**Solución del reto técnico**

**Desafíos**

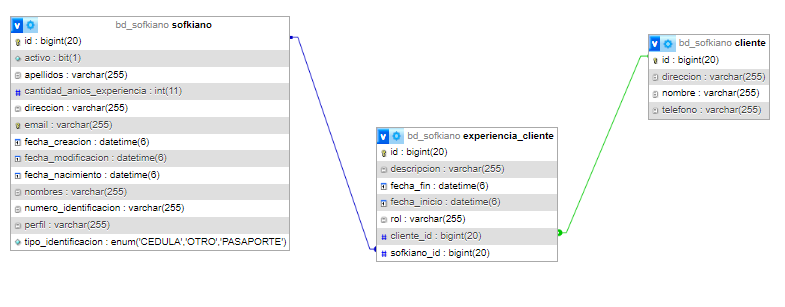
**Nivel 1:**

Se realiza Backend en java 17 con el Framework SpringBoot y Base de Datos MySql (En AWS Aurora) y DynamocDB, se Uso Apache Camel y Rabbit MQ para la integración de los Microservicios se usa el estilo de Arquitectura Clean Code Hexagonal o de puertos y adaptadores.

**Se crearon Microservicios:**

1. **Microsofkiano**: Realiza el Crud de Sofkiano y el Crud de Clientes, no fue solicitado en las HU pero lo hice para facilitar la asociación de los Sofkianos con los Clientes, se asocian Sofkianos a un Cliente para crear la experiencia (Ingresos), y se Desasocian Sofkianos del Cliente para culminar la experiencia (Egresos). Al momento de Asociar a un Sofkiano con un Clliente se genera la experiencia con la fecha de inicio y el rol que desempeñara, y al momento de desasociar un Sofkiano del Cliente, se genera la fecha fin de la experiencia y una descripción de la misma.

**A continuación el Modelo Entidad Relación**



1. **Microestadisiticas**: Lee los datos en la Cola Los Almacena en Dynamoc para generar las estadísticas solicitadas. La estructura del documento Dynamoc es:

**[**

**{**

**"sofkianoId": "string",**

**"nombre": "string",**

**"clientesIngreso": [**

**{**

**"clienteId": "string",**

**"nombreCliente": "string",**

**"fecha": "string"**

**}**

**],**

**"clientesEgreso": [**

**{**

**"clienteId": "string",**

**"nombreCliente": "string",**

**"fecha": "string"**

**}**

**]**

**}**

**]**

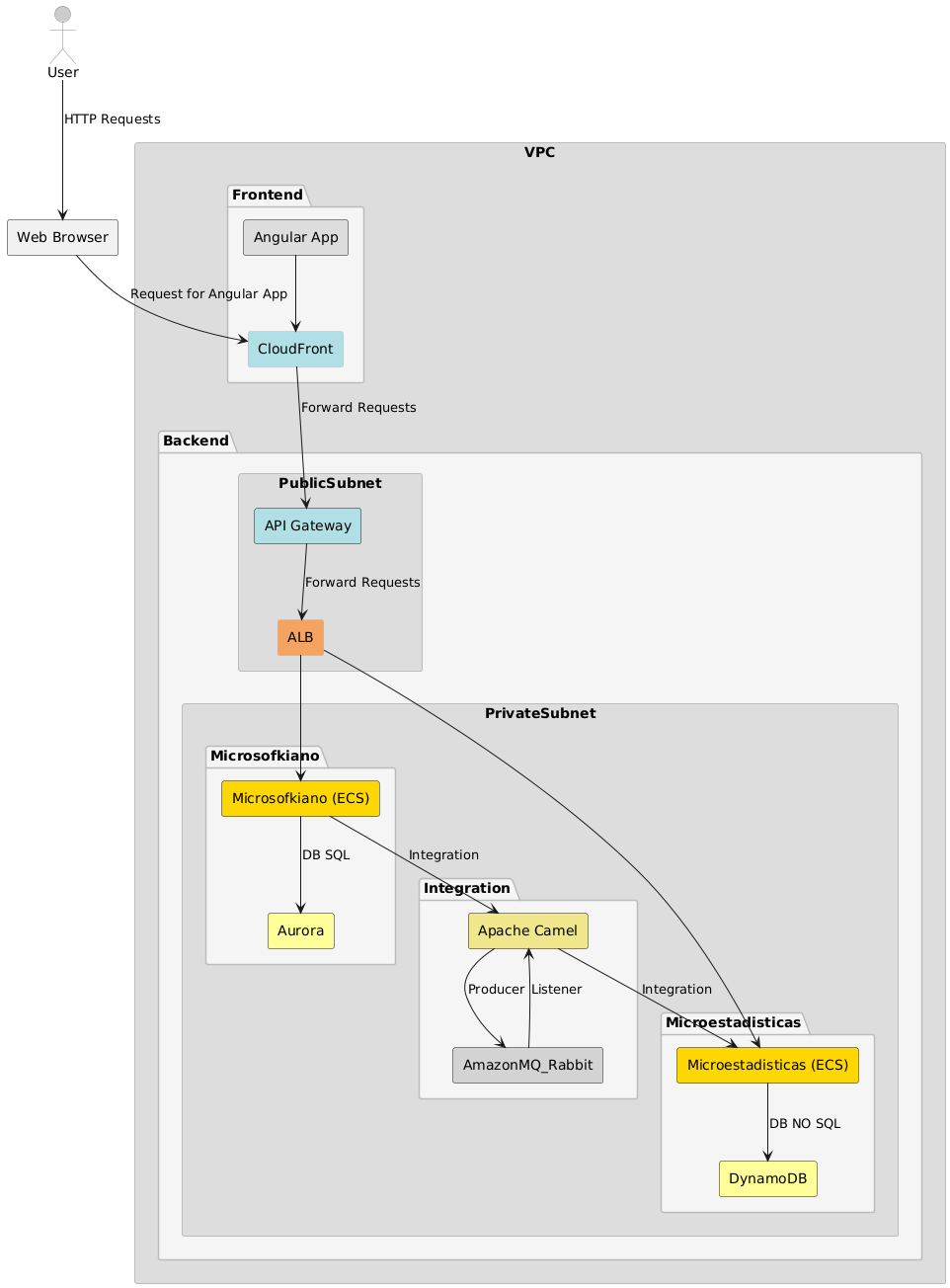
Donde se logra visualizar por sofkiano a que clientes Ingresó y asi como los de Egresó

Se utiliza Api Gateway de AWS, para la redirección de las peticiones del Fontend a los microservicios solicitados.

