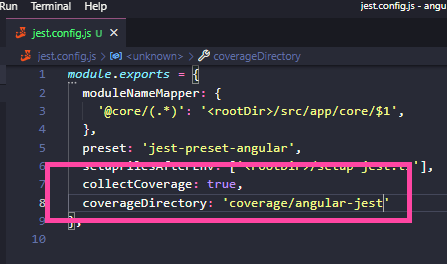
1. Modificar el archivo tsconfig.spec.json para que apunte a Jest



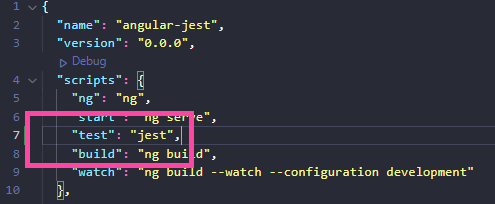
1. Adicionar los siguientes flags en el tsconfig.spec.json



1. Modificar el archivo jest.config.js para adicionar coverage.



1. Actualizar el fichero package.json



*Observaciones:* En algunos casos os apararezca en el fichero "setup-jest.ts" el siguiente import:

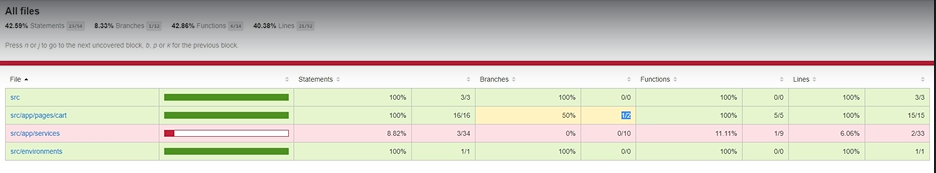
import 'jest-preset-angular’

Es muy importante que lo cambiéis por este otro que indico a continuación.

import 'jest-preset-angular/setup-jest';

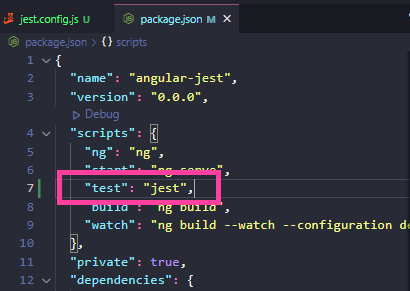
## Carpeta Coverage

*Coverage*, se puede interpretar como el porcentaje de líneas de código que tenemos cubierto.

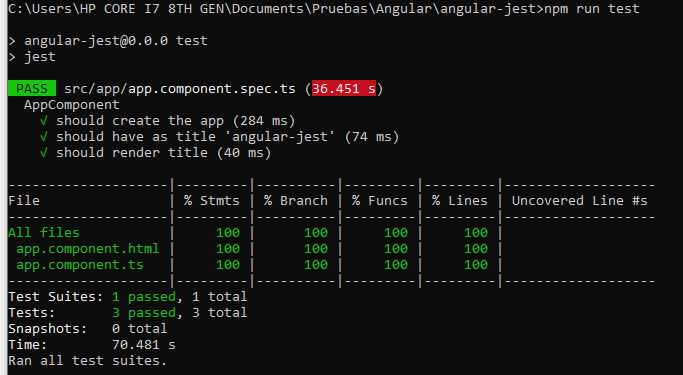


### Correr test unitarios

Anteriormente, habíamos configurado un script dentro del package.json



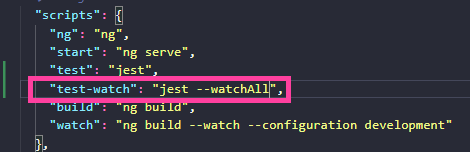
Para ejecutarlo, corremos en terminal



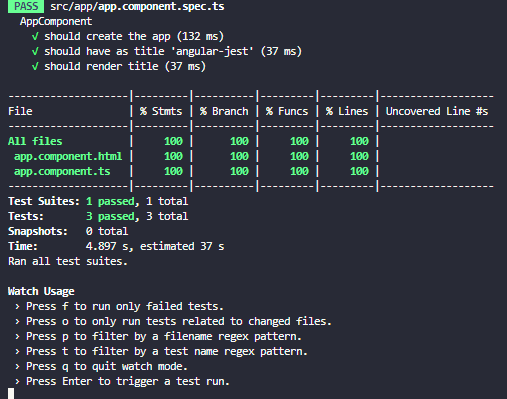
* Test suites: ficheros spec.
* Tests: Conjunto de test dentro de cada fichero de spec.

