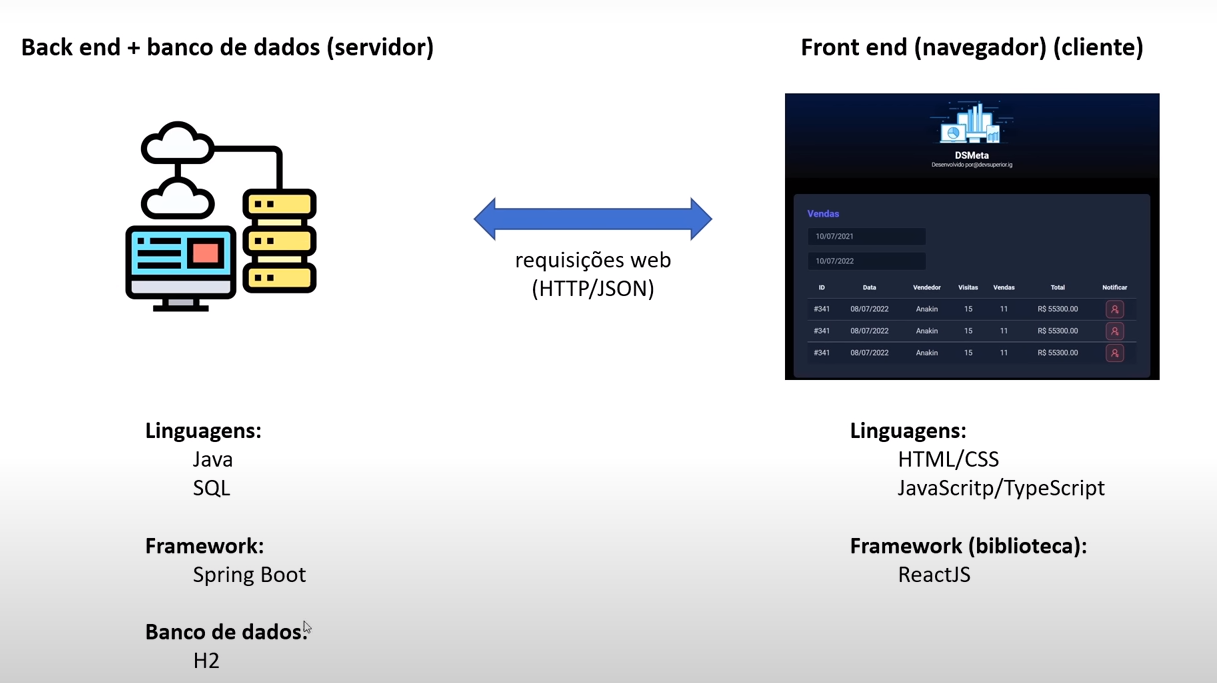
# Projeto - Semana Spring React



|  |
| --- |
| Episódio 1 |

## Objetivos do projeto para esta aula

* Criar projetos backend e frontend
* Salvar os projeto no Github em monorepo
* Montar o visual estático do front end

## Design Figma

<https://www.figma.com/file/PehiT8Dw4Lv5ioTSULZeRI/DSMeta3>

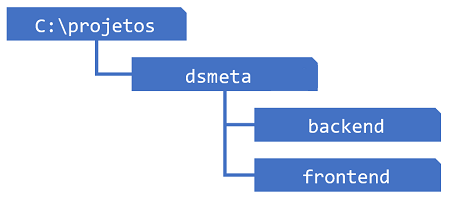
<https://www.figma.com/file/Yu2RHFmirHQ4BIVZM2XxY6/DSMeta2>

<https://www.figma.com/file/EN1zFtk4eY3Jgmpgi9YaMG/DSMeta1>

## Ferramentas

* Git
* Nodejs 16 e Yarn (vídeo: <https://youtu.be/x5tgFTS-CYA> )
* STS (vídeo: <https://youtu.be/TGQ0K9QsX88> )
* VS Code e as extensões:
  + IntelliCode
  + ESLint
  + JSX HTML <tags/>