**Nivel 2**: se crean las plantillas de cloud formation para el despliegue de la arquitectura siguiente:

**Despliegue en AWS:**

**Arquitectura**



**Plantillas Cloud Formation**

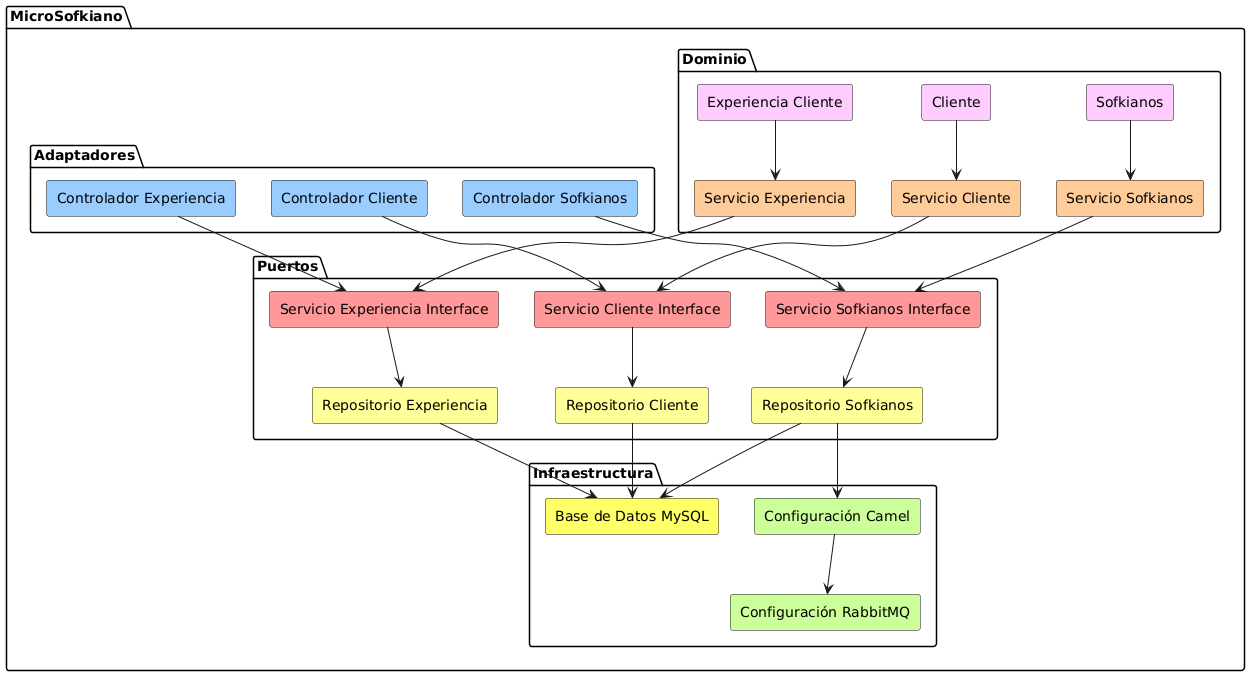
|  |  |
| --- | --- |
| Despliegue | Plantilla Cloud Formation |
| Configuración de VPC y Sub Redes, grupos de seguridad, enrutamiento y ALB | AWSTemplateFormatVersion: '2024-09-21'  Description: Plantilla para desplegar la arquitectura de Microservicios en AWS  Parameters:  VpcCIDR:  Type: String  Default: "10.0.0.0/16"  Description: CIDR block para la VPC  PublicSubnetCIDR:  Type: String  Default: "10.0.1.0/24"  Description: CIDR block para la Subred Pública  PrivateSubnetCIDR:  Type: String  Default: "10.0.2.0/24"  Description: CIDR block para la Subred Privada  Resources:  MyVPC:  Type: 'AWS::EC2::VPC'  Properties:  CidrBlock: !Ref VpcCIDR  EnableDnsSupport: true  EnableDnsHostnames: true  Tags:  - Key: Name  Value: MyVPC  # Subred Pública  PublicSubnet:  Type: 'AWS::EC2::Subnet'  Properties:  VpcId: !Ref MyVPC  CidrBlock: !Ref PublicSubnetCIDR  MapPublicIpOnLaunch: true  AvailabilityZone: !Select [0, !GetAZs '']  Tags:  - Key: Name  Value: PublicSubnet  # Subred Privada  PrivateSubnet:  Type: 'AWS::EC2::Subnet'  Properties:  VpcId: !Ref MyVPC  CidrBlock: !Ref PrivateSubnetCIDR  AvailabilityZone: !Select [0, !GetAZs '']  Tags:  - Key: Name  Value: PrivateSubnet  # Internet Gateway  InternetGateway:  Type: 'AWS::EC2::InternetGateway'  Properties:  Tags:  - Key: Name  Value: MyInternetGateway  AttachGateway:  Type: 'AWS::EC2::VPCGatewayAttachment'  Properties:  VpcId: !Ref MyVPC  InternetGatewayId: !Ref InternetGateway  NATGateway:  Type: AWS::EC2::NatGateway  Properties:  AllocationId: !GetAtt EIP.AllocationId  SubnetId: !Ref PublicSubnet  # Tabla de Rutas para Subred Pública  PublicRouteTable:  Type: 'AWS::EC2::RouteTable'  Properties:  VpcId: !Ref MyVPC  Tags:  - Key: Name  Value: PublicRouteTable  PublicRoute:  Type: 'AWS::EC2::Route'  Properties:  RouteTableId: !Ref PublicRouteTable  DestinationCidrBlock: '0.0.0.0/0'  GatewayId: !Ref InternetGateway  PublicSubnetRouteTableAssociation:  Type: 'AWS::EC2::SubnetRouteTableAssociation'  Properties:  SubnetId: !Ref PublicSubnet  RouteTableId: !Ref PublicRouteTable  # ALB  MyLoadBalancer:  Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  Properties:  Name: MyALB  Subnets:  - !Ref PublicSubnet  SecurityGroups:  - !Ref ALBSecurityGroup  Scheme: internet-facing  ALBSecurityGroup:  Type: 'AWS::EC2::SecurityGroup'  Properties:  GroupDescription: Allow HTTP and HTTPS traffic  VpcId: !Ref MyVPC  SecurityGroupIngress:  - IpProtocol: tcp  FromPort: '80'  ToPort: '80'  CidrIp: '0.0.0.0/0'  - IpProtocol: tcp  FromPort: '443'  ToPort: '443'  CidrIp: '0.0.0.0/0'  # Grupo de destino para el ALB  MyTargetGroup:  Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  Properties:  Name: MyTargetGroup  Port: 8080  Protocol: HTTP  VpcId: !Ref MyVPC  TargetType: ip  MyLoadBalancerListener:  Type: 'AWS::ElasticLoadBalancingV2::Listener'  Properties:  LoadBalancerArn: !Ref MyLoadBalancer  Port: 80  Protocol: HTTP  DefaultActions:  - Type: forward  TargetGroupArn: !Ref MyTargetGroup |
| Configuración de Api Gateway | ApiGateway:  Type: 'AWS::ApiGateway::RestApi'  Properties:  Name: MyApiGateway  Description: API Gateway para Microservicios  # Recursos y métodos para Sofkiano  ApiGatewayResourceSofkiano:  Type: 'AWS::ApiGateway::Resource'  Properties:  ParentId: !GetAtt ApiGateway.RootResourceId  RestApiId: !Ref ApiGateway  PathPart: sofkianos  ApiGatewayMethodGetAllSofkianos:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: GET  ResourceId: !Ref ApiGatewayResourceSofkiano  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: GET  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/sofkianos"  