AUTOMATION DataCamp

GUIDE GRATUIT · ÉDITION 2026 · DONNÉES VÉRIFIÉES EN OCTOBRE 2026

Guide QA Automation 2026.

Devenir testeur automaticien, ou progresser dans le métier : la feuille de route, les salaires sourcés, le choix des outils, les certifications utiles, le financement et un portfolio qui se montre. Avec exemples de code Playwright et un plan de 90 jours.

6 chapitres
carrière, salaires, outils, certifications, financement, portfolio

Playwright
exemples en TypeScript, prêts à copier

90 jours
plan semaine par semaine pour démarrer

Sources
chiffres datés et vérifiés sur la source primaire



POUR QUI

Testeurs, développeurs, personnes en reconversion

Édition octobre 2026, version 1.0
contact@automationdatacamp.com

SOMMAIRE

Ce que contient ce guide

00	**Introduction**	à qui s'adresse le guide, comment l'utiliser	03
01	**Roadmap carrière QA**	quatre étapes, compétences, pièges, transition	04
02	**Salaires et marché**	repères sourcés, compétences demandées	06
03	**Choisir ses outils**	Playwright, Selenium, Cypress, WebdriverIO sur 12 critères	08
04	**Premiers pas avec Playwright**	installation, tests web et API, Page Object, configuration	10
05	**Certifications**	ISTQB, CPACC, AWS : formats, prix, ordre conseillé	13
06	**Financer sa formation**	les options réelles aujourd'hui, en quatre étapes	15
07	**Portfolio GitHub**	dix projets, README, structure, intégration continue	16
08	**Plan de 90 jours**	treize semaines pour passer à l'action	19
09	**Glossaire**	les termes à connaître en entretien	20
10	**Aller plus loin avec ADC**	les deux façons de se former avec nous	21
11	**À propos d'ADC**	qui a écrit ce guide	22
12	**Sources**	références numérotées et datées	23
13	**Licence et mentions**	droits d'utilisation, marques citées	25

> Les appels entre crochets, comme [1], renvoient à la page Sources, où chaque référence porte son adresse et sa date de consultation. Les chiffres de ce guide ont été vérifiés le 22 octobre 2026 ; ils évoluent, en particulier les versions d'outils.

00 · INTRODUCTION

Pourquoi ce guide, et comment le lire

Le test logiciel automatisé est un métier où l'on apprend vite à condition de savoir dans quel ordre avancer. Ce guide rassemble ce que nous répondons chaque semaine aux personnes qui veulent entrer dans le métier ou y progresser : quelles compétences viser, ce que paie réellement le marché, quel outil apprendre d'abord, quelles certifications valent l'effort et comment montrer son niveau à un recruteur.

À qui il s'adresse

Vous partez de zéro

Reconversion, études, premier emploi. Lisez les chapitres dans l'ordre, puis suivez le plan de 90 jours.

Vous testez déjà à la main

Vous connaissez les cas de test et les anomalies. Allez droit aux chapitres 3, 4 et 7 : outils, code et portfolio.

Vous développez

Vous savez coder. Les chapitres 1, 5 et 7 vous montrent ce qui distingue un bon testeur d'un développeur qui écrit des tests.

Comment l'utiliser

- **Comme une carte.** Le chapitre 1 situe votre point de départ ; chaque chapitre suivant répond à une question concrète.
- **Comme un cahier d'exercices.** Les encadrés « À faire » et les exemples de code sont pensés pour être refaits sur votre machine, pas seulement lus.
- **Comme une référence.** Le glossaire et la page Sources vous servent à préparer un entretien ou à vérifier un chiffre.

Ce que ce guide n'est pas

Ce n'est pas un cours complet : un guide ne remplace ni la pratique régulière ni les retours d'un formateur sur votre code. Ce n'est pas non plus une promesse d'emploi ou de salaire. Les chiffres cités décrivent le marché à une date donnée, avec leurs limites, que nous indiquons à chaque fois.

> **Notre parti pris.** Nous recommandons Playwright avec TypeScript pour démarrer, parce que c'est l'outil le plus cité dans les offres QA françaises que nous avons analysées [2] et qu'il couvre le web et les API avec un seul outil. Le chapitre 3 explique dans quels cas un autre choix est plus pertinent.

01 · ROADMAP CARRIÈRE QA

De testeur à Quality Architect

La carrière QA se lit en quatre étapes. Les durées indiquées sont des repères courants, pas des seuils : on change d'étape quand on maîtrise les compétences, pas quand on a « fait ses années ».



É1 Testeur QA (manuel)

repère : 0 à 2 ans

Comprendre ce qu'il faut tester et pourquoi. Écrire des cas de test clairs, les exécuter, rédiger des rapports d'anomalie reproductibles.

COMPÉTENCES CLÉS

Techniques de conception (partitions d'équivalence, valeurs limites, tables de décision), rédaction d'anomalies, outil de gestion des tests (Jira, Xray, Squash TM), SQL de base.

REPÈRE DE SALAIRE

En début de carrière, un testeur logiciel gagne en moyenne 25 000 à 28 000 € bruts par an selon HelloWork [1].

É2 QA Automation Engineer

repère : 2 à 5 ans

Transformer les tests répétitifs en tests automatisés fiables, qui tournent seuls à chaque modification du code.

COMPÉTENCES CLÉS

Un langage (TypeScript, Python ou Java), un framework (Playwright, Selenium ou Cypress), Git, tests d'API, intégration continue (GitLab CI, Jenkins, GitHub Actions).

CE QUI FAIT LA DIFFÉRENCE

Des tests stables : localisateurs robustes, aucune attente fixe, données de test maîtrisées, rapports lisibles par toute l'équipe.

É3 Senior QA ou QA Lead

repère : 5 à 8 ans

Concevoir l'architecture des suites de tests, fixer la stratégie qualité d'une équipe et faire progresser les autres.

COMPÉTENCES CLÉS

Stratégie et plan de test, choix d'outils argumenté, revue de code, mentorat, indicateurs qualité, tests de performance et d'accessibilité.

