TEST AUTOMATION ENGINEER – FORMACIÓN INTEGRAL

{desafío} latam_

CURSO:

TEST AUTOMATION ENGINEER – FORMACIÓN INTEGRAL

- Módulo 1: FUNDAMENTOS DEL TESTING DE SOFTWARE
- Módulo 2: CONTROL DE VERSIONES Y ENTORNOS DE DESARROLLO
- Módulo 3: FUNDAMENTOS DE PROGRAMACIÓN APLICADOS AL TESTING (JAVASCRIPT)
- Módulo 4: CARACTERISTICAS DE CYPRESS – AUTOMATIZACIÓN WEB I
- Módulo 5: CARACTERISTICAS DE CYPRESS – AUTOMATIZACIÓN WEB II
- Módulo 6: PRUEBAS DE APIS CON POSTMAN Y SUPERTEST

{desafío} latam_

CURSO:

TEST AUTOMATION ENGINEER – FORMACIÓN INTEGRAL

- Módulo 7: HERRAMIENTAS DE PLAYWRIGHT Y PRUEBAS CON MÚLTIPLES NAVEGADORES
- Módulo 8: DESARROLLO GUIADO EN EL COMPORTAMIENTO (BDD) CON CUCUMBER.JS
- Módulo 9: HERRAMIENTAS DE AUTOMATIZACIÓN MÓVIL CON APPIUM
- Módulo 10: HERRAMIENTAS DE INTEGRACIÓN DE PRUEBAS EN CI/CD
- **Módulo 11: HERRAMIENTAS DE DOCKER, ENTORNOS VIRTUALIZADOS Y PRUEBAS EN LA NUBE**
- HERRAMIENTAS DE AUTOMATIZACIÓN DE UN FLUJO COMPLETO WEB + API + CI/CD

Te encuentras aquí

{desafío} latam_

Módulo 11: Herramientas de DOCKER, entornos virtualizados y pruebas en la nube.

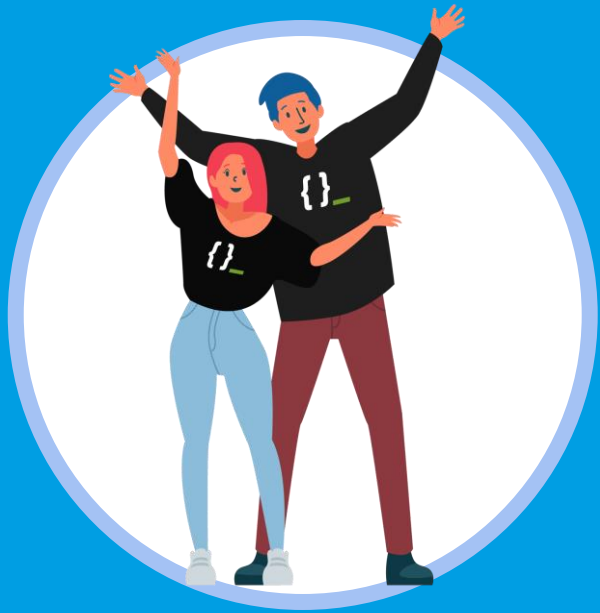
{desafío}
latam_

OBJETIVO ESPECÍFICO DEL MÓDULO

- UTILIZAR HERRAMIENTAS DE DOCKER, ENTORNOS VIRTUALIZADOS Y PRUEBAS EN LA NUBE, DE ACUERDO A LAS APLICACIONES WEB, MÓVILES Y APIS.

{desafío} latam_

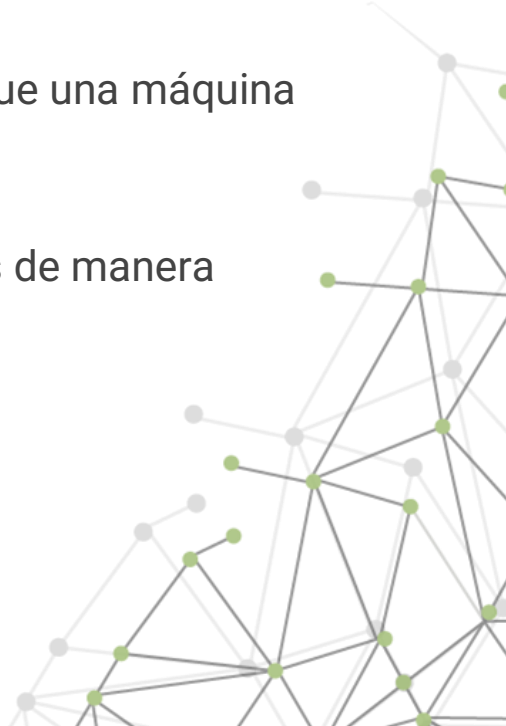
¿Qué beneficios imagina que puede ofrecer el uso de contenedores y entornos aislados en los procesos de prueba automatizada, especialmente en equipos de QA?