ApiGatewayMethodCreateSofkiano:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: POST  ResourceId: !Ref ApiGatewayResourceSofkiano  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: POST  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/sofkianos"  ApiGatewayMethodUpdateSofkiano:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: PUT  ResourceId: !Ref ApiGatewayResourceSofkiano  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: PUT  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/sofkianos"  ApiGatewayMethodDeleteSofkiano:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: DELETE  ResourceId: !Ref ApiGatewayResourceSofkiano  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: DELETE  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/sofkianos"  # Recurso para Excel  ApiGatewayResourceExcel:  Type: 'AWS::ApiGateway::Resource'  Properties:  ParentId: !GetAtt ApiGateway.RootResourceId  RestApiId: !Ref ApiGateway  PathPart: excel  ApiGatewayMethodUploadExcel:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: POST  ResourceId: !Ref ApiGatewayResourceExcel  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: POST  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/excel"  # Recurso para Cliente  ApiGatewayResourceCliente:  Type: 'AWS::ApiGateway::Resource'  Properties:  ParentId: !GetAtt ApiGateway.RootResourceId  RestApiId: !Ref ApiGateway  PathPart: clientes  ApiGatewayMethodGetAllClientes:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: GET  ResourceId: !Ref ApiGatewayResourceCliente  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: GET  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/clientes"  ApiGatewayMethodCreateCliente:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: POST  ResourceId: !Ref ApiGatewayResourceCliente  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: POST  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/clientes"  ApiGatewayMethodUpdateCliente:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: PUT  ResourceId: !Ref ApiGatewayResourceCliente  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: PUT  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/clientes"  ApiGatewayMethodDeleteCliente:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: DELETE  ResourceId: !Ref ApiGatewayResourceCliente  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: DELETE  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/clientes"  # Recurso para Estadísticas  ApiGatewayResourceEstadisticas:  Type: 'AWS::ApiGateway::Resource'  Properties:  ParentId: !GetAtt ApiGateway.RootResourceId  RestApiId: !Ref ApiGateway  PathPart: estadisticas  ApiGatewayMethodGetEstadisticas:  Type: 'AWS::ApiGateway::Method'  Properties:  HttpMethod: GET  ResourceId: !Ref ApiGatewayResourceEstadisticas  RestApiId: !Ref ApiGateway  AuthorizationType: NONE  Integration:  IntegrationHttpMethod: GET  Type: AWS\_PROXY  Uri: !Sub "http://${MyLoadBalancer.DNSName}/estadisticas" |
| RDS para la Base de Datos | AuroraDBCluster:  Type: AWS::RDS::DBCluster  Properties:  Engine: aurora-mysql  MasterUsername: admin  MasterUserPassword: mypassword  DBSubnetGroupName: !Ref PrivateSubnet  VpcSecurityGroupIds:  - !Ref DBSecurityGroup  BackupRetentionPeriod: 7    DBSecurityGroup:  Type: 'AWS::EC2::SecurityGroup'  Properties:  GroupDescription: Allow access to the database  VpcId: !Ref MyVPC  SecurityGroupIngress:  - IpProtocol: tcp  FromPort: '3306'  ToPort: '3306'  CidrIp: '0.0.0.0/0' |
| Dynamoc DB | EstadisticaCambioClienteTable:  Type: 'AWS::DynamoDB::Table'  Properties:  TableName: EstadisticaCambioCliente  AttributeDefinitions:  - AttributeName: sofkianoId  AttributeType: S  KeySchema:  - AttributeName: sofkianoId  KeyType: HASH  ProvisionedThroughput:  ReadCapacityUnits: 5  WriteCapacityUnits: 5  BillingMode: PAY\_PER\_REQUEST  Outputs:  EstadisticaCambioClienteTableName:  Description: "Nombre de la tabla DynamoDB"  Value: !Ref EstadisticaCambioClienteTable |
