Yabi - Yet Another Business Inteligence

Vitório Miguel Prieto Cilia - 40920

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Albano Alves

Prof. Lúcio Valentin

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2017-2018

Yabi - Yet Another Business Inteligence

Vitório Miguel Prieto Cilia - 40920

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Albano Alves

Prof. Lúcio Valentin

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

2017-2018

Dedicatória

(Facultativo) Dedico este trabalho a ...

Agradecimentos

(Facultativo) Agradeço a ...

Resumo

O resumo (no máximo com 250 palavras), permite a avaliação do interesse de um documento e facilita a sua identificação na pesquisa bibliográfica em bases de dados onde o documento se encontre referenciado.

É recomendável que o resumo aborde, de forma sumária:

- Objetivos principais e tema ou motivações para o trabalho;
- Metodologia usada (quando necessário para a compreensão do relatório);
- Resultados, analisados de um ponto de vista global;
- Conclusões e consequências dos resultados, e ligação aos objetivos do trabalho.

Como este modelo de relatório se dirige a trabalhos cujo foco incide, maioritariamente, no desenvolvimento de software, algumas destas componentes podem ser menos enfatizadas, e acrescentada informação sobre análise, projeto e implementação do trabalho.

O resumo não deve conter referências bibliográficas.

Palavras-chave: termos (no máximo 4), que descrevem o trabalho.

Abstract

Direct translation (maximum of 250 words) to English of the section “Resumo”.

Keywords: direct translation of “Palavras-chave”

Contents

1 Introduction **1**

2 Context **3**

3 Objective **5**

4 Concepts and Technologies **7**

4.1 Front-end 7

4.1.1 Typescript 7

4.1.2 HTML 7

4.1.3 CSS 8

4.1.4 SASS 8

4.1.5 Angular 8

4.1.6 Angular Material 9

4.1.7 Sb-Admin-Material 9

4.2 Back-end 14

4.2.1 Java 14

4.2.2 Spring 14

4.2.3 Rest 14

4.2.4 Maria Db 14

4.2.5 LDAP 14

4.3 Development 14

4.3.1 Apache Netbeans . . . 14
4.3.2 Maven . . . 14
4.3.3 Lombok . . . 14
4.3.4 Apache Directory Studio . . . 14
4.3.5 Visual Studio Code . . . 14
4.3.6 Docker . . . 14
4.3.7 Docker-compose . . . 14
4.3.8 Chinook Database . . . 14
4.3.9 Angular CLI . . . 14
4.3.10 Firefox . . . 14
4.3.11 postman . . . 14

5 Project **15**

5.1 Use-cases . . . 15
5.2 Class Diagram . . . 15
5.3 Template Sb-Admin-Material . . . 15
5.4 Multi-Database Support . . . 15

6 Implementation and Results **17**

6.1 Front-end . . . 18
6.1.1 Component Structure . . . 18
6.1.2 Generic Form Control Builder . . . 18
6.1.3 Spring HATEOAS Classes . . . 18
6.1.4 Temporal Caching Repository . . . 18
6.1.5 Error Handler . . . 18
6.1.6 Database Reader . . . 18
6.2 Back-end . . . 18
6.2.1 Entities . . . 18
6.2.2 Spring Configuration . . . 18
6.2.3 Custom Controllers & View Models . . . 18

6.2.4 Spring Repositories 18
6.2.5 ORM Generated Database 18
6.2.6 Multi-Database Support 18
6.3 Development Environment 18
6.3.1 Apache Directory 18
6.3.2 Multi-Database Support 18
6.3.3 Testing File 18
6.3.4 Postman Tests 18

7 Conclusion **19**

8 Future Work **21**

List of Tables

List of Figures

4.1 Login Screen 10
4.2 Dashboard with collapsed side menu 10
4.3 Dashboard with visible side menu 11

Siglas

CSS Cascading Style Sheet. 8

HTML Hypertext Markup Language. 7, 8

LTS Long Term Support. 8

SASS Syntactically Awesome Style Sheets. 8, 12

UI User Interface. 9

Chapter 1

Introduction

Chapter 2

Context

Chapter 3

Objective

Chapter 4

Concepts and Technologies

4.1 Front-end

4.1.1 Typescript

“ A super-set of JavaScript that compiles to plain JavaScript ”[1], Typescript is a language maintained by Microsoft and developed by *Anders Hejlsberg* in 2012 with the goal of improving the quality and manageability of JavaScript code bases with features such as static typing and object-orientated qualities[2]. Ultimately, Typescript must be compiled to JavaScript before being executed, for compatibility reasons, the default JavaScript target is version ES3 but newer back-ends are also available.

4.1.2 HTML

The Hypertext Markup Language (HTML), the “ World Wide Web’s core markup language ”[3] is a declarative language through which the vast majority of online content is structured, shared and accessed. It is a specification of elements that can be used to structure the content of web pages, such as headings, images, link to other documents, buttons and many others[4].

