



Automatización de pruebas con Cypress

Used by Developers and QA

- ▶ Cypress es un framewok de testing automatizado para realizar pruebas sobre cualquier aplicación web



Que podemos obtener:

- ▶ pruebas automatizadas:
- ▶ Pruebas de integración
- ▶ Pruebas unitarias

Documentación

<https://docs.cypress.io/>

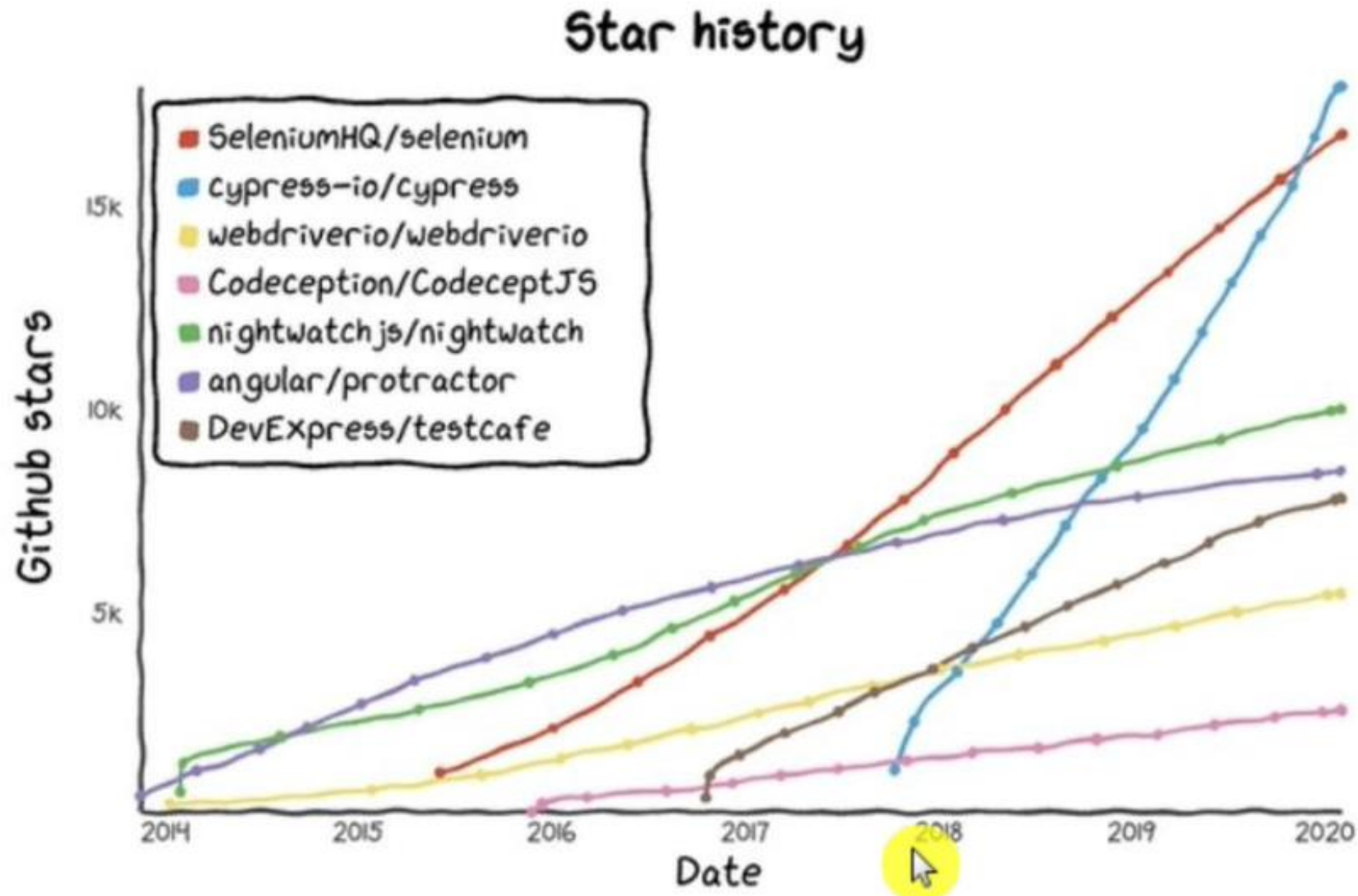
Requirements:

Instalar editor visual Studio Code	
instalar Node JS (interpreta el código de JavaScript permite la compilación)	
Instalar el manejador de paquetes NPM	
Instalar Cypress como dependencia en el proyecto a probar	

Advantages

- ▶ Runner especial para correr pruebas en diferentes navegadores, cualquier browser se puede usar para correr las pruebas.
- ▶ Buena documentación y soporte.
- ▶ Permite Crear fotos videos de las pruebas.
- ▶ Permite generar reportes en Html
- ▶ It's free!
- ▶ All in one testing framework:
assertion library

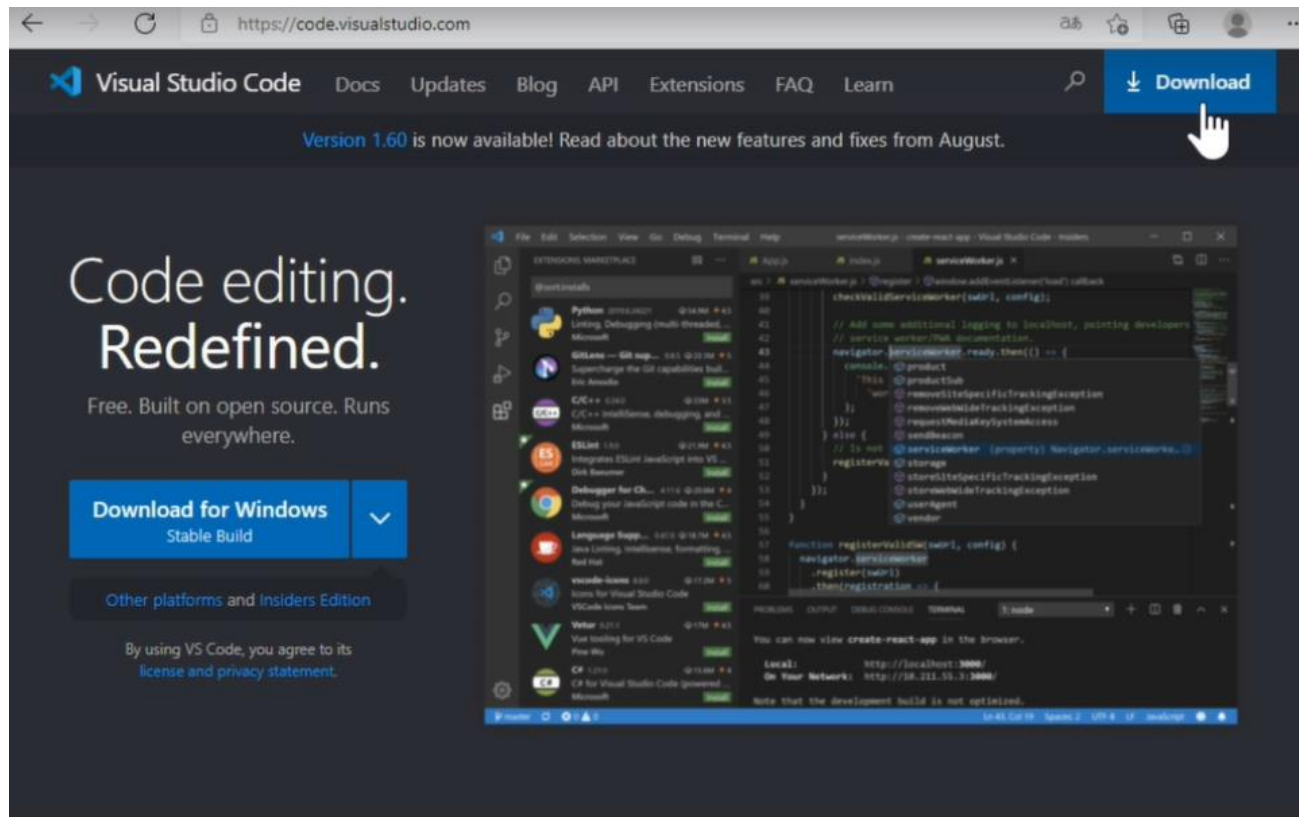
Global rating according to Github



Pasos para la instalación:

- Podemos realizar un proyecto, crear los casos de prueba los cuales se ejecutara en el navegador para validar la funcionalidad de una aplicación web.

Descargar Visual Studio Code



Instalar NodeJs

<https://nodejs.org> > ... ▼

[Node.js](#)

Node.js® es un entorno de ejecución para JavaScript construido con V8, motor de JavaScript de Chrome. Descargar para Linux (x64).

Descarga

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript ...

Documentación

Características de ES6; Guías. Documentación de Referencia ...

Versiones

El estado de lanzamiento de LTS es "long-term support", de ...

Acerca

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript ...

[Más resultados de nodejs.org »](#)

Descargar para Windows

Node.js® es un entorno de ejecución para JavaScript construido con V8, motor de JavaScript de Chrome.

Node.js 18 is now available!

Descargar para Windows (x64)

16.14.2 LTS

Recomendado para la mayoría

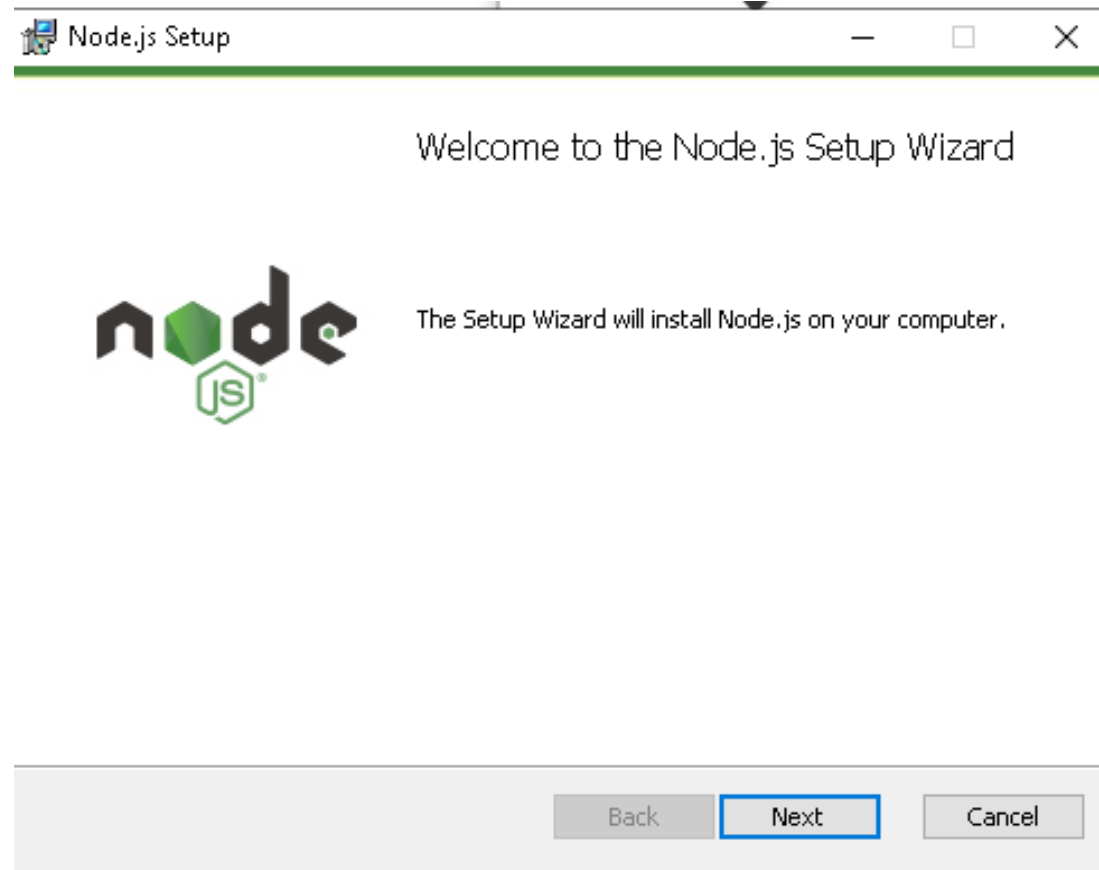
[Otras Descargas](#) | [Cambios](#) | [Documentación de la API](#)

18.0.0 Actual

Últimas características

[Otras D](#) [Descarga 18.0.0 Actual](#) [Documentación de la API](#)

Instalar valores por defecto



Node.js Setup

Completed the Node.js Setup Wizard



Click the Finish button to exit the Setup Wizard.

Node.js has been successfully installed.