| ECR para Microservicios, Cluster ECS | EcrRepositorySofkianos:  Type: 'AWS::ECR::Repository'  Properties:  RepositoryName: sofkianos  EcrRepositoryEstadisticas:  Type: 'AWS::ECR::Repository'  Properties:  RepositoryName: estadisticas  EcrRepositoryFrontend:  Type: 'AWS::ECR::Repository'  Properties:  RepositoryName: frontend    MyECSCluster:  Type: 'AWS::ECS::Cluster'  Properties:  ClusterName: MicroserviciosCluster  # Definición de Tarea para Sofkianos  TaskDefinitionSofkiano:  Type: 'AWS::ECS::TaskDefinition'  Properties:  Family: SofkianosService  ContainerDefinitions:  - Name: SofkianosContainer  Image: !Sub "${EcrRepository}.dkr.ecr.${AWS::Region}.amazonaws.com/sofkianos:latest"  Memory: 512  Cpu: 256  Essential: true  PortMappings:  - ContainerPort: 7080  HostPort: 7080  Environment:  - Name: AURORA\_DB\_ENDPOINT  Value: !GetAtt AuroraDBCluster.Endpoint.Address # Endpoint de Aurora  # Definición del servicio para Sofkianos  SofkianosService:  Type: 'AWS::ECS::Service'  Properties:  Cluster: !Ref EcsCluster  DesiredCount: 2  LaunchType: FARGATE  TaskDefinition: !Ref TaskDefinitionSofkianos  NetworkConfiguration:  AwsvpcConfiguration:  AssignPublicIp: ENABLED  Subnets:  - !Ref PublicSubnet1  - !Ref PublicSubnet2  SecurityGroups:  - !Ref EcsSecurityGroup  # Definición de Tarea para Estadísticas  TaskDefinitionEstadisticasSofkianos:  Type: 'AWS::ECS::TaskDefinition'  Properties:  Family: EstadisticasService  ContainerDefinitions:  - Name: EstadisticasContainer  Image: !Sub "${EcrRepository}.dkr.ecr.${AWS::Region}.amazonaws.com/estadisticas:latest"  Memory: 512  Cpu: 256  Essential: true  PortMappings:  - ContainerPort: 7081  HostPort: 7081  RequiresCompatibilities:  - FARGATE  NetworkMode: awsvpc  Cpu: "256"  Memory: "512"  ExecutionRoleArn: !Ref EcsTaskExecutionRole  TaskRoleArn: !Ref EcsTaskRole  Environment:  - Name: DYNAMODB\_TABLE\_NAME  Value: EstadisticaCambioCliente # Nombre de la tabla de DynamoDB  # Definición del servicio para Estadísticas  EstadisticasService:  Type: 'AWS::ECS::Service'  Properties:  Cluster: !Ref EcsCluster  DesiredCount: 2  LaunchType: FARGATE  TaskDefinition: !Ref TaskDefinitionEstadisticas  NetworkConfiguration:  AwsvpcConfiguration:  AssignPublicIp: ENABLED  Subnets:  - !Ref PublicSubnet1  - !Ref PublicSubnet2  SecurityGroups:  - !Ref EcsSecurityGroup  Outputs:  ApiUrl:  Description: URL del API Gateway  Value: !Sub "https://${ApiGateway}.execute-api.${AWS::Region}.amazonaws.com/prod/"  LoadBalancerURL:  Description: URL del Application Load Balancer  Value: !GetAtt MyLoadBalancer.DNSName |
| Roles y permisos | MicroservicioRole:  Type: 'AWS::IAM::Role'  Properties:  AssumeRolePolicyDocument:  Version: '2024-09-21'  Statement:  - Effect: Allow  Principal:  Service: ecs-tasks.amazonaws.com  Action: sts:AssumeRole  Policies:  - PolicyName: MicroservicioPolicy  PolicyDocument:  Version: '2024-09-21'  Statement:  - Effect: Allow  Action:  - rds:DescribeDBInstances  - rds:Connect  Resource: "\*"    - Effect: Allow  Action:  - dynamodb:PutItem  - dynamodb:GetItem  - dynamodb:UpdateItem  - dynamodb:DeleteItem  - dynamodb:BatchWriteItem  - dynamodb:BatchGetItem  - dynamodb:Query  - dynamodb:Scan  Resource: arn:aws:dynamodb:${AWS::Region}:${AWS::AccountId}:table/EstadisticaCambioCliente |
| Broker | RabbitMQBroker:  Type: AWS::AmazonMQ::Broker  Properties:  AutoMinorVersionUpgrade: true  BrokerName: RabbitMQBroker  DeploymentMode: SINGLE\_INSTANCE  EngineType: RABBITMQ  EngineVersion: "3.8.6"  HostInstanceType: mq.t3.micro  PubliclyAccessible: false  SubnetIds:  - !Ref PrivateSubnet  Users:  - Username: rabbitUser  Password: rabbitPassword |

