### MICROSERVICE E MENSAGERIA COM SPRING BOOT CLOUD E DOCKER ###

1. DISCOVERY SERVER
   1. Criar projeto em Spring Initializr <https://start.spring.io/> e adicionar as dependências:
      1. Eureka Server
   2. Em src/main/java/EurekaserverApplication adicionar a annotation
      1. @EnableEurekaServer;
   3. Em src/main/Resources alterar a extensão para application.yml e configura-lo
2. GATEWAY
   1. Criar projeto em Spring Initializr <https://start.spring.io/> e adicionar as dependências:
      1. Eureka Discovery Client
   2. Configurar o arquivo pom.xml
      1. Adicionar <dependencyManagement>
      2. Alterar <dependency> spring-boot-starter-webflux
      3. Adicionar <dependency>org.springframework.cloud => spring-cloud-starter-gateway
      4. Adicionar <dependency>spring-cloud-starter-netflix-eureka-client
   3. Em src/main/Resources alterar a extensão para application.yml e configura-lo
   4. Em src/main/java/MscloudgatewayApplication adicionar a annotation
      1. @EnableEurekaClient
      2. @EnableDiscoveryClient;
   5. Criar em src/main/java/MscloudgatewayApplication:
      1. Função RouteLocator routes(RouteLocatorBuilder builder)
      2. Adicionar a annotation @Bean
      3. Referenciar as rotas dos microserviços
3. MICROSERVICE Clientes
   1. Criar projeto em Spring Initializr <https://start.spring.io/> e adicionar as dependências:
      1. Spring WEB
      2. Spring Data JPA
      3. Lombok
      4. Spring Boot DevTools
      5. H2 Database
      6. Eureka Discovery Client
   2. Configurar o arquivo pom.xml
   3. Em src/main/Resources alterar a extensão para application.yml e configura-lo
   4. Em src/main/java/EurekaserverApplication adicionar a annotation
      1. @EnableEurekaClient;
   5. Criar no pacote raiz => domain.Cliente
   6. Criar no pacote raiz => infra.repository.IClienteRepository
   7. Criar no pacote application => ClienteService
   8. Criar no pacote raiz => application.ClienteResource ou ClienteController
   9. Criar no pacote application => representation.ClienteSaveRequest ou dto.ClienteSaveRequest
   10. Realizar o registro da rota no microservice GATEWAY
4. MICROSERVICE Cartoes
   1. Criar projeto em Spring Initializr <https://start.spring.io/> e adicionar as dependências:
      1. Spring WEB
      2. Spring Data JPA
      3. Lombok
      4. Spring Boot DevTools
      5. H2 Database
      6. Eureka Discovery Client
   2. Configurar o arquivo pom.xml
   3. Em src/main/Resources alterar a extensão para application.yml e configura-lo
   4. Em src/main/java/MscartoesApplication adicionar a annotation
      1. @EnableEurekaClient;
   5. Criar no pacote raiz => domain.Cartao
   6. Criar no pacote raiz domain => BandeiraCartao
   7. Criar no pacote raiz => infra.repository.ICartaoRepository
   8. Criar no pacote application => CartaoService
   9. Criar no pacote raiz => application.CartoesResource ou CartoesController
   10. Criar no pacote application => representation.CartaoSaveRequest ou dto.CartaoSaveRequest
   11. Realizar o registro da rota no microservice GATEWAY
5. MICROSERVICE AvaliadorCredito
   1. Criar projeto em Spring Initializr <https://start.spring.io/> e adicionar as dependências:
      1. Spring WEB
      2. Spring Data JPA
      3. Lombok
      4. Spring Boot DevTools
      5. Eureka Discovery Client
      6. OpenFeign
   2. Configurar o arquivo pom.xml
   3. Em src/main/Resources alterar a extensão para application.yml e configura-lo
   4. Em src/main/java/MscartoesApplication adicionar a annotation
      1. @EnableEurekaClient;
      2. @EnableFeignClients
   5. Criar no pacote raiz => domain.SituacaoCliente
   6. Criar no pacote domain => DadosCliente
   7. Criar no pacote domain => CartaoCliente
   8. Criar no pacote application => AvaliadorCreditoService
   9. Criar no pacote raiz => infra.clients.ClienteResourceClient
   10. Criar no pacote raiz => application.AvaliadorCreditoController
   11. Realizar o registro da rota no microservice GATEWAY