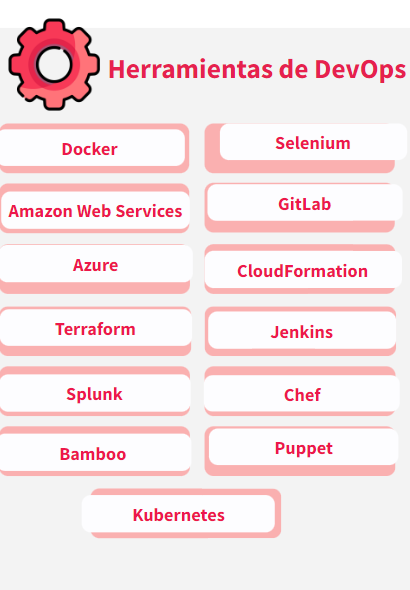
Infraestructura II

SEMANA 1

DevOps: Es la combinación de filosofías, prácticas y herramientas que incrementan la velocidad a la que una organización entrega aplicaciones y servicios, permitiendo mejorar los productos a un ritmo más rápido que las organizaciones que usen procesos de desarrollo e infraestructura tradicionales. Esta velocidad le permite a las organizaciones entregar más valor a sus clientes y ser más competitivas en el mercado.

En la cultura DevOps, los equipos de desarrollo (Dev) y operaciones (Ops) no trabajan por separado, sino que se comunican de forma constante. A veces los equipos se fusionan en un único equipo que trabaja en el ciclo de vida completo de la aplicación, desde el desarrollo y las pruebas hasta el despliegue y la operación. En algunos casos, la calidad y la seguridad también se integran con el desarrollo y las operaciones.

Estos equipos hacen foco en automatizar procesos que históricamente fueron manuales y lentos. Para eso, utilizan un stack de tecnologías y herramientas que los ayuda a operar y evolucionar aplicaciones de forma rápida y confiable. Además, colaboran a que una sola persona pueda realizar tareas que normalmente requerirían de una combinación de varias personas, como desplegar código o aprovisionar infraestructura.



PERFILES DevOps:

* Desarrollador
* Analista de calidad
* Analista de infraestructura
* Analista de redes
* Analista de seguridad
* Analista de CI/CD
* Arquitecto de nube
* Ingeniero de confiabilidad de sitio (SRE)
* Gerente de entregas

EXPECTATIVAS DevOps:

* Automatizar
* Medir y monitorear
* Reducir la fricción
* Aumentar la comunicación
* Ser dueños de la solución

CI/CD:

Integración continua(CI): Integrar a main frecuentemente.

Despliegue continuo(CD): Desplegar a producción frecuentemente.

MONITOREO:

Monitorear servidores: Disponibilidad, carga.

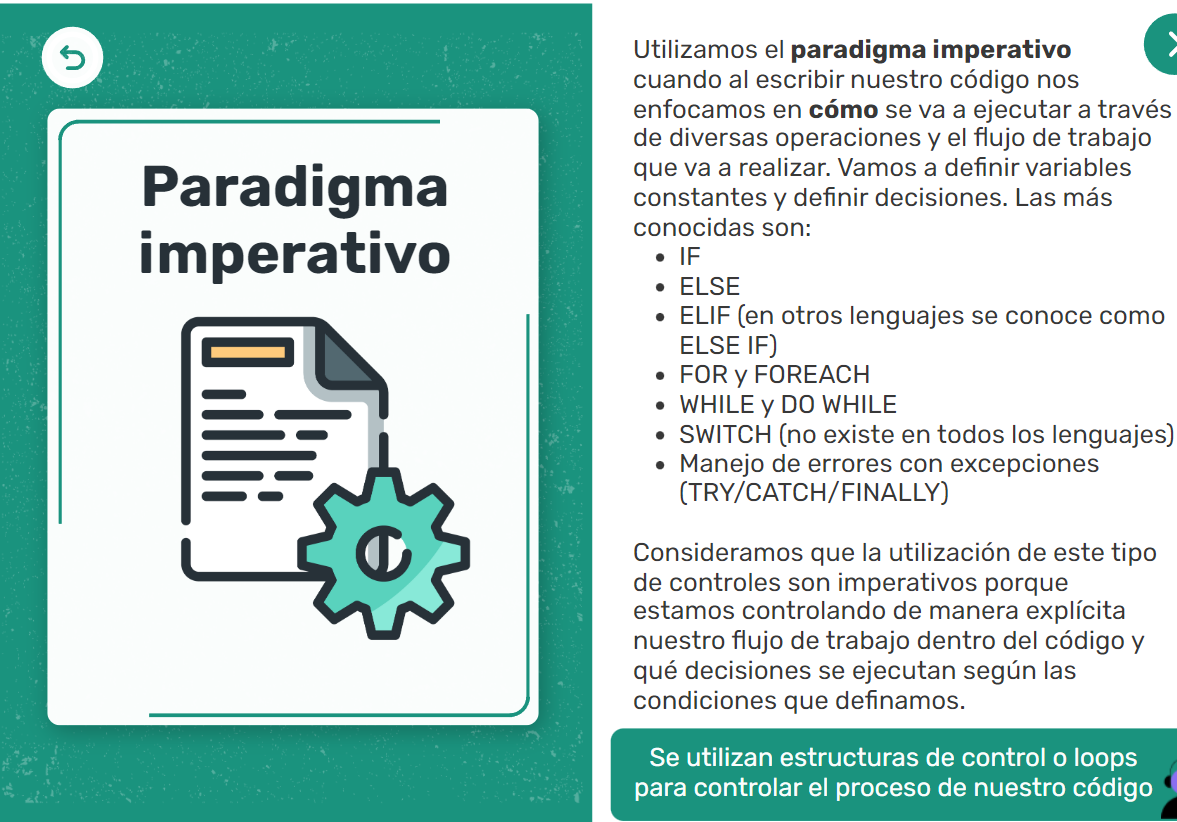
Monitorear aplicaciones: Performance, experiencia de usuario.

