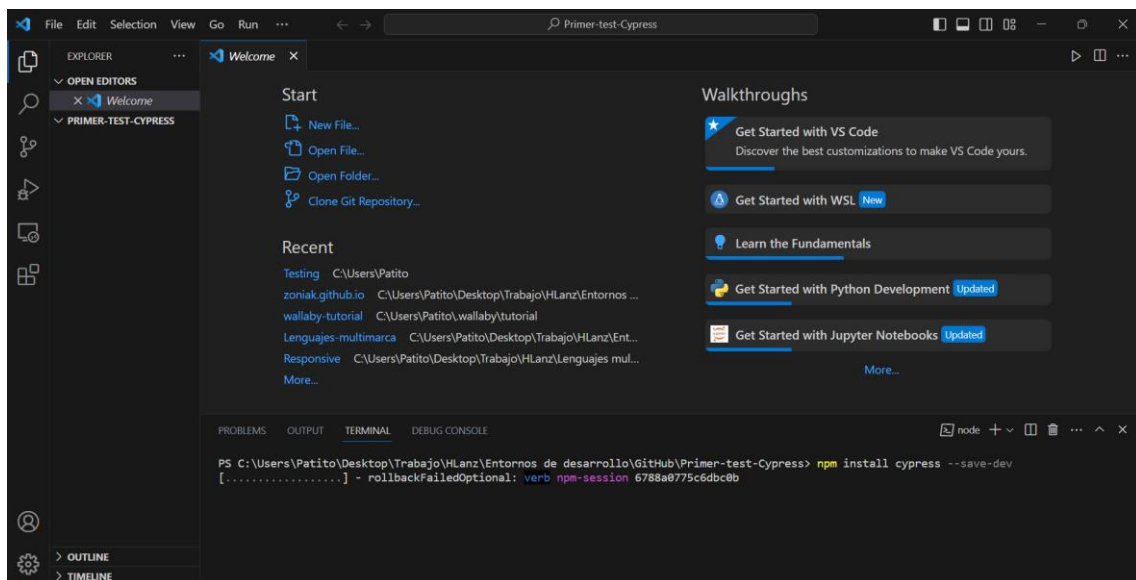
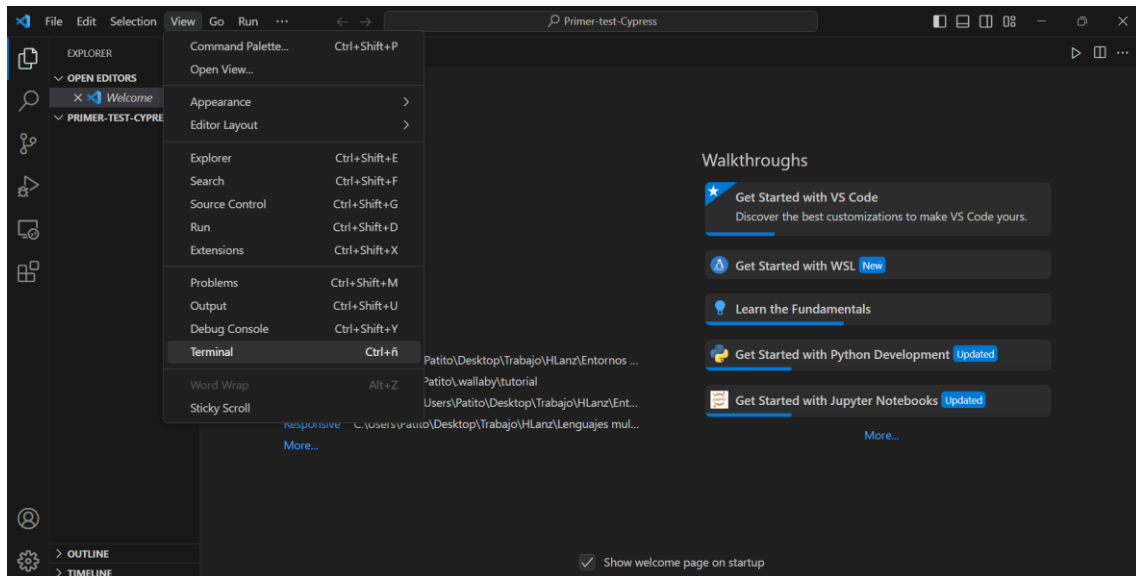