4.1.3 CSS

Cascading Style Sheet (CSS) is another declarative language that pairs with HTML. It's purpose is to describe how the elements present in a web page are presented. Some of the definitions handle colors, fonts, element arranging, visibility, interaction and many others[4].

4.1.4 SASS

Syntactically Awesome Style Sheets (SASS) is a augmentation of CSS with features that are similar to a object-oriented languages, with loops, variables, functions and rule nesting [5]. SASS files need to be compiled into plain CSS before deployment, there are many of such compilers, some re-generate CSS files upon file changes.

4.1.5 Angular

Front-end web framework developed as a side project at Google that proved itself as a valuable tool for modern application development. The core idea is that HTML faults when it comes to declare dynamic content[6], therefore a new middle-ware is introduced between the rendered page and the underling code so that all the elements and events in the HTML document are captured and made available to it's components. Such binding goes both ways, so if the state of the underling code changes, the document is re-rendered to reflect the new state.

The first version of Angular is now called AngularJs and can be included in a HTML document just like any other JavaScript library. This version proved it's value but was considered confusing and some times, slow. Since then it entered Long Term Support (LTS) stage and no features are added. Angular version 2 and up is a Typescript re-write that includes some new features that aid in the architecture and development of scalable and reusable code, namely, the introduction of Components, Router, Ahead-of-Time compilation and Observables[7].

TODO: Router

TODO: Components

TODO: Template

TODO: angular cli

4.1.6 Angular Material

Material Design is a set of guidelines and principles made by Google for designing User Interface (UI) that aims to bring natural and consistent interactions between users and computers. The guiding principle is based on paper and ink but it is not limited to what they can do in the physical world[8].

Angular Material[9] is the implementation made by Google of components like buttons, text input and separators that follow the Material Design guidelines to be used by Angular applications, providing a consistent look across devices.

4.1.7 Sb-Admin-Material

To accelerate the development speed and have faster working prototypes, many web-based projects begin form a ready-made template. This saves time by keeping developers from re-writing common pieces of code commonly referred as "boilerplate".

SB Angular Material is a re-write of the famous SB Admin template[10], a free and open source template developed by Start Bootstrap[11] in Angular using components developed in the previously discussed Angular Material project.

As the name implies this template tries to assess the need for an administrator panel, and in doing so it provides a few ready-made components, to name a few, a login component as seen in figure 4.1; the main screen with a top and a collapsible side navigation

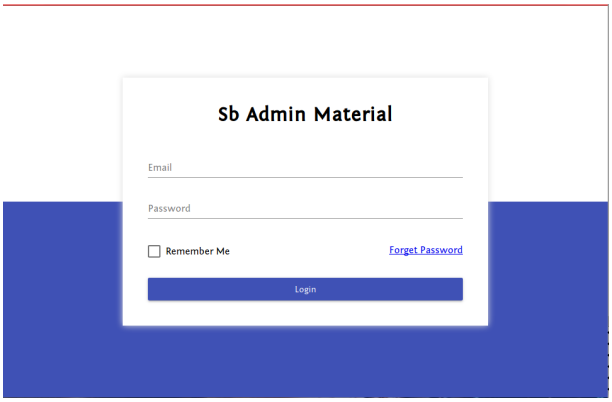

Figure 4.1: Login Screen

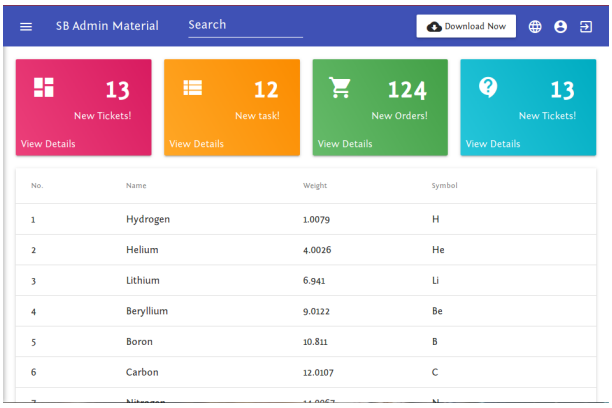

Figure 4.2: Dashboard with collapsed side menu

components, seen in figure 4.2 and 4.3. This template already encompass some amount of responsive design by toggling the ability of said side navigational panel to be collapsed depending on the user's screen width.

TODO: referenciamento da próxima secção, isso esta certo ?

In the following subsection this template's folder structure will be explained so that one can understand where what are the main parts in which it can be extended to fit any particular project.

