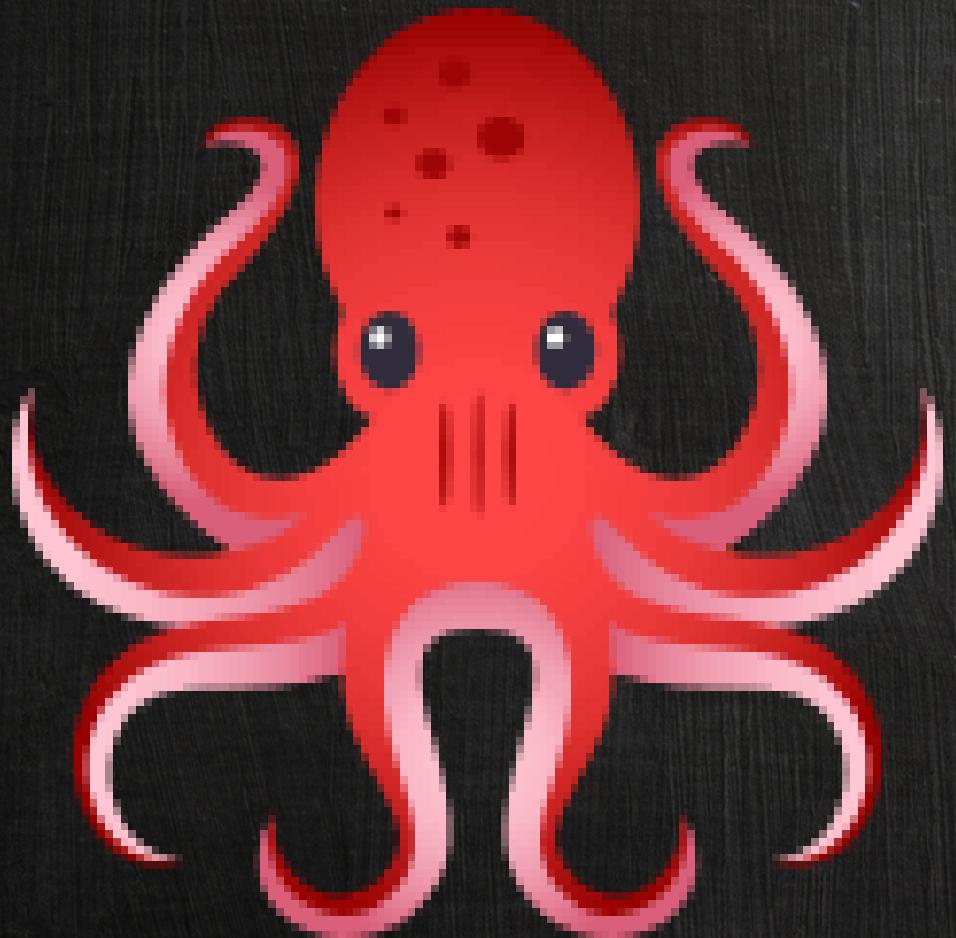


PROJET 10 - 724

DÉBUGGEZ LE SITE D'UNE
AGENCE D'ÉVÉNEMENTIEL



ALBERTO RAMOS
MUNOZ

INTÉGRATEUR WEB





METTEZ EN PLACE VOTRE
ENVIRONNEMENT DE
DÉVELOPPEMENT

INSTALLEZ REACT DEVELOPER
TOOLS

RÉPAREZ LES BUGS

RÉDIGEZ LE CAHIER DE RECETTE

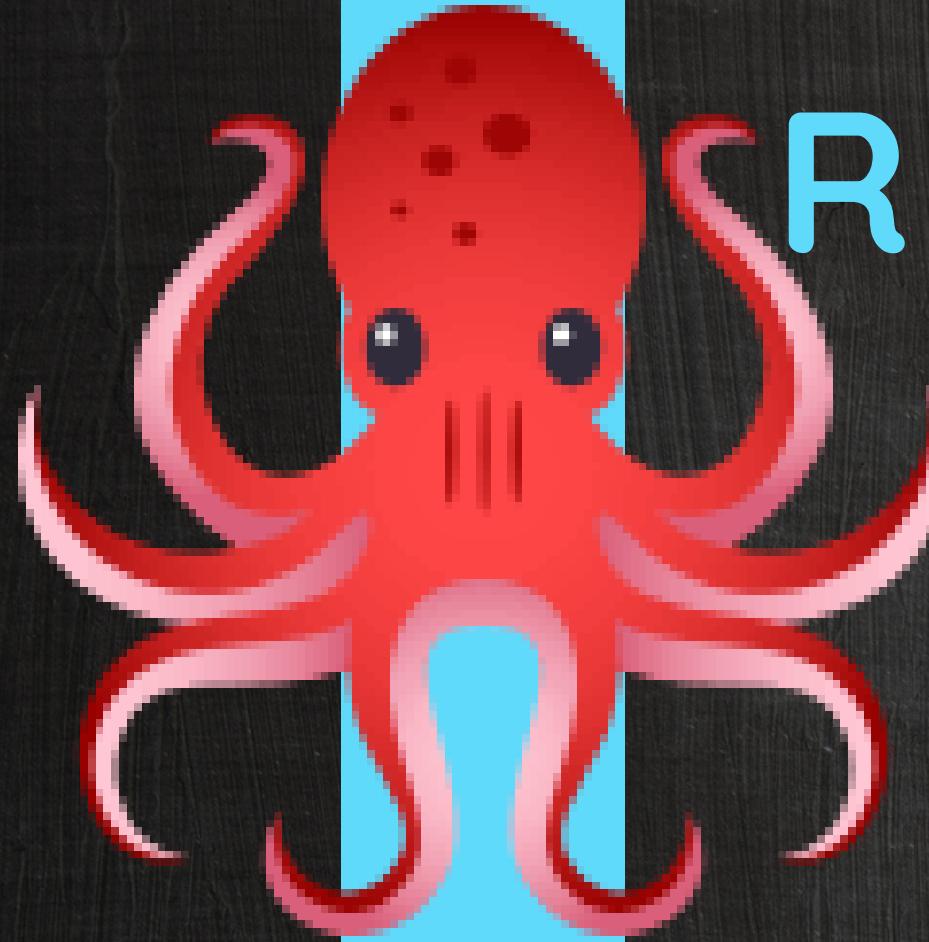
JEST

Jest.js est un framework de test JavaScript utilisé pour tester les applications.



```
function sum(a, b) {
  return a + b;
}

test('adds 1 + 2 to equal 3', () => {
  expect(sum(1, 2)).toBe(3);
});
```



REACT TESTING LIBRARY

React Testing Library est un utilitaire de test pour les applications React qui se concentre sur le test des interactions et du comportement des utilisateurs.

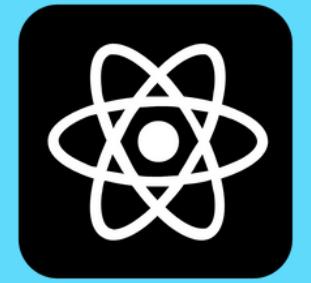
```
test("Counter increments when button is clicked",
() => {
  const { getByTestId } = render(<Counter />);

  fireEvent.click(getByTestId("increment-btn"));

  expect(getByTestId("counter")).toHaveTextContent
    ("2");
});
```



- La console peut être utilisée pour le déboggage en insérant des instructions, en inspectant les valeurs des variables et en traçant le flux d'exécution.
- React Development Tools peut être utilisé pour le déboggage en fournissant des informations sur les hiérarchies de composants, les changements d'état etc.
- L'inspecteur Web peut être utilisé pour inspecter et modifier HTML, CSS et JavaScript en temps réel.



STRATÉGIES



Analysez
l'application
complète.



yarn test –
watch



Tous les tests
réussis



Les points à respecter

- 
- A partir des tests échoués, notre objectif est de réparer les fonctionnalités cassées.
 - Analysez le .json pour identifier comment les données sont structurées
 - Remplir le cahier de recette.



Tests ajoutés

INTÉGRATION

- Completer les 3 tests d'intégration de la page Home.

UNITAIRE

- Completer le test unitaire de Helper.
- Ajout d'un test unitaire à EventCard qui vérifie le "alt" d'une image pour garantir l'accessibilité du site.
- Ajout d'un nouveau test unitaire qui vérifie que l'on peut correctement changer la pagination dans le composant Events.



Avant

```
Test Suites: 5 failed, 11 passed, 16 total
Tests:      5 failed, 3 skipped, 41 passed, 49 total
Snapshots:  0 total
Time:       50.24 s
```



Après

```
PASS  src/components/ServiceCard/index.test.js
PASS  src/components/PeopleCard/index.test.js
PASS  src/components/Button/index.test.js
PASS  src/containers/Events/index.test.js (6.759 s)
PASS  src/components/Icon/index.test.js
PASS  src/helpers/Date/index.test.js
```

```
Test Suites: 16 passed, 16 total
Tests:      5 skipped, 48 passed, 53 total
Snapshots:  0 total
```

Compétences à acquérir

Dans ce projet, nous avons écrit les tests basés sur le BDD - "Behavior-Driven Development", afin de pouvoir imaginer les actions des utilisateurs sur notre application.

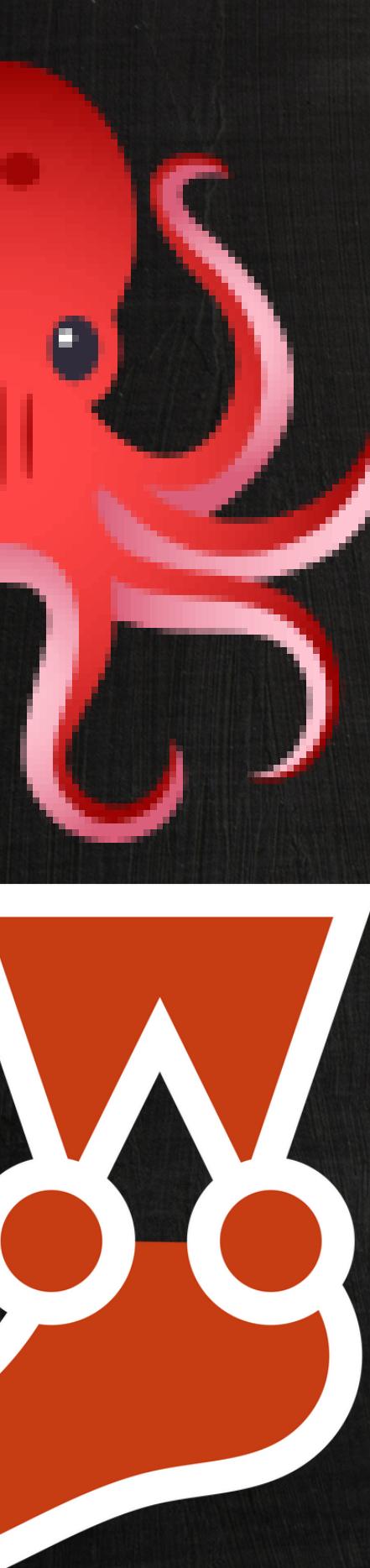
À l'avenir, j'adorerais développer avec TDD - "Test-Driven Development",

1
RED
Write a test
that fails

TDD

3
REFACTOR
Eliminate
redundancy

2
GREEN
Make the
code work



Conclusion

