# Reto Automatización

## Objetivo

El finalizar el reto los analistas contaran con la habilidad de automatizar cualquier API REST o servicio, entendiendo de forma clara y objetiva el desarrollo que deben certificar, igualmente contaran con la habilidad de realizar las pruebas para diferentes ambientes a través de Azure Pipelines, manejando variables a través de las diferentes opciones de almacenamiento y de configuración que provee Azure.

## Herramientas de DevOps incluidas en el reto

Coding, Building, Testing, Packaging

## Roles:

En el marco del Reto se contarán con los siguientes Roles:

* Equipo de Calidad: parte del equipo solucionador
* Desarrollador
* Aprovisionador de infraestructura: soporte para el aprovisionamiento de infraestructura requerida por el equipo de QA y Desarrollador
* Soporte DevOps: soporte para los recursos desplegados sobre Azure DevOps
* Scrum/PO: define el requerimiento y aprueba los entregables.

## Definición del Reto

El grupo de analistas tendrá como recurso principal una definición de un requerimiento, por parejas, deberán definir escenarios, casos de prueba y recursos necesarios para asegurar el proceso de aseguramiento de calidad. La estrategia de prueba será definida por el equipo, teniendo en cuenta los tiempos de entrega y el alcance definido para las pruebas. Como parte del pilar de Pruebas Continuas, el equipo debe construir en Azure DevOps, Repositorios, Pipelines y demás recursos que se requieren para cumplir con esta tarea.

Para el aprovisionamiento de infraestructura y DevOps se contará con una persona para resolver dudas y solucionar problemas.

El Equipo debe reportar todas las inconsistencias y bugs encontrados en el aplicativo, debido a que se tomaran en cuenta los reportes de QA y la calidad de los hallazgos encontrados.

Al final del reto el equipo deberá exponer de forma clara las herramientas, metodologías y estrategias definidas para la solución.

Se realizará además una evaluación técnica del código construido, en base a los principios Solid y buenas practicas.

Tener en cuenta que el reto está definido para fomentar el autoestudio y la búsqueda de soluciones por parte del equipo.

**Importante**: el consumo de los servicios puede cambiar durante el sprint, la automatización debe estar construida de tal forma que el impacto por estos cambios sea el mínimo.

## Restricciones:

El tiempo de entrega será definido al momento de presentar el reto.

Las personas involucradas en la evaluación serán definidas al momento de presentar el reto.

## Definición del requerimiento

## 

### Prerrequisitos:

* Maven/Gradle
* Java JDK, JRE
* IDE (Eclipse, Intellij…)
* Herramienta de consumo de servicios (Postman, Soap UI, Powershell)
* Git Bash
* Se recomienda además el uso de Visual Studio Code con el complemento Yaml y Open Open Api Swagger Editor.
* Se pueden usar herramientas adicionales de acuerdo a la estrategia definida por el equipo.

### Especificación.

Para el área de mercadeo se requiere desarrollar un api tipo CRUD que administre las operaciones sobre la base de datos de Clientes, esto con el fin de agregar una capa de seguridad sobre el proceso y segmentar responsabilidades dentro del flujo de administración de clientes. El equipo de calidad debe asegurarse que el desarrollo cuente con las funciones descritas y además cumpla con el contrato definido, debido a que este api debe ser expuesta para el consumo de un tercero.

Cada operación en el servicio de clientes debe almacenar el log de trazabilidad de la función consumida.

Dado que la compañía se encuentra en la transformación a DevOps, el desarrollo debe pasar por los dos ambientes que se tienen contemplados en el momento (pruebas, producción), antes del paso a producción se deben asegurar las funciones del desarrollo a través de las pruebas automatizadas.

## Puntos adicionales.

Mínimo de errores en SonarLint.

Usar la estrategia de Todo-Como-Código para la definición de pipelines en Azure DevOps.

Encontrar los defectos en seguridad del aplicativo.