**Configurar jest para que siempre se quede a la escucha:**

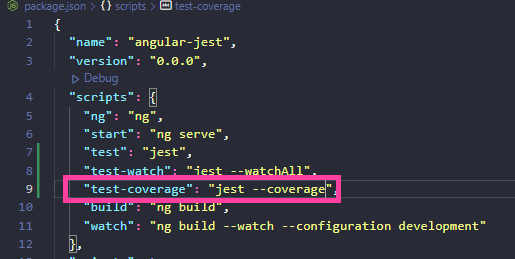
Configuramos otro comando dentro del fichero package.json:



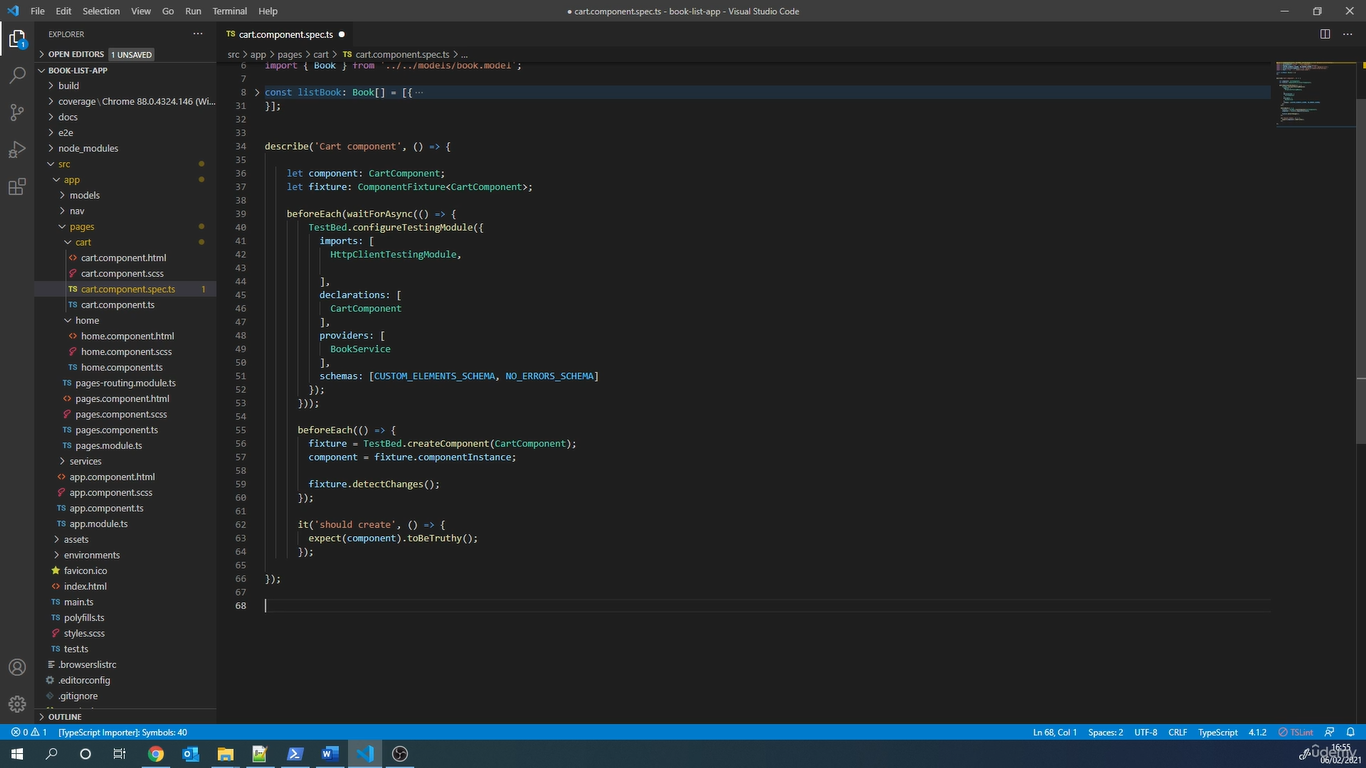
Luego, de ejecutar los test, Jest se queda esperando con un listado de comando de interacción.



**Para crear la carpeta coverage:**

****

## Ejemplo de estructura de un fichero de prueba



# Configurar fichero test

* Un fichero de test va asociado a un componente o servicio.
* Los test unitarios solo ejecutan métodos de su propia clase.
* Si se necesita de un método externo, se debe simular.
* Usualmente, todo nuestro tests estarán contenidos en el describe del fichero.
  + Consideraciones, ocupar el HttpClientTestingModule para no hacer peticiones reales.

## xit, fit, it, only, describe, fdescribe.

* **xit**: Permite pasar “por alto” o anular un test.
* **xdescribe:** Permite pasar “por alto” o anular un test suite y su correspondiente conjunto de tests.
* **fit / it.only:** Permite que se ejecute únicamente el test indicado y los demás son anulados no importando si estaban con it o xit. Es una priorización de test en concreto. Recomendando al ir construyendo test.
* **fdescribe (Jest):** Permite que se ejecuten todos los tests.

## BeforeEach, BeforteAll, AfterEach, AfterAll.

* BeforeEach: Función que se ejecuta antes de cada test.
* BeforeAll: Se ejecuta una vez antes de todo.
* AfterEach: Función que se ejecuta después/terminar cada test.
* AfterAll: Se ejecutará al finalizar todos los test.