Back

Finish

Cancel

Comprobar que se encuentre instalado el nodejs

- Abrir el cmd o por el termina del visual studio code y escribir:

node -v (muestra la versión instalada del nodejs)

npm -v (muestra la versión instalada del manejador de paquetes)

 C:\WINDOWS\system32\cmd.exe

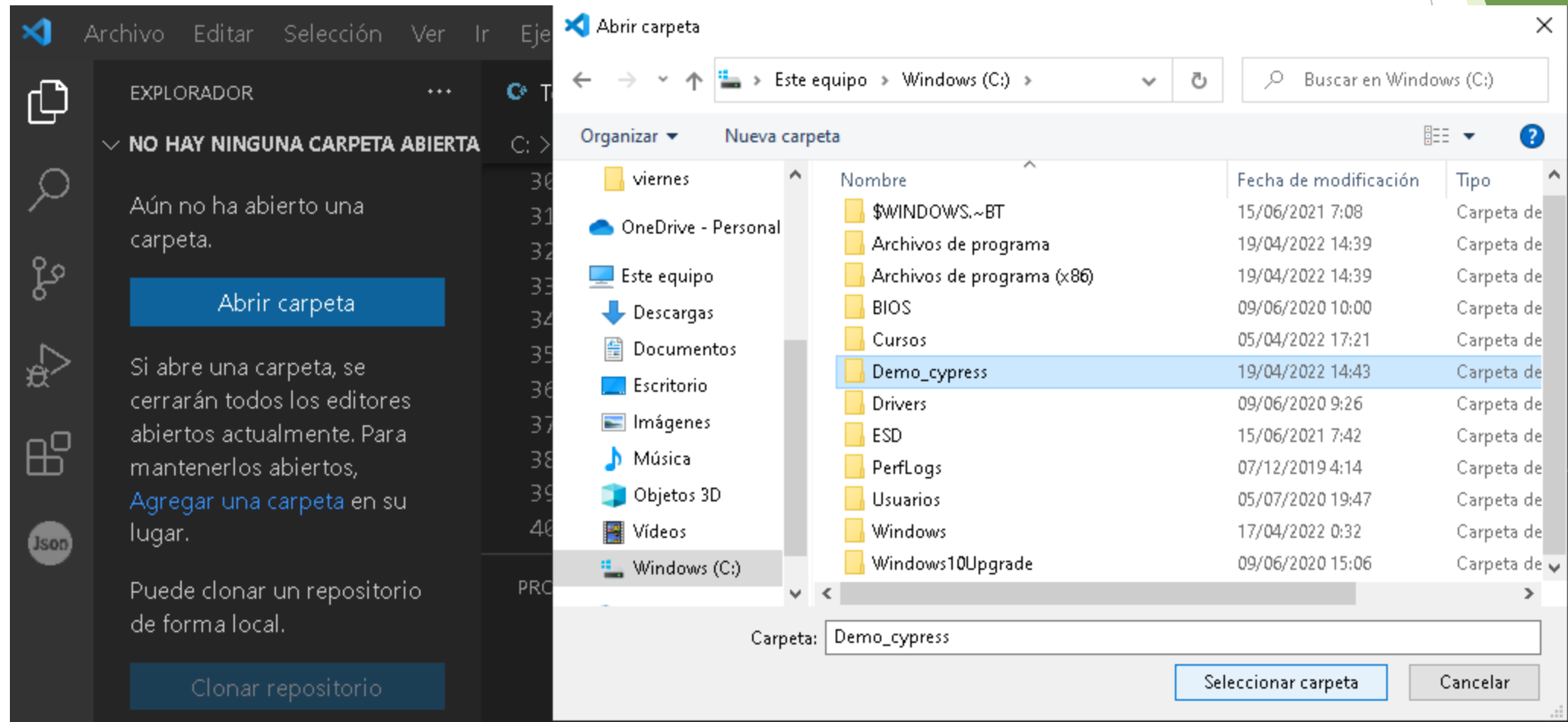
```
Microsoft Windows [Versión 10.0.19043.1645]  
(c) Microsoft Corporation. Todos los derechos reservados.
```

```
C:\Users\kalex>npm -v  
8.6.0
```

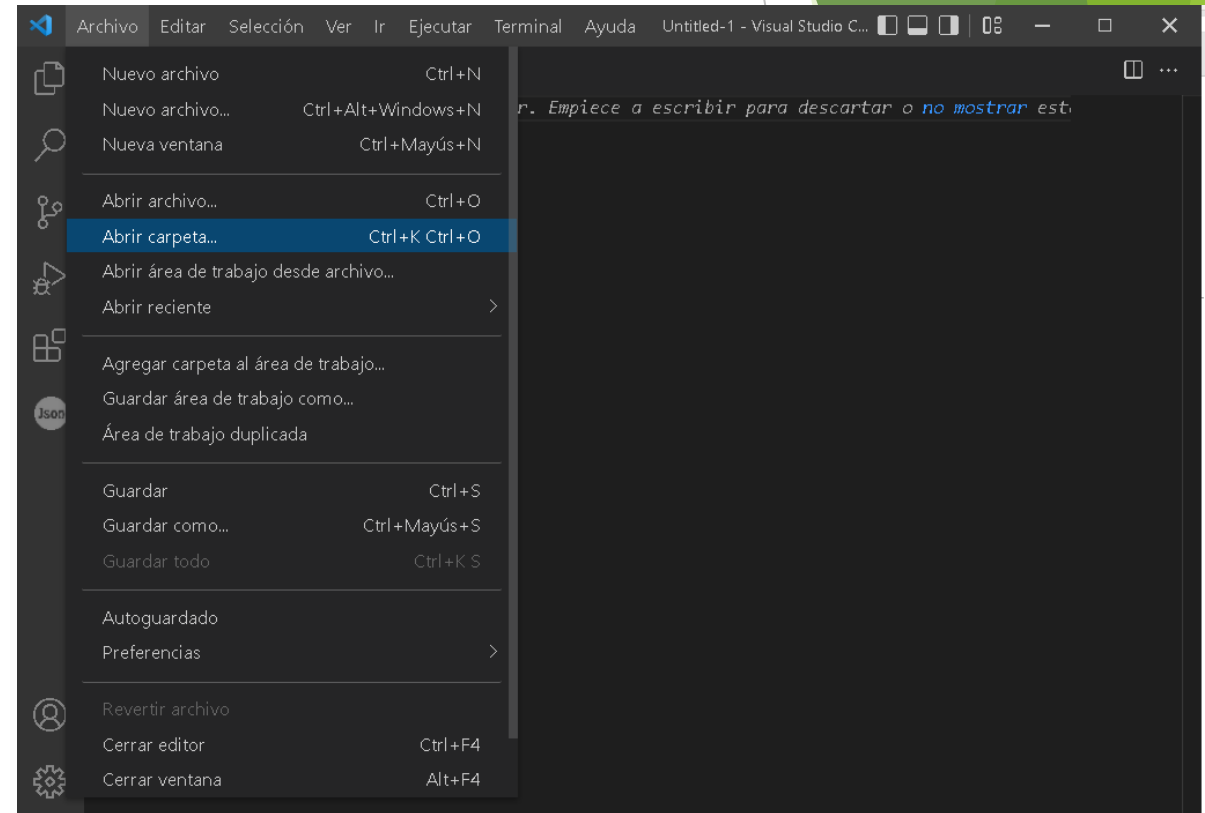
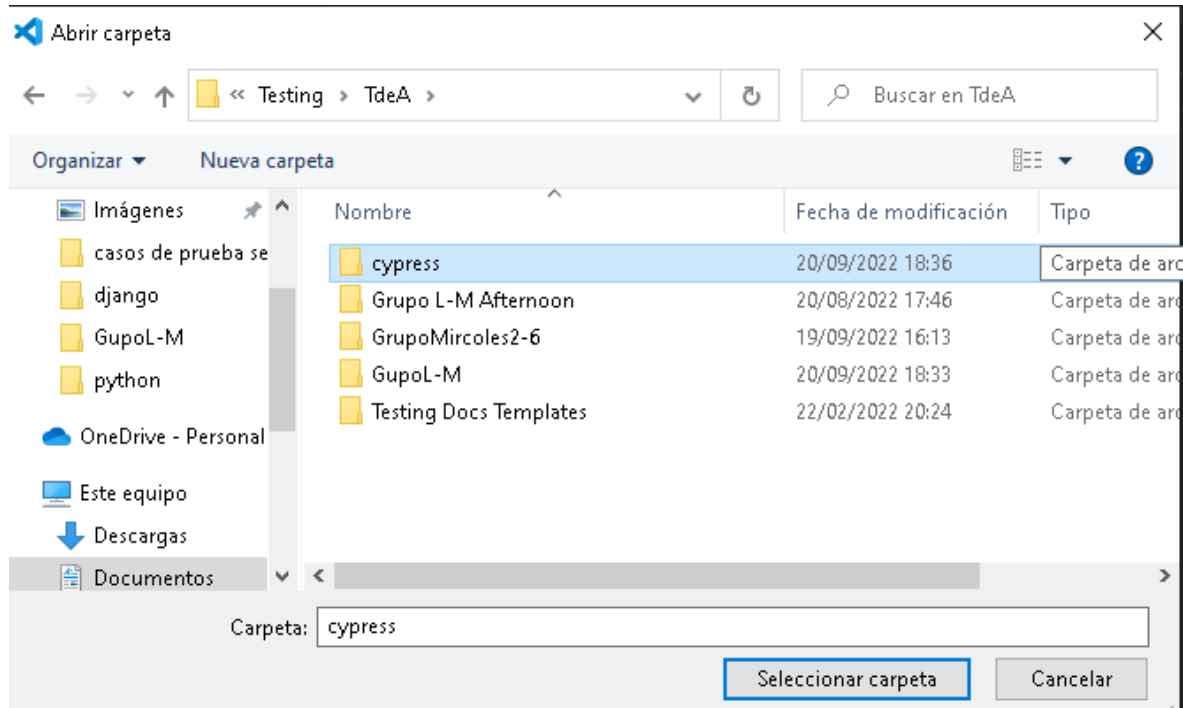
```
C:\Users\kalex>
```

Usar cypress dentro del proyecto:

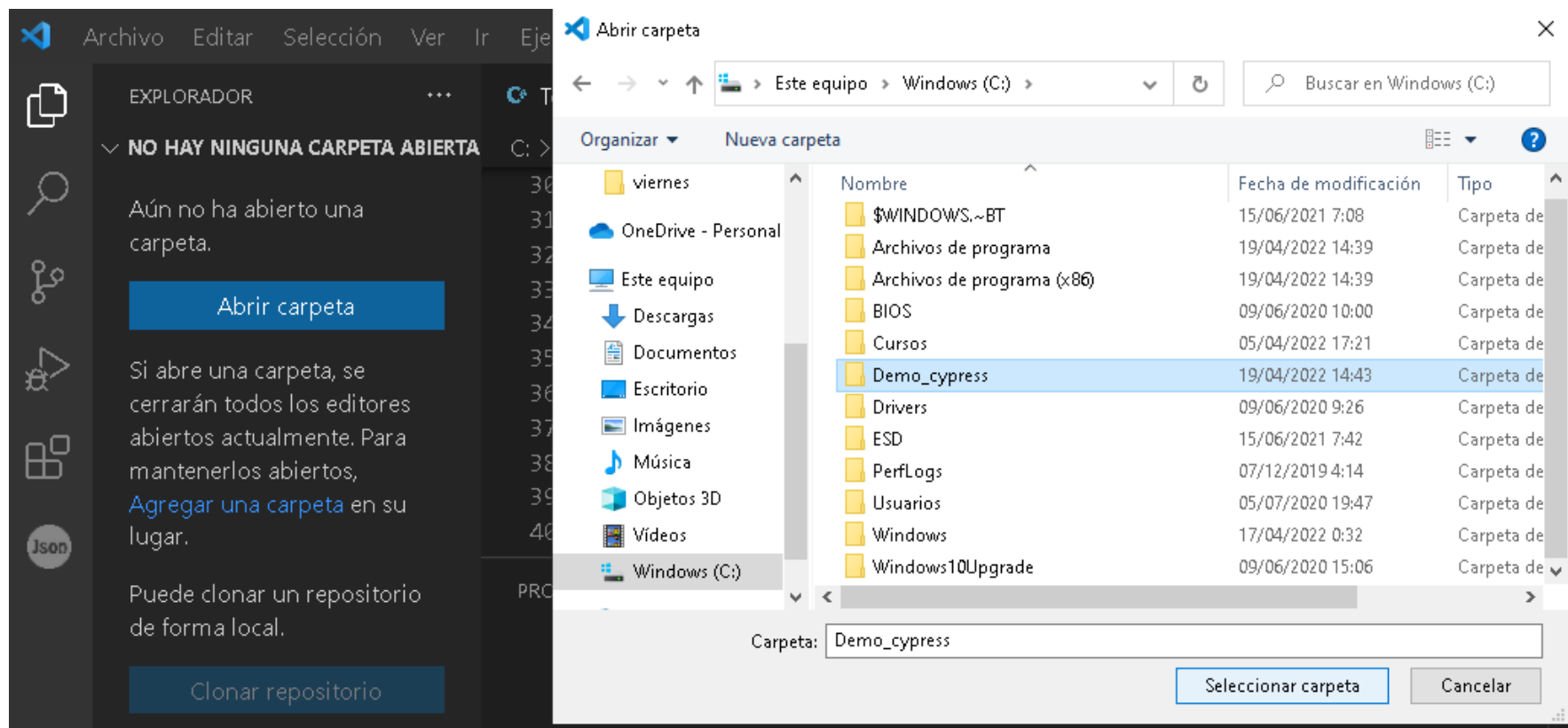
► Abrir visual studio code:



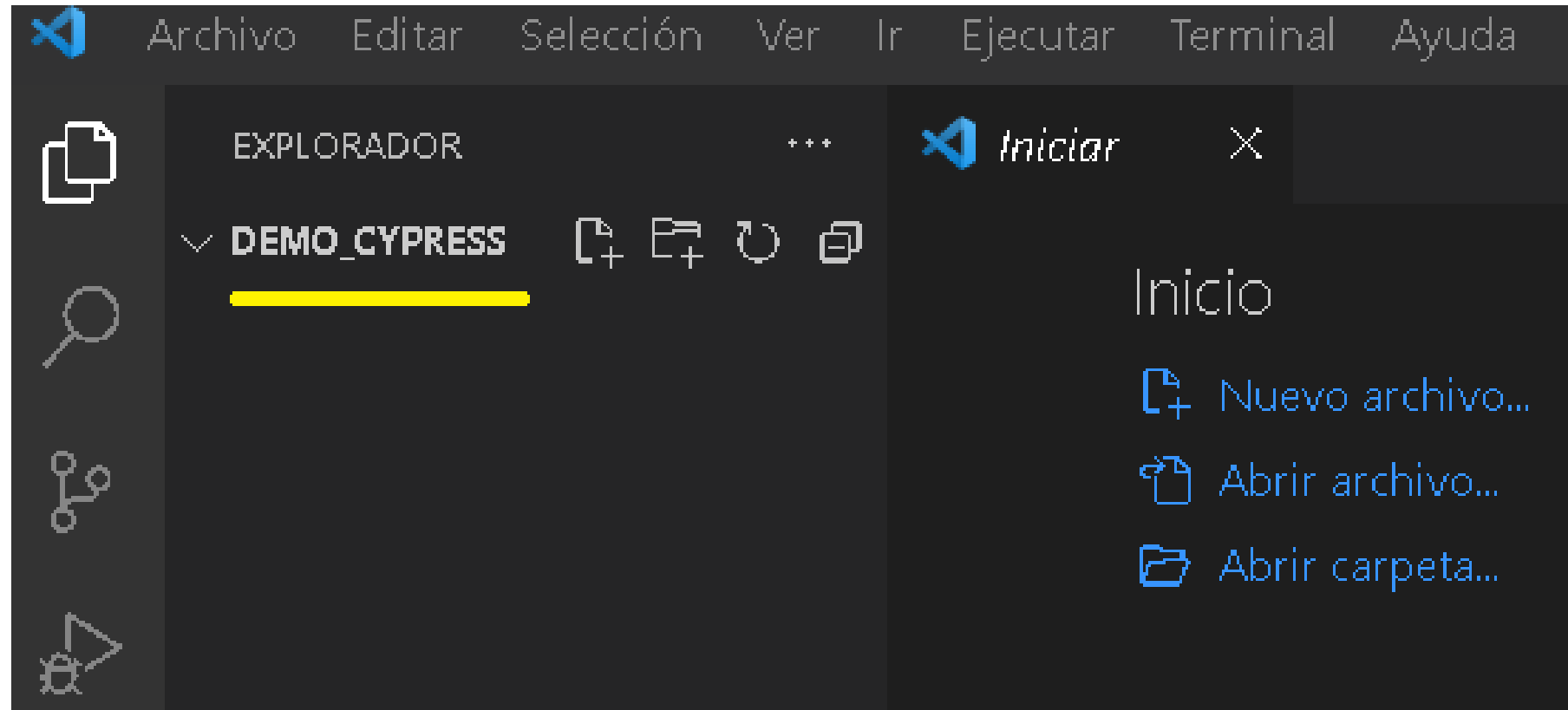
Crear carpeta y abrirla desde Visual Studio:



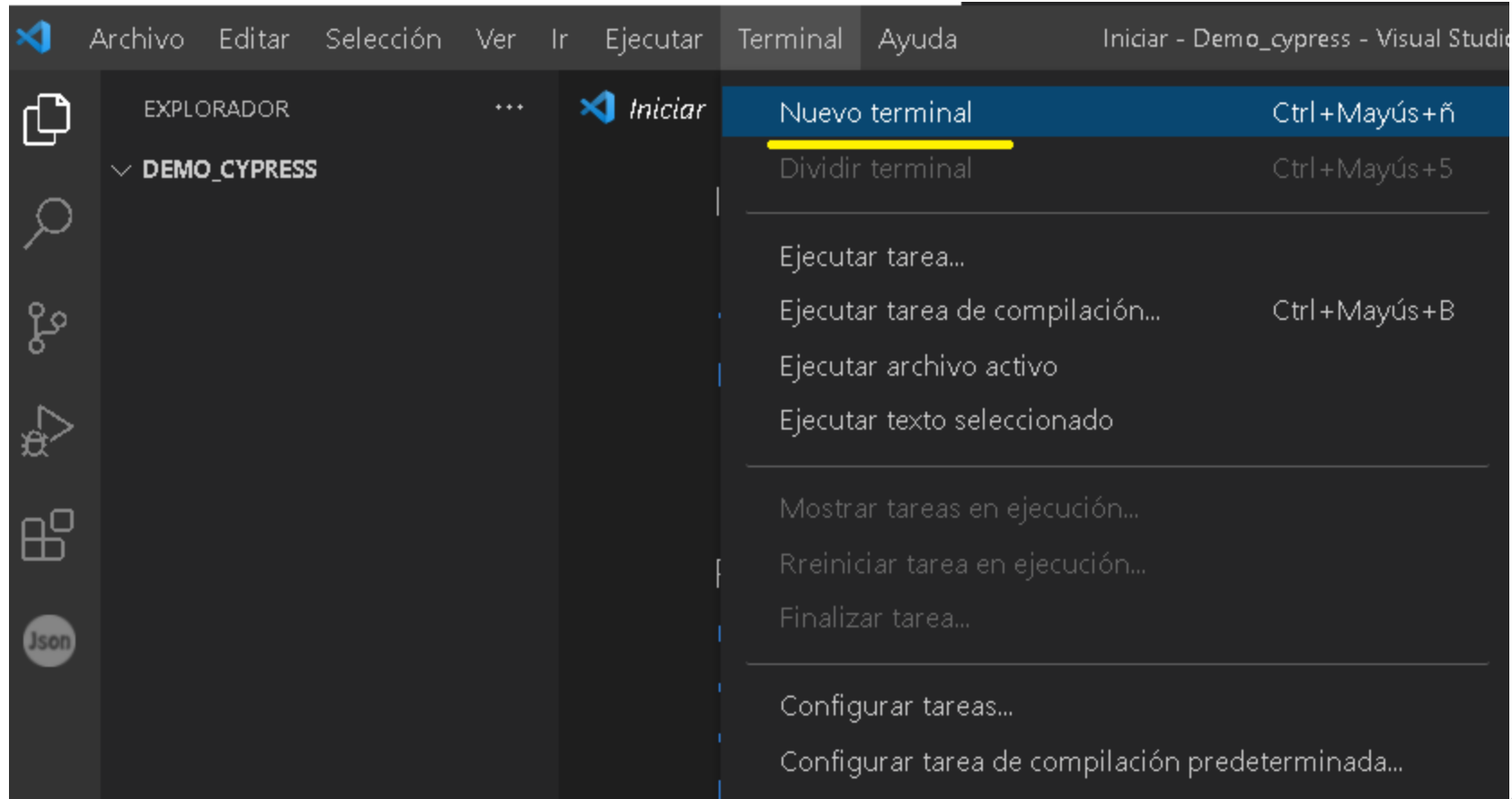
Abrir carpeta para proyecto con Cypress



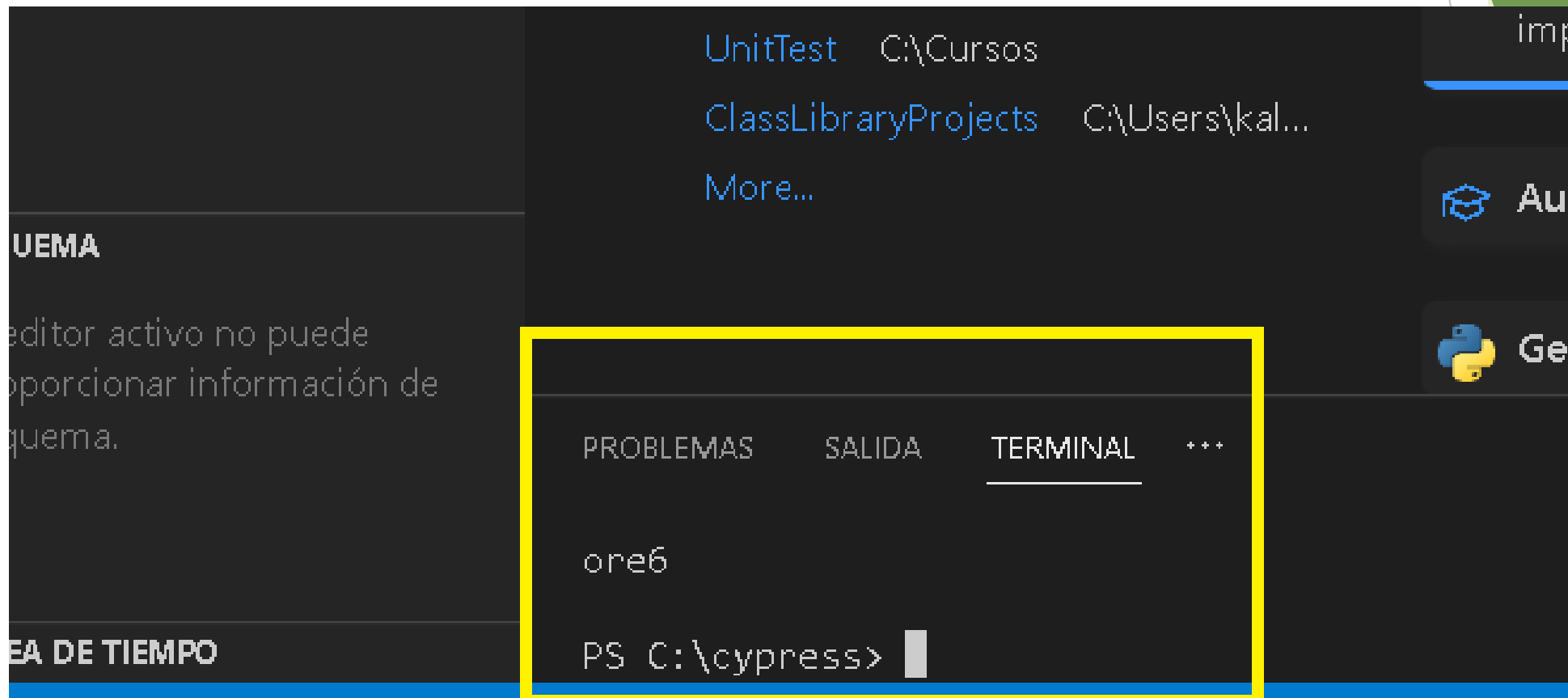
Inicializar proyecto

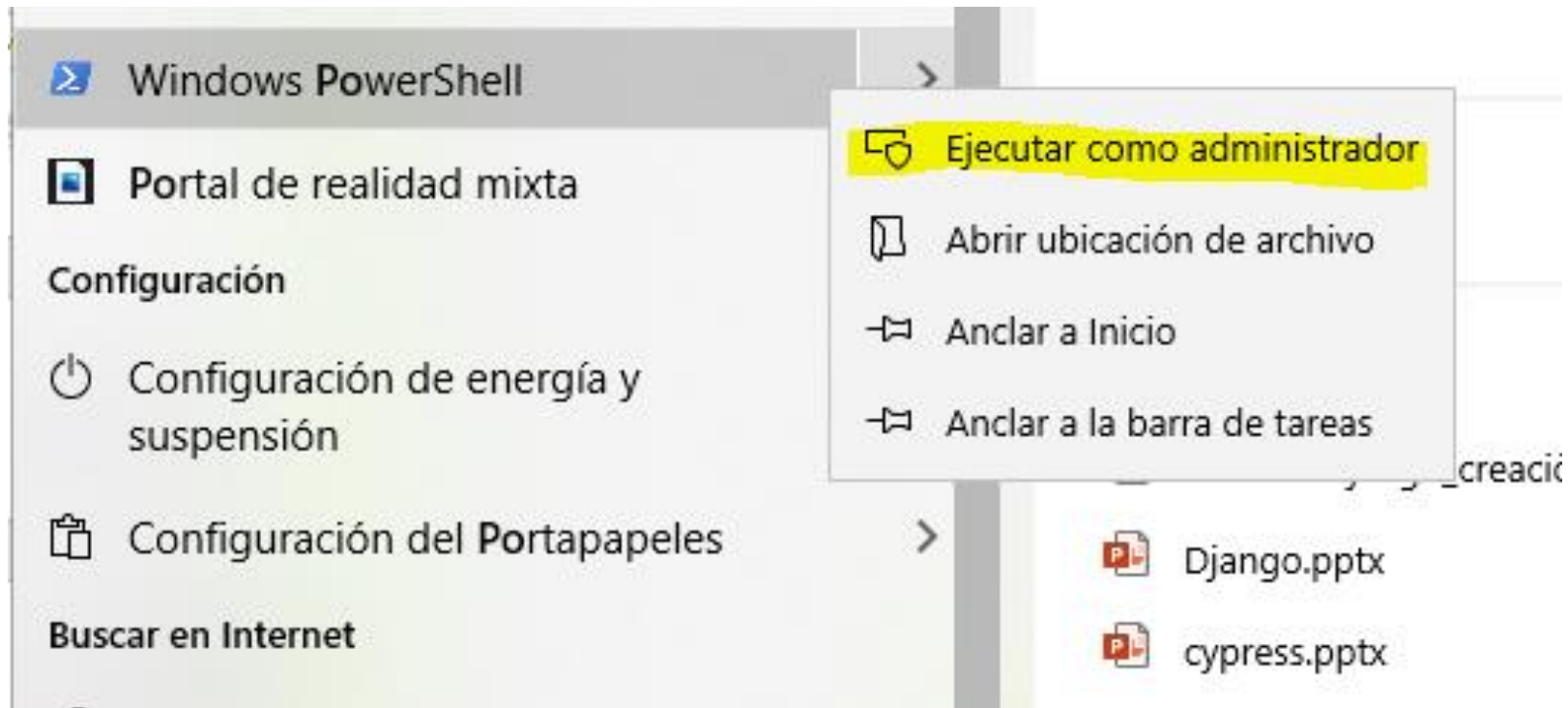


Click en nuevo terminal:



- Ingresar desde el terminal a la ruta de la carpeta creada:





Ejecutar el siguiente comando dentro del powershell como admin, para habilitar scripting

Set-ExecutionPolicy RemoteSigned

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

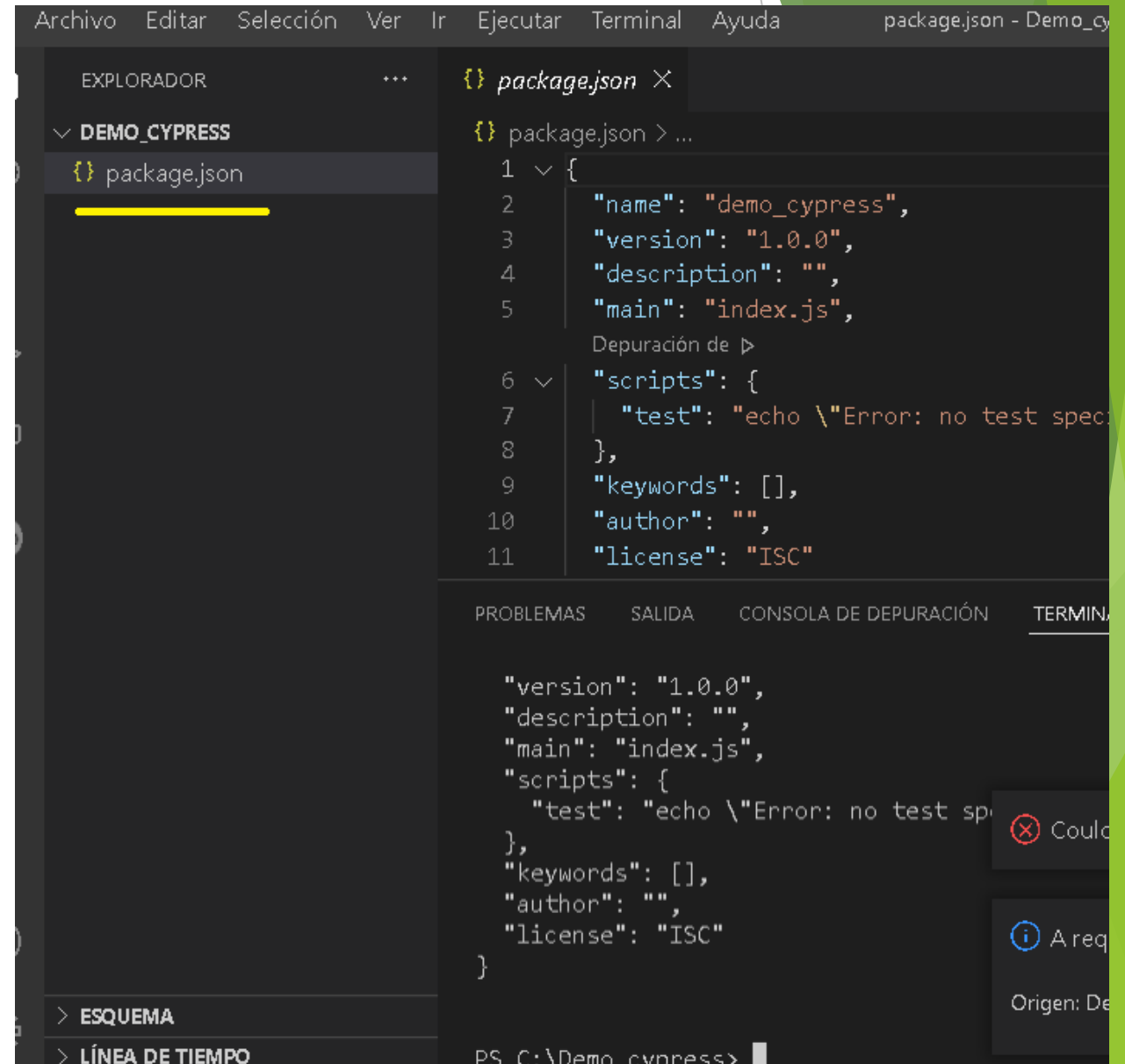
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\kalex> cd..
PS C:\Users> cd..
PS C:\> cd windows
PS C:\windows> cd system32
PS C:\windows\system32> Set-ExecutionPolicy RemoteSigned
```

Crear dependencias(librerías) de la solución:

- Digitar el siguiente comando desde el terminal para configurar el archivo de inicialización:

npm init



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project named 'DEMO_CYPRESS' with a file named 'package.json' highlighted. The main editor displays the content of 'package.json' with line numbers 1 through 11. The file contains the following JSON:

```
1 {  
2   "name": "demo_cypress",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \\\"Error: no test specified\\\"",  
8   },  
9   "keywords": [],  
10  "author": "",  
11  "license": "ISC"  
}
```

Below the editor, the 'TERMINAL' pane shows the output of the 'npm init' command, which is the same JSON structure as the one in the editor. The terminal prompt is 'PS C:\Demo_cypress>'. On the right side of the terminal, there are two notifications: a red 'X' icon with the text 'Could not find a package.json file' and a blue 'i' icon with the text 'A required package.json file was not found'.

npm init

Crear dependencias(librerías) de la solución:

Digitar la tecla enter cuando pregunte el nombre

Version (confirme digitando enter)

Descripción: proyecto de pruebas

Entry point (se deja vacío y se da enter)

Test command se deja vacío(enter)

Git repository se deja vacío(enter)

Keywords(se deja vacío)

Autor(nuestro nombre)

```
PROBLEMAS  SALIDA  TERMINAL  ...
node + - [] X
PS C:\cypress> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (cypress)
```

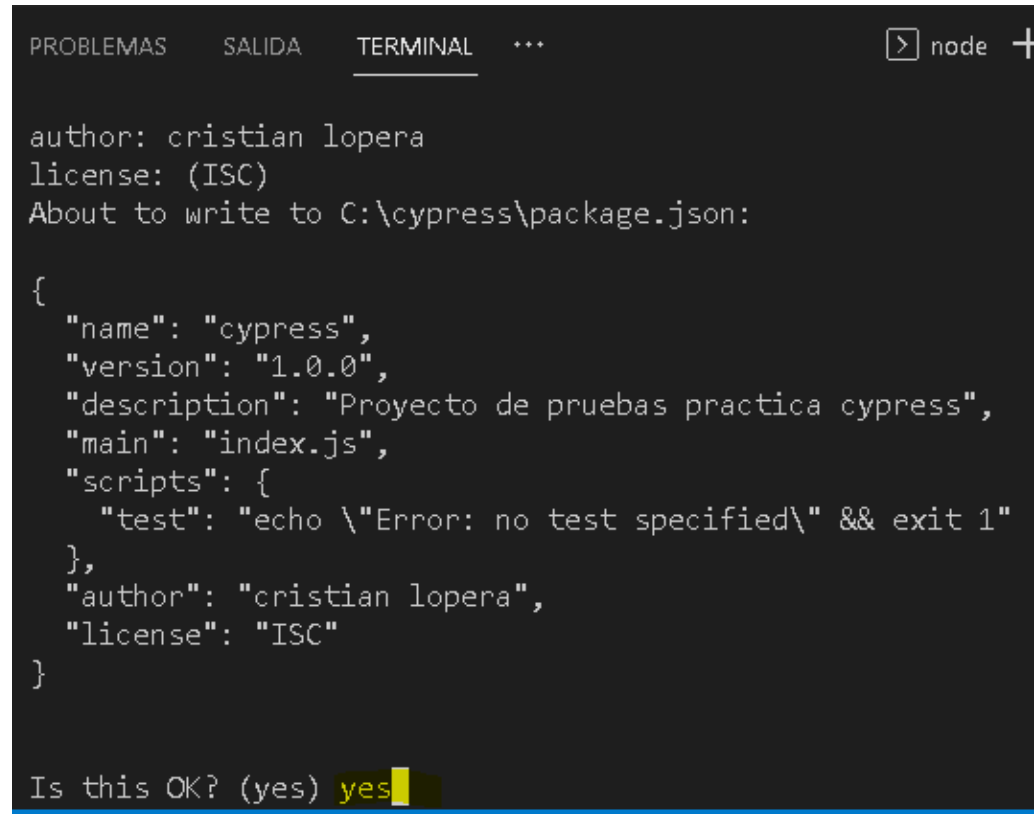
```
version: (1.0.0)
description: Proyecto de pruebas practica cypress
```

```
entry point: (index.js)
```

```
test command:
```

```
git repository:
keywords:
author: cristian lopera
```

Click en si:



```
PROBLEMAS  SALIDA  TERMINAL  ...  > node +
author: cristian lopera
license: (ISC)
About to write to C:\cypress\package.json:
{
  "name": "cypress",
  "version": "1.0.0",
  "description": "Proyecto de pruebas practica cypress",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "cristian lopera",
  "license": "ISC"
}

Is this OK? (yes) yes
```

Instalar cypress dentro del proyecto:

- Para Instalar Cypress dentro del proyecto ejecute el siguiente comando dentro del terminal:

Se descargan todas las librerías y casos de prueba disponibles

```
npm install cypress --save -dev
```

Validar que se cree automáticamente el archivo
package.json

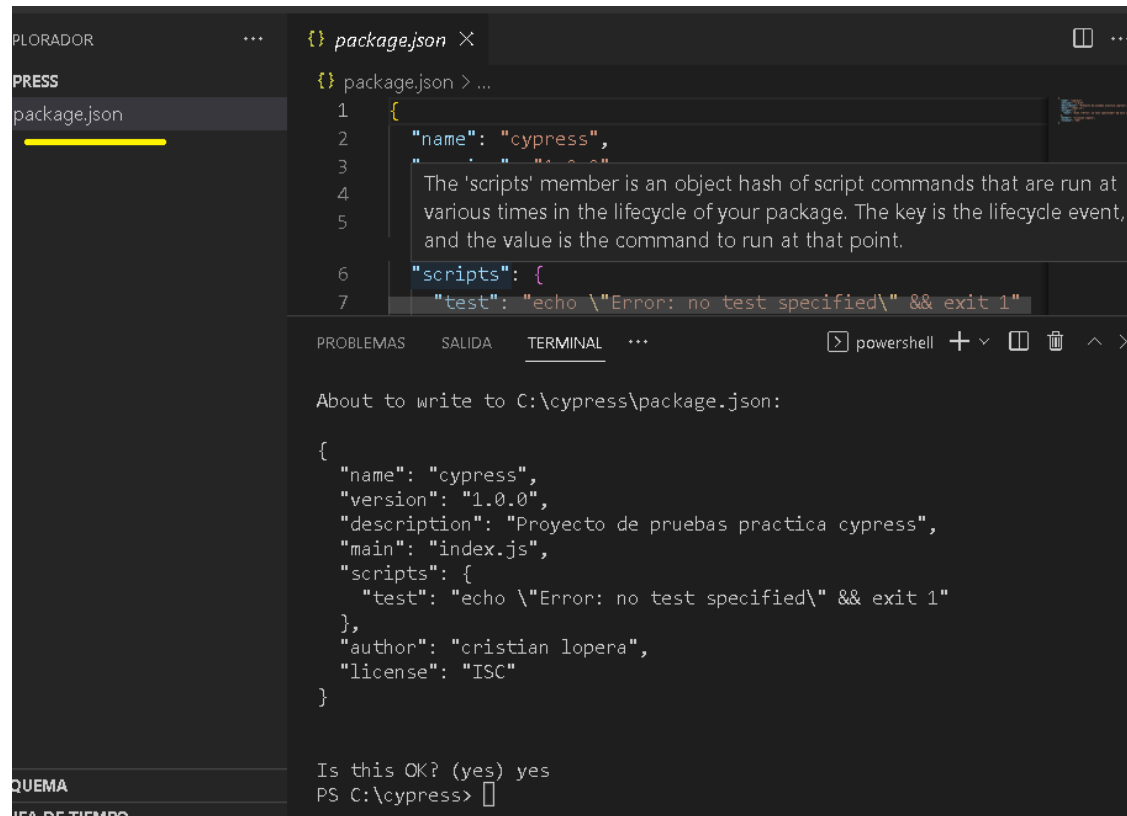
```
PS C:\cypress> npm install cypress --save -dev
npm WARN install Usage of the `--dev` option is deprecated. Use `--include=dev` instead.

added 165 packages, and audited 166 packages in 19s

27 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\cypress>
```

Validar archivo package.json