Project Structure

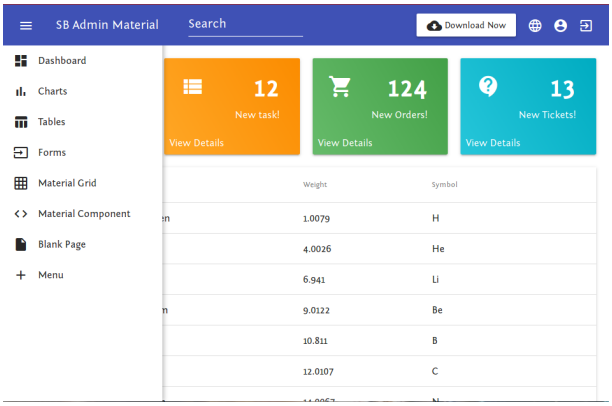

Figure 4.3: Dashboard with visible side menu

TODO: ambientes descriptions aninhados ficaram bons ?

root This item is not a folder but the root of the project. In here there are configurations for code linter, JavaScript dependency descriptor and the license statement.

dist Once the project is built for deployment, this directory will hold all the assets and optimized code ready for production, including the main `index.html` file that bootstraps the whole project.

e2e This holds the source code for End-to-End test cases, hence the name.

src This is the heart of this template, a directory that holds all the structure, content and behavior needed per application.

app The Angular entry-point and application wide router module.

layout All the components used to compose the navigational elements and menus and their subsequent pages plus some example pages.

black-page A inaccessible component that does nothing, probably unfinished.

blank-page An example component that is white.

charts A component that display chart capabilities of the integrated JavaScript module chartjs[1].

components Omnipresent page elements such as the Top Bar and the collapsible Side Bar.

topnav The blue navigation top bar as seen on Figure 4.2.

sidebar The Menu on the left side of the screen as seen on Figure 4.3.

dashboard The page in which the user is redirected after logging in.

forms Demonstration of the many different input methods such as Auto Complete text input, Date picker, Text Area and others.

grid A demo of the available page subdivisions.

material-components An example page displaying the main components of Angular Material such as buttons, Dialog and Notifications.

nav

tables

login This is the Login component as see on Figure 4.1

shared Code that can be used in a application wide manner so that higher abstractions and code reuse can be achieved.

assets Static content directory. Images, fonts, and i18n translations.

environments Depending on how the project is run, either in development or in production mode, a the respective configuration file that holds environment constants is used, allowing developers to use the same reference name throughout the code base no matter the environment.

styles SASS files that define the look and feel.

[1] Available at https://www.chartjs.org/

4.2 Back-end

4.2.1 Java

View Model

4.2.2 Spring

Dependency Injection

Boot

Data

Web

HATEOAS

Security

4.2.3 Rest

4.2.4 Maria Db

4.2.5 LDAP

4.3 Development

4.3.1 Apache Netbeans

4.3.2 Maven

4.3.3 Lombok

4.3.4 Apache Directory Studio

4.3.5 Visual Studio Code

4.3.6 Docker

4.3.7 Docker-compose

Chapter 5

Project

5.1 Use-cases

5.2 Class Diagram

5.3 Template Sb-Admin-Material

5.4 Multi-Database Support

Chapter 6

Implementation and Results

6.1 Front-end

6.1.1 Component Structure

Services

Modules

Dialogs

6.1.2 Generic Form Control Builder

6.1.3 Spring HATEOAS Classes

Entity Class

Acessor Class

Repository Class

Repository Service Class

6.1.4 Temporal Caching Repository

6.1.5 Error Handler

6.1.6 Database Reader

Chapter 7

Conclusion

Chapter 8

Future Work

Bibliography

[1] typescriptlang team, *Typescript website*, http://www.typescriptlang.org, May 2019.

[2] D. Maharry and T. Meister, *Typescript revealed*, 1st. Apress, lda, 2013, ISBN: 978-1-4302-5725-7.

[3] w3c, *Html 5.2, w3c recommendation*, https://www.w3.org/TR/html52/introduction.html, Dec. 2017.

[4] ——, *Html & css*, https://www.w3.org/standards/webdesign/htmlcss, Dec. 2017.

[5] J. Anne and N. Weizenbaum, *Sass documentation*, https://sass-lang.com/documentation, May 2019.

[6] A. Team, *Angularjs homepage*, https://angularjs.org/, May 2019.

[7] S. K. Kasagoni, *Building Modern Web Applications Using Angular*, 1st. Packt Publishing Ltd., 2017, ISBN: 978-1-78588-072-8.

[8] I. G. Clifton, *Android User Interface Design*, 2nd. Pearson Education Inc., 2016, ISBN: 978-0-134-19140-9.

[9] A. M. Team, *Angular material homepage*, https://material.angular.io/, May 2019.

[10] F. Martino and N. K. Mishra, *Sb admin material*, https://github.com/start-javascript/sb-admin-material, May 2019.

[11] D. Miller, Kouceyla, A. Kumar, and B. Clees, *Sb material*, https://github.com/BlackrockDigital/startbootstrap-sb-admin, May 2019.