REPÈRE DE SALAIRE

Médiane de 50 000 € bruts annuels dans les offres senior ou lead qui affichent un salaire (31 offres sur 89), baromètre ADC d'octobre 2026 [2].

É4 Principal QA ou Quality Architect

repère : 8 ans et plus

Porter la qualité au niveau du produit et de l'organisation : plusieurs équipes, plusieurs produits, des choix qui engagent sur plusieurs années.

COMPÉTENCES CLÉS

Architecture de test à l'échelle, plateformes d'exécution, observabilité, gouvernance de l'IA dans les tests, arbitrage coût et risque.

POSTURE

Convaincre sans imposer : la qualité devient une responsabilité partagée, pas une étape en fin de chaîne.

Les pièges qui ralentissent une carrière

Rester trop longtemps sur le manuel

Le test manuel est une base indispensable, pas une destination. Commencez à automatiser vos propres cas répétitifs dès que vous les maîtrisez.

Ignorer Git

Un test qui n'est pas versionné n'existe pas pour l'équipe. Branches, commits lisibles et demandes de fusion font partie du métier.

Négliger la revue de code

Vos tests sont du code : ils se relisent, se refactorisent et se maintiennent. Demandez des revues et faites-en.

Collectionner les outils

Mieux vaut un framework maîtrisé en profondeur que cinq survolés. Les recruteurs regardent la qualité de vos tests, pas la longueur de la liste.

Transition réaliste. Passer du test manuel à l'automatisation demande en général de 6 à 18 mois selon le temps disponible, avec une formation ciblée et des projets personnels (estimation ADC). Le plan de 90 jours du chapitre 8 couvre la première marche.

À faire : situer votre point de départ

- [ ] Je sais écrire un cas de test avec préconditions, étapes et résultat attendu.
- [ ] Je sais rédiger une anomalie qu'un développeur reproduit du premier coup.
- [ ] J'utilise Git au quotidien (clone, branche, commit, push, demande de fusion).
- [ ] J'ai écrit au moins un test automatisé qui tourne dans une CI.
- [ ] Je sais expliquer pourquoi un test est instable et comment le corriger.
- [ ] J'ai défendu une stratégie de test devant une équipe.

Une ou deux cases cochées : étape 1. Trois ou quatre : étape 2. Cinq ou six : vous visez l'étape 3.

02 · SALAIRES ET MARCHÉ

Ce que paie et ce que demande le marché

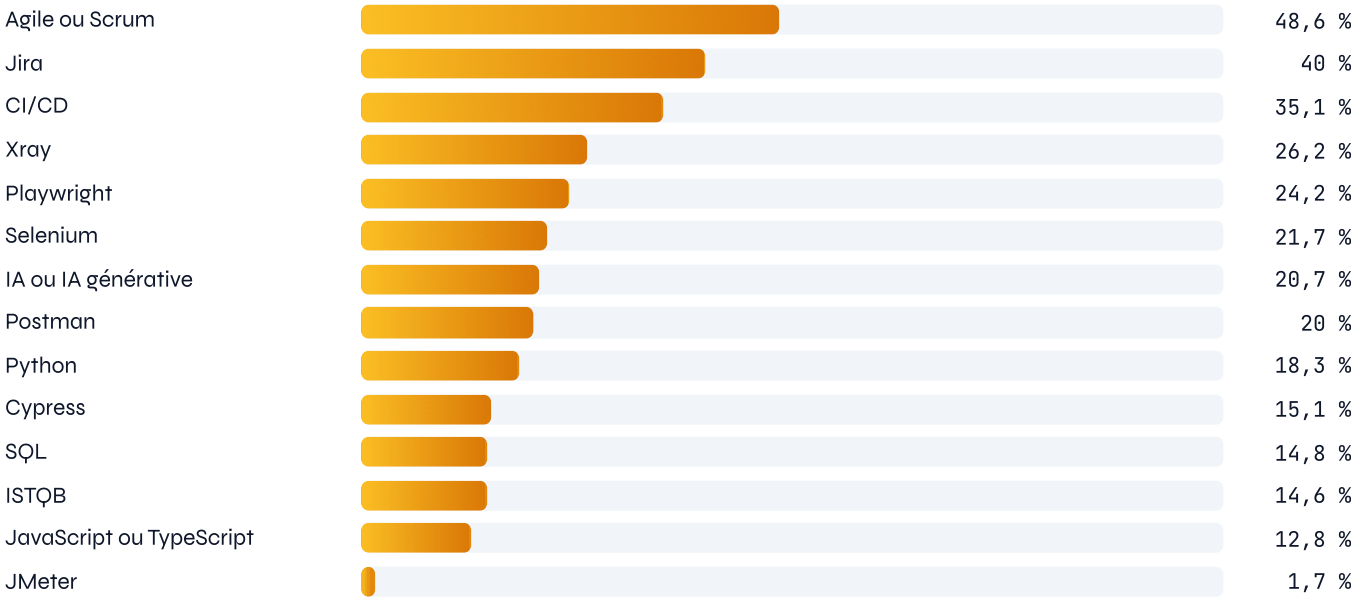
Deux sources se complètent. HelloWork donne des repères de salaire par métier [1]. Le baromètre ADC compte ce que disent les offres d'emploi publiées en France, à partir de l'API publique de l'agrégateur Adzuna [2].

Repères de salaire en France

Repère	Valeur	Source
Testeur logiciel en début de carrière	25 000 € à 28 000 € bruts par an	HelloWork [1]
Salaire médian du testeur logiciel	Environ 32 500 € bruts par an (environ 2 700 € bruts par mois)	HelloWork [1]
Testeur après plusieurs années	Jusqu'à environ 38 500 € bruts par an	HelloWork [1]
Offres de test et QA qui affichent un salaire (229 offres, tous niveaux)	Médiane 42 500 € bruts annuels ; la moitié entre 40 000 € et 45 000 €	Baromètre ADC [2]
Offres senior ou lead qui affichent un salaire	Médiane 50 000 € bruts annuels	Baromètre ADC [2]

Les compétences les plus citées dans les offres QA

Part des 405 offres françaises dont l'intitulé contient « QA » qui citent chaque compétence, relevé du 2 octobre 2026 sur 60 jours [2].



