

**UNIVERSIDAD DEL VALLE DE GUATEMALA**  
**FACULTAD DE INGENIERÍA**  
**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION**  
**ADMINISTRACION Y MANTENIMIENTO DE SISTEMAS**



**Análisis y Diseño de una Solución en la Nube**

**Pablo Coutiño 18817**

**Contexto:** se contrata a una empresa para provea el servicio de mercaderistas a los distintos puntos de venta de un cliente al que se le venden productos de consumo masivo. Parte de las responsabilidades que se le asignan en una visita a una persona pueden ser : acomodo de productos, reporte de inventarios, reporte de actividades promocionales, toma de fotos de ejecuciones con geolocalización, cálculos de espacios en la tienda, reporte de check in y check out, entre otras. Estos puntos de venta se visitan todas las semanas al menos 1 vez, dependiendo de la importancia de la tienda se asignan el numero de horas semanales que se da servicio.

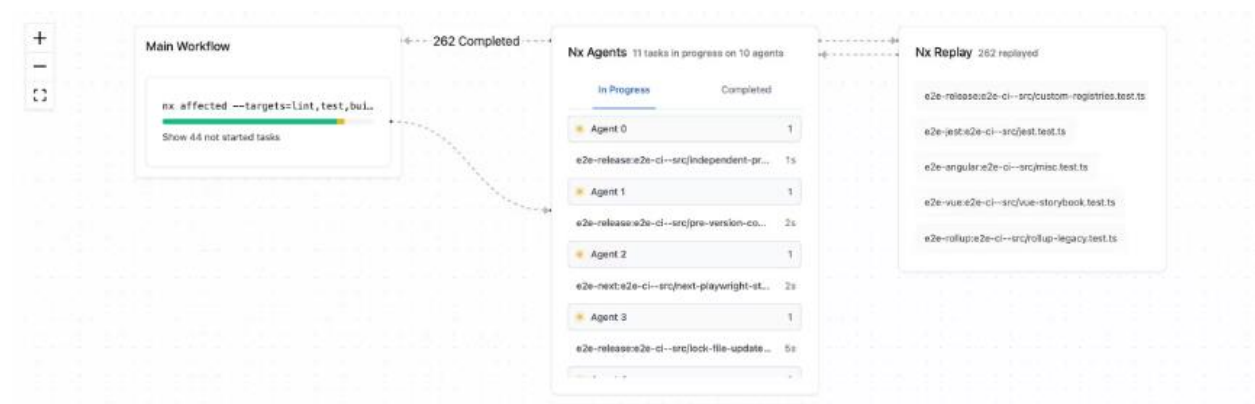
El proyecto en el que se trabajara el diseño y la propuesta es una aplicación para reporteria de las visitas de los mercaderías a los puntos de ventas en React native.

## 1 y 2. Flujo de Integración Continua y Despliegue continuo

Para este proyecto Configurare en github actions Nx cloud porque haría eficiente la integración con su uso de cache remota, de manera que para cada cambio realizado, revisa si ya ha ejecutado las acciones antes o no, y si ya se ha realizado esta acción, entonces utiliza los artefactos de la corrida anterior.

La ventaja de usar esta opción es que con la configuración implementada, se pueden agregar más agentes para que escale adecuadamente con su distribución inteligente.

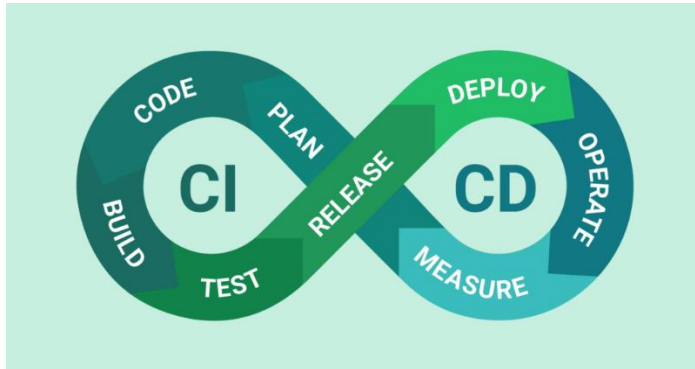
En el reporte del bot de Nx cloud podemos revisar los errores que se generaron al realizar alguna acción o incluso revisar código con interdependencias o código redundante.



