



Simulación completa de
la experiencia de usuario



Manuel López Camarena
Abril, 2021
AT Sistemas S.A.

Interfaz

Estado de los tests



The screenshot shows a web browser window titled 'blogs_application-actions' with the address 'http://localhost:8888/#/'. The left sidebar displays test results for 'TodoMVC'.

Test Results:

- Tests: 1 passed (green checkmark), 0 failed (red X), 0 pending (grey circle). Duration: 0.69s.
- TodoMVC
 - Adds items (spy example)
 - calls inform (passed)
 - SPIES / STUBS (1)

| Type | Function | Alias(es) | # Calls |
|-------|----------|-----------|---------|
| spy-1 | inform | inform | 1 |
 - BEFORE EACH
 - 1 VISIT /
 - TEST
 - 1 WINDOW
 - 2 - ITS .model
 - 3 WINDOW
 - 4 - ITS .model
 - 5 - INVOKE .addTodo(buy some cheese, feed the... (SPY-1) inform() (inform)
 - 6 GET .todo-list li (todos 3)
 - 7 GET @inform
 - 8 - ASSERT expected inform to have been called exactly once
 - (NEW URL) http://localhost:8888/#/

The right pane shows the 'todos' application with the heading 'What needs to be done?'. It lists three items: 'buy some cheese', 'feed the cat', and 'book a doctors appointment'. At the bottom, it shows '3 items left' and filters for 'All', 'Active', and 'Completed'.

Cronología



Double-click to edit a todo
Created by petehunt
TodoMVC



Vista previa app

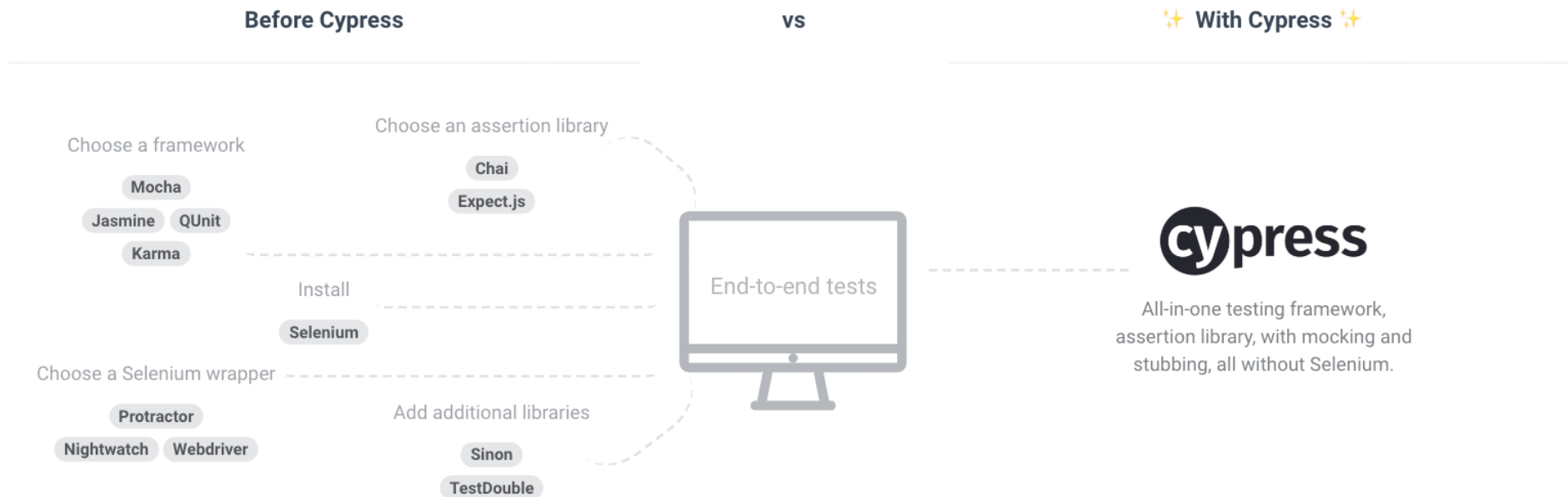
Navegación por snapshot en la cronología de ejecución

The screenshot displays a web browser window at `http://localhost:5000/`. On the right, a form is visible with fields for 'Name' (Molly), 'Email' (molly@dev.dev), and 'Your message' (Mind you if I ask some silly question?). A 'SEND' button is at the bottom. On the left, a 'Tests' panel shows a 'Form test' with a green checkmark and a duration of 1.43. Below the test name, a list of steps is shown:

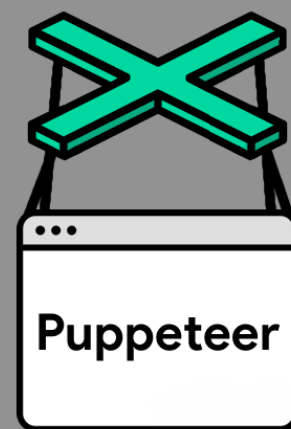
| Step | Method | URL/Action |
|------|----------|--|
| 1 | VISIT | / |
| 2 | GET | form |
| 3 | GET | input[name="name"] |
| 4 | - TYPE | Molly |
| 5 | - ASSERT | expected <input#name> to have value Molly |
| 6 | GET | input[name="email"] |
| 7 | - TYPE | molly@dev.dev |
| 8 | - ASSERT | expected <input#email> to have value molly@dev.dev |
| 9 | GET | textarea |
| 10 | - TYPE | Mind you if I ask some silly question? |
| 11 | - ASSERT | expected <textarea#message> to have value Mind you if I ask some silly question? |

A hand cursor icon points to step 8, with the text 'Click!' in red. A large black arrow points from the 'Click!' text to the 'SEND' button in the form.