Comment lire ces chiffres

- HelloWork décrit le métier de testeur logiciel en général, sans date de mise à jour affichée sur sa page [1]. Le baromètre ne porte que sur les offres qui affichent un salaire : 229 sur 605, soit 37,9 % [2].
- L'écart entre les deux sources s'explique en partie par le périmètre : les offres « QA » visent souvent des profils d'automatisation, mieux payés que le test manuel.
- Le baromètre ne compare pas les salaires par ville ni par spécialité. L'Île-de-France concentre 31,9 % des offres de test et de QA [2].

Ce que cela change pour vous

L'outillage compte autant que le code

Jira (40 %) et Xray (26,2 %) devancent tous les frameworks. Savoir organiser et tracer ses tests est une compétence attendue, pas un détail.

La CI est la norme

Une offre QA sur trois cite l'intégration continue. Un test qui ne tourne que sur votre poste ne convainc pas.

L'IA entre dans les fiches de poste

Une offre QA sur cinq la mentionne : tester des produits qui intègrent de l'IA, ou utiliser des assistants pour écrire des tests.

La performance reste une spécialité

JMeter (1,7 %), k6 et Gatling (1 % chacun) sont rarement demandés à un testeur généraliste. À garder pour plus tard.

> **Limites du baromètre.** Une même offre peut apparaître dans plusieurs bases ; les offres sans salaire affiché ne sont pas extrapolées ; le relevé porte sur 60 jours et sur la France uniquement. Méthode complète dans l'article source [2].

03 · CHOISIR SES OUTILS

Playwright, Selenium, Cypress, WebdriverIO

Les quatre frameworks sont gratuits, open source et utilisés en production. Le bon choix dépend du langage de l'équipe, des navigateurs à couvrir et de ce que demandent les offres que vous visez. Voici les faits, puis notre recommandation.



Comparatif sur 12 critères

Critère	Playwright	Selenium	Cypress	WebdriverIO
1. Licence	Apache-2.0 [3]	Apache-2.0 [4]	MIT [5]	MIT [6]
2. Dernière version au 22/10/2026	1.64.1 (14/10) [3]	4.51.0 (09/10) [4]	16.2.0 (14/10) [5]	10.1.0 (18/10) [6]
3. Langages	JavaScript et TypeScript, Python, Java, .NET [9]	Java, Python, C#, Ruby, JavaScript, Kotlin [13]	JavaScript (et TypeScript) uniquement [12]	JavaScript et TypeScript
4. Navigateurs	Chromium, Firefox, WebKit [8]	Chrome, Edge, Firefox, Safari [13]	Famille Chrome, Edge, Firefox ; WebKit expérimental [11]	Chrome, Firefox, Safari, Edge [14]
5. Protocole de pilotage	Propre à Playwright	WebDriver (norme W3C)	Exécution dans le navigateur [12]	WebDriver et WebDriver BiDi [14]
6. Mobile	Émulation Chrome Android et Mobile Safari [8]	Applications natives via Appium (projet distinct)	Pas d'application native	Applications iOS et Android natives [14]
7. Onglets et navigateurs multiples	Oui	Oui	Un seul navigateur à la fois ; onglets via un plugin [12]	Oui
8. Tests d'API intégrés	Oui (contexte `request`)	Non, bibliothèque à part	Oui (`cy.request`)	Non, bibliothèque à part

Critère	Playwright	Selenium	Cypress	WebdriverIO
9. Tests de composants	Expérimental	Non	Oui	Oui (React, Vue, Svelte…) [14]
10. Débogage	Trace Viewer, mode UI	Outils du langage choisi	Retour dans le temps dans l'interface	Commande `debug`, rapports
11. Téléchargements npm, sept. 2026	261,3 millions [7]	8,7 millions [7]	27,2 millions [7]	11,4 millions [7]
12. Offres QA en France qui le citent	24,2 % [2]	21,7 % [2]	15,1 % [2]	non mesuré

Lecture : les téléchargements npm mesurent l'installation des paquets JavaScript (y compris par les chaînes d'intégration continue), pas le nombre d'utilisateurs ; ils ne comptent pas Selenium en Java ou en Python. Les cellules sans appel décrivent le fonctionnement documenté de l'outil.

Notre recommandation

Vous débutez : Playwright en TypeScript

Le plus cité dans les offres QA analysées, un seul outil pour le web et les API, trois moteurs de navigateur, une attente automatique des éléments qui réduit les tests instables, et un Trace Viewer qui fait gagner des heures de débogage.

Votre équipe est sur Selenium : restez-y

L'écosystème est mûr, les langages nombreux et Selenium reste très demandé (21,7 % des offres QA). Apprenez Playwright en parallèle sur vos projets personnels.

Projet front-end JavaScript : Cypress

Confortable pour tester une application web d'une seule origine, avec des tests de composants. Ses limites (un navigateur à la fois, JavaScript seul) comptent moins dans ce cadre.

Web et mobile natif : WebdriverIO

Il s'appuie sur les normes WebDriver et WebDriver BiDi et couvre aussi les applications iOS et Android, ce qui évite deux outils distincts.

04 · PREMIERS PAS AVEC PLAYWRIGHT

Votre premier projet, en TypeScript

Ces exemples sont courts et exacts : ils tournent tels quels avec Playwright 1.64. Les sites utilisés sont des sites publics prévus pour l'entraînement.

1. Installer

Playwright demande Node.js 22, 24 ou 26, sous Windows 11, macOS 14 ou une distribution Linux récente [8]. Dans un dossier vide :

```
# crée le projet, la configuration, un exemple de test et (au choix) le workflow GitHub Actions
npm init playwright@latest

# lance tous les tests, puis ouvre le rapport HTML
npx playwright test
npx playwright show-report
```

2. Un premier test web

```
// tests/accueil.spec.ts
import { test, expect } from '@playwright/test';

test("la page d'accueil affiche le bon titre", async ({ page }) => {
  await page.goto('https://playwright.dev/');
  await expect(page).toHaveTitle(/Playwright/);
});
```

Deux réflexes à prendre dès le premier test : une assertion par intention, et des `expect` « web-first » qui attendent d'eux-mêmes que la condition soit vraie, sans `waitForTimeout`.

