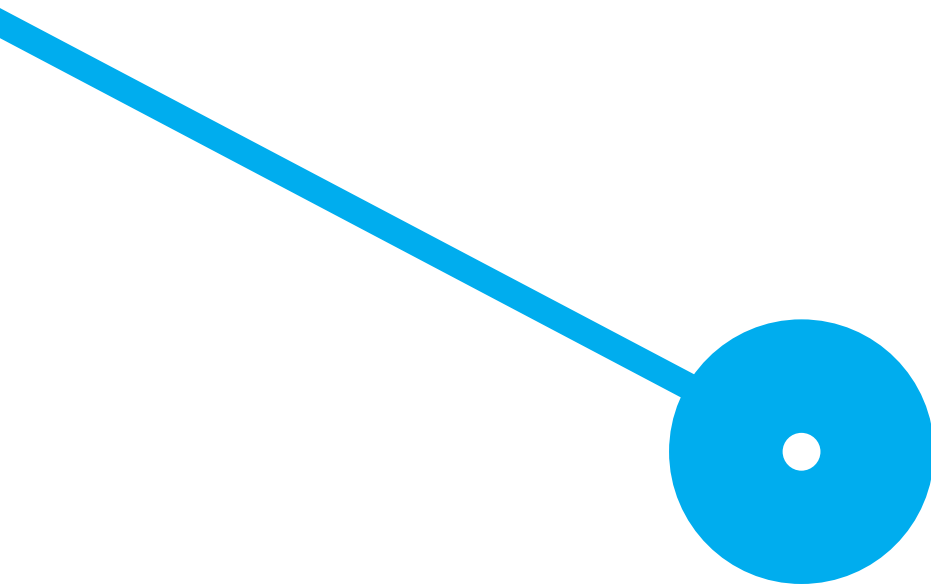


RELATÓRIO DE ESTÁGIO

Sérgio Luís Lopes Félix

2021/2022



Basicamente Website

Sérgio Luís Lopes Félix

Paulo César de Magalhães da Rocha Melo

Resumo

O presente documento serve para fazer balanço do estágio curricular realizado no âmbito do Curso Técnico Superior Profissional de Desenvolvimento para a Web e Dispositivos Móveis lecionados nos anos 2020/21 e 2021/2022.

Com a crescente importância da experiência profissional na qualificação dos trabalhadores no mercado profissional, o Curso Técnico Superior Profissional é valorizado pelo facto de essa competência ser cada vez mais valorizada pelos empregadores. A oportunidade de, em âmbito curricular, ter contacto com as metodologias de trabalho de uma empresa é uma mais-valia para o currículo.

O estágio decorreu no período 21 março a 05 de agosto de 2022 na Basicamente, empresa de Tecnologia e Marketing Digital.

Este projeto teve como objetivo principal o desenvolvimento de uma solução que utilizasse as tecnologias mais recentes de forma a obter um produto com mais longevidade, performance e facilidade no processo de inserção de conteúdos.

Teve também como objetivo o desenvolvimento de qualidades pessoais, tais como o trabalho em equipa e ética em local de trabalho.

Agradecimentos

Com a finalização do relatório de estágio curricular gostaria de agradecer a algumas pessoas que, direta ou indiretamente, me ajudaram nesta fase importante da vida académica.

Em primeiro lugar, gostaria de agradecer ao Instituto Politécnico do Porto (IPP), mais concretamente à Escola Superior de Tecnologia e Gestão (ESTG) e aos seus professores, pela oportunidade que me proporcionou, por todo o apoio prestado ao longo desta formação e por todo o conhecimento que me foi transmitido tanto a nível profissional como pessoal.

Gostaria de agradecer à empresa Basicamente Digital por ter proporcionado a realização deste estágio nas suas instalações e por todo o apoio e disponibilidade prestada ao longo destes meses. Um especial agradecimento ao meu supervisor José António Carvalho Coelho pela dedicação e apoio, que desde o início me ajudou em todas as atividades e ao representante Luís Fernando Mesquita Matos por se mostrar sempre disponível na resolução de qualquer dúvida.

Quero agradecer ao meu orientador de estágio, o Professor Paulo César de Magalhães da Rocha Melo, por toda a ajuda, simpatia, dedicação e atenção prestada ao longo destes meses, contribuindo para que fosse possível a concretização deste estágio curricular.

Por fim, agradeço à minha família, em especial aos meus pais, que sempre me apoiaram e ajudaram nesta etapa da minha vida e também aos meus amigos e colegas de curso por todo o apoio contribuindo para o sucesso durante este percurso.