Entregables.

● Diseño técnico de la arquitectura del software para dar soporte a las decisiones

tomadas en el proceso.

Adicional a la Arquitectura de AWS, Muestro Diagrama de Arquitectura de los componentes de software para microsofkianos:



El digrama de microservicio estadisticas es analogo.

● Código Fuente (En repositorio git):

https://github.com/milymoreno/sofka.git

● Instrucciones de cómo ejecutar el proyecto (En README del proyecto).

● Swagger del proyecto.

Crud de Sofkianos



Jon para crear Sofkiano:

{

"id": 0,

"tipoIdentificacion": "CEDULA",

"numeroIdentificacion": "string",

"nombres": "string",

"apellidos": "string",

"direccion": "string",

"activo": true,

"email": "string",

"perfil": "string",

"cantidadAniosExperiencia": 0,

"fechaNacimiento": "2024-09-24T08:53:33.794Z",

"fechaCreacion": "2024-09-24T08:53:33.794Z",

"fechaModificacion": "2024-09-24T08:53:33.794Z"

}

Listados de Sofkianos con sus experiencias:

[

{

"id": 46,

"sofkiano": {

"id": 1,

"tipoIdentificacion": "CEDULA",

"numeroIdentificacion": "1020846521",

"nombres": "Mildred Maria",

"apellidos": "Moreno Liscano",

"direccion": "bello, Antioquia",

"activo": true,

"email": "mily@gmail.com",

"perfil": "arquitecto",

"cantidadAniosExperiencia": 10,

"fechaNacimiento": "1980-01-03T14:13:21",

"fechaCreacion": "2024-09-20T09:28:51",

"fechaModificacion": "2024-09-20T14:13:21"

},

"cliente": {

"id": 1,

"nombre": "Seguros la previsora",

"direccion": "Medellin",

"telefono": "3044123136"

},

"fechaInicio": "2024-09-24T02:41:19",

"fechaFin": "2024-09-24T02:58:59",

"rol": "QA",

"descripcion": "fffffffff"

},

{

"id": 47,

"sofkiano": {

"id": 1,

"tipoIdentificacion": "CEDULA",

"numeroIdentificacion": "1020846521",

"nombres": "Mildred Maria",

"apellidos": "Moreno Liscano",

"direccion": "bello, Antioquia",

"activo": true,

"email": "mily@gmail.com",

"perfil": "arquitecto",

"cantidadAniosExperiencia": 10,

"fechaNacimiento": "1980-01-03T14:13:21",

"fechaCreacion": "2024-09-20T09:28:51",

"fechaModificacion": "2024-09-20T14:13:21"

},

"cliente": {

"id": 2,

"nombre": "Colombina",

"direccion": "zona industrial Bogota",

"telefono": "3138904348"

},

"fechaInicio": "2024-09-24T02:41:33",

"fechaFin": "2024-09-24T02:59:05",

"rol": "QA manual",

"descripcion": "fffffffff"