3. Des localisateurs lisibles

Ciblez les éléments comme un utilisateur les voit (rôle, libellé, texte) plutôt que par des sélecteurs CSS fragiles.

```
// tests/connexion.spec.ts : site de démonstration public saucedemo.com
import { test, expect } from '@playwright/test';

test('un utilisateur standard accède au catalogue', async ({ page }) => {
  await page.goto('https://www.saucedemo.com/');
  await page.getByPlaceholder('Username').fill('standard_user');
  await page.getByPlaceholder('Password').fill('secret_sauce');
  await page.getByRole('button', { name: 'Login' }).click();
  await expect(page).toHaveURL(/inventory/);
});
```

Les identifiants `standard_user` et `secret_sauce` sont publiés sur la page d'accueil du site de démonstration ; ce ne sont pas de vrais comptes.

4. Un test d'API, sans navigateur

```
// tests/api.spec.ts : API publique de démonstration JSONPlaceholder
import { test, expect } from '@playwright/test';

test('GET /posts/1 renvoie le bon article', async ({ request }) => {
  const response = await request.get('https://jsonplaceholder.typicode.com/posts/1');
  expect(response.ok()).toBeTruthy();
  const body = await response.json();
  expect(body).toHaveProperty('id', 1);
});
```

5. Le modèle Page Object

Dès que plusieurs tests partagent un écran, regroupez ses localisateurs et ses actions dans une classe. Un changement d'interface ne se corrige alors qu'à un seul endroit.

```
// pages/LoginPage.ts
import { type Page, type Locator } from '@playwright/test';

export class LoginPage {
  readonly username: Locator;
  readonly password: Locator;
  readonly submit: Locator;

  constructor(private readonly page: Page) {
    this.username = page.getByPlaceholder('Username');
    this.password = page.getByPlaceholder('Password');
    this.submit = page.getByRole('button', { name: 'Login' });
  }

  async goto() {
    await this.page.goto('/');
  }

  async login(user: string, pass: string) {
    await this.username.fill(user);
    await this.password.fill(pass);
    await this.submit.click();
  }
}
```

```
// tests/connexion-pom.spec.ts
import { test, expect } from '@playwright/test';
import { LoginPage } from '../pages/LoginPage';

test('connexion via la page objet', async ({ page }) => {
  const login = new LoginPage(page);
  await login.goto();
  await login.login('standard_user', 'secret_sauce');
  await expect(page).toHaveURL(/inventory/);
});
```

6. Une configuration utile

```ts
// playwright.config.ts
import { defineConfig, devices } from '@playwright/test';

export default defineConfig({
  testDir: './tests',
  retries: process.env.CI ? 2 : 0,      // relances seulement en CI
  reporter: 'html',
  use: {
    baseURL: 'https://www.saucedemo.com',  // utilisé par page.goto('/')
    trace: 'on-first-retry',               // trace pour comprendre un échec
  },
  projects: [
    { name: 'chromium', use: { ...devices['Desktop Chrome'] } },
    { name: 'firefox',  use: { ...devices['Desktop Firefox'] } },
    { name: 'webkit',   use: { ...devices['Desktop Safari'] } },
  ],
});
```

Checklist : un test automatisé de qualité

- [ ] Son nom décrit le comportement attendu, pas la technique.
- [ ] Il est indépendant : il ne dépend ni de l'ordre d'exécution ni d'un autre test.
- [ ] Il ne contient aucune attente fixe (`waitForTimeout`).
- [ ] Ses localisateurs sont fondés sur le rôle, le libellé ou un identifiant de test.
- [ ] Il crée ses données ou les isole, et ne laisse pas de trace derrière lui.
- [ ] Il échoue pour une seule raison, avec un message compréhensible.
- [ ] Il passe dix fois de suite en local et en CI avant d'être fusionné.

05 · CERTIFICATIONS

Quelles certifications, dans quel ordre

Une certification ne remplace pas un portfolio, mais elle structure vos connaissances et rassure sur le vocabulaire commun. L'ISTQB est citée dans 14,6 % des offres QA françaises : un atout, rarement une condition éliminatoire [2].

Certification	Format de l'examen	Prérequis	Prix
ISTQB CTFL v4.0 Foundation Level	40 questions, 40 points, réussite à 26 ; 60 minutes, 25 % de temps en plus hors langue maternelle [15]	Aucun ; formation facultative [15]	Fixé par l'organisme d'examen : à demander au CFTL ou à un fournisseur agréé
ISTQB CT-GenAI v1.1 tester avec l'IA générative	40 questions, 46 points, réussite à 30 (65 %) ; 60 minutes, 25 % en plus hors langue maternelle [16]	CTFL [16]	Fixé par l'organisme d'examen
ISTQB CTAL-TAE v2.0 Test Automation Engineering	40 questions, 66 points, réussite à 43 ; 90 minutes [17]	CTFL et une expérience pratique suffisante [17]	Fixé par l'organisme d'examen
IAAP CPACC accessibilité, compétences de base	100 questions, 2 heures [18]	Aucun	510 USD ; 410 USD pour les membres IAAP [18]
AWS Certified Developer - Associate	65 questions à choix unique ou multiple, 130 minutes [19]	Aucun	150 USD, hors taxes et change [19]

> **Changement de version AWS.** L'examen passe de DVA-C02 à DVA-C03 : inscriptions à la nouvelle version à partir du 27 octobre 2026, dernier jour pour passer DVA-C02 le 30 novembre 2026 [19]. Choisissez vos supports de révision en conséquence.

L'ordre que nous conseillons

1. **CTFL d'abord**

   C'est le socle commun et le prérequis de toutes les autres certifications ISTQB. Le syllabus v4.0 existe en français, publié par le CFTL [20].

