

GitHub Actions

Pipeline CI/CD

Julian F. Latorre

August 15, 2024

¿Qué es GitHub Actions?

- Herramienta de automatización de flujos de trabajo
- Integrada directamente en GitHub
- Permite automatizar:
 - Compilación
 - Pruebas
 - Despliegue
- Configuración basada en YAML

Ventajas de GitHub Actions

- Fácil configuración y uso
- Integración nativa con GitHub
- Ejecución en múltiples entornos
- Flujos de trabajo paralelos
- Secretos y variables de entorno seguros

Estructura de un workflow

```
name: CI
on: [push, pull_request]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Use Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'
      - run: npm ci
      - run: npm run build
      - run: npm test
```

Eventos de activación

- `push`
- `pull_request`
- `schedule`
- `release`
- `workflow_dispatch`
- `repository_dispatch`

Jest: Framework de pruebas para JavaScript

- Desarrollado por Facebook
- Enfocado en simplicidad
- Características principales:
 - Pruebas unitarias
 - Mocking
 - Cobertura de código
 - Snapshots

Configuración de Jest en GitHub Actions

```
name: Jest Tests
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Use Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'
      - run: npm ci
      - run: npm test
      - name: Upload coverage
        uses: actions/upload-artifact@v3
        with:
          name: coverage
          path: coverage
```

Cypress: Framework de pruebas E2E

- Pruebas de extremo a extremo (E2E)
- Enfocado en aplicaciones web modernas
- Características principales:
 - Ejecución en tiempo real
 - Depuración sencilla
 - Esperas automáticas
 - Control del estado de la aplicación

Configuración de Cypress en GitHub Actions

```
name: Cypress Tests
on: [push, pull_request]
jobs:
  cypress-run:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Cypress run
        uses: cypress-io/github-action@v3
        with:
          build: npm run build
          start: npm start
          wait-on: http://localhost:3000
```

Workflow completo con Jest y Cypress

```
name: CI/CD Pipeline
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Use Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'
      - run: npm ci
      - run: npm run test:unit
      - run: npm run build
      - name: Cypress run
        uses: cypress-io/github-action@v3
        with:
          start: npm start
```

Mejores prácticas

- Usar acciones oficiales y verificadas
- Mantener los workflows simples y modulares
- Utilizar caché para dependencias
- Configurar notificaciones adecuadas
- Revisar y actualizar regularmente las acciones
- Utilizar variables de entorno y secretos

Consejos avanzados

- Paralelización de pruebas
- Matriz de pruebas para múltiples entornos
- Integración con herramientas de análisis de código
- Despliegue automático tras pruebas exitosas
- Creación de acciones personalizadas

Recapitulación

- GitHub Actions ofrece una potente automatización
- Jest y Cypress son herramientas complementarias
- La integración mejora la calidad del código
- Facilita la implementación de CI/CD
- Aumenta la productividad del equipo

Recursos adicionales

- Documentación oficial de GitHub Actions
- Marketplace de GitHub Actions
- Documentación de Jest
- Documentación de Cypress
- Comunidades y foros de desarrolladores