```
{  
  "name": "cypress",  
  "version": "1.0.0",  
  "description": "Proyecto de pruebas practica cypress",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "cristian lopera",  
  "license": "ISC"  
}
```

package.json > ...

1 {
2 "name": "cypress",
3 "version": "1.0.0",
4 "description": "Proyecto de pruebas practica cypress",
5 "main": "index.js",
6 "scripts": {
7 "test": "echo \\\"Error: no test specified\\\" && exit 1"

PROBLEMAS SALIDA TERMINAL ...

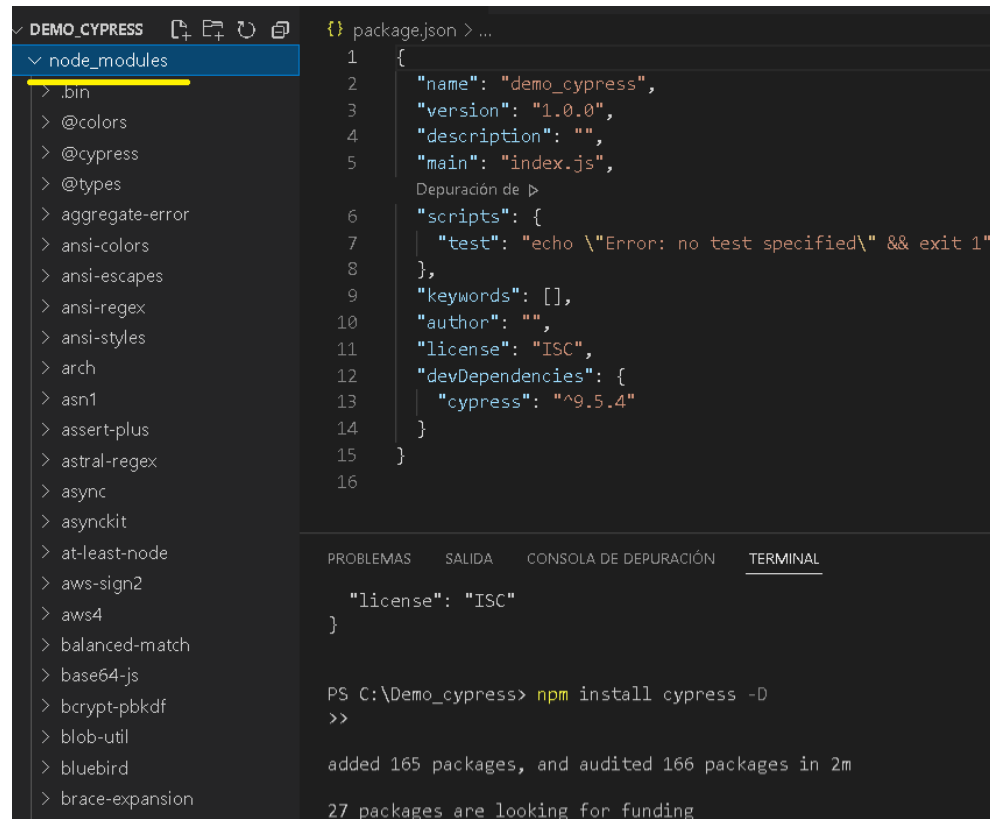
About to write to C:\cypress\package.json:

```
{  
  "name": "cypress",  
  "version": "1.0.0",  
  "description": "Proyecto de pruebas practica cypress",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "author": "cristian lopera",  
  "license": "ISC"  
}
```

Is this OK? (yes) yes
PS C:\cypress>

Validar que se encuentren creadas las dependencias **node_modules**

- Contiene todas las librerías



The screenshot shows a VS Code editor with a project named 'DEMO_CYPRESS'. The left sidebar displays the file explorer with the 'node_modules' directory expanded, showing a list of installed packages including .bin, @colors, @cypress, @types, aggregate-error, ansi-colors, ansi-escapes, ansi-regex, ansi-styles, arch, asn1, assert-plus, astral-regex, async, asynckit, at-least-node, aws-sign2, aws4, balanced-match, base64-js, bcrypt-pbkdf, blob-util, bluebird, and brace-expansion. The main editor area shows the 'package.json' file with the following content:

```
1 {
2   "name": "demo_cypress",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "cypress": "^9.5.4"
14  }
15 }
```

Below the code editor, the 'TERMINAL' tab is active, showing the output of the command 'npm install cypress -D':

```
PS C:\Demo_cypress> npm install cypress -D
>>
added 165 packages, and audited 166 packages in 2m
27 packages are looking for funding
```

Crear primer caso de prueba

Realizar un proyecto que creando casos de prueba los cuales se ejecutan en el navegador para validar la funcionalidad de la aplicación web.

Abrir cypress:

- En el terminal de visual studio code digitar el el siguiente comando para preparar el enviroment:

```
npx cypress open
```

```
PS C:\Demo_cypress> npx cypress open
It looks like this is your first time using Cypress: 9.5.4

✓ Verified Cypress! C:\Users\kalex\AppData\Local\Cypress\Cache\9.5.4\Cypress

Opening Cypress...

[3120:0419/163426.037:ERROR:gpu_init.cc(453)] Passthrough is not supported, GL is disabled, ANGLE is
```

Seleccionar E2E

Cypress

File Edit View Window Help Developer Tools



v10.8.0

Docs

Log In

Welcome to Cypress!

[Review the differences between each testing type →](#)



E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

Not Configured



Beta

Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

Not Configured

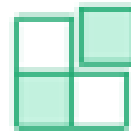
Configurar E2E



E2E Testing

Build and test the entire experience of your application from end-to-end to ensure each flow matches your expectations.

◉ Not Configured



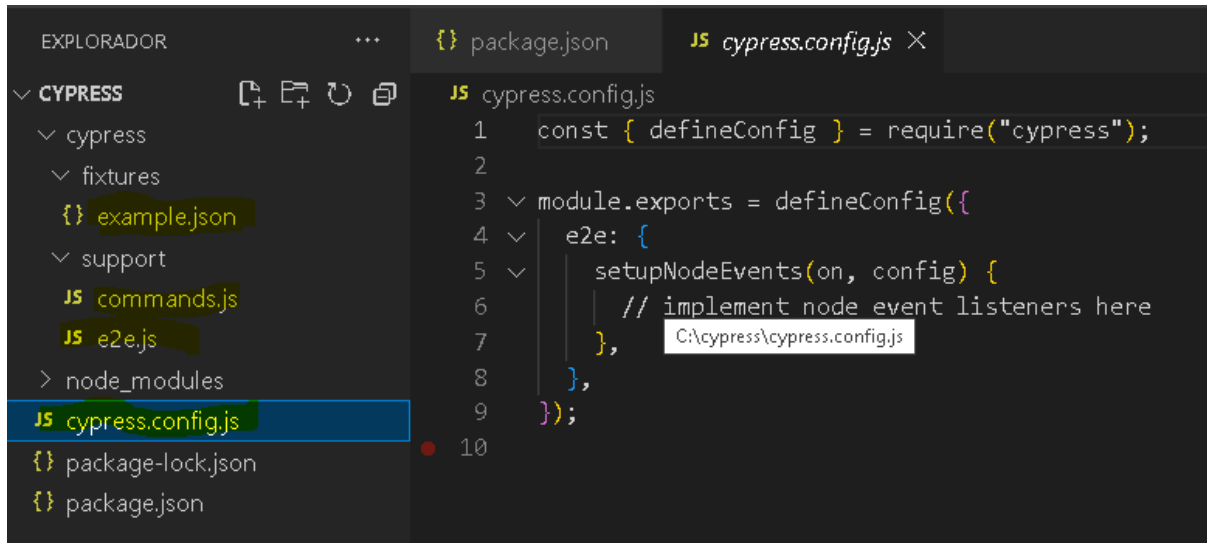
Component Testing

Build and test your components from your design system in isolation in order to ensure each state matches your expectations.

◉ Not Configured

Beta

Validar que quede instalados agregados al la solución:







The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure under the 'CYPRESS' folder: 'cypress' (containing 'fixtures' with 'example.json'), 'support' (containing 'commands.js' and 'e2e.js'), and 'node_modules'. The 'cypress.config.js' file is selected and highlighted in blue. The main editor shows the content of 'cypress.config.js', which includes a configuration object for Cypress, defining 'e2e' settings and a 'setupNodeEvents' function. The code is as follows:

```
1 const { defineConfig } = require("cypress");
2
3 module.exports = defineConfig({
4   e2e: {
5     setupNodeEvents(on, config) {
6       // implement node event listeners here
7     },
8   },
9 });
```

Configuration Files

We added the following files to your project:

-  [cypress.config.js](#)
The Cypress config file for E2E testing.
-  [cypress\support\e2e.js](#)
The support file that is bundled and loaded before each E2E spec.
-  [cypress\support\commands.js](#)
A support file that is useful for creating custom Cypress commands and overwriting existing ones.
-  [cypress\fixtures\example.json](#)
Added an example fixtures file/folder

[Continue](#)

Seleccionar navegador:

Choose a Browser

Choose your preferred browser for E2E testing.



Chrome

v105



Edge

v105



Electron


v102





Start E2E Testing in Chrome


← Switch testing type

Crear casos de prueba


 cypress


 Specs

 Runs

 Settings


Specs

 v10.8.0

 Chrome 105


Create your first spec

Since this project looks new, we recommend that you use the specs and tests that we've written for you to get started.



Scaffold example specs

We'll generate several example specs to help guide you on how to write tests in Cypress.











Create new empty spec

We'll generate an empty spec file which can be used to start testing your application.

Validar la creación:

Great! We added the following files to your project: — [Need help](#) ?

- +  cypress\e2e
- +  1-getting-started
- +  todo.cy.js
- +  2-advanced-examples
- +  actions.cy.js
- +  aliasing.cy.js
- +  assertions.cy.js
- +  connectors.cy.js

Okay, I got it!

- ▼ CYPRESS
 - ▼ cypress
 - ▼ e2e
 - > 1-getting-started
 - ▼ 2-advanced-examples
 - JS actions.cy.js
 - JS aliasing.cy.js
 - JS assertions.cy.js
 - JS connectors.cy.js
 - JS cookies.cy.js
 - JS cypress_api.cy.js
 - JS files.cy.js
 - JS local_storage.cy.js
 - JS location.cy.js
 - JS misc.cy.js
 - JS navigation.cy.js
 - JS network_requests.cy.js
 - JS querying.cy.js
 - JS spies_stubs_clocks.cy.js
 - JS traversal.cy.js
 - JS utilities.cy.js
 - JS viewport.cy.js

Ir a la interfaz y correr la prueba de ejemplo todo

The screenshot displays the Cypress web interface. On the left, a dark sidebar contains the following menu items: 'cypress' (with a terminal icon), 'Specs' (with a document icon), 'Runs' (with a list icon), and 'Settings' (with a gear icon). The main content area is titled 'Specs' and features a search bar labeled 'Search Specs'. Below the search bar, the section 'E2E specs' is shown with a 'Last updated ?' link. The file tree is expanded to show the following structure:

- ▼ cypress\e2e
 - ▼ 1-getting-started
 - todo.cy.js** (highlighted in yellow, with a tooltip showing 'todo.cy.js') 5 minutes ago
 - ▼ 2-advanced-examples
 - actions.cy.js 5 minutes ago
 - aliasing.cy.js 5 minutes ago
 - assertions.cy.js 5 minutes ago
 - connectors.cy.js 5 minutes ago
 - cookies.cy.js 5 minutes ago

```
describe('example to-do app', () => { Agrupar todas las pruebas
```

```
  beforeEach(() => {
```

```
    // Cypress starts out with a blank slate for each test
```

```
    // so we must tell it to visit our website with the `cy.visit()` command.
```

```
    // Since we want to visit the same URL at the start of all our tests,
```

```
    // we include it in our beforeEach function so that it runs before each test
```

```
    cy.visit('https://example.cypress.io/todo')
```

```
  })
```

las pruebas se implementan con el it y cada uno de los it es una prueba independiente

```
  it('displays two todo items by default', () => {
```

```
    // We use the `cy.get()` command to get all elements that match the selector.
```

```
    // Then, we use `should` to assert that there are two matched items,
```

```
    // which are the two default items.
```

```
    cy.get('.todo-list li').should('have.length', 2)
```

```
    // We can go even further and check that the default todos each contain
```

```
    // the correct text. We use the `first` and `last` functions
```

```
    // to get just the first and last matched elements individually,
```

Validar caso de prueba actions ejecutar caso de prueba llamado actions

```
actions.cy.js 00:17

Actions
✓ .type() - type into a DOM element
  BEFORE EACH
  1 visit https://example.cypress.io/commands/actions

  TEST BODY
  1 get .action-email
  2 - type fake@email.com
  3 - assert expected
    <input#email1.form-control.action-email> to have
    value fake@email.com
  4 - type {leftarrow}{rightarrow}{uparrow}{downarrow}
  5 - type {del}{selectall}{backspace}
  6 - type {alt}{option}
```

Actions

Examples of actions being performed on DOM elements in Cypress, for a full reference of commands, go to docs.cypress.io

.type()

To type into a DOM element, use the `.type()` command.