2. **CT-GenAI ensuite**

   Elle répond directement à la demande des offres (une offre QA sur cinq cite l'IA [2]) et se prépare dans la foulée du CTFL, dont elle reprend le vocabulaire.

3. **CTAL-TAE après de la pratique**

   Elle porte sur l'architecture de l'automatisation et demande une expérience réelle. Visez-la quand vous avez maintenu une suite de tests en CI sur la durée.

4 CPACC ou AWS selon votre spécialité

CPACC si vous visez l'accessibilité : l'Acte européen sur l'accessibilité s'applique depuis juin 2025 [21], et le RGAA encadre l'accessibilité numérique en France. AWS si vos tests tournent sur une infrastructure cloud.

Se préparer sans dépenser

- Les syllabus et les examens blancs officiels sont téléchargeables gratuitement sur les pages de chaque certification ISTQB [15] [17].
- Pour le CPACC, l'IAAP conseille de prévoir 5 à 10 heures par semaine pendant 6 à 8 semaines et publie des questions d'exemple [18].
- Faites les examens blancs en conditions réelles, chronomètre lancé, et relisez chaque erreur dans le syllabus plutôt que de mémoriser la réponse.

À faire : réviser le CTFL v4.0 en six semaines

Une semaine par chapitre du syllabus [20], puis un examen blanc officiel en fin de semaine 6.

- [ ] Semaine 1 : fondamentaux du test (pourquoi tester, les sept principes, le processus de test).
- [ ] Semaine 2 : le test tout au long du cycle de développement (niveaux et types de test, maintenance).
- [ ] Semaine 3 : le test statique (revues, analyse statique).
- [ ] Semaine 4 : analyse et conception des tests (techniques boîte noire, boîte blanche, fondées sur l'expérience).
- [ ] Semaine 5 : gestion des activités de test (planification, estimation, risques, suivi, défauts).
- [ ] Semaine 6 : outils de test, puis examen blanc chronométré et reprise de chaque erreur.

06 · FINANCER SA FORMATION

Les options réelles, aujourd'hui

Ce chapitre décrit le financement des formations ADC au 22 octobre 2026. Nous préférons être précis : CPF et OPCO prochainement. Aujourd'hui, une formation ADC se finance personnellement ou par l'entreprise sur son budget formation [22].

Financement personnel

1 500 € TTC

Formation Testeur QA Automatisation & IA (399 h, cours du soir), payable en 1, 2, 3 ou 4 fois sans frais, soit 4 × 375 € [22].

Financement entreprise

7 500 € TTC

Formule à temps plein, prise en charge par l'employeur sur son budget formation, avec devis et convention de formation [22].

Les quatre étapes

1. **Parlez d'abord à votre employeur**

   Si vous êtes salarié, présentez le projet à votre responsable ou aux ressources humaines, programme détaillé à l'appui.

2. **Demandez un devis et une convention**

   ADC fournit le devis et la convention de formation. Écrivez à contact@automationdatacamp.com en précisant la formule et les dates visées.

3. **Sinon, financez en plusieurs fois**

   En financement personnel, le paiement en 4 fois sans frais étale la dépense sur quatre mensualités. Le programme de parrainage réduit le prix de votre propre formation : -10 % au premier filleul inscrit, -20 % au deuxième, -50 % au troisième, -60 % au quatrième, formation offerte au cinquième [24].

4. **Calez le calendrier**

   Réservez tôt l'entretien préalable obligatoire de 30 minutes, pour que le financement soit réglé avant le démarrage.

CPF et OPCO prochainement. Ces dispositifs ne financent pas encore les formations ADC. Méfiez-vous de toute offre qui vous promet le contraire pour un organisme qui n'y est pas inscrit : vérifiez toujours sur la plateforme officielle Mon Compte Formation avant de vous engager.

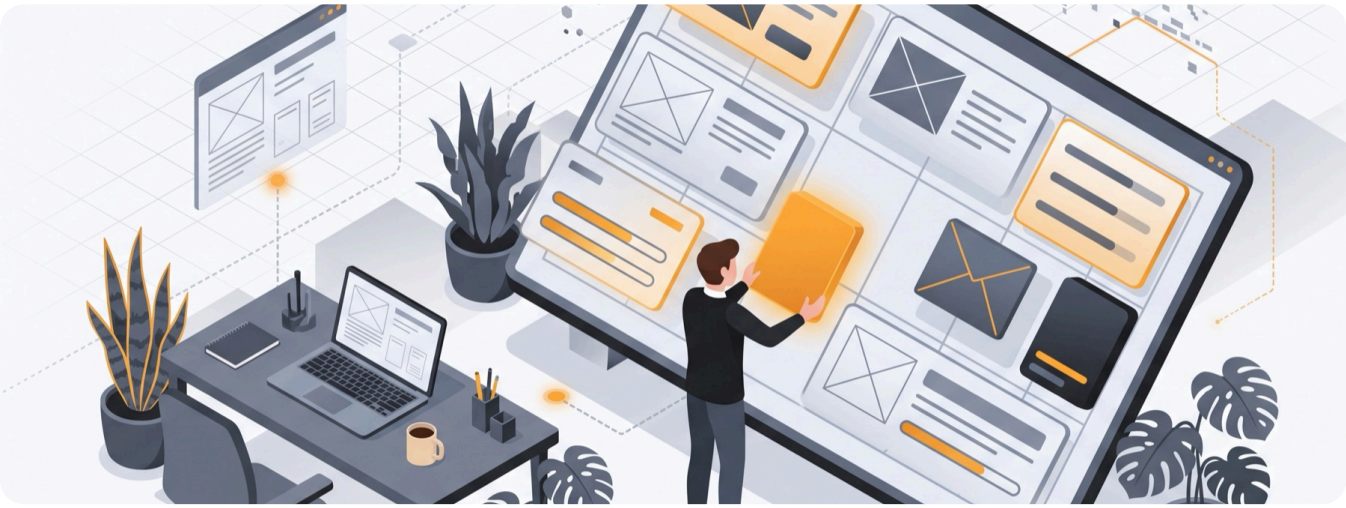
Checklist avant de payer une formation, quelle qu'elle soit

- [ ] Le programme détaillé (heures, contenus, livrables) est écrit et téléchargeable.
- [ ] Ce qui est certifié est précis : certificat de l'organisme, examen officiel inclus ou non.
- [ ] Le prix total TTC et les modalités de paiement sont affichés.
- [ ] Vous savez combien d'heures se font en direct avec un formateur.
- [ ] Les conditions d'annulation et de remboursement sont écrites.
- [ ] Vous pouvez parler à quelqu'un avant de vous inscrire.

