

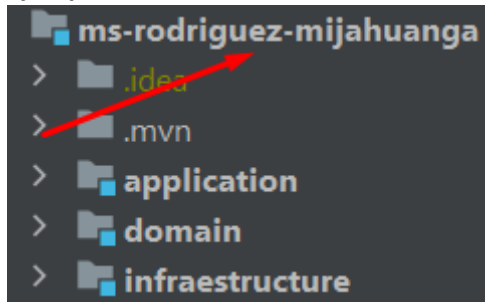
Examen G4

Yo como usuario requiero la creación de un api, para realizar el mantenimiento de empresas y personas asociadas a la empresa. Adicional a ello requiero automatizar el proceso de registro con tan solo brindar el número de ruc de la empresa poder registrar los datos de está consultando con los datos en sunat. Finalmente, también requiero agilizar el tiempo de respuesta de las consultas para cuando busque una empresa o persona por id, esto utilizando almacenamiento en cache por un tiempo de 10 min.

Consideraciones:

1. El proyecto que se genere debe seguir la siguiente estructura para el nombre: **ms-apellidoPaterno-ApellidoMaterno**

Ejemplo:



2. Para el arquetipo del api a construir, por disposición del arquitecto de software recomienda que se utilice el arquetipo **Hexagonal modular (ver imagen anterior)**
3. Para el registro de empresas deberá consultar al siguiente endPoint:

- <https://api.apis.net.pe/v2/sunat/ruc?numero=10720320083>

Nota: Recuerde que debe generar su token, en caso no pueda, puede usar el siguiente token: apis-token-8325.t-IC0AzPK3f6DagMhJl5obD-DIF49oB2

Recomendación: genere su propio token.

Ejemplo de Respuesta del endPoint:

```
{
  "razonSocial": "RODRIGUEZ MIJAHUANGA PAUL NIKE",
  "tipoDocumento": "6",
  "numeroDocumento": "10720320083",
  "estado": "ACTIVO",
  "condicion": "HABIDO",
  "direccion": "-",
  "ubigeo": "-",
  "viaTipo": "-",
  "viaNombre": "-",
  "zonaCodigo": "-",
  "zonaTipo": "-",
  "numero": "-",
  "interior": "-",
  "lote": "-",
  "dpto": "-",
  "manzana": "-",
```

```

    "kilometro": "-",
    "distrito": "",
    "provincia": "",
    "departamento": "",
    "EsAgenteRetencion": false
  }

```

4. Para el registro de una persona deberá consultar al endPoint:

- <https://api.apis.net.pe/v2/reniec/dni?numero=72032002>

Nota: Recuerde que debe generar su token, en caso no pueda puede usar el siguiente token: apis-token-8325.t-IC0AzPK3f6DagMhJl5obD-DIF49oB2

Recomendación: genere su propio token.

Ejemplo de Respuesta del EndPoint:

```

{
  "nombres": "SARA MARA",
  "apellidoPaterno": "ZAMBACHE",
  "apellidoMaterno": "CAÑO HUA",
  "tipoDocumento": "1",
  "numeroDocumento": "72032003",
  "digitoVerificador": ""
}

```

5. Parámetros de entrada para métodos crearEmpresa, debe considerar tener como request tipoDocumento y numero de ruc

Ejemplo:

```

{
  "tipoDoc": "RUC",
  "numDoc": "10720320083"
}

```

6. Parámetros de entrada para métodos crearPersona, debe considerar tener como request tipoDocumento, numero de dni y el ruc de la empresa al que se va asociar.