```
cy.get('.action-email')
  .type('fake@email.com').should('have.value', 'fake@email.com')

// .type() with special character sequences
.type('{leftarrow}{rightarrow}{uparrow}{downarrow}')
.type('{del}{selectall}{backspace}')

// .type() with key modifiers
.type('{alt}{option}') //these are equivalent
.type('{ctrl}{control}') //these are equivalent
.type('{meta}{command}{cmd}') //these are equivalent
.type('{shift}')
```

// Delay each keypress by 0.1 sec
.type('slow.typing@email.com', { delay: 100 })

Email address

fake@email.com

Disabled Text area

Pinned Highlights

Crear primer caso de prueba:

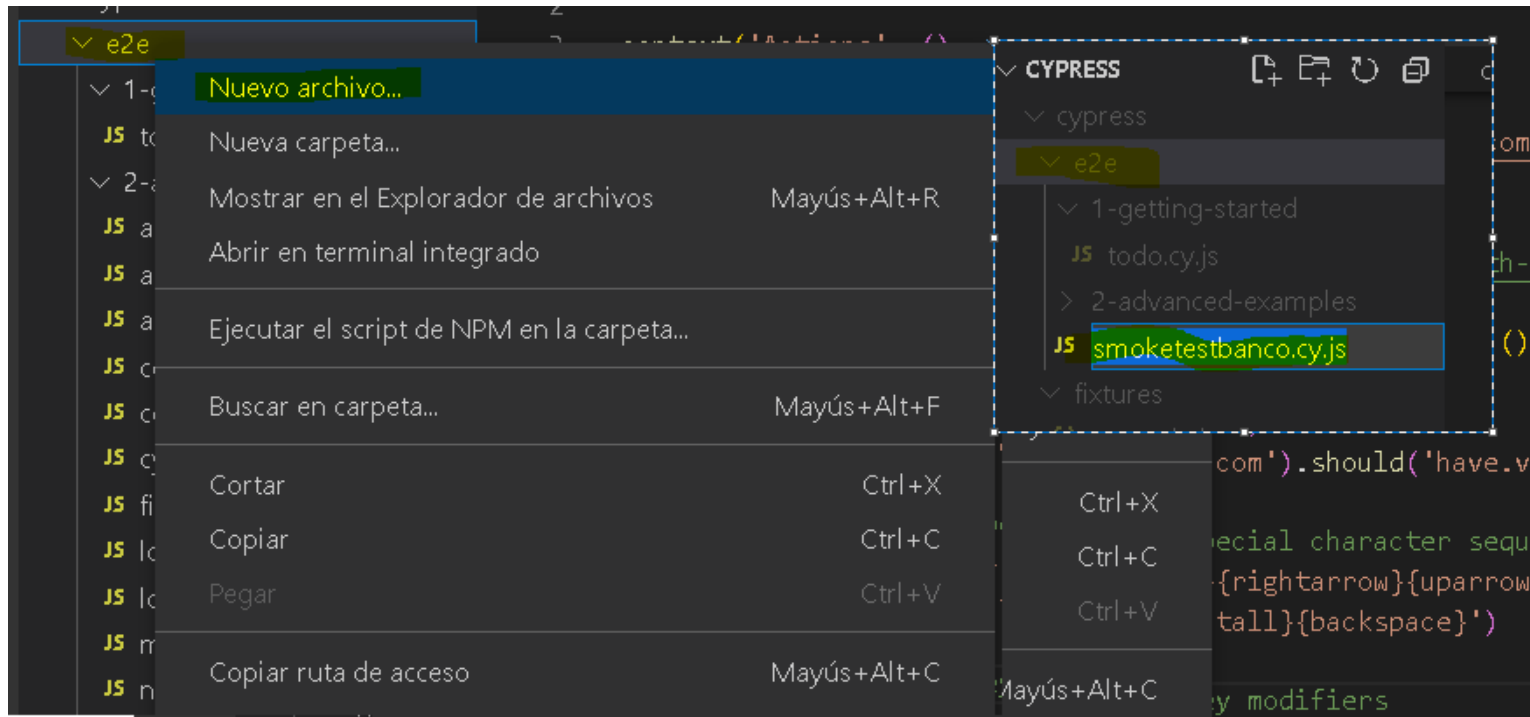
- Ingresar a la siguiente pagina: <http://zero.webappsecurity.com/>

Caso de prueba:

- Validar que cargue la pagina, el banner superior y el texto online banking

Pasos:

- Ir a visual studio crear en la carpeta end2end crear una prueba llamada smoketestbanco.cy.js



```
// Definir el nombre de la suite de pruebas
describe ('test suite- pruebas pagina de inicio', function(){
  //crear prueba
  it ('validación encabezado superior', function(){
    // Abrir sitio web
    cy.visit('http://zero.webappsecurity.com')
    // comandos en Cypress
    // Todos se inicializan con cy.get()
    // cy.get ('dentro del paréntesis va el selector entre comillas').('la
    acción que se desea hacer sobre el elemento')
    // validar que la imagen exista
    cy.get('.active > img').should('be.visible')

    // validar que el texto se encuentre en el encabezado
    cy.get('.active > .custom > h4').contains('Online Banking')
  })
})
```

Ctrl+S (guardar los cambios y ejecutar)

JS assertions.spec.js

JS Smoketest.spec.js •

cypress > integration > JS Smoketest.spec.js > ...

```
1 // atocompletar todos los códigos que usemos en con cypress con IntelliSense
2 /// <reference types= "Cypress" />
3 // definir los nombres de la prueba
4 describe(['pruebas pagina de incicio', function(){
5   it('validación encabezado superior', function(){
6     // Abrir sitio web
7     cy.visit('http://zero.webappsecurity.com')
8     // comandos en Cypress
9     // Todos se inicializan con cy.get()
10    // cy.get ('dentro del parentesis va el selector entre comillas').('la acción que se desea hacer sobre el elemento')
11    // validar que la imagen exista
12    cy.get('.active > img').should('be.visible')
13
14    // validar que el texto se encuentre en el encabezado
15    cy.get('.active > .custom > h4').contains('Online Banking')
16  })
17 }])
```

Validar ejecución:

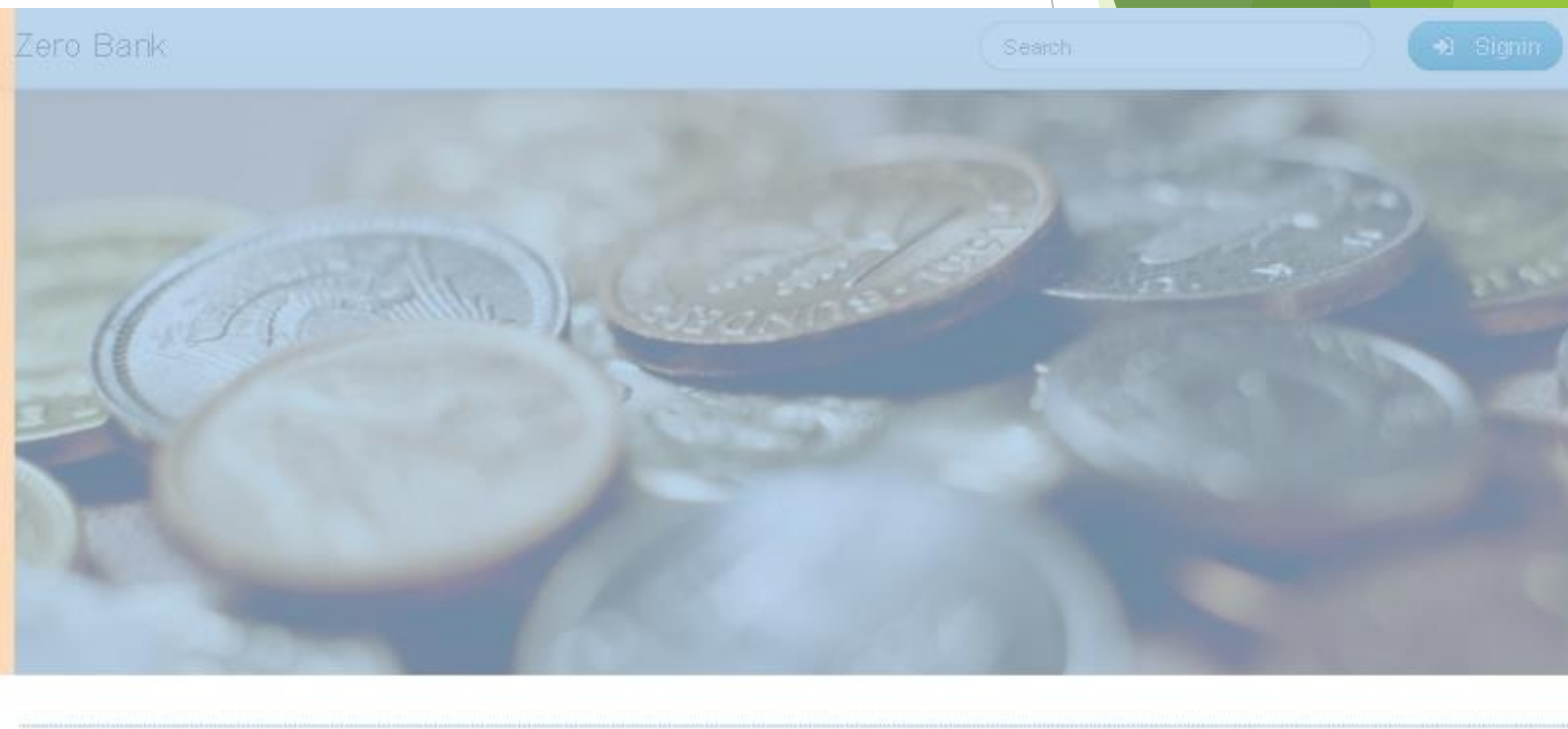
cypress/integration/Smoketest.spec.js

▼ pruebas pagina de inicio

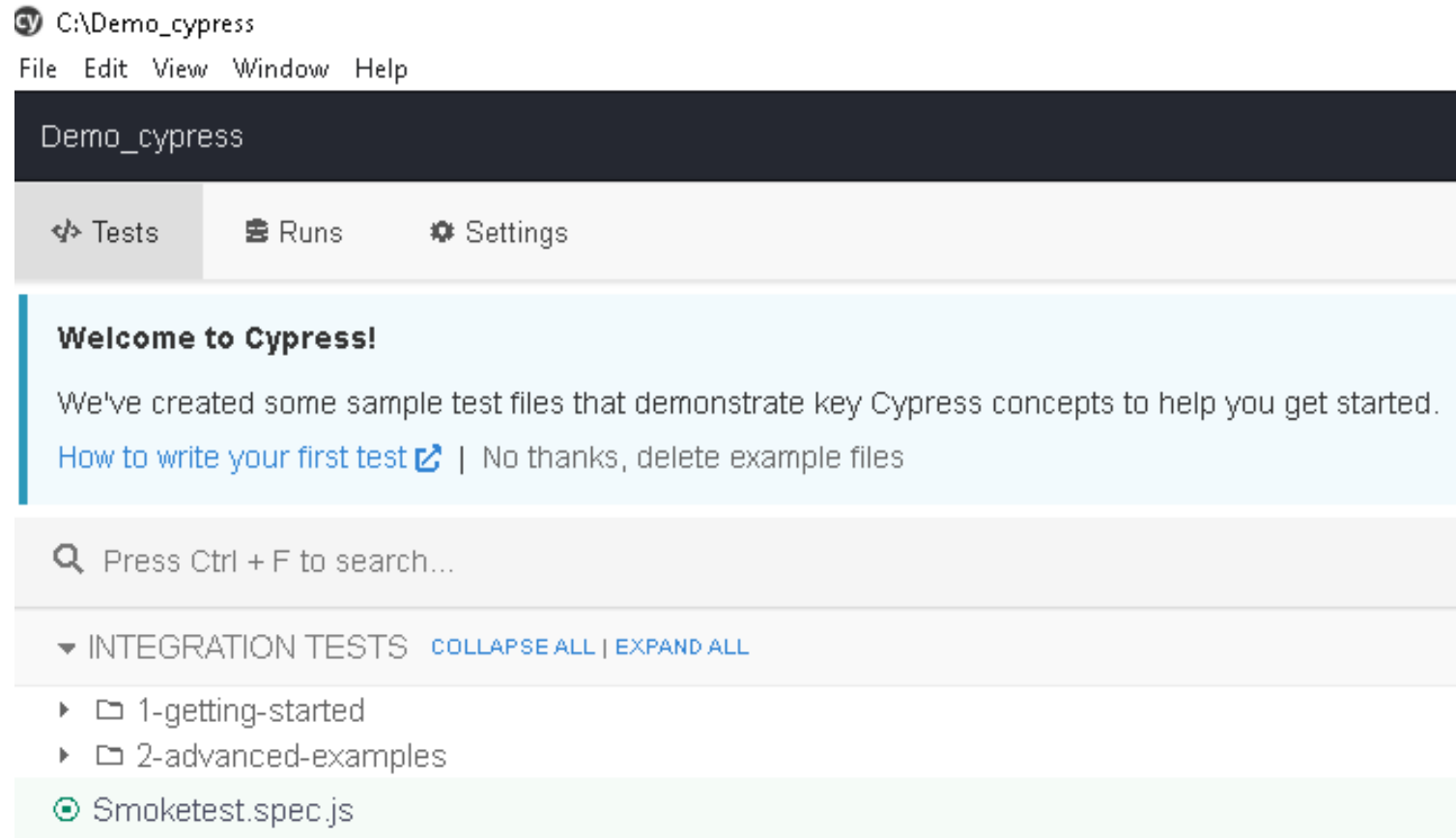
✓ validación encabezado superior

▼ TEST BODY

1	visit	http://zero.webappsecurity.com
2	get	.active > img
3	- assert	expected to be visible
4	get	.active > .custom > h4
5	- contains	Online Banking



Cada que se guarda cypress ejecuta la prueba



Realizar ejercicio prueba transferencia de fondos:

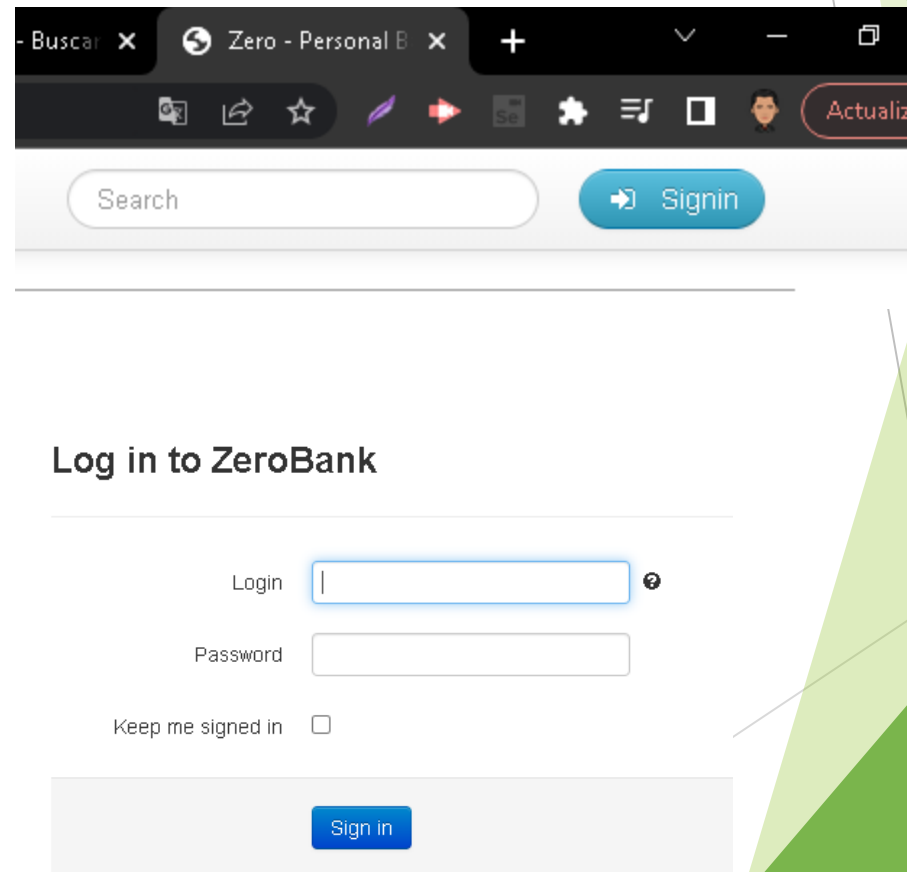
- Ingresar a la siguiente pagina de pruebas:

<http://zero.webappsecurity.com>

Loguearse:

username

password



The screenshot shows a web browser window with the title 'Zero - Personal B'. The address bar shows the URL 'http://zero.webappsecurity.com'. The page has a search bar and a 'Signin' button. Below the search bar, the text 'Log in to ZeroBank' is displayed. There are two input fields: 'Login' and 'Password'. Below the 'Password' field, there is a checkbox labeled 'Keep me signed in'. At the bottom, there is a 'Sign in' button.

Log in to ZeroBank

Login

Password

Keep me signed in ☐

Sign in

Ejemplo prueba transferencia.

- Ir al menú transferencia de fondos:

Zero Bank

Search

Settings

username


HOME

ONLINE BANKING

FEEDBACK

Online Banking

Welcome to Zero Online Banking. Zero provides a greener and more convenient way to manage your money. Zero enables you to check your account balances, pay your bills, transfer money, and keep detailed records of your transactions, wherever there is an internet connection.



Online Banking

Click the button below to view online banking features.

[More Services](#)

Checking Account Activity

Use Zero to view the most up-to-date listings of your deposits, withdrawals, interest payments, and a number of other useful transactions.

Transfer Funds

Use Zero to safely and securely transfer funds between accounts. There is no hold placed on online money transfers, so your funds are available when you need them.

My Money Map

Use Zero to set up and monitor your personalized money map. A money map is an easy-to-use online tool that helps you manage your finances efficiently. With Money Map, you can create a budget, sort your finances into spending and savings categories, check the interest your accounts are earning, and gain new

Seleccionar cuenta origen y de destino y dar click en enviar:

Zero Bank [Settings](#)

[Account Summary](#) [Account Activity](#) [Transfer Funds](#) [Pay Bills](#) [My Money Map](#) [Online Statements](#)

Transfer Money & Make Payments

From Account	<input type="text" value="Savings(Avail. balance = \$ 1548)"/>
To Account	<input type="text" value="Credit Card(Avail. balance = \$ -265)"/>
Amount	\$ <input type="text" value="200"/>
Description	<input type="text" value="prueba de transferencia"/>

Descriptions appear for checking, savings, money market or market rate accounts only.

[Continue](#)

Dar click en enviar:

Zero Bank

Search

Settings

[Account Summary](#)

[Account Activity](#)

[Transfer Funds](#)

[Pay Bills](#)

[My Money Map](#)

[Online Statements](#)

Transfer Money & Make Payments - Verify

Please verify that the following transaction is correct by selecting the **Submit** button below.

From Account

Savings

To Account

Credit Card

Amount

\$ 200

Description

prueba de transferencia

Cancel

Submit

Validar mensaje de confirmación:

Transfer Money & Make Payments - Confirm

You successfully submitted your transaction.

From Account	Savings
To Account	Credit Card
Amount	\$ 200

[View transfers or make another transfer](#)

Automatizar prueba en cypress

- ▶ Obtener el código del botón Signin
- ▶ `cy.get('#signin_button')`

The image shows a Cypress test runner interface on the left and a browser view on the right. The browser view shows the Zero Bank website with the 'Signin' button highlighted by a yellow box. The Cypress test runner shows a successful test run for 'Prueba E2E Transferencia de fondos' with a duration of 03.57. The test body shows the following steps:

```
1 visit http://zero.webappsecurity.com/
  (page load) --page loaded--
  (new url) http://zero.webappsecurity.com/Login.html
  (page load) --page loaded--
  (new url) http://zero.webappsecurity.com/
```

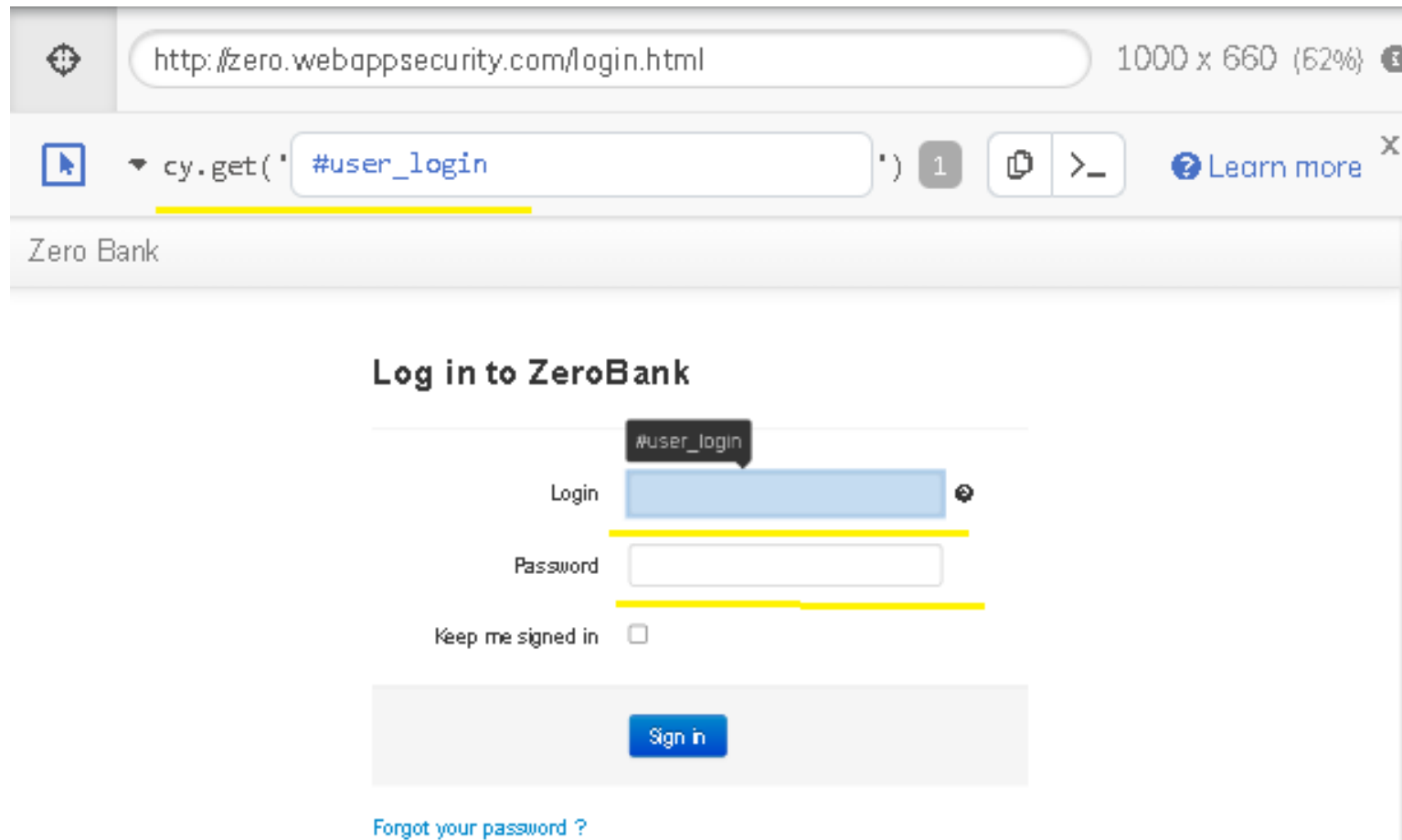
The browser view shows the Zero Bank website with the 'Signin' button highlighted by a yellow box. The Cypress command log shows the command `cy.get('#signin_button')` with a count of 1. The browser address bar shows the URL `http://zero.webappsecurity.com/`.

Realizar acción evento click sobre el botón loguearse:

- Se obtiene el objeto con el identificador del botón y le adicionamos el evento click

```
cypress > integration > JS transferencia.spec.js > describe('Prueba E2E Transferencia de fondos ') callback > it('Caso de prueba 1 transferencia de fondos de ahorros a tarjeta de credito ', function(){
1  // Prueba E2E Transferencia de fondos
2  describe ('Prueba E2E Transferencia de fondos ', function(){
3
4
5
6
7
8  it ('Caso de prueba 1 transferencia de fondos de ahorros a tarjeta de credito ', function(){
9
10 //primer paso ir a la página
11 cy.visit('http://zero.webappsecurity.com/')
12
13 //1.obtener el objeto en este caso el botón y asigne el evento click
14 cy.get('#signin_button').click()
15
```

Obtener el selector de los botones:



Agregar código para obtener los botones de ingresar el login

```
//2.ingresar login y password  
cy.get('#user_login').type('username')  
cy.get('#user_password').type('password')
```

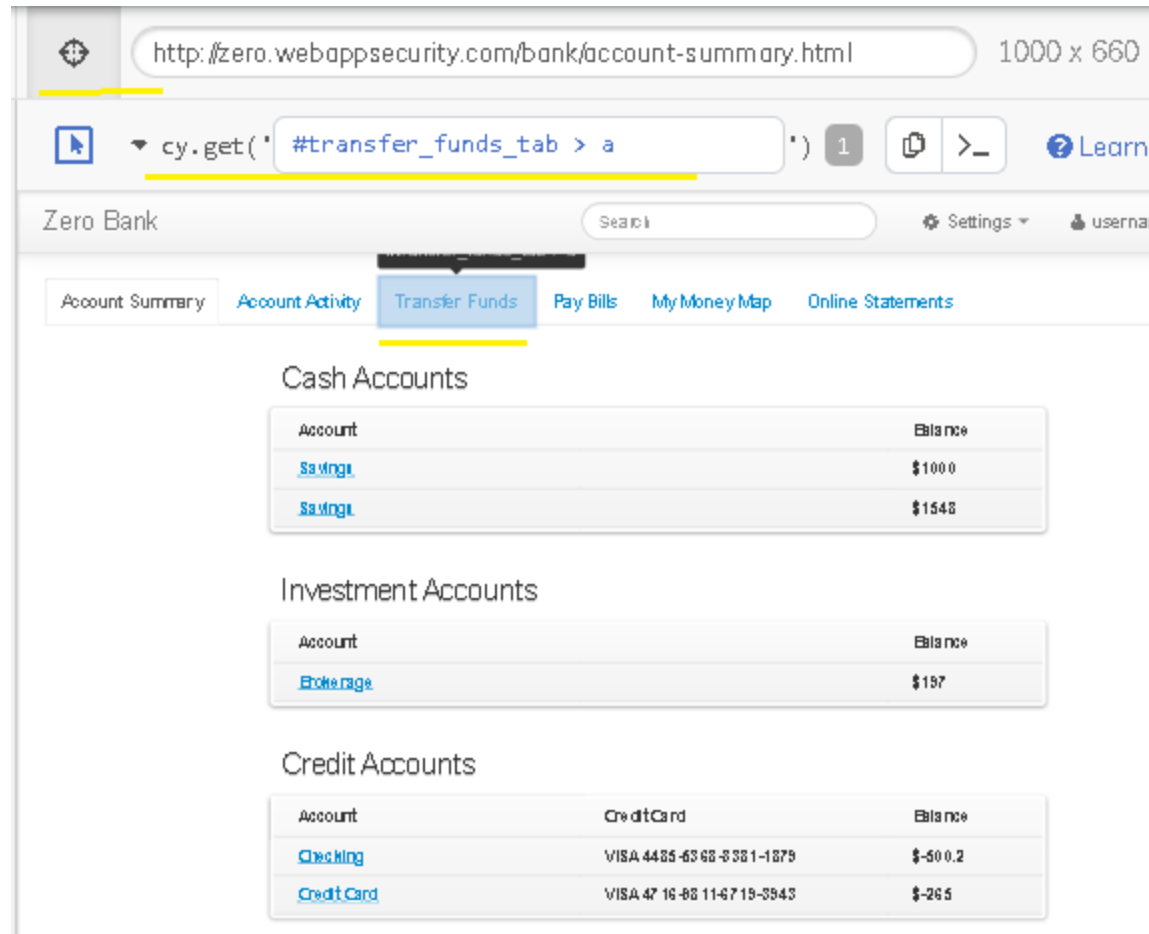
```
//2.ingresar login y password  
cy.get('#user_login').type('username')  
cy.get('#user_password').type('password')
```

```
//3.obtener el selector del botón y  
asignarle el evento click  
cy.get('.btn').click
```

```
//3.obtener boton y asignarle el evento click  
cy.get('.btn').click  
  
})
```

Click en link transferir fondos:

- Click al link transferencia de fondos y obtener el link y pegarlo en el visual studio code.