Índice

Resumo	3
Agradecimentos	4
Lista de Abreviaturas e Siglas	8
Abreviaturas	8
Siglas	8
1. Introdução	9
1.1. Apresentação da entidade de acolhimento	9
1.2. Âmbito/Contextualização	9
1.3. Objetivos	9
1.4. Resultados.....	10
2. Visão geral do projeto	11
2.1. Perspetiva do Produto	11
2.2. Perspetiva do Utilizador.....	11
2.3. Pressupostos de Restrições	12
2.4. Dependência	13
3. Requisitos Específicos.....	14
3.1. Requisitos Funcionais	14
3.2. Requisitos Não Funcionais.....	16
3.3. Requisitos de Interface	17
4. Design da Solução.....	18
4.1. Arquitetura Conceptual.....	18
4.2. Diagrama Caso de Uso	20
4.3. Diagrama de Sequência	21
4.4. Protótipos de Baixa-Fidelidade.....	23
4.5. Protótipos de Alta-Fidelidade.....	25
4.6. Estrutura do Projeto.....	29
4.7. Cliente Apollo	30
4.8. Definição de Rotas	31
4.9. Multilíngue	32
4.10. Fetch dos dados em conformidade com a linguagem	33

4.11.	Definição dos modelos de conteúdos.....	34
4.12.	Envio de e-mails.....	36
5.	Gestão e acompanhamento do projeto.....	39
5.1.	Planeamento	39
5.2.	Acompanhamento do projeto.....	40
5.3.	Ferramentas de gestão	41
6.	Resultados	42
6.1.	Cenários de validação	42
6.1.1.	Pedir Proposta	42
6.1.2.	Entrar em Contacto	42
6.1.3.	Mudar de Linguagem	43
6.1.4.	Voltar para o topo.....	44
6.1.5.	Listagem de Serviços	44
6.1.6.	Listagem de Projetos.....	45
6.1.7.	Introdução de conteúdos.....	45
6.2.	Análise de resultados	46
7.	Conclusões	48
8.	Bibliografia e Webgrafia.....	49
9.	Anexos.....	50

Índice de ilustrações

Figura 1 - Processo da aplicação	18
Figura 2 - Árvore de componentes	19
Figura 3 - Diagrama de Caso de Uso	20
Figura 4 - Diagrama de Sequência de Contacto	21
Figura 5 - Diagrama de Sequência de Proposta	22
Figura 6 - Protótipo inicial de componentes gerais.....	23
Figura 7 - Protótipo inicial de alguns componentes de determinadas páginas. 24	
Figura 8 - Homepage (Versão Desktop).....	25
Figura 9 - Homepage (Versão Mobile)	26
Figura 10 - Página de Pedir Proposta (Versão Desktop)	27
Figura 11 - Página de Pedir Proposta (Versão Mobile)	28
Figura 12 - Estrutura do projeto.....	29
Figura 13 - Criação do cliente Apollo (ficheiro index.tsx)	30
Figura 14 - Criação das rotas (ficheiro app.tsx).....	31
Figura 15 - Código fonte do contexto multilingue	32
Figura 16 – Uso do hook useQuery em conformidade com o valor multilingue atual	33
Figura 17 - Consulta GraphQL com valor da linguagem como parâmetro	33
Figura 18 - Modelos de conteúdo (Contentful Dashboard).....	34
Figura 19 - Campos do modelo de conteúdo (Contentful Dashboard)	35
Figura 20 - Formulário de contato (com os dados preenchidos e validados) ...	36
Figura 21 - Serviço SMTP utilizado (EmailJS Dashboard)	37
Figura 22 - E-mail template e e-mail Auto-Reply Template (Dashboard do EmailJS).....	38
Figura 23 - Formulário de contato submetido (pop-up aberta)	38
Figura 24 - Cronograma de Desenvolvimento do Projeto Inicial	39
Figura 25 - Cronograma de Desenvolvimento do Projeto Final.....	40
Figura 26 - Repositório do projeto no GitHub	41
Figura 27 - Resultados do Lighthouse.....	47

Lista de Abreviaturas e Siglas

Abreviaturas

API – Application Programming Interface

B2B – Business to Business

B2C – Business to Consumer

CMS – Content Management System

CRUD – Create Read Update Delete

CLI – Command Line Interface

CSS – Cascading Style Sheets

HTML – Hyper Text Markup Language

JS – JavaScript

SEO – Search Engine Optimization

SMTP – Simple Mail Transfer Protocol

SaaS – Software as a Service

TS – TypeScript

UI – User Interface

UX – User Experience

Siglas

IPP - Instituto Politécnico do Porto

ESTG – Escola Superior de Tecnologia e Gestão

CTeSP – Curso Técnico Superior Profissional

1. Introdução

1.1. Apresentação da entidade de acolhimento

A Basicamente Digital, é uma empresa que está presente no mercado há sensivelmente 8 anos. Conta assim com cerca de 50 colaboradores e a sua presença distribuída pelo distrito de Lisboa sendo aí a sua sede e no distrito do Porto.