07 · PORTFOLIO GITHUB

Dix projets qui montrent votre niveau

Un recruteur technique passe quelques minutes sur votre GitHub. Il cherche des tests qui tournent, un README qui explique, et des choix justifiés. Trois projets soignés valent mieux que dix ébauches : choisissez dans cette liste ceux qui correspondent aux postes visés.



Projet	Outils	Ce qu'il doit montrer
1. Suite E2E web	Playwright, TypeScript	Parcours d'achat sur saucedemo.com, Page Object, fixtures, rapport HTML.
2. Tests d'API	Playwright `request` ou Postman et Newman	Codes de retour, schéma JSON, cas d'erreur, exécution en CI.
3. Pipeline CI complet	GitHub Actions	Lint, tests, rapport publié en artefact, badge de statut dans le README.
4. Tests visuels	Playwright `toHaveScreenshot`	Comparaison de captures, gestion des différences acceptées.
5. Audit d'accessibilité automatisé	@axe-core/playwright	Violations détectées, priorisées et reliées aux critères WCAG.
6. Test de charge	k6	Scénario sur une API publique, seuils de réussite, lecture des résultats.
7. Framework BDD	Cucumber ou playwright-bdd	Scénarios Gherkin lisibles par le métier, étapes réutilisables.
8. Tests de contrat	Pact	Contrat consommateur et vérification côté fournisseur.
9. Test mobile	Appium ou WebdriverIO	Un parcours sur émulateur Android, rapport lisible.
10. Scan de sécurité	ZAP	Scan automatisé d'une application d'entraînement, rapport commenté.

Testez uniquement des sites et des API publics prévus pour l'entraînement, ou vos propres applications. Un scan de sécurité ou un test de charge sur un site tiers sans autorisation est interdit.

Structure de projet conseillée

```
mon-portfolio-playwright/
├── .github/workflows/playwright.yml   # intégration continue
├── pages/                             # Page Objects
│   └── LoginPage.ts
├── tests/
│   ├── e2e/                           # parcours utilisateur
│   └── api/                           # tests d'API
├── fixtures/                          # données de test
├── playwright.config.ts
├── package.json
└── README.md
```

Le workflow GitHub Actions

C'est le fichier que génère `npm init playwright@latest` si vous acceptez l'option GitHub Actions [10]. Il installe les dépendances et les navigateurs, lance les tests et conserve le rapport 30 jours.

```yaml
# .github/workflows/playwright.yml
name: Playwright Tests
on:
  push:
    branches: [ main, master ]
  pull_request:
    branches: [ main, master ]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v6
    - uses: actions/setup-node@v6
      with:
        node-version: lts/*
    - name: Install dependencies
      run: npm ci
    - name: Install Playwright Browsers
      run: npx playwright install --with-deps
    - name: Run Playwright tests
      run: npx playwright test
    - uses: actions/upload-artifact@v4
      if: ${{ !cancelled() }}
      with:
        name: playwright-report
        path: playwright-report/
        retention-days: 30
```

Modèle de README

```
# Tests automatisés de saucedemo.com

![Tests](https://github.com/<compte>/<dépôt>/actions/workflows/playwright.yml/badge.svg)

## En bref
Suite Playwright (TypeScript) : 12 tests E2E et 6 tests d'API,
exécutés sur Chromium, Firefox et WebKit à chaque push.

## Lancer les tests (3 commandes)
npm ci
npx playwright install --with-deps
npx playwright test

## Choix techniques
- Page Object pour isoler les écrans
- Localisateurs par rôle et libellé, aucune attente fixe
- Traces conservées au premier échec en CI

## Rapport
Capture du rapport HTML et lien vers le dernier artefact de CI.
```

Ce qu'un recruteur regarde en premier

- [ ] Le badge de CI est vert, et l'historique montre des commits réguliers.
- [ ] Le README se lit en une minute et les tests se lancent en trois commandes.
- [ ] Les noms de tests décrivent des comportements métier.
- [ ] Vous expliquez un choix technique et ce que vous feriez autrement.

08 · PLAN DE 90 JOURS

Treize semaines pour passer à l'action

Ce plan suppose 8 à 10 heures par semaine, par exemple deux soirées et une demi-journée le week-end. Chaque semaine se termine par un livrable concret, poussé sur GitHub. Si vous prenez du retard, décalez le plan plutôt que de sauter une étape.

Sem.	Objectif	Livrable de fin de semaine
1	Les bases du test : niveaux, types, techniques de conception	20 cas de test écrits pour un site de démonstration
2	Anomalies et outillage : rédiger, prioriser, tracer	5 rapports d'anomalie et un tableau de suivi
3	Git et GitHub : branches, commits, demandes de fusion	Dépôt public avec README et première demande de fusion
4	TypeScript : types, fonctions, async et await	10 petits exercices corrigés dans le dépôt
5	Playwright : installation, premiers tests, assertions	5 tests web qui passent sur trois navigateurs
6	Localisateurs robustes et débogage avec le Trace Viewer	Un test instable corrigé, avec l'explication dans le commit
7	Page Object et fixtures	Suite refactorisée, sans duplication
8	Tests d'API et données de test	6 tests d'API, dont 3 cas d'erreur
9	Intégration continue avec GitHub Actions	Pipeline vert, rapport en artefact, badge dans le README
10	Tests visuels et accessibilité	Un test visuel et un audit axe intégrés à la CI
11	IA comme assistant : générer, puis relire chaque test	3 tests générés avec un assistant, revus et corrigés
12	Révision du syllabus CTFL et examens blancs	Deux examens blancs officiels, erreurs analysées
13	Portfolio et candidatures	README final, profil à jour, 10 candidatures ciblées

Chaque semaine

- Une séance pour apprendre, une pour pratiquer, une pour livrer.
- Un commit par jour de travail, même petit.
- Une question posée à quelqu'un de plus expérimenté.