```
// ingresar al link transferencia de fondos  
cy.get('#transfer_funds_tab > a').click()  
})
```

Transfer Money & Make Payments

From Account Savings(Avail. balance = \$ 1548) ▼

To Account Credit Card(Avail. balance = \$ -265) ▼

Amount \$ 500

Description Prueba de transferencia de ahorros a t.cre

Descriptions appear for checking, savings, money market or market rate accounts only.

e destino de la

```
//5.Selecionar la cuenta de origen y destino:
```

```
//ingresamos el id del selector cuenta de origen from account y la cuenta en este caso el elemento numero 3  
cy.get('#tf_fromAccountId').select(3)
```

```
//ingresamos el id para la cuenta destino con el indice en la pocición 5  
cy.get('#tf_toAccountId').select(5)
```

```
//ingresar un valor en el cuadro de texto  
cy.get('#tf_amount').type('500')
```

```
//ingresar la descripción  
cy.get('#tf_description').type('Prueba de transferencia de ahorros a t.credito')
```

```
//click al boton continuar  
cy.get('#btn_submit').click()
```

Confirmar transferencia

Transfer Money & Make Payments - Verify

Please verify that the following transaction is correct by selecting the Submit button below.

From Account

Loan

To Account

Brokerage

Amount

\$ 500

Description

Prueba de transferencia de ahorros a t.credit

#btn_submit

Cancel

Submit

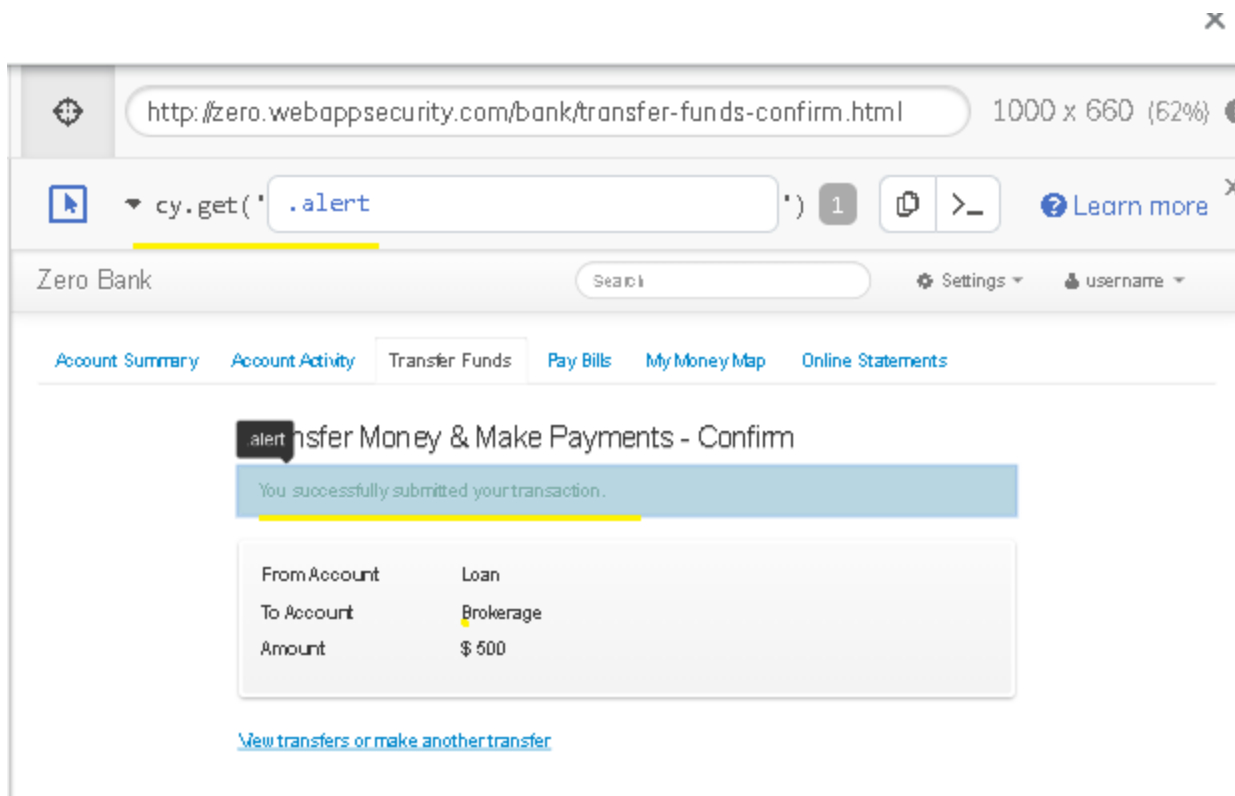
```
//Confirmar transferencia  
cy.get('#btn_submit').click()
```

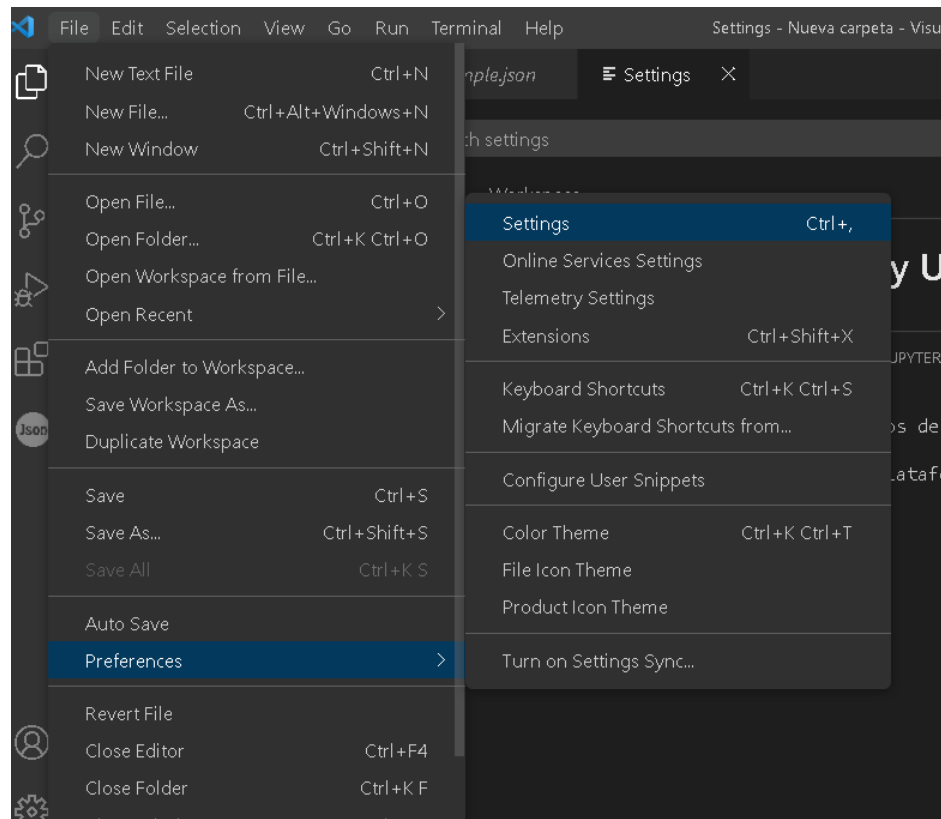
Validar mensaje exitoso

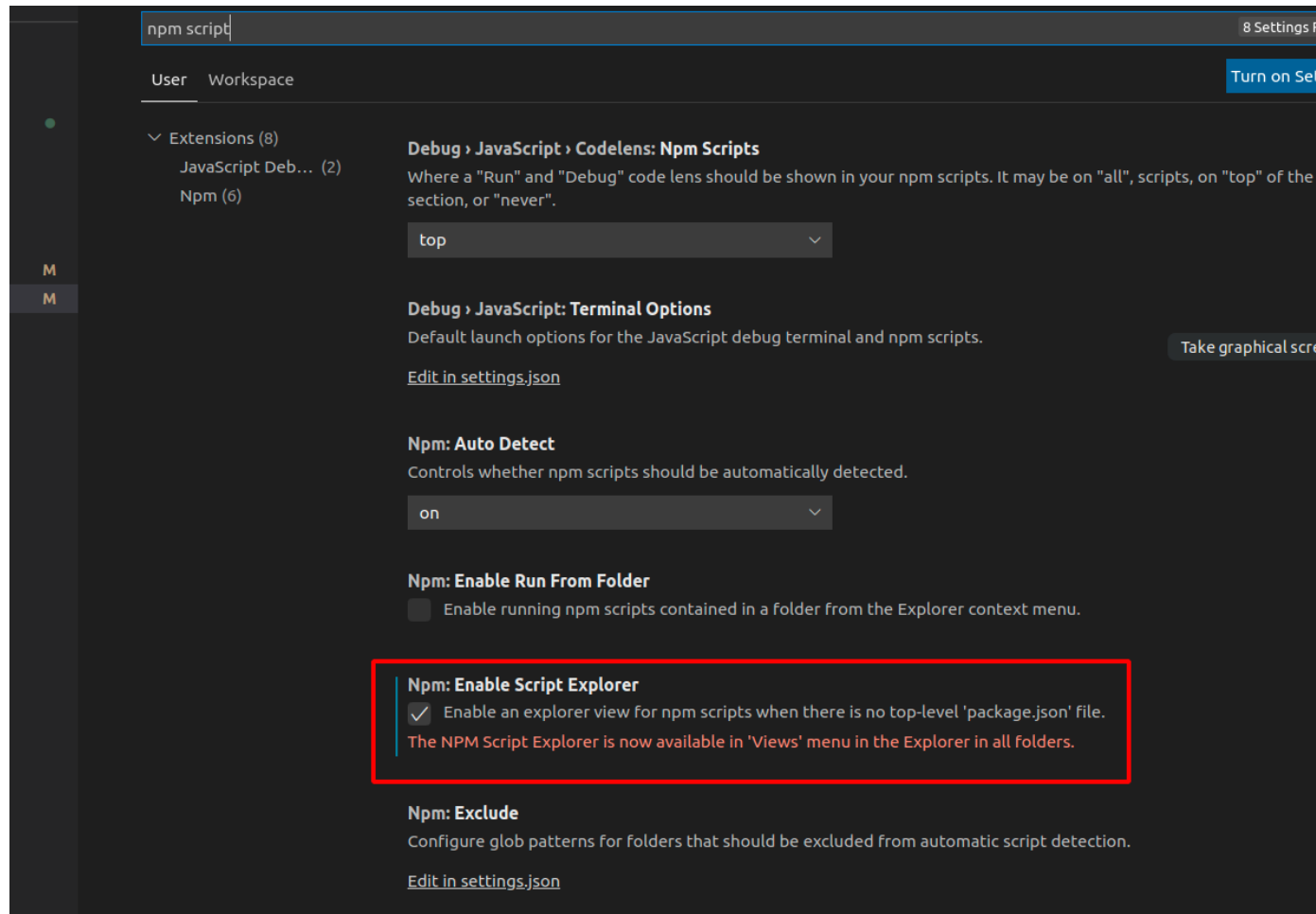
```
// validar mensaje
```

```
// validar mensaje
```

```
cy.get('.alert').should('contain.text', 'You successfully submitted your transaction.')
```









Register

First Name

Last Name

Username

Password

[Register](#) [Cancel](#)

Angular 6 User Registration and Login Example & Tutorial
JasonWatmore.com



Login

Username

Password

[Login](#) [Register](#)

Angular 6 User Registration and Login Example & Tutorial
JasonWatmore.com



Hi n!

You're logged in with Angular 6!!

All registered users:

- rr (n rr) - [Delete](#)

[Logout](#)

Angular 6 User Registration and Login Example & Tutorial
JasonWatmore.com

Selenium review

- ▶ How to locate webElements using CSS selectors
- ▶ how to locate Submit button id element using CSS Selector in Selenium
- ▶ Python Webdriver Manager-How to import the web driver for python.
- ▶ Asignación de variables
- ▶ Action commands