

DOCUMENTACIÓN PROYECTO APPCLEVER

a- Definición general:

Se desarrolló la primera entrega de la app solicitada por Clever Business. En la misma se crearon los servicios principales para lograr el MVP del proyecto, siendo el mismo, registrar los empleados con sus respectivos ingresos y egresos en las distintas sucursales, y poder hacer consultas básicas a la base de datos.

b- Procedimientos de instalación del proyecto:

El proyecto fue desarrollado con el lenguaje Java y su framework Spring Boot, utilizando como base de datos relacional MySQL, y fue documentado mediante Swagger. Para poder poner en funcionamiento el proyecto, es necesario tener establecida la conexión entre el IDE a utilizar (NetBeans) y MySQL, y tener instalado Apache Maven.

Una vez clonado el repositorio y creada la base de datos, se deberán instalar las dependencias (Build with Dependencies) y luego levantar el proyecto (spring-boot run). Por último, accedemos a <http://localhost:8080/swagger-ui/> mediante el navegador, para poder hacer las pruebas necesarias.

c- Descripción de Servicios Solicitados:

RegisterService

1- SERVICE 1 - REGISTER

```
public Register register (Integer idEmployee, LocalDate date, RegisterType registerType, BusinessLocation businessLocation)
```

Este servicio, se encarga de registrar el Ingreso o Egreso de un empleado, guardando la fecha y sucursal del evento. Recibe por parámetros el ID del empleado, la fecha (formato AAAA-MM-DD), el tipo de registro (Ingreso o Egreso) y la sucursal.

2- SERVICE 2 – SEARCH REGISTER

```
public List<Register> findByDateBetweenWithFilter (LocalDate dateFrom,
LocalDate dateTo, String descriptionFilter, BusinessLocation businessLocation)
```

Este servicio, se encarga de listar los Ingresos y Egresos dada una fecha, pudiendo filtrar la búsqueda por sucursal y nombre o apellido. Recibe como parámetros una fecha inicial y un límite (formato AAAA-MM-DD), un Sting para poder filtrar nombres o apellidos, y por último la sucursal.

3- SERVICE 3 – AVERAGE

```
public Map<String, Double> averageByLocation (LocalDate dateFrom, LocalDate
dateTo, BusinessLocation businessLocation)
```

Este servicio, calcula el promedio de Hombres y Mujeres que ingresan o egresan en determinada fecha, por cada sucursal. Recibe como parámetros una fecha inicial y un límite (formato AAAA-MM-DD), y por último la sucursal que queremos filtrar.

4- LIST REGISTER

```
public List<Register> listRegister ()
```

Lista el total de registros existentes.

5- FIND REGISTER BY ID

```
public Register findById (Integer id)
```

Busca mediante el ID, un registro determinado.

EmployeeService

1- REGISTER EMPLOYEE

```
public Employee register (String firstName, String lastName, Gender gender)
```

Este servicio registra un nuevo empleado. Recibe como parámetros, el nombre, apellido y género.

2- LIST EMPLOYEE

```
public ArrayList<Employee> listEmployees ()
```

Lista el total de empleados existentes.

3- SEARCH EMPLOYEE BY ID

```
public Employee searchById (Integer id)
```

Busca mediante el ID, un empleado determinado.