## Passo: criar projeto ReactJS

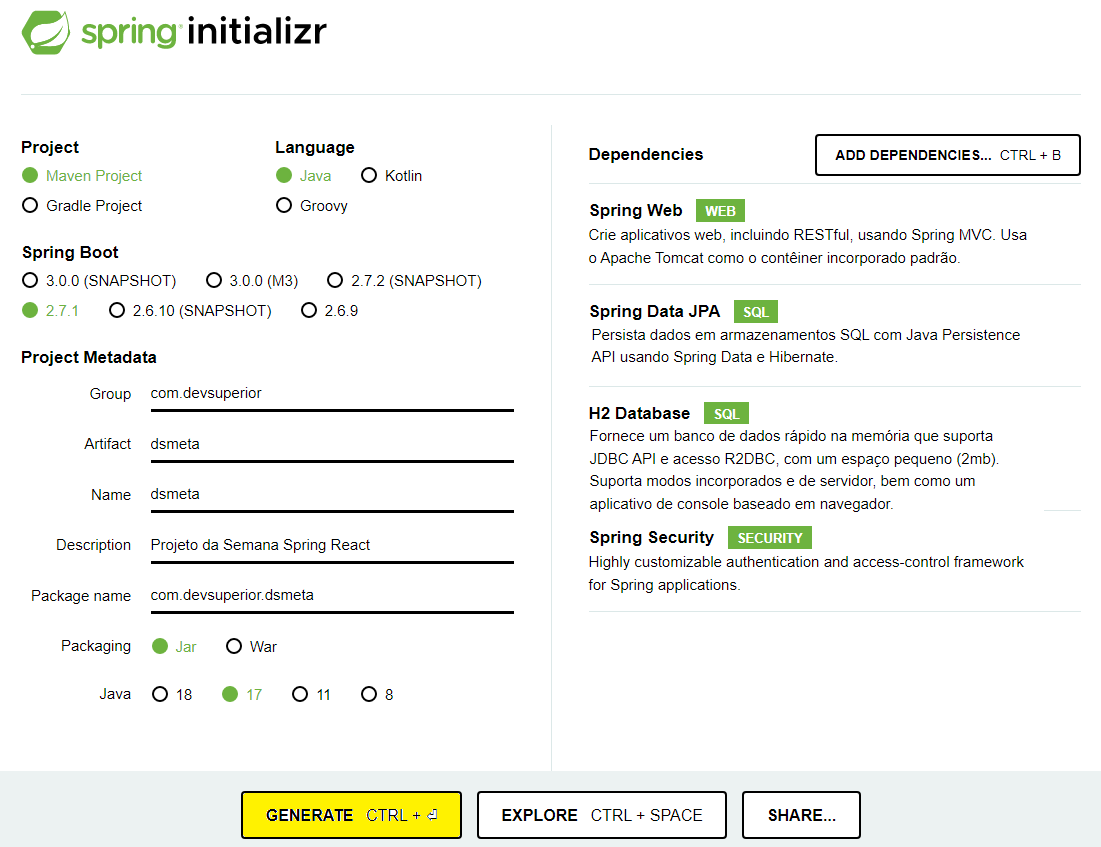
[](https://raw.githubusercontent.com/devsuperior/bds-assets/main/sds/pastas-dsmeta.png)

* Criar a pasta **dsmeta**
* Dentro da pasta abrir o Gitbash
* Colocar o seguinte comando:
* yarn create vite frontend --template react-ts
* O Yarn irá criar a pasta **frontend** e toda a estrutura do projeto react-ts
* Abrir a pasta frontend no Visual Studio Code
* Abrir o terminal Gitbash da pasta frontend e colocar o comando:
* yarn
* Ele irá instalar automaticamente as dependências descritas no arquivo **package.son** do projeto. Também irá criar a pasta **node\_modules** para guardar as dependências.
* Conforme descrito no arquivo **package.son** na parte de **scripts**, coloque os comandos no Gitbash:
* yarn dev
* Ele irá rodar a aplicação e mostrar o login para colocar no navegador. funcionada com o Node aberto.
* http://127.0.0.1:5173/

## Passo: criar projeto Spring Boot

* Criar projeto Spring Boot no **Spring Initializr** com as seguintes dependências:
  + Web
  + JPA
  + H2
  + Security

Abrir o site do **Spring Initializr** <https://start.spring.io/> e crie um projeto:



* Ele fará o dowload da pasta zip com a estrutura do projeto spring boot.
* Extrair a pasta interna do zip para dentro da pasta **dsmeta**.
* Renomear a pasta interna para **backend**
* Abrir o **programa SpringToolSuit4**
* Abrir o **workspace - pasta projetos**
* Importar o projeto backend pelo **Maven**(gerenciador de dependências)
* Assim como o **Yarn** baixou as dependências para o Front o **Maven** irá baixar as dependências do backend da internet
* Ajuste no arquivo pom.xml:
* Dentro do **SpringToolSuit4 abrir** o arquivo **pom.xml** adicione o código abaixo dentro da tag <plugins>.

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-resources-plugin</artifactId>

<version>3.1.0</version><!--$NO-MVN-MAN-VER$ -->

</plugin>

* Para formatar pressione ctrl +shif + f
* Saltar em ctrl + s
* Botão direito no projeto -> Maven -> Update project (force update)
* Esse comando atualiza o projeto com o novo plugin.

## Passo: salvar primeira versão no Github

* Dentro do Gitbash pasta **ds meta**, insira esses comandos:

git init

git add .

git commit -m "Project created"

git branch -M main

git remote add origin git@github.com:seuusuario/seurepositorio.git

git push -u origin main

## Passo: "limpar" o projeto ReactJS e estilizar

Vamos pegar o CSS que fizemos nas aulas de preparação:

<https://github.com/acenelio/dsmeta-css>

* **COMMIT: Project clean**

## Passo: Cria o Primeiro componente

* **COMMIT: First component**

## Passo: Outros componentes (Header, card, button)

* **COMMIT: Other components**

## Passo: Datepicker

Colocar data e hora:

Documentação: <https://github.com/Hacker0x01/react-datepicker>

No Gitbash frontend:

yarn add react-datepicker@4.8.0 @types/react-datepicker@4.4.2

Substituir o input da data no cardSales index:

import DatePicker from "react-datepicker";

import "react-datepicker/dist/react-datepicker.css";

<DatePicker

selected={new Date()}

onChange={(date: Date) => {}}

className="dsmeta-form-control"

dateFormat="dd/MM/yyyy"

/>

* **COMMIT: Datepicker**

## Passo: React Hook useState para manter estado das datas

Macete para criar uma data de X dias atrás:

const date = new Date(new Date().setDate(new Date().getDate() - 365));

* **COMMIT: useState**

|  |
| --- |
| Episódio 2 |

## Objetivos do projeto para esta aula

* Implementar o back end
* Acesso a banco de dados
* Criar endpoints da API REST
* Integração com SMS
* Implantação na nuvem

## Ferramentas

* Postman (Vídeo: <https://youtu.be/CWKLVapcnCU> )
* Heroku CLI (Vídeo: <https://youtu.be/70LUh5KNaEk> )

## Passo: configuração de segurança

* Cross-Origin Resource Sharing ou CORS é um mecanismo que permite que que a aplicação backend que está hospedada no heroku seja acessada pelo frontend que está hospedado em outro lugar.
* No **SpringToolSuite** no DsmetaAplication.java colocar a seguinte configuração:

import java.util.Arrays;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.http.SessionCreationPolicy;

import org.springframework.security.web.SecurityFilterChain;

import org.springframework.web.cors.CorsConfiguration;

import org.springframework.web.cors.CorsConfigurationSource;

import org.springframework.web.cors.UrlBasedCorsConfigurationSource;

@Configuration

@EnableWebSecurity

public class SecurityConfig {

@Bean

public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

http.headers().frameOptions().disable();

http.cors().and().csrf().disable();

http.sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS);

http.authorizeHttpRequests((auth) -> auth.anyRequest().permitAll());

return http.build();

}

@Bean

CorsConfigurationSource corsConfigurationSource() {

CorsConfiguration configuration = new CorsConfiguration().applyPermitDefaultValues();

configuration.setAllowedMethods(Arrays.asList("POST", "GET", "PUT", "DELETE", "OPTIONS"));

final UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();

source.registerCorsConfiguration("/\*\*", configuration);

return source;

}

}

* **COMMIT: Security config**

### Passo: banco de dados

* Criar entidade Sale no SpringToolSuite
  + Clicar em ~~ctrl + shif + O~~ para importar o tipo LocalDate
  + Ver vídeo de encapsulamento de Orientação a Objeto
* Fazer mapeamento objeto-relacional (JPA)
  + Fazer anotations que é são pré processamentos:

@Entity

@Table (name = "tb\_sales")

@Id

@GeneratedValue

* + Escrever o @nome da anotation e clicar em ~~ctrl + shif + O~~ para importar, ou ~~ctrl + espaço~~ para completar preenchimento
* Configurar dados de conexão do banco de dados ***H2*** no arquivo application.properties
* Fazer seed do banco de dados

#### application.properties

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.username=sa

spring.datasource.password=

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

#### import.sql

* Criar o arquivo **import.sql** dentro da pasta **resources** para colocar os comandos abaixo e preencher o banco de dados com as 117 vendas.

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',121,67,18196.0,'2022-06-16');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',26,14,4255.0,'2022-06-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',55,42,13249.0,'2022-06-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',73,65,20751.0,'2022-06-10');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',47,25,7318.0,'2022-06-04');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',72,60,15608.0,'2022-06-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',97,68,8901.0,'2022-06-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',68,26,13231.0,'2022-06-02');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',73,53,19476.0,'2022-05-22');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',28,23,20530.0,'2022-05-18');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',83,44,4864.0,'2022-05-13');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',82,43,21753.0,'2022-05-06');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',43,26,7362.0,'2022-05-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',54,23,10549.0,'2022-04-28');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',125,84,13333.0,'2022-04-25');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',44,26,7431.0,'2022-04-23');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',46,25,21099.0,'2022-04-19');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',42,28,7217.0,'2022-04-19');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',52,21,10107.0,'2022-04-18');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',121,90,18174.0,'2022-04-17');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',65,14,8095.0,'2022-04-12');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',107,74,11507.0,'2022-04-12');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',140,74,11709.0,'2022-04-09');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',95,91,8288.0,'2022-04-08');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',68,43,17016.0,'2022-04-07');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',21,20,17126.0,'2022-04-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',38,15,7957.0,'2022-03-31');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',53,29,20903.0,'2022-03-29');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',19,10,3987.0,'2022-03-28');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',78,34,20795.0,'2022-03-27');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',83,44,4938.0,'2022-03-26');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',32,12,6926.0,'2022-03-13');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',64,33,8193.0,'2022-03-13');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',39,39,10557.0,'2022-03-05');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',158,84,21601.0,'2022-03-02');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',12,6,7625.0,'2022-02-29');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',82,82,22465.0,'2022-02-27');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',68,56,12595.0,'2022-02-17');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',27,13,4636.0,'2022-02-16');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',52,33,10155.0,'2022-02-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',142,81,13610.0,'2022-02-13');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',81,45,15306.0,'2022-02-08');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',64,35,17460.0,'2022-02-07');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',48,24,21413.0,'2022-02-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',150,82,6505.0,'2022-01-26');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',138,95,7983.0,'2022-01-18');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',70,48,9564.0,'2022-01-16');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',162,84,7302.0,'2022-01-15');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',57,54,9126.0,'2022-01-12');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',78,60,5253.0,'2022-01-06');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',81,53,11553.0,'2022-01-04');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',168,92,6299.0,'2021-12-28');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',48,13,22411.0,'2021-12-26');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',107,67,9788.0,'2021-12-24');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',106,62,18942.0,'2021-12-20');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',40,26,11731.0,'2021-12-18');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',101,68,19882.0,'2021-12-18');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',185,100,14618.0,'2021-12-17');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',82,47,7951.0,'2021-12-15');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',86,45,4147.0,'2021-12-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',95,88,12943.0,'2021-12-09');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',75,58,18747.0,'2021-12-02');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',96,50,12624.0,'2021-12-01');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',79,40,14770.0,'2021-11-21');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',73,46,14124.0,'2021-11-20');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',92,58,20953.0,'2021-11-20');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',43,30,9690.0,'2021-11-18');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',58,30,11396.0,'2021-11-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',49,14,5119.0,'2021-11-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',53,23,8206.0,'2021-11-12');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',49,25,8269.0,'2021-11-10');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',53,29,17984.0,'2021-11-09');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',43,26,3056.0,'2021-11-08');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',51,21,8624.0,'2021-11-06');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',64,41,10959.0,'2021-11-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',75,23,15883.0,'2021-10-30');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',51,44,14038.0,'2021-10-27');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',141,81,13535.0,'2021-10-26');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',135,98,17241.0,'2021-10-25');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',116,66,16415.0,'2021-10-19');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',60,44,5329.0,'2021-10-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',63,32,16618.0,'2021-10-07');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',176,100,5062.0,'2021-10-01');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',118,45,22235.0,'2021-09-29');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',150,97,14484.0,'2021-09-26');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',115,63,18081.0,'2021-09-24');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',159,88,16101.0,'2021-09-23');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',76,45,11150.0,'2021-09-22');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',65,63,17982.0,'2021-09-09');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',90,68,15927.0,'2021-09-08');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',22,12,9793.0,'2021-09-06');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',19,11,4185.0,'2021-09-05');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',68,21,15541.0,'2021-09-04');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',64,47,7287.0,'2021-09-04');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',153,92,17913.0,'2021-09-04');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',93,53,12648.0,'2021-09-02');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',78,32,12021.0,'2021-08-30');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',94,49,18787.0,'2021-08-29');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',54,28,3974.0,'2021-08-28');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',45,25,5681.0,'2021-08-26');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',11,1,4008.0,'2021-08-14');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',118,80,5218.0,'2021-08-13');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',52,21,21220.0,'2021-08-09');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',127,23,8831.0,'2021-08-06');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',78,23,13900.0,'2021-08-04');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',102,52,22086.0,'2021-08-03');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',54,53,15731.0,'2021-07-31');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Barry Allen',173,93,10816.0,'2021-07-22');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',60,45,17633.0,'2021-07-20');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',138,72,14528.0,'2021-07-19');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',147,96,21582.0,'2021-07-17');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Logan',32,12,9751.0,'2021-07-13');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',83,59,8496.0,'2021-07-08');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',58,48,5283.0,'2021-07-07');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Kal-El',55,35,20474.0,'2021-07-05');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Anakin',84,34,5787.0,'2021-07-01');