{desafío} latam_

¿QUÉ ES UN CONTENEDOR?

- Un contenedor es una unidad de software que agrupa código y sus dependencias en un solo paquete ejecutable.
- Se basa en virtualización a nivel de sistema operativo, más ligera que una máquina virtual.
- Herramientas como Docker permiten crear y ejecutar contenedores de manera consistente en cualquier entorno.

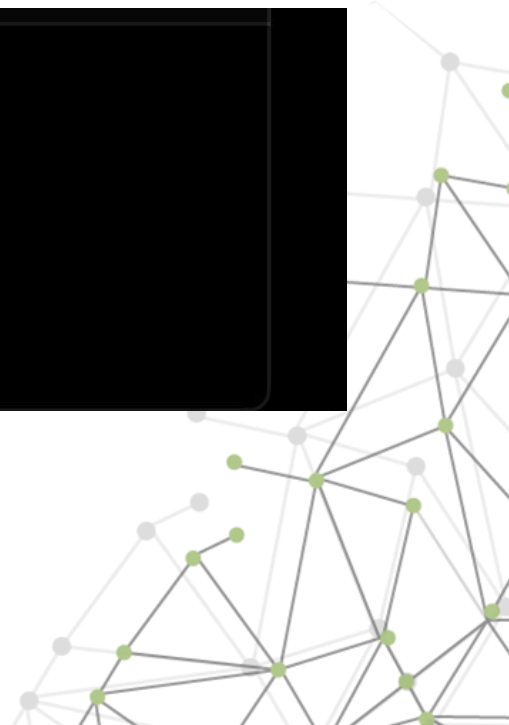


- **Beneficios para QA:**
- Entornos limpios y replicables
- Aislamiento de pruebas
- Rápido despliegue y rollback
- Ideal para integración continua

CREAR Y EJECUTAR IMÁGENES DE PRUEBAS CON DOCKER

- **Dockerfile:** define los pasos para construir una imagen personalizada

```
FROM node:18
WORKDIR /app
COPY . .
RUN npm install
CMD ["npm", "test"]
```



- **Comandos básicos:**
- docker build -t mi-test . → crea la imagen
- docker run mi-test → ejecuta el contenedor
- Permite correr pruebas unitarias, APIs, Cypress o Playwright en contenedores independientes del equipo local.

- Con Docker, todo el entorno de pruebas se puede compartir o ejecutar en servidores CI/CD sin errores de configuración.

AUTOMATIZACIÓN DE PRUEBAS DENTRO DE CONTENEDORES

- **Ejecución de pruebas automatizadas sin necesidad de configurar el entorno local:**
- Cypress con navegador en modo headless
- Postman usando Newman
- Jest o Supertest para APIs

- **Ejemplo:**

```
docker run --rm mi-proyecto-cypress
```

- Se puede conectar con otros contenedores usando docker-compose (ej: app + base de datos + runner de tests)

- Esta práctica garantiza coherencia en resultados, sin importar el sistema operativo o configuración del desarrollador o QA.

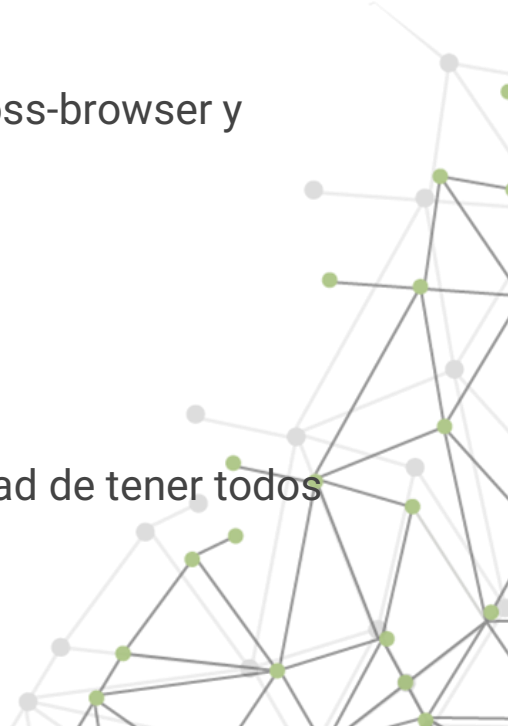
PLATAFORMAS CLOUD PARA TESTING DISTRIBUIDO

- **BrowserStack:** pruebas en tiempo real y automatizadas en cientos de navegadores/dispositivos reales.
- Compatible con Selenium, Cypress, Playwright.

- **Sauce Labs:** plataforma avanzada para pruebas automatizadas cross-browser y móviles.
- Integración con frameworks de QA y CI/CD.

- **AWS Device Farm:** pruebas en dispositivos físicos Android/iOS.
- Sube tu app o test script, y se ejecuta en hardware real.