},

{

"id": 48,

"sofkiano": {

"id": 2,

"tipoIdentificacion": "CEDULA",

"numeroIdentificacion": "3529299",

"nombres": "Maria Esperanza",

"apellidos": "Liscano",

"direccion": "bello, Antioquia",

"activo": true,

"email": "eperanza@gmail.com",

"perfil": "desarrollador",

"cantidadAniosExperiencia": 15,

"fechaNacimiento": "1951-03-25T14:13:21",

"fechaCreacion": "2024-09-20T10:45:30",

"fechaModificacion": "2024-09-20T14:13:21"

},

"cliente": {

"id": 2,

"nombre": "Colombina",

"direccion": "zona industrial Bogota",

"telefono": "3138904348"

},

"fechaInicio": "2024-09-24T02:41:48",

"fechaFin": null,

"rol": "lider",

"descripcion": null

},

{

"id": 49,

"sofkiano": {

"id": 2,

"tipoIdentificacion": "CEDULA",

"numeroIdentificacion": "3529299",

"nombres": "Maria Esperanza",

"apellidos": "Liscano",

"direccion": "bello, Antioquia",

"activo": true,

"email": "eperanza@gmail.com",

"perfil": "desarrollador",

"cantidadAniosExperiencia": 15,

"fechaNacimiento": "1951-03-25T14:13:21",

"fechaCreacion": "2024-09-20T10:45:30",

"fechaModificacion": "2024-09-20T14:13:21"

},

"cliente": {

"id": 1,

"nombre": "Seguros la previsora",

"direccion": "Medellin",

"telefono": "3044123136"

},

"fechaInicio": "2024-09-24T02:42:07",

"fechaFin": null,

"rol": "Arquitecto",

"descripcion": null

},

{

"id": 50,

"sofkiano": {

"id": 3,

"tipoIdentificacion": "CEDULA",

"numeroIdentificacion": "14695408",

"nombres": "Arina Eliette",

"apellidos": "Mendoza Moreno",

"direccion": "Bello, Antioquia, Colombia",

"activo": true,

"email": "ariana.mendoza@gmail.com",

"perfil": "Analista QA",

"cantidadAniosExperiencia": 5,

"fechaNacimiento": "2017-11-21T16:24:56",

"fechaCreacion": "2024-09-23T11:27:16",

"fechaModificacion": "2024-09-23T16:24:56"

},

"cliente": {

"id": 2,

"nombre": "Colombina",

"direccion": "zona industrial Bogota",

"telefono": "3138904348"

},

"fechaInicio": "2024-09-24T02:59:36",

"fechaFin": null,

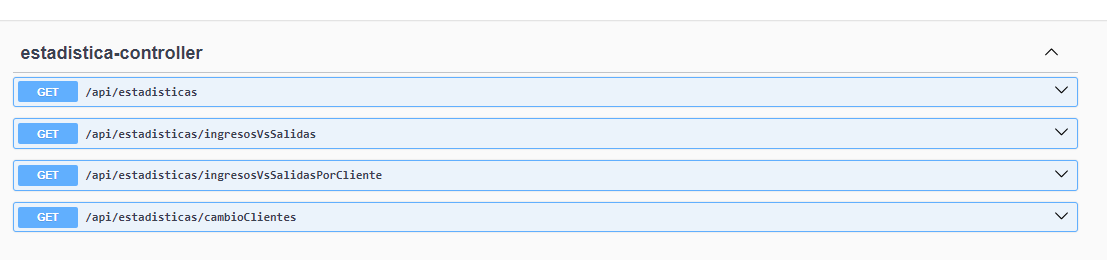
"rol": "rol",

"descripcion": null

}

]

Microestadistica:



Para Cambio de cliente por rango de fechas se muestra:

[

{

"sofkianoId": "string",

"nombreSofkiano": "string",

"clientesEgreso": [

{

"clienteId": "string",

"nombreCliente": "string",

"fecha": "2024-09-24T09:17:20.799Z"

}

],

"clientesIngreso": [

{

"clienteId": "string",

"nombreCliente": "string",

"fecha": "2024-09-24T09:17:20.799Z"

}

]

}

]

**Se Considera Cambio de Cliente, si en el rango de fechas Egreso de un cliente, es decir termino su experiencia laboral e Ingresa a otro.**

● Colección de Postman con ejemplo de consumo de las apis.

No creo que sean necesarias, no use Apis Externas, y los json están en los swagers

● Evidencia de Code coverage.

Microsofkiano

Microestadisticas: 