### 3. Pruebas Automatizadas

Se deben probar los cambios realizados previo a subirse a la, para esto podemos hacer uso de git hooks, y en la configuración del .git/hooks

También es necesario en este proyecto configurar snyk como parte de las github actions para reducir los riesgos que pueden presentar y planear en las siguientes implementaciones que pruebas automatizadas hay que configurar para saber si ya se ha logrado corregir el error o no.



Estas pruebas se deben realizar sobre todos los requisitos funcionales .

### 4. Gestión de la Configuración.

Dentro de uno de los hooks necesarios en este proyecto es la protección de las contraseñas maestras para evitar que se suba esta información a

Dentro de la lista de palabras a detectar en los commit Diff debería venir contraseñas , usuarios, variables de entorno o información que pueda representar un riesgo.

y también se me debería notificar como el administrador, que por descuido o malicia se estaba quemando una contraseña o información sensible en el commit.

Se puede configurar esta notificación por correo electrónico.

## 5. Gestión de Versionamiento

Previo a subir los cambios en componentes, se deben realizar pruebas unitarias del código. Para esto sería bueno utilizar la librería de Jest , que debería venir por defecto en la configuración del proyecto.



```
{
  "scripts": {
    "test": "jest"
  },
  "jest": {
    "preset": "react-native"
  }
}
```

```
__tests__/Intro-test.js

import React from 'react';
import renderer from 'react-test-renderer';
import Intro from '../Intro';

test('renders correctly', () => {
  const tree = renderer.create(<Intro />).toJSON();
  expect(tree).toMatchSnapshot();
});
```

Esto nos debería generar un snapshot con el cual el equipo puede determinar los errores previos a empujar los cambios al repositorio.

Otra opción para asegurar que los cambios cumplan con estructuras recomendadas y

## 6. Gestión de la Seguridad (DevSecOps)

En Visual studio Code se puede instalar la extensión de codeql, para hacer revisión de vulnerabilidades con el motor de análisis de código, para poder exponer vulnerabilidades o malas prácticas en el código generado. Es importante que se haga revisiones constantes a lo largo de todo el proceso de , con este acercamiento, todo el equipo es responsable y se compromete a mantener una buena calidad de código y que los errores se puedan detectar tempranamente.

## **7. Gestión Financiera (FinOps)**

En este proyecto si es necesario una implementación del marco de trabajo de FinOps ya que la cantidad de recursos a emplear en la nube puede ser variable por estacionalidades o por horas laborales, entonces la comunicación de cuantas instancias se deben utilizar para gestionar el trafico debe ir de la mano de una conversación con la data a analizar del trafico hacia el backend.

Es necesario cuantificar el valor que aporta tener o no un servicio en la nube en distintos momentos del día, para poder reservar presupuestos, forecastear trafico y planear ventanas de mantenimiento también.

---

### **1. Cuáles serán las ventajas obtenidas?**

Tener un flujo de integración y despliegue continuo agiliza la validación de que los cambios producidos sean limpios y no destruyan nada por incompatibilidad, también se obtiene la ventaja de que todos los cambios tendrán estándares rigurosos en los que no se publicara código con vulnerabilidades o riesgos de divulgación de información sensible. Aprovechar las acciones de github también nos permitirá revisar para todas las contribuciones que realice validaciones automáticas.

También puedes definir y preparar los ambientes para que sean compatibles con la versión específica de la tecnología que estas usando.

### **2. Cuáles fueron los retos más complicados?**

Una de las cosas más difíciles en cualquier proyecto es el habito humano, las personas en general tienen dificultad rompiendo hábitos o creando nuevos, en este contexto, la configuración de seguridad y la implementación de practicas seguras en todo el proceso de desarrollo continuo será el reto más grande.

Manejar las dependencias continuamente, para evaluar los riesgos que pueden presentar también es una practica que se debe adoptar y se debe tener mente abierta para no “casarse” con una tecnología o con una herramienta.

### **3. Cuáles serían sus propuestas para mejorar el resultado?**

En general, empezar con investigar cuales son los workflows más recomendados para la combinación de tecnologías que se usaran para un determinado proyecto.

Dependiendo de la escala, validar que financieramente haga sentido la administración de los recursos, posiblemente deshabilitar maquinas en tiempos de baja demanda.