En cas de blocage

- Lisez le message d'erreur en entier, puis la trace.
- Réduisez le problème au plus petit test qui échoue.
- Notez la solution dans le README : c'est aussi une preuve de méthode.

09 · GLOSSAIRE

Les termes à connaître

Assertion	Vérification d'un résultat attendu dans un test, par exemple `expect(page).toHaveTitle()`.
CI/CD	Intégration et livraison continues : chaque modification du code déclenche automatiquement la construction et les tests.
Cas de test	Ensemble de préconditions, d'étapes et de résultats attendus qui vérifie un comportement précis.
E2E (bout en bout)	Test qui parcourt l'application comme un utilisateur, de l'interface jusqu'aux données.
Fixture	Élément préparé avant un test et nettoyé après : page ouverte, utilisateur connecté, données de départ.
Localisateur	Moyen de retrouver un élément de la page : rôle, libellé, texte, identifiant de test.
Page Object	Classe qui regroupe les éléments et les actions d'un écran, pour éviter la duplication dans les tests.
Pyramide des tests	Principe d'équilibre : beaucoup de tests unitaires, moins de tests d'API, peu de tests E2E, plus lents et plus fragiles.
Régression	Défaut qui réapparaît ou apparaît dans une fonction qui marchait, après une modification.
Test instable	Test qui réussit ou échoue sans changement du code (en anglais « flaky ») ; souvent dû à une attente mal gérée.
Test de contrat	Vérifie qu'un fournisseur d'API respecte ce qu'attend son consommateur.
Test de non-régression	Ensemble de tests relancés après chaque modification pour s'assurer que rien n'a cassé.
Trace Viewer	Outil Playwright qui rejoue un test pas à pas : captures, réseau, console, action par action.
WCAG et RGAA	Référentiels d'accessibilité numérique : international pour les WCAG, français pour le RGAA.
WebDriver BiDi	Évolution bidirectionnelle du protocole WebDriver du W3C, qui permet d'écouter les événements du navigateur.
Xray, Squash TM	Outils de gestion des tests : campagnes, exécutions, traçabilité avec les exigences.

10 · ALLER PLUS LOIN AVEC ADC

Se former avec un mentor

Ce guide donne la carte. Si vous voulez parcourir le chemin avec un formateur qui relit votre code, ADC propose deux formats en ligne, en français.

Formation Testeur QA Automatisation & IA

automationdatacamp.com/formation-qa-adc

- 399 h sur 12 semaines, projet d'entreprise réel en fin de parcours.
- Particuliers : 1 500 € TTC, cours du soir, en 1 à 4 fois sans frais.
- Entreprises : 7 500 € TTC, temps plein, devis et convention fournis.
- Formation certifiante : certificat ADC CP-QAI après soutenance, et examens ISTQB CTFL et CT-GenAI inclus, deuxième tentative offerte [22].

QA Odyssey

automationdatacamp.com/qa-odyssey

- Formations en ligne accompagnées, courtes, moyennes et longues, jusqu'à 12 semaines.
- Disponible aujourd'hui : devenir testeur QA automaticien en 12 semaines, 1 500 € TTC.
- Au moins 3 réunions avec le formateur, selon la durée de la formation.
- Certificat ADC à la fin et ebook ADC offert pour préparer ses prochaines certifications ; frais d'examen ISTQB non inclus [23].

CPF et OPCO prochainement. Pour toute question, ou pour réserver l'entretien préalable de 30 minutes : **contact@automationdatacamp.com**.

11 · À PROPOS D'ADC

AutomationDataCamp

AutomationDataCamp (ADC) est un organisme de formation en ligne, en français, spécialisé dans le test logiciel et l'automatisation des tests augmentée par l'IA. ADC se présente comme une plateforme d'apprentissage des métiers intelligents, c'est-à-dire des métiers que l'intelligence artificielle transforme. Les missions sont menées par l'équipe ADC.

NOM	AutomationDataCamp, abrégé ADC
ENTITÉ JURIDIQUE	AutomationDataCamp LLC, société de droit américain, État du Wyoming (États-Unis)
CRÉATION	14 mai 2026
SIÈGE SOCIAL	75 E 3rd St, Ste 7, Sheridan, WY 82801, États-Unis
DIRECTION	Oussama Belakhdar, fondateur
ACTIVITÉS	Formation en ligne au test logiciel et à l'IA appliquée au test ; conseil qualité, services IA et recrutement QA pour les entreprises
LANGUE	Français
CONTACT	contact@automationdatacamp.com · automationdatacamp.com

> Malgré la ressemblance des noms, AutomationDataCamp est une entreprise indépendante, sans aucun lien avec DataCamp : ni filiale, ni marque, ni partenaire, ni revendeur [25].

Sources

Chaque fait externe a été vérifié sur la source primaire. Toutes les références ont été consultées le 22 octobre 2026 ; en cas d'écart sur l'offre ADC, les pages de automationdatacamp.com font foi.