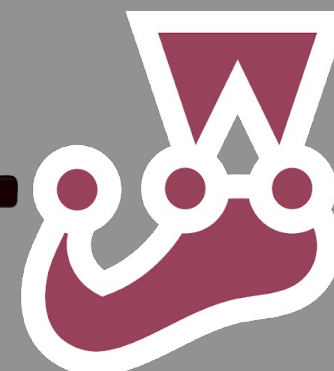
Cypress, todo en uno.



End-to-end testing. Cypress como alternativa a otras librerías.



+



Integración con Testing Library.



js? react? 極?

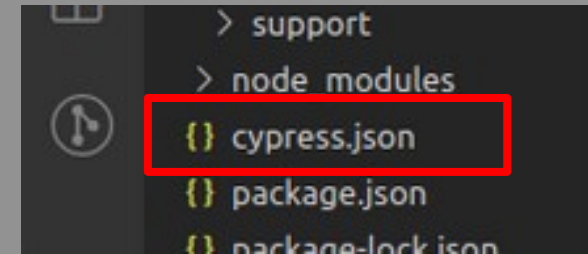
```
cy.findByRole('button', { name: /Jackie Chan/i }).click()
cy.findByRole('button', { name: /Button Text/i }).should('exist')
cy.findByRole('button', { name: /Non-existing Button Text/i }).should(
  'not.exist'
)
cy.findByLabelText(/Label text/i, { timeout: 7000 }).should('exist')

// findAllByText _inside_ a form element
cy.get('form')
  .findByText('button', { name: /Button Text/i })
  .should('exist')
cy.findByRole('dialog').within(() => {
  cy.findByRole('button', { name: /confirm/i })
})
```

Experiencia de usuario real y completa,
se puede testear casi sin saber Javascript,
el código es similar a escribir en inglés, con cierta nomenclatura de programación

Estructura y variables de entorno.

| | |
|---|---|
| | |
| fixtures example.json | Carga de datos fijos/mock |
| integration examples actions.spec.js aliasing.spec.js ... 17 more files ... | Tests |
| plugins index.js | Manipulación del comportamiento interno de Cypress |
| support commands.js index.js | Creación de comandos propios o integración de comandos de otras librerías (Testing library) |

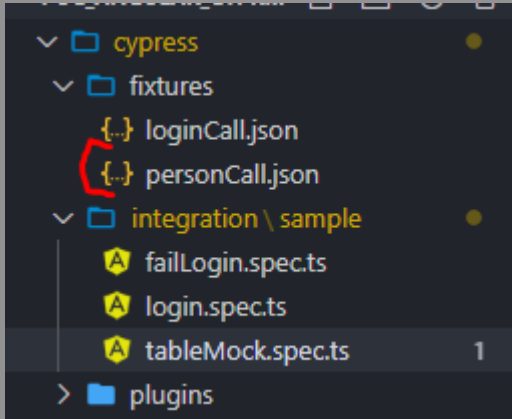


```
{  
  "host": "veronica.dev.local",  
  "api_server": "http://localhost:8888/api/v1/"  
}
```



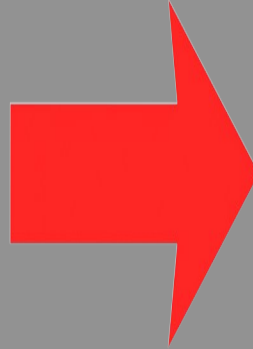
```
Cypress.env() // {host: 'veronica.dev.local', api_server: 'http://localhost:  
Cypress.env('host') // 'veronica.dev.local'  
Cypress.env('api_server') // 'http://localhost:8888/api/v1/'
```

Fixtures



```
[
  {
    "name": "Juan",
    "age": 30
  },
  {
    "name": "Vicente",
    "age": 53
  }
]
```

```
cy.intercept(`${Cypress.env('host')}${Cypress.env('data')}`, {fixture: 'personCall.json'}).as('peopleCall');
```



| Name | × Headers | Preview | Response | Initiator | Timing |
|----------------------|-----------|---------|---|-----------|--------|
| main.js | | | ▼ [{name: "Juan", age: 30}, {name: "Vicente", age: 53}] | | |
| info?t=1617891720909 | | | ▶ 0: {name: "Juan", age: 30} | | |
| websocket | | | ▶ 1: {name: "Vicente", age: 53} | | |
| login | | | | | |
| login | | | | | |
| data | | | | | |
| login | | | | | |
| bootstrap.min.css | | | | | |
| all.css | | | | | |
| animate.min.css | | | | | |

Documentación

Cypress:

<https://docs.cypress.io/guides/overview/why-cypress>

<https://github.com/cypress-io/cypress>

Testing library:

<https://testing-library.com/docs/>

<https://github.com/testing-library/cypress-testing-library>