A Basicamente Digital está presente nas seguintes indústrias:

- Serviços de Tecnologia da Informação
- Consultoria de Tecnologia da Informação

Tendo como suas especialidades o comércio eletrónico B2C e B2B, aplicações na nuvem, desenvolvimento de websites, marketing digital, entre muitos outros.

1.2. Âmbito/Contextualização

A Basicamente Digital pretende fazer um *re-branding* do seu *website* de modo que seja utilizado as últimas tecnologias e de modo que se tenha uma maior e mais eficaz alcance e visibilidade das suas soluções.

Sendo também necessário fazer a respetiva gestão de conteúdos.

1.3. Objetivos

Com objetivo de alcançar um maior número possível de clientes será necessário o desenvolvimento de um *website* com as funcionalidades de

contacto, envio de proposta e visualização de serviços prestados de forma que o novo website dê a perspetiva que a entidade acolhedora pretende.

1.4. Resultados

A Basicamente espera através deste *website*, dar aos seus utilizadores uma leitura da informação mais clara e objetiva, uma maior facilidade de entrar em contacto com a empresa e ainda um processo mais ágil de introdução de conteúdos através do uso de tecnologias mais recentes.

2. Visão geral do projeto

2.1. Perspetiva do Produto

O *website* tem como perspetiva facilitar a leitura da informação, ter um aspeto mais agradável e moderno, poder esclarecer eventuais dúvidas, receber contacto de possíveis clientes, agilizar o processo da introdução de conteúdos e como se tratar de um *website* desenvolvido com tecnologias recentes e escaláveis possuir uma maior longevidade.

2.2. Perspetiva do Utilizador

Os utilizadores do *website* de modo geral serão pessoas ou entidades com interesse de transformar, inovar e/ou escalar os seus negócios no mundo digital.

Através do *website* o utilizador anonimo vai conseguir ter uma perceção dos serviços prestados pela Basicamente Digital e bem como, dos trabalhos anteriormente realizados. Será também possível saber os ideais e valores da empresa, onde dezenas de marcas confiam na mesma, por fim, possibilidade de entrar em contacto ou fazer um pedido de proposta através dos respetivos formulários.

Já os utilizadores administradores e editores através da plataforma Contentful dependendo do seu papel irão poder criar, editar ou eliminar respetivos conteúdos.

2.3. Pressupostos de Restrições

Neste projeto foram estabelecidas algumas restrições técnicas, sendo a maior delas o desenvolvimento da aplicação com a mais recente e popular tecnologia React, uma biblioteca JavaScript de código aberto, desenvolvida pelo Facebook que solidifica ainda mais a sua escolha devido a ter uma grande comunidade de desenvolvedores e a própria empresa Facebook a manter a mesma sempre atualizada, fora as inúmeras outras vantagens como fácil aprendizagem, *SEO-friendly*, componentes reusáveis, performance, escalabilidade, cross-platform, etc...

Tendo em consideração a existência de conteúdo dinâmico foi estipulada a necessidade da utilização de um sistema de gerenciamento de conteúdo, como tal foi definido a utilização da plataforma Contentful.

Os pontos determinantes para essa decisão foi o facto de se tratar de um *headless* CMS possuindo assim uma customização ilimitada devido ao facto da lógica de armazenamento de dados e da interface de utilizador CRUD estar de-copulada da lógica de visualização de dados, outro fator decisivo foi o custo a segurança performance e escalabilidade.

Após estabelecidas as ferramentas e linguagem de programação a utilizar, era imprescindível um local para hospedar o código-fonte e outros arquivos com um controlo de versão usando o sistema GIT, assim sendo, foi decretado o uso da plataforma GitHub de forma a contornar esse problema, fornecendo assim posteriormente suporte a plataforma Vercel que facilita o *deploy* e a hospedagem da aplicação.

2.4. Dependência

A aplicação contém como dependências externas:

- Contentful – Um serviço que oferece uma infraestrutura de conteúdo para os projetos, permitindo assim através das suas API's criar, gerenciar e publicar conteúdos para qualquer canal digital.
- ReCAPTCHA – Um serviço gratuito oferecido pela Google que protege *websites* contra spam e abuso, utilizando técnicas avançadas de análise de risco para diferentes humanos robôs.
- EmailJS – Um serviço que permite criar um conjunto predefinido de e-mails através do painel de controlo e depois só acionar os e-mails a partir do JavaScript.
- Google Analytics – Um serviço gratuito oferecido pela Google que permite aos proprietários dos *websites* avaliarem o retorno sobre o investimento de publicidade, bem como acompanhar outras estatísticas dos seus *websites*.

