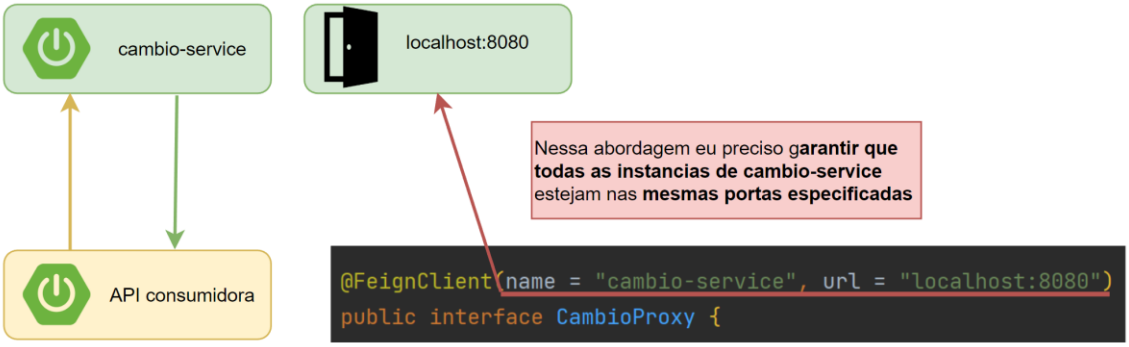
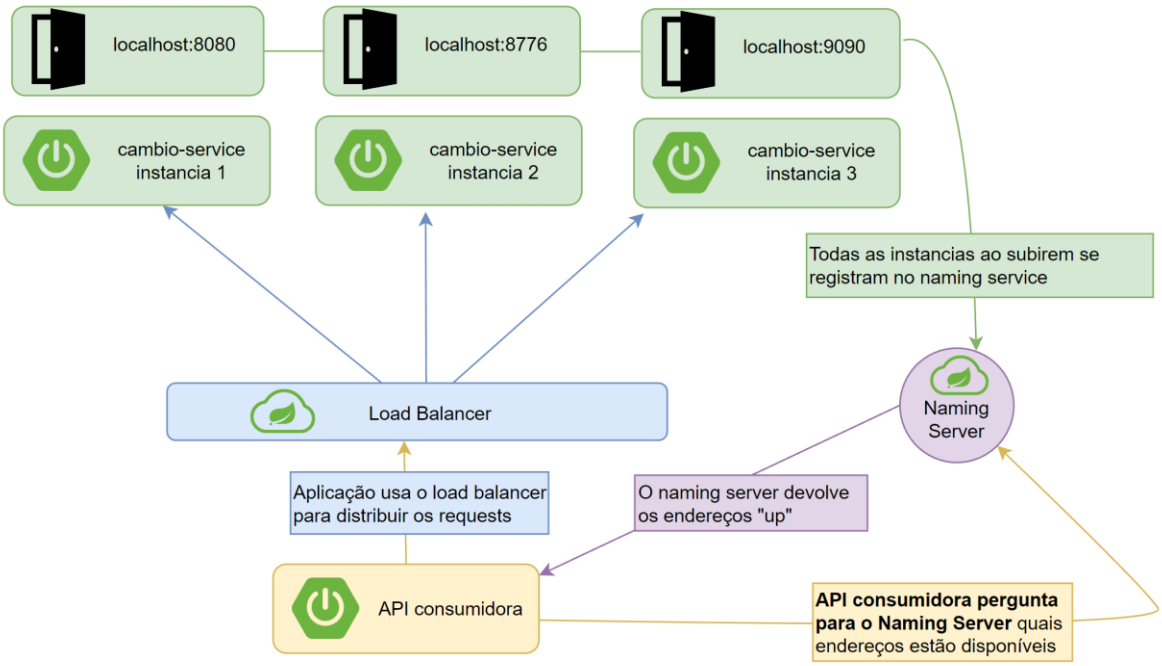
Consumindo Outras API's - **Inteligentemente**

Um dos problemas no consumo de API's é o uso do ***"hard code"*** para se referenciar ao endereço de um **micro serviço**. Seguindo essa abordagem eu **sempre vou me referenciar ao mesmo IP** e tenho que torcer para os **micros serviços** estarem up por lá:

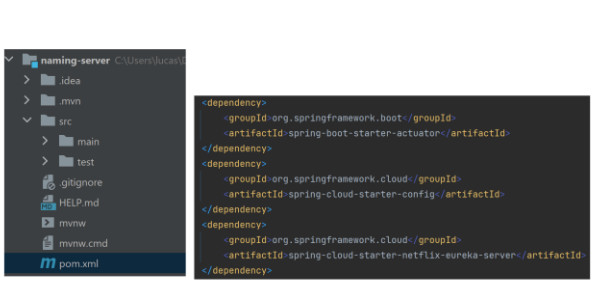


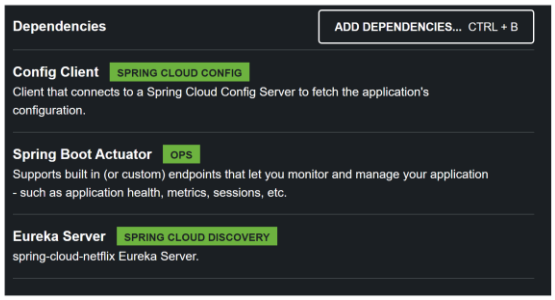
Pensando no cenário onde micro serviços podem **ter várias instâncias em vários servidores**, instâncias essas que "morrem e nascem" diversas vezes, essa abordagem é completamente ineficiente. O caminho mais "correto" é fazer isso dinamicamente com um **load balancer** e **um naming server**:



Naming Server – Spring cloud Netflix Eureka Server

Obviamente o **Naming server** é outra API Spring Boot que precisa de algumas dependências, a principal obviamente é a Eureka Server:



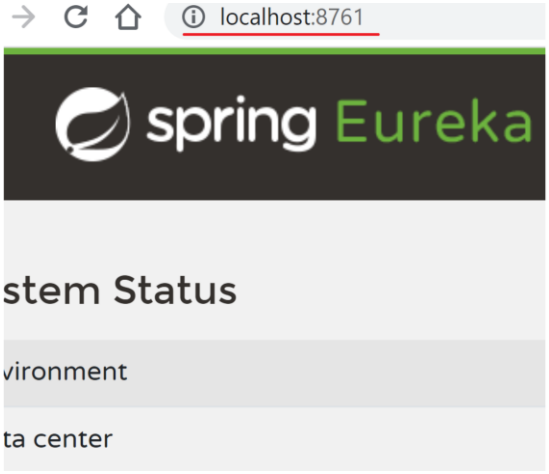


Uma configuração interessante do eureka impede que ele registre a si mesmo, já que ele próprio pode ter várias instâncias, isso não é interessante num ambiente local (em prod sim):

```
@SpringBootApplication
@EnableEurekaServer
public class NamingServerApplication {

    public static void main(String[] arg
```

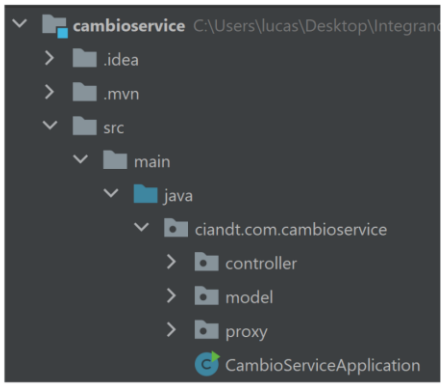
```
server:
  port: 8761

spring:
  application:
    name: naming-server

eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
```

Configurações para ambiente de desenvolvimento



Registrando os micros serviços no naming server – Spring cloud Netflix Eureka Client

Uma vez que o **naming server** está operante, é necessário fazer com que os **micros serviços** se registrem nele, é ridiculamente fácil fazer isso. Só precisamos que o micro serviço tenha a dependência de Client e pronto:




```
server:
  port: 8000

spring:
  application:
    name: cambio-service

eureka:
  client:
    service-url:
      defaultzone: localhost:8761
```

Essa propriedade é opcional

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>
</dependency>
```

Quando o **micro serviço** iniciar ele vai se registrar no **naming service** automaticamente: