

Configuracion y Ejecucion en CI-CD

Diagrama del Flujo CI/CD

```
graph TD; Trigger["Trigger (Push/Pull Request)"] --> Checkout["Checkout Code"]; Checkout --> Setup["Setup Node.js Environment"]; Setup --> Install["Install Dependencies"]; Install --> RunTests["Run Cypress Tests for UI"]; RunTests --> GenerateReport["Generate Test Report"]; GenerateReport --> DeployQA["Deploy to QA (if tests pass)"]; DeployQA --> DeployProd["Deploy to Production"];
```

github.yaml:

```
name: CI/CD Pipeline for Payment UI Tests

on:
  push:
    branches:
      - qa
  pull_request:
    branches:
      - qa

jobs:
  ui-tests:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
```

- ```
uses: actions/checkout@v3
```
- name: Set up Node.js

```
uses: actions/setup-node@v3
with:
 node-version: '18'
```
  - name: Install Cypress and dependencies

```
run: npm ci
```
  - name: Run Cypress Tests

```
uses: cypress-io/github-action@v5
with:
 config-file: cypress.json
 record: false
 browser: chrome
```
  - name: Generate Cypress Report

```
run: npm run generate-report
env:
 CI: true
```

#### deploy-qa:

```
needs: ui-tests
runs-on: ubuntu-latest
if: success()
```

#### steps:

- name: Checkout code

```
uses: actions/checkout@v3
```
- name: Deploy to QA

```
run: |
 # Replace with deployment commands
```

#### deploy-prod:

```
needs: deploy-qa
runs-on: ubuntu-latest
if: github.event_name == 'push'
```

#### steps:

- name: Checkout code

```
uses: actions/checkout@v3
```
- name: Deploy to Production

```
run: |
```

```
Replace with deployment commands
```

## Descripcion Detallada de las Etapas

### 1. Trigger:

- El pipeline se activad cuando se realiza un `push` or un `pull_request` en la rama de `qa`.
- Esto asegura que los cambios realizados en QA sean probados y validados antes de proceder al ambiente de Produccion.

### 2. Preparacion de Entorno:

- Node.js: Se configura con las version 18
- Dependencias:
  - La instalacion de las dependencias se realiza con el comando `npm ci`.
- Incluye las dependecias de Cypress para ejecutar las preubas automatizadas.

### 3. Ejecucion de Pruebas:

- Cypress ejecuta las pruebas de UI para validar el sistema de pagos.
- Las pruebas se aran en el navegador Chrome.
- La pruebas deben pasar con exito para permitir el avance del pipeline.

### 4. Generacion de Reportes:

- Se generan reportes de los resultados en formato como HTML y JSON.
- Son accesibles para el analisis.

### 5. Push a QA:

- Si todas la pruebas son exitosas, el pipeline realiza el pusho al entorno de QA.
- Al validar primero en QA, se valida no afectar produccion.

### 6. Push to Produccion:

- El push a Produccion es condicional y depende el la aprobacion Manual y el exito completo de todos los tests
- Proceso Final: Al finalizar la aprobacion, los cambios validados se implementan en produccion.