3. Requisitos Específicos

3.1. Requisitos Funcionais

RF-001: Pedir proposta	
Descrição	Esta funcionalidade permite ao utilizador entrar em contacto facilmente com a empresa através de um formulário para apresentar o seu projeto ou o serviço que necessita.
Prioridade	Urgente - SIM Importante -SIM
Autores	Utilizador anónimo
Restrições	N/A
Critérios de aceitação	Os elementos assinalados como obrigatórios devem estar devidamente preenchidos e respeitando o formato do tipo de dados.
RF-002: Contactar	
Descrição	Esta funcionalidade permite ao utilizador entrar em contacto facilmente com a empresa através de um formulário para apresentar as suas questões.
Prioridade	Urgente - SIM Importante -SIM
Autores	Utilizador anónimo
Restrições	N/A
Critérios de aceitação	Os elementos assinalados como obrigatórios devem estar devidamente preenchidos e respeitando o formato do tipo de dados.
RF-003: Mudar linguagem	
Descrição	Esta funcionalidade permite ao utilizador alterar entre português e inglês o conteúdo apresentado.
Prioridade	Urgente - NÃO Importante -SIM
Autores	Utilizador anónimo

Restrições	N/A
Critérios de aceitação	N/A
RF-004: Voltar para o topo	
Descrição	Esta funcionalidade permite ao utilizador através de um botão (apresentado no canto inferior do ecrã) voltar para o topo da página.
Prioridade	Urgente - NÃO Importante -NÃO
Autores	Utilizador anónimo
Restrições	N/A
Critérios de aceitação	Apenas deverá ficar visível apos serem movidos no eixo Y 300 pixéis.
RF-005: Listagem dos serviços	
Descrição	Esta funcionalidade permite ao utilizador visualizar a lista dos principais serviços prestados pela entidade.
Prioridade	Urgente - SIM Importante -SIM
Autores	Utilizador anónimo
Restrições	N/A
Critérios de aceitação	N/A
RF-006: Listagem dos projetos	
Descrição	Esta funcionalidade permite ao utilizador visualizar a os principais projetos desenvolvidos anteriormente pela entidade.
Prioridade	Urgente - SIM Importante -SIM
Autores	Utilizador anónimo
Restrições	N/A
Critérios de aceitação	N/A

3.2. Requisitos Não Funcionais

RNF-001: Linguagem de programação	
Descrição	A aplicação deve ser desenvolvida em TypeScript para evitar futuros erros, com a biblioteca React.
RNF-002: UI	
Descrição	A UI (<i>User Interface</i>) deve ser amigável, responsiva, apelativa, etc... para tal deve ser planeada na ferramenta Figma e transposta através da framework Tailwind CSS para a aplicação.
RNF-003: UX	
Descrição	A UX (<i>User Experience</i>) deve ser estar bem estruturada e organizada de forma que o utilizador tenha uma boa experiência e não se sinta perdido ou confuso acerca do que necessita para tal será utilizado o CMS Contentful.
RNF-004: Desempenho	
Descrição	Devem ser tidos em conta formas de melhorar a performance da aplicação e o SEO (<i>Search Engine Optimization</i>), seja através do uso de imagens com formato “.webp” e <i>loading</i> do tipo <i>lazy</i> , a <i>bundles</i> de “.css”, “.js”, purgados e minificados de forma a reduzir o <i>payload</i> de cada <i>request</i> , etc...
RNF-005: Compatibilidade	
Descrição	A aplicação deve ser completamente funcional sem muita discrepância entre os diferentes web browsers.
RNF-006: Segurança	
Descrição	Todos os formulários devem estar acompanhados pelo método reCaptcha de modo a prevenir ataques automatizados.
RNF-007: Manutenibilidade	
Descrição	A aplicação deve conter uma arquitetura não muito diferente do padrão habitual, devendo ser possível uma fácil expansão, permitindo assim uma rápida evolução da aplicação para se adaptar a situações possíveis no futuro.

3.3. Requisitos de Interface

RI-001: Tipografia	
Descrição	O tipo de letra deve ser Poppins e o tamanho da fonte não deve ser menor que 0.875rem/14px.
RI-002: Acessibilidade	
Descrição	O sistema deve respeita as diretrizes de acessibilidade para o conteúdo da web 2.1.
RI-003: Formulários	
Descrição	Os formulários devem ser claros, curtos e agrupados de forma lógica.
RI-004: Cor	
Descrição	As cores devem estar em conformidade da Basicamente Digital, ou seja, branco(#ffffff) e azul(#0c5eac).
RI-005: Formato de imagens	
Descrição	O formato das imagens deve ser do tipo “.webp”.

4. Design da Solução

4.1. Arquitetura Conceptual

