

Demonstração Github Terminal Linux: <https://www.youtube.com/watch?v=0s699q5Sja4>

Demonstração Github Terminal Windows <http://www.youtube.com/watch?v=DaydwPB2WSI>

Tutorial Basico GitHub com Eclipse e EGit Usando Chave SSH - <https://youtu.be/fFBSazTSGZw>

Vídeos Suporte:

Parte 1: Config Spring: <https://youtu.be/T-z4bLI0vHk>

Parte 2: Projeto Web: <https://youtu.be/Cs5NGGu2W1c>

Parte 3: Maven Install e Maven Site: https://youtu.be/1_JzxbELb3o

Exercícios:

Implementar em Spring Web o Exercício desenvolvido em Java Web com Maven

1) Para fazer um cadastro de cliente, conforme diagrama abaixo, o CPF deve ser válido.






Criar uma procedure que permita fazer o insert, update e delete de um cliente garantindo que:

- Para insert, o CPF deve ser válido e não pode ser 11 números repetidos (11111111111, 22222222222, etc)
- Para update, não se pode alterar o CPF
- Atomizar a stored procedure principal chamando outras procedures para facilitar o entendimento e a manutenção;

Preencher a tabela com dados para testes e usar o https://www.4devs.com.br/gerador_de_cpf para gerar CPF válidos com finalidade de teste.

Link com a lógica da validação de CPF:

http://www.leandrocolevati.com.br/downloadmateriais?idFile=0ByaHyIR4Cic0ZkI3SEZHMWFveTg&arquivo=Aula_02_-_Exercicio_CPF.pdf

Cliente		
	CPF	char(11)
	Nome	varchar(100)
	Email	varchar(200)
	Limite_de_credito	decimal(7, 2)
	Dt_Nascimento	date

Criar, em Spring MVC, uma solução Java Web que permita fazer o CRUD de cliente, utilizando as chamadas de procedures para tratar as regras de negócio ao invés de usar os PreparedStatements para Insert, Update e Delete.