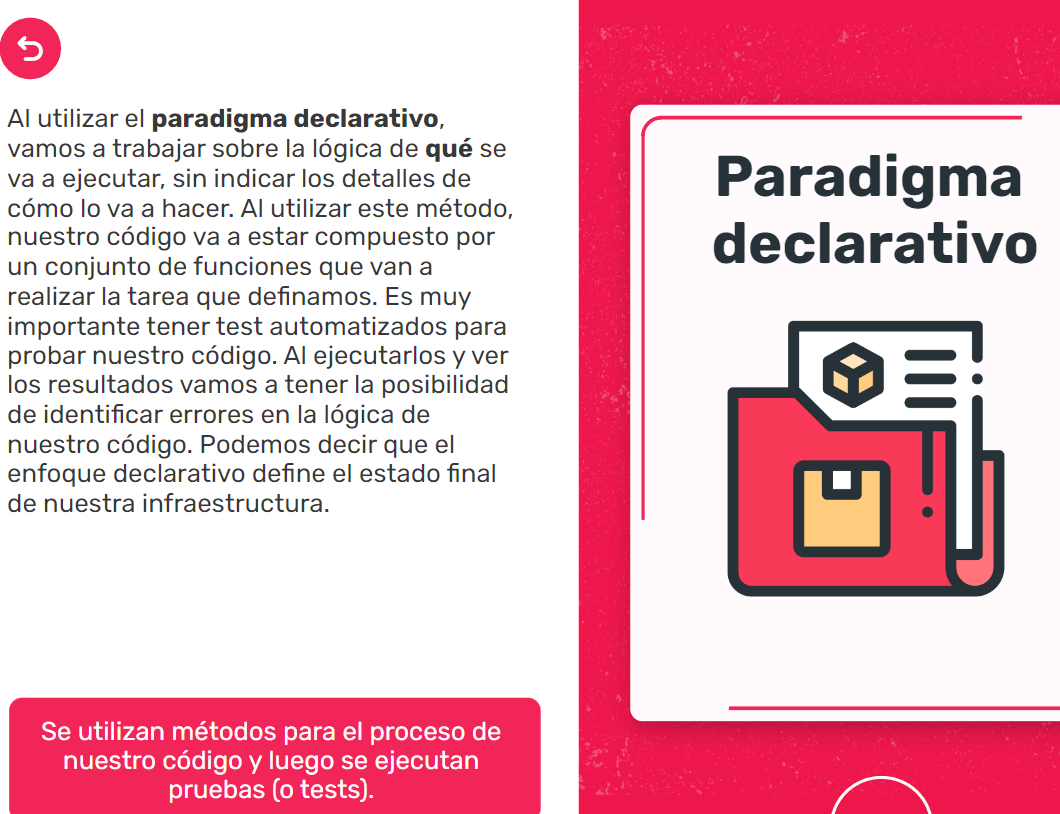
SEMANA 2

INFRAESTRUCTURA COMO CODIGO (IaC): Gestión de la infraestructura a través de templates que tienen la capacidad de ser versionados. De esta forma vamos a poder automatizar los procesos manuales que se requieren para lograr el objetivo final que buscamos.

Al momento de realizar los análisis para implementar una infraestructura para nuestra aplicación, lo primero a definir es la arquitectura que necesitamos: qué servidor es el adecuado o qué base de datos necesitamos. Una vez seleccionado el tipo de servidor, avanzamos en la configuración e instalación de aquello que nuestro sistema operativo requiere para estar operativo.

PARADIGMAS DE PROGRAMACION APLICADOS A LA INFRA COMO CODIGO





ECOSISTEMA DE HERRAMIENTAS IaC

**TERRAFORM**

Terraform es un software de código libre desarrollado por HashiCrop. Es una herramienta declarativa de aprovisionamiento y orquestación de infraestructura que permite automatizar el aprovisionamiento de todos los aspectos de la infraestructura, tanto para la nube como la infraestructura on-premise (en los mismos datacenter). Tiene algunas características interesantes, como comprobar el estado de la infraestructura antes de aplicar los cambios. Es la herramienta más popular porque es compatible con todos los proveedores de nube sin realizar modificaciones en nuestros templates.

**AWS CLOUDFORMATION**

AWS CloudFormation es la solución nativa de AWS para aprovisionar recursos en esta nube. En este caso se pueden definir templates en formato JSON o YAML. Se pueden utilizar para crear, actualizar y eliminar recursos las veces que sea necesario. Una ventaja de CloudFormation es que, al ser un servicio propio de Amazon, tiene una integración completa con los demás servicios de AWS, por lo que es nuestra mejor opción si solo utilizamos este proveedor de nube.

**AZURE RESOURCE MANAGER**

ARM es la herramienta nativa en Azure para implementar infraestructura como código, Azure Resource Manager (ARM Templates). Estas plantillas llevan una sintaxis declarativa en formato JSON, que nos permiten definir los recursos y las propiedades que conforman la infraestructura.

**GOOGLE CLOUD DEPLOYMENT MANAGER**

Google Cloud Deployment Manager es la herramienta IaC para la plataforma Google Cloud —lo mismo que CloudFormation es para AWS—. Con esta herramienta los usuarios de Google pueden administrar fácilmente mediante archivos de configuración YAML.

**ANSIBLEANSIBLE**

Ansible es una herramienta de automatización de infraestructuras creada por Red Hat. Ansible modela nuestra infraestructura describiendo cómo se relacionan sus componentes y el sistema entre sí, en lugar de gestionar los sistemas de forma independiente.

INFRAESTRUCTURA COMO CODIGO EN AWS (CloudFormation):

CloudFormation crea y configura la infraestructura que definimos previamente en una plantilla (o template) de acuerdo a los requisitos que necesitamos. Esto nos ofrece algunas ventajas, como crear repositorios con nuestros templates para que sean accesibles o que se puedan realizar entregas rápidas de los recursos de infraestructura.

CONCEPTOS IMPORTANTES:

-Plantillas o templates: es un archivo de texto con formato JSON (JavaScript Object Notation) que describe los recursos que queremos crear junto a sus propiedades.

-Pilas: es una unidad que genera CloudFormation para la creación ordenada de los recursos.

-Cambios: es un resumen de los cambios que se proponen para anticiparnos al resultado final.



LINKS UTILES PARA CLOUDFORMATION:

<https://docs.aws.amazon.com/es_es/AWSCloudFormation/latest/UserGuide/sample-templates-applications-us-west-1.html>

<https://s3.us-west-1.amazonaws.com/cloudformation-templates-us-west-1/WordPress_Single_Instance.template>