Marché et salaires

[1] HelloWork, *Combien gagne un testeur logiciel* : 25 000 à 28 000 € bruts par an en début de carrière, médiane d'environ 32 500 € bruts par an, jusqu'à environ 38 500 € après plusieurs années. La page n'indique pas de date de mise à jour.
https://www.hellowork.com/fr-fr/metiers/testeur-logiciel.html

[2] AutomationDataCamp, *Baromètre des compétences QA en France*, relevé du 2 octobre 2026 à partir de l'API publique d'Adzuna (offres publiées en France sur 60 jours : 405 offres « QA », 1 180 offres « test », échantillon de 605 offres uniques pour les salaires).
https://www.automationdatacamp.com/blog-post-barometre-competences-qa

Outils

[3] Registre npm, paquet *@playwright/test* : version 1.64.1 du 14/10/2026, licence Apache-2.0.
https://www.npmjs.com/package/@playwright/test

[4] Registre npm, paquet *selenium-webdriver* : version 4.51.0 du 09/10/2026, licence Apache-2.0.
https://www.npmjs.com/package/selenium-webdriver

[5] Registre npm, paquet *cypress* : version 16.2.0 du 14/10/2026, licence MIT.
https://www.npmjs.com/package/cypress

[6] Registre npm, paquet *webdriverio* : version 10.1.0 du 18/10/2026, licence MIT.
https://www.npmjs.com/package/webdriverio

[7] API de statistiques npm, téléchargements du 1er au 30 septembre 2026 (exemple pour Playwright ; même adresse pour les trois autres paquets).
https://api.npmjs.org/downloads/point/2026-09-01:2026-09-30/@playwright/test

[8] Playwright, *Installation* : commande d'installation, versions de Node.js et de systèmes, navigateurs pris en charge, émulation mobile.
https://playwright.dev/docs/intro

[9] Playwright, *Supported languages*.
https://playwright.dev/docs/languages

[10] Playwright, *Setting up CI* : workflow GitHub Actions de référence.
https://playwright.dev/docs/ci-intro

[11] Cypress, *Launching Browsers*.
https://docs.cypress.io/app/references/launching-browsers

[12] Cypress, *Trade-offs* : un seul navigateur piloté à la fois, onglets via @cypress/puppeteer, JavaScript seul langage.
https://docs.cypress.io/app/references/trade-offs

[13] Selenium, *Documentation* : langages des exemples et pages par navigateur.
https://www.selenium.dev/documentation/

[14] WebdriverIO, page d'accueil : WebDriver et WebDriver BiDi, navigateurs, applications mobiles, tests de composants.
https://webdriver.io/

Certifications et réglementation

[15] ISTQB, *Certified Tester Foundation Level (CTFL) v4.0*.
https://istqb.org/certifications/certified-tester-foundation-level-ctfl-v4-0/

[16] ISTQB, *Certified Tester Testing with Generative AI (CT-GenAI)*, syllabus v1.1.
https://istqb.org/certifications/gen-ai/

[17] ISTQB, *Certified Tester Advanced Level Test Automation Engineering (CTAL-TAE) v2.0*.
https://istqb.org/certifications/certified-tester-advanced-level-test-automation-engineering-ctal-tae-v2-0/

[18] IAAP, *CPACC* : format, tarifs, conseils de préparation.
https://www.accessibilityassociation.org/cpacc

[19] AWS, *AWS Certified Developer - Associate* : prix, durée, format, passage à DVA-C03.
https://aws.amazon.com/certification/certified-developer-associate/

[20] CFTL, syllabus ISTQB Niveau Fondation v4.0 en français.
https://cftl.fr/wp-content/uploads/2023/11/ISTQB-Syllabus-Niveau-Fondation-v4.0-Francais.pdf

[21] Commission européenne, *EU becomes more accessible for all*, 31 juillet 2025 : l'Acte européen sur l'accessibilité s'applique depuis juin 2025.
https://commission.europa.eu/news-and-media/news/eu-becomes-more-accessible-all-2025-07-31_en

Références ADC

[22] Formation Testeur QA Automatisation & IA : prix, formules, certifications, financement.
https://www.automationdatacamp.com/formation-qa-adc

[23] QA Odyssey : formations en ligne accompagnées.
https://www.automationdatacamp.com/qa-odyssey

[24] Programme de parrainage.
https://www.automationdatacamp.com/parrainage

[25] Qui est ADC, et quel lien avec DataCamp ? ; mentions légales.
https://www.automationdatacamp.com/a-propos-automation-datacamp · /legal

Licence

[26] Creative Commons, *Attribution - Pas d'Œuvre dérivée 4.0 International (CC BY-ND 4.0)*, licence de ce document.
https://creativecommons.org/licenses/by-nd/4.0/deed.fr

13 · LICENCE ET MENTIONS

Licence et mentions

CC BY-ND 4.0

© 2026 AutomationDataCamp LLC. Document distribué sous licence Creative Commons BY-ND 4.0 (partage autorisé sans modification, avec attribution).

Vous pouvez copier et redistribuer ce guide, sur tout support et dans tout format, à condition de citer AutomationDataCamp LLC comme auteur, d'indiquer le lien vers la licence et de ne pas le modifier. Texte officiel de la licence [26] :

`https://creativecommons.org/licenses/by-nd/4.0/deed.fr`

Marques citées

ISTQB® est une marque de l'International Software Testing Qualifications Board. CPACC est une certification de l'International Association of Accessibility Professionals (IAAP). AWS est une marque d'Amazon.com, Inc. ou de ses filiales. Playwright, TypeScript, Selenium, Cypress, WebdriverIO, Appium, Node.js, npm, GitHub, GitHub Actions, GitLab, Jenkins, Jira, Xray, Squash TM, Postman, Newman, k6, JMeter, Gatling, Pact, Cucumber, axe-core, ZAP, HelloWork et Adzuna sont des marques de leurs détenteurs respectifs. Elles sont citées à titre d'information. AutomationDataCamp n'est affilié à aucun de ces détenteurs et n'est pas lié à DataCamp.

Exemples de code

Les exemples de code de ce guide peuvent être réutilisés librement dans vos propres projets. Les sites saucedemo.com et jsonplaceholder.typicode.com sont des services publics de démonstration édités par des tiers.

Illustrations

Illustrations réalisées pour ADC avec un outil de génération d'images.

Nature des certifications

Le Certificat Professionnel ADC CP-QAI est un certificat privé d'organisme de formation, non enregistré au RNCP. Les certifications ISTQB, IAAP et AWS sont délivrées selon les règles de leurs organismes respectifs.

Éditeur

AutomationDataCamp LLC · 75 E 3rd St, Ste 7, Sheridan, WY 82801, États-Unis · contact@automationdatacamp.com · automationdatacamp.com
Édition octobre 2026, version 1.0.

AUTOMATION DataCamp