- Estas plataformas permiten validar compatibilidad real sin necesidad de tener todos los dispositivos físicamente.



{desafío} latam_

QA EN ENTORNOS AISLADOS Y CONTROLADOS

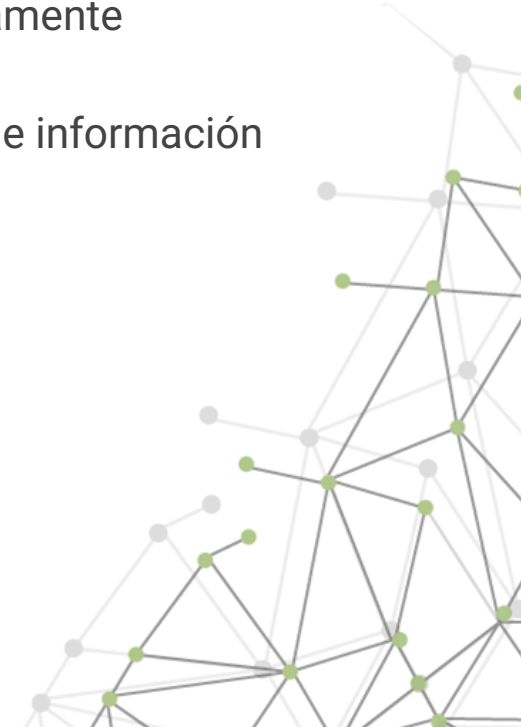
- **Entorno aislado:** espacio dedicado donde se ejecutan pruebas sin afectar otros procesos o datos reales.

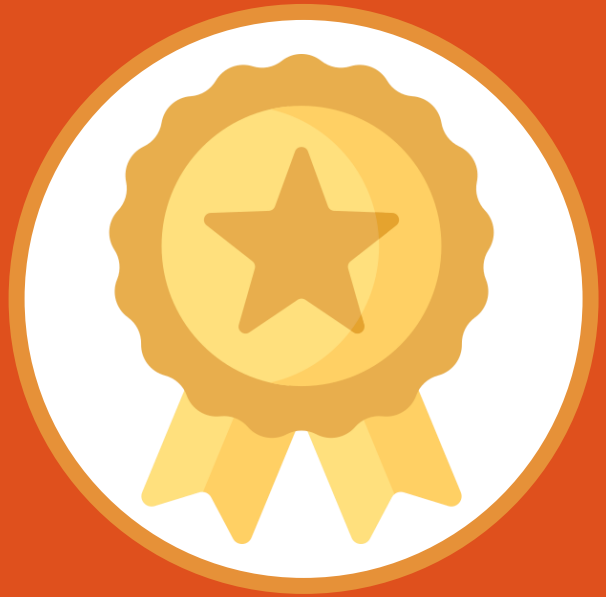
- **Formas de aislamiento:**
- Contenedores Docker
- Máquinas virtuales
- Ambientes cloud (staging, sandbox)

{desafío} latam_

- **Buenas prácticas:**
- Restablecer el estado entre pruebas
- Usar datos ficticios o anonimizados
- Separar ambientes QA de producción
- Integrar con CI/CD para levantar y destruir ambientes automáticamente

- Esto previene errores irreversibles, asegura repetibilidad, y protege información sensible durante pruebas.

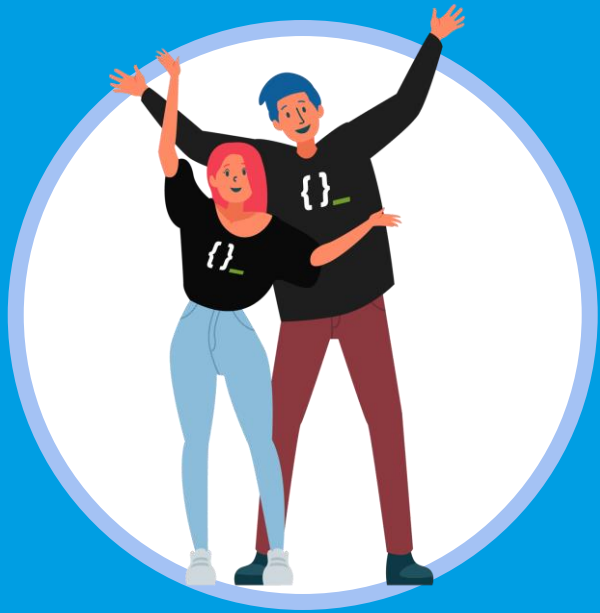




Recuerde que debe desarrollar los ejercicios prácticos del Módulo…

{desafío} latam_

¿Cómo contribuyen herramientas como Docker, BrowserStack o Device Farm a garantizar pruebas consistentes, reproducibles y escalables en distintos entornos, y qué criterios consideraría para elegir una solución u otra?



Éxito en la evaluación parcial y en la Prueba Final...

{desafío}
latam_

Academia de
talentos digitales