Ejemplo:

```

{
  "tipoDoc": "DNI",
  "numDoc": "72032003",
  "empresa": "10720320083"
}

```

7. Para la mejora del tiempo de respuesta de sus endPoints que buscan por ID debe usar redis:

- Debe iniciar su servidor local de redis:

```
C:\Users\Nike Rodriguez>redis-server
[1080] 01 May 13:01:15.996 # Warning: no config file specified, using the default config. In order to specify a config
file use redis-server /path/to/redis.conf

Redis 3.0.504 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 1080

http://redis.io

[1080] 01 May 13:01:16.002 # Server started, Redis version 3.0.504
[1080] 01 May 13:01:16.011 * DB loaded from disk: 0.008 seconds
[1080] 01 May 13:01:16.013 * The server is now ready to accept connections on port 6379
```

- Luego debe configurar su proyecto para que se comunice con redis:

```
spring.data.redis.host=localhost
spring.data.redis.port=6379
```

-Debe generar su service para poder interactuar con redis:

Ejemplo:

```
@Service
@RequiredArgsConstructor
public class RedisService {
    private final StringRedisTemplate stringRedisTemplate;

    1 usage  NikeRodriguez
    public void saveInRedis(String key, String value, int exp){
        stringRedisTemplate.opsForValue().set(key, value);
        stringRedisTemplate.expire(key, exp, TimeUnit.MINUTES);
    }

    1 usage  NikeRodriguez
    public String getFromRedis(String key){
        return stringRedisTemplate.opsForValue().get(key);
    }

    no usages NikeRodriguez
    public void deleteKey(String key) { stringRedisTemplate.delete(key); }
}
```

Por último debe aplicar la lógica para buscar antes en redis y luego en Base de datos.
No olvide la clase Útil para la transformación de objetos a string y viceversa.

8. Importante tener en cuenta las auditorias para cada endPoint(crear, actualizar, eliminar)
9. Como base de datos debe usar la siguiente:

```
-- Creación de la tabla empresa_info
```

```

CREATE TABLE empresa_info (
    id SERIAL PRIMARY KEY,
    razonSocial VARCHAR(255) NOT NULL,
    tipoDocumento VARCHAR(5) NOT NULL,
    numeroDocumento VARCHAR(20) UNIQUE NOT NULL,
    estado INT NOT NULL,
    condicion VARCHAR(50) NOT NULL,
    direccion TEXT,
    distrito VARCHAR(100),
    provincia VARCHAR(100),
    departamento VARCHAR(100),
    EsAgenteRetencion BOOLEAN NOT NULL DEFAULT false,
    usuaCrea VARCHAR(255) NOT NULL,
    dateCreate TIMESTAMP WITH TIME ZONE NOT NULL,
    usuaModif VARCHAR(255),
    dateModif TIMESTAMP WITH TIME ZONE,
    usuaDelet VARCHAR(255),
    dateDelet TIMESTAMP WITH TIME ZONE
);

-- Creación de la tabla persona con una columna para la clave foránea
CREATE TABLE persona (
    id SERIAL PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    apellido VARCHAR(255) NOT NULL,
    tipoDocumento VARCHAR(5) NOT NULL,
    numeroDocumento VARCHAR(20) UNIQUE NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    telefono VARCHAR(15),
    direccion TEXT,
    estado INT NOT NULL,
    usuaCrea VARCHAR(255) NOT NULL,
    dateCreate TIMESTAMP WITH TIME ZONE NOT NULL,
    usuaModif VARCHAR(255),
    dateModif TIMESTAMP WITH TIME ZONE,
    usuaDelet VARCHAR(255),
    dateDelet TIMESTAMP WITH TIME ZONE,
    empresa_id INT NOT NULL,
    CONSTRAINT fk_empresa FOREIGN KEY (empresa_id) REFERENCES
    empresa_info(id)
);

```

10. Debe configurar sus entidades de manera relacionada.

PersonaEntity

```

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "empresa_id", nullable = false)
private Empresa empresa;

```

Indispensable:

- Creación de los 10 endPoints (3 puntos):

Empresa

- crearEmpresa
- buscarId
- buscar todos
- actualizar
- eliminar

Persona:

- crearPersona
- buscarId
- buscar todos
- actualizar
- eliminar

- Uso de auditoria (3 puntos)
- Uso de Redis (3 puntos)
- Uso de feignClient (3 puntos)
- Uso de arquetipo Hexagonal modular (3 puntos)
- Estructura nombre proyecto (1 punto)
- Entrega de examen de acuerdo con lo acordado en clase (1 punto)
- Plus de algo nuevo que este aplicando en su examen (3 puntos)