INSERT INTO tb\_sales(seller\_name,visited,deals,amount,date) VALUES ('Padme',79,68,11976.0,'2021-06-27');

* **COMMIT: Database**

### Passo: Primeiro teste de endpoint da API REST

* Criar repository: componente responsável por acessar o banco de dados
  + Assistir aula JpaRepository interface boladona spring
* Criar service: componente responsável por implementar operações de negocios. Ex. operação de buscar as vendas
* Criar controller: componente responsável por disponibilizar a API.
  + Usar o aplicativo **postman** para testar a API
* **COMMIT: API test**

### Passo: Consulta por data

Consulta customizada:

@Query("SELECT obj FROM Sale obj WHERE obj.date BETWEEN :min AND :max ORDER BY obj.amount DESC")

Page<Sale> findSales(LocalDate min, LocalDate max, Pageable pageable);

* **COMMIT: Date select**

## Passo: Envio de SMS

* Fazer cadastro na **Twilio**
* Dentro do **Spring Tool Suite**, arquivo **pom.xml**, tag **<dependencies>,** colocar as Dependências Maven do Twilio, como informa abaixo:

<dependency>

<groupId>com.twilio.sdk</groupId>

<artifactId>twilio</artifactId>

<version>8.31.1</version>

</dependency>

* ~~Ctrl + shift + F~~ para arrumar a endentação.
* Configurar as **variáveis de ambiente** no application.properties para ocultar dados sensíveis como telefone pessoal:
* Dentro do **Spring Tool Suite**, arquivo **application.properties**

twilio.sid=${TWILIO\_SID}

twilio.key=${TWILIO\_KEY}

twilio.phone.from=${TWILIO\_PHONE\_FROM}

twilio.phone.to=${TWILIO\_PHONE\_TO}

* Criar a Classe SmsService dentro do **Spring Tool Suite**:

@Service

public class SmsService {

@Value("${twilio.sid}")

private String twilioSid;

@Value("${twilio.key}")

private String twilioKey;

@Value("${twilio.phone.from}")

private String twilioPhoneFrom;

@Value("${twilio.phone.to}")

private String twilioPhoneTo;

public void sendSms() {

Twilio.init(twilioSid, twilioKey);

PhoneNumber to = new PhoneNumber(twilioPhoneTo);

PhoneNumber from = new PhoneNumber(twilioPhoneFrom);

Message message = Message.creator(to, from, "Teste").create();

System.out.println(message.getSid());

}

}

* **COMMIT: Twilio SMS**

### Passo: Implantação no Heroku

* Entrar no site do Heroku e colocar as Config Vars igual o que está no , arquivo application.properties, TWILIO\_SID, TWILIO\_KEY etc.
* Criar os a Arquivo **system.properties** no STS.

Conteudo do arquivo: java.runtime.version=17

* Criar app no Heroku
* Definir variáveis de ambiente:
  + TWILIO\_SID
  + TWILIO\_KEY
  + TWILIO\_PHONE\_FROM
  + TWILIO\_PHONE\_TO

Colocar no terminal gitbash:

heroku -v

heroku login

heroku git:remote -a <nome-do-app>

git remote -v

git subtree push --prefix backend heroku main

heroku git:remote -a dsmeta-giselle

heroku login -i