|  |  |  |
| --- | --- | --- |
| **Nombre**: Kevin Francisco Garza Rodriguez | | **Matricula**: 02887531 |
| **Nombre del curso:** Fundamentos DevOps | **Nombre del profesor**: Prof. Carlos Morales | |
| **Actividad:** Actividad 01 | **Fecha**: 28/10/2024 | |

La empresa Axanet es muy popular y CNC la considera una de las mejores proveedoras de servicios de manufactura. La compañía guarda los datos de sus clientes en un archivo, sin embargo, ha solicitado tus servicios para desarrollar una solución basada en Linux, ya que busca generar los archivos de los nuevos clientes y, a su vez, revisar la información de los antiguos.

La empresa ha tenido un crecimiento exponencial en los últimos años, a tal grado que triplica el número de clientes cada mes, pero también mantiene el servicio que ofrece a los más antiguos. La atención de la empresa funciona de la siguiente manera: si una persona o negocio solicita el servicio, se verifica si se trata de un cliente nuevo o de uno recurrente. En el primer caso, se genera un nuevo archivo con la información de la persona y con la descripción del servicio solicitado; en el segundo escenario, se busca el archivo ya elaborado y únicamente se agrega la descripción de la nueva solicitud. Cualquier usuario del sistema debe ser capaz de acceder al archivo de un cliente, ya sea introduciendo su nombre directamente o buscándolo en una lista que los contiene a todos.

1. **Diseña los microservicios necesarios que requiere esta empresa; para ello, básate en una arquitectura de microservicios y utiliza la metodología de diseño determinado por dominios (DDD) como fundamento.**

**Gestion de Cliente.**

* Endpoints:
  + /nuevoCliente
  + /buscarCliente

**Gestion de Servicios**

* Endpoints:
  + /nuevoServicio
  + /historialServicios.

**Gestion de Archivos**

* Endpoints:
  + /crearArchivoCliente
  + actualizarArchivoCliente

**Consulta de Clientes**

* **/**listarClientes
* /verArchivocliente

1. **Define e identifica las entidades que participan en el proceso para planificar los microservicios que puedes llegar a necesitar; en este caso, asegúrate de que las entidades tengan un contexto definido, atributos y métodos (acciones que pueden afectarlas).**

**Nota**. Recuerda que solo la entidad puede acceder y modificar los datos, mientras que las otras deben acceder a ellos a través de invocaciones a los métodos, es decir, por medio de una API (*Application Program Interfaz*). También debes considerar que las entidades tienen una identidad única y pueden ser invocadas en otros microservicios.

* 1. **Cliente**
* Atributos:
  + id\_cliente
  + nombre
  + contacto
  + fehcaRegistro
  + historialServicios
* Métodos
  + consultar\_historial();
  + registrar\_servicio();
  1. **Servicio**
* Atributos:
  + id\_servicio
  + tipoServicio
  + fechaSolicitud
* **Métodos:**
  + consultar\_detalle()

1. **Define e identifica los objetos de valor que deben incluirse en el proceso; con ello, tendrás un panorama más claro de los microservicios que necesitas utilizar.**

**Nota**. Recuerda que los objetos de valor no tienen identidad propia y, por ende, solo adquieren sentido dentro del contexto de un microservicio.

* 1. **ArchivoCliente:**
* Contiene la informacion del cliente y su historial de servicios
* No tiene identidad propia y su función es complementar al cliente como un recurso de almacenamiento
  1. **ListaClientes:**
* Representa la lista de clientes existentes en el sistema.
* Facilita la búsqueda de clientes para su consulta o actualización de la informacion.

1. **Define e identifica los agregados en el modelo para planificar la administración de los procesos.**

**Nota.** Recuerda que los agregados son colecciones que aglutinan varias entidades alrededor de un elemento raíz.

* **ClienteAgregado.**
  + Funciona para administrar los datos de cada cliente y el historial de servicios en un archivo individual.
  + Es responsable de ordenar las acciones que afectan tanto a la información el cliente como a la adición de los nuevos servicios.

1. **Aplica la filosofía DevOps para planificar la implementación y la mejora continua de la aplicación; de esta manera, tendrás una imagen clara de la estructura de tu proyecto.**
   1. **Automatización de Despliegue**
      * Uso de contenedored de Docker
      * Integraicon con Gitlab
   2. **Infraestructura como Código**
      * Uso de herramientas para automatizar la configuración de la infraestructura.
   3. **Monitoreo**
      * Integración con herramientas de monitoreo.
   4. **Pruebas**
      * Automatización de pruebas
   5. **Escalabilidad**
      * Implementación de Kubernetes para gestionar escalabilidad.
2. **Genera un *script* en Bash (video) que permita visualizar y modificar el contenido del archivo de un cliente, o bien, crear uno para un nuevo comprador. No olvides explicar tu código de manera precisa para una mejor comprensión y, además, asegúrate de aplicar la planeación generada en los puntos anteriores.**