CREACIÓN DEL PRIMER TEST CON CYPRESS

ELEMENTOS NECESARIOS PARA LA PRACTICA

Lo primero de todo va a ser tener instalado nvm para Windows. Si aun no lo tienes te dejo este [link](#) donde puedes encontrar los archivos necesarios para instalarlo. Una vez lo tengamos, navegamos hasta la carpeta donde este el proyecto (con CMD y cd o simplemente abriendo el terminal en el proyecto en Visual Studio Code), y procedemos a instalar Cypress con el siguiente comando:

npm install cypress --save-dev



Algo curioso que os puede venir bien para el futuro es que con nvm podemos crear la estructura de un proyecto, para ello vamos a ejecutar el siguiente comando:

npm init vite

Al ejecutar esto nos pedirá datos como el nombre del proyecto o qué tipo de proyecto queremos crear:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

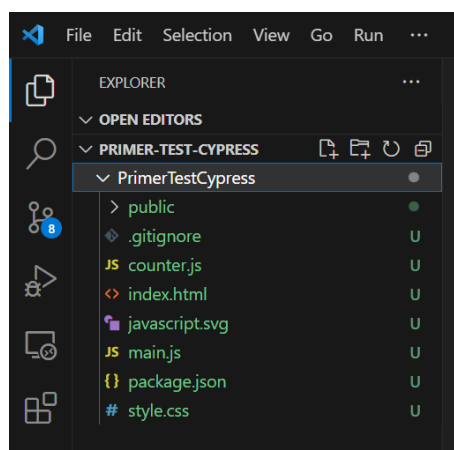
✓ Project name: ... PrimerTestCypress
✓ Package name: ... primertestcypress
? Select a framework: » - Use arrow-keys. Return to submit.
> Vanilla
  Vue
  React
  Preact
  Lit
  Svelte
```

Vamos a crear un proyecto Vanilla, y vamos a elegir como lenguaje JavaScript:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

✓ Select a framework: » Vanilla
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
> JavaScript
```

Ahora ya tenemos nuestro proyecto creado:

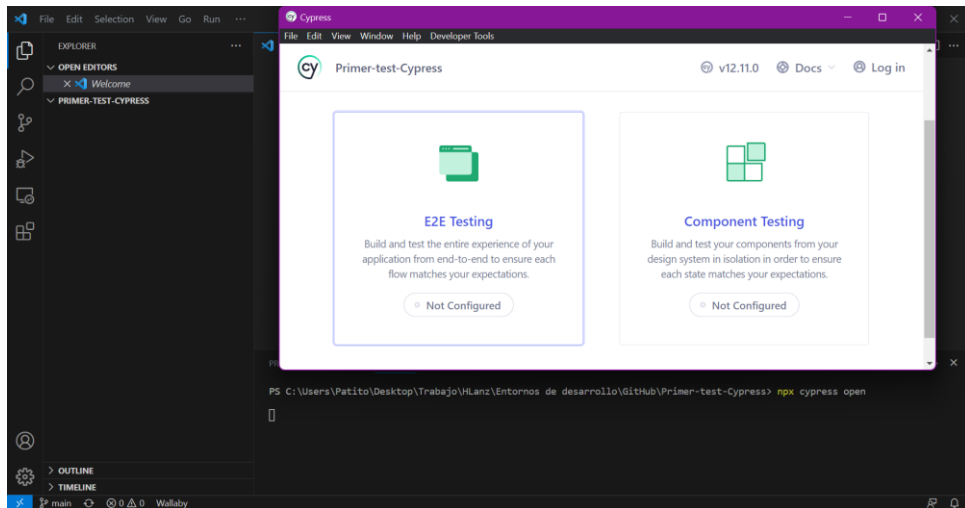


EMPEZANDO A ESCRIBIR EL PRIMER TEST

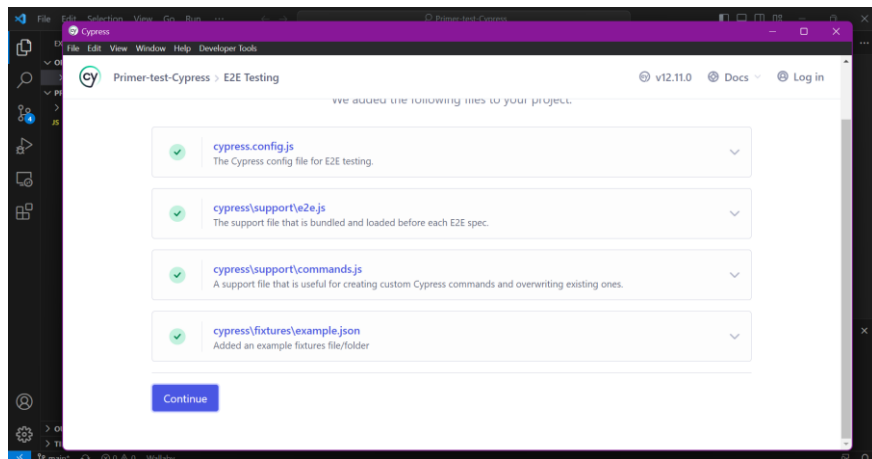
Vale ahora que tenemos instalado nvm y Cypress vamos a lanzar nuestro primer test. Para ello creamos un repositorio en GitHub para esta clase y empezamos a trabajar en el en Visual Studio Code. En el terminal ejecutamos este comando:

`npx cypress open`

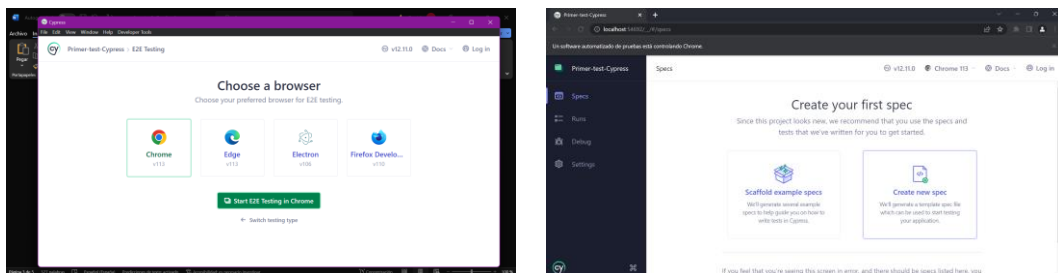
Se nos abrirá la aplicación de Cypress, vamos a crear el entorno para test E2E:



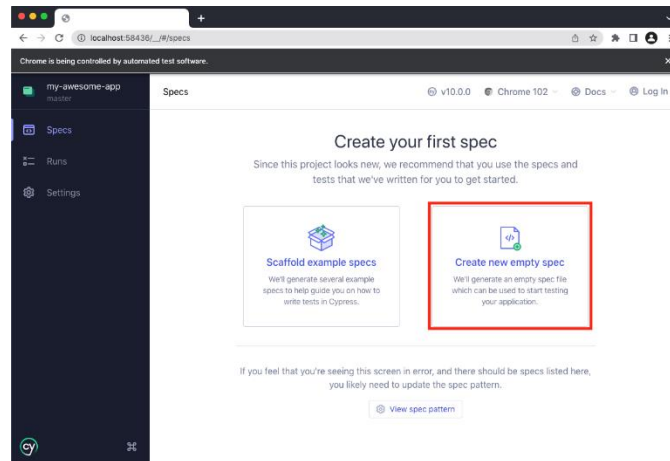
Le damos a continue:



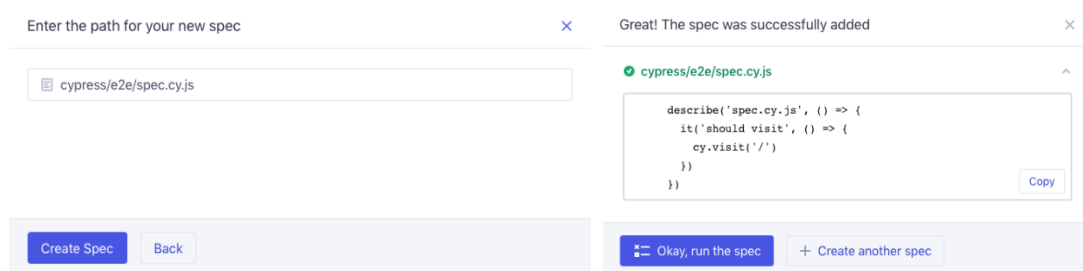
Y ejecutamos el programa en el navegador:



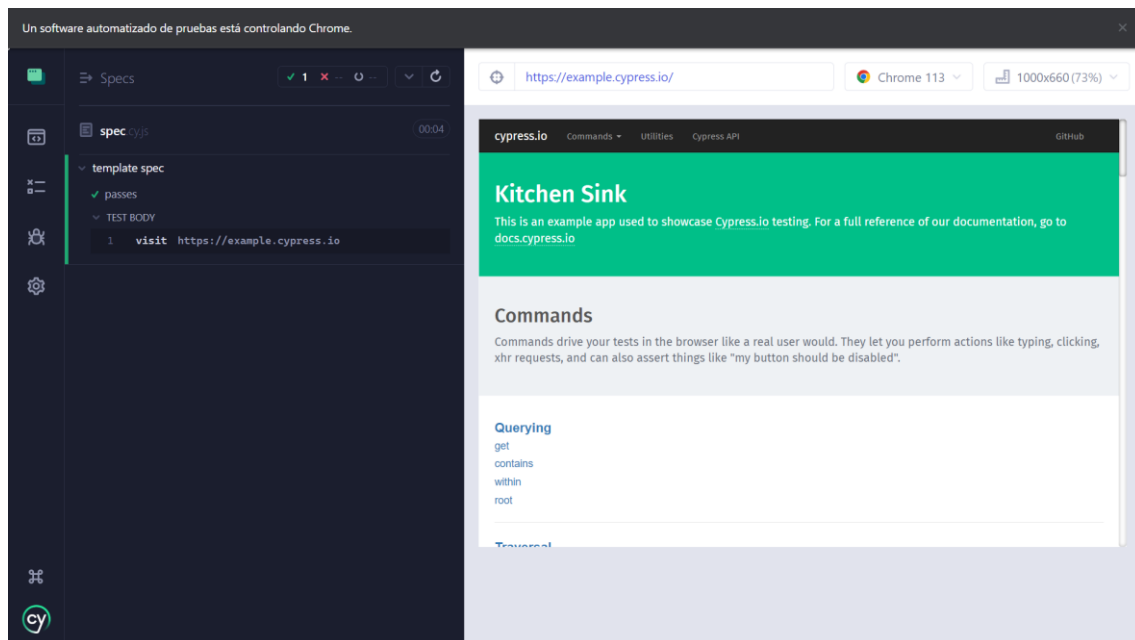
Ahora vamos a crear el primer test para comprobar que funciona. Para eso primero hacemos click en *Create new spec*



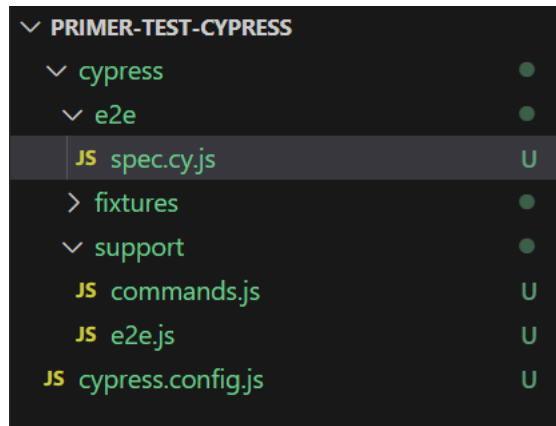
Procedemos a dejar todo como viene por defecto:



Y a continuación se abrirá este dónde se ejecutará y nos dará un resultado:



Ahora vamos a escribir nosotros nuestras pruebas. Para ello buscamos el archivo que se ha creado el cual es spec.cy.js que se encuentra en la siguiente ruta:



Vamos a escribir una aplicación simple con la que testear el código. Lo primero es crear las siguientes funciones:

```
//-- Dummy application code
let add = (a,b) => a+b;
let subtract = (a,b) => a-b;
let divide = (a,b) => a/b;
let multiply = (a,b) => a*b;
```

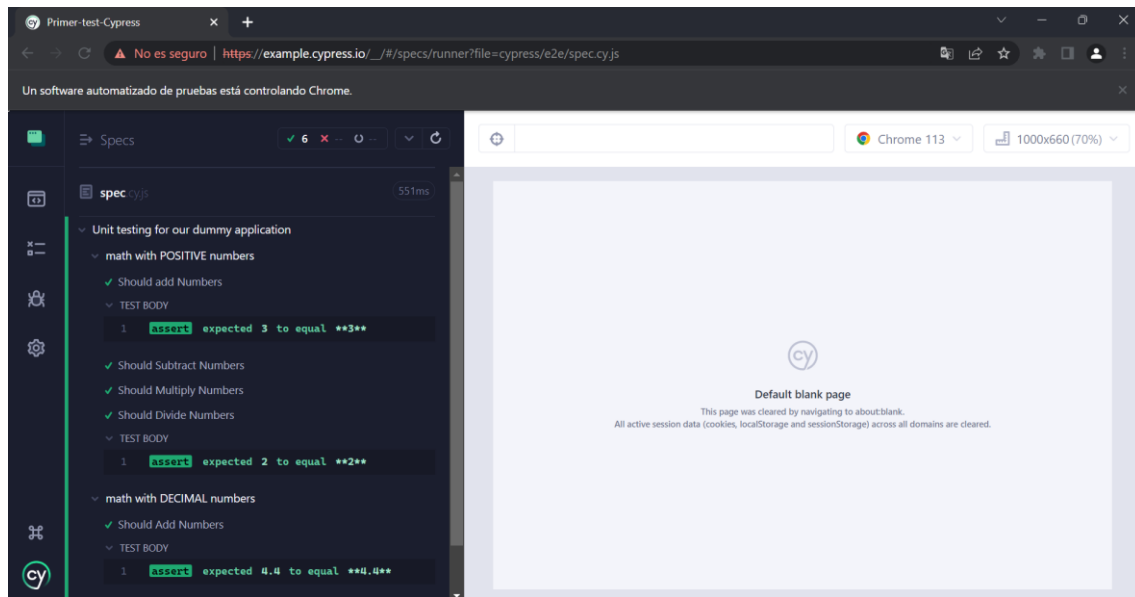
[Si pinchas en la imagen te llevara al código](#)

Después vamos a preparar los test, y para esto vamos a usar **describe**. Este es un método que nos sirve para agrupar en Cypress uno o más test relacionados. Procedemos a escribir el siguiente código:

```
describe('Unit testing for our dummy application', () => {
  context('math with POSITIVE numbers', () => {
    it('Should add Numbers', () => {
      expect(add(1,2)).to.eq(3);
    });
    it('Should Subtract Numbers', () => {
      expect(subtract(2,1)).to.eq(1);
    });
    it('Should Multiply Numbers', () => {
      expect(multiply(2,2)).to.eq(4);
    });
    it('Should Divide Numbers', () => {
      expect(divide(4,2)).to.eq(2);
    });
  });
  context('math with DECIMAL numbers', () => {
    it('Should Add Numbers', () => {
      expect(add(2.2,2.2)).to.eq(4.4);
    });
    it('Should Subtract Numbers', () => {
      expect(subtract(4.4, 2.2)).to.eq(2.2);
    });
  });
});
```

[Si pinchas en la imagen te llevara al código](#)

Según los vayamos escribiendo el navegador se ira actualizando automáticamente:



Ahora es vuestro turno. Mirar en la [documentación de Cypress](#) distintos tipos de test y probar estos. Por ejemplo con el siguiente código va a visitar una pagina y hacer click :

```
describe('My First Test', () => {
  it('clicks the link "type"', () => {
    cy.visit('https://example.cypress.io')
    cy.contains('type').click()
  })
})
let multiply = (a,b) => a*b;
```

Yo he probado a ejecutarlo con mi portfolio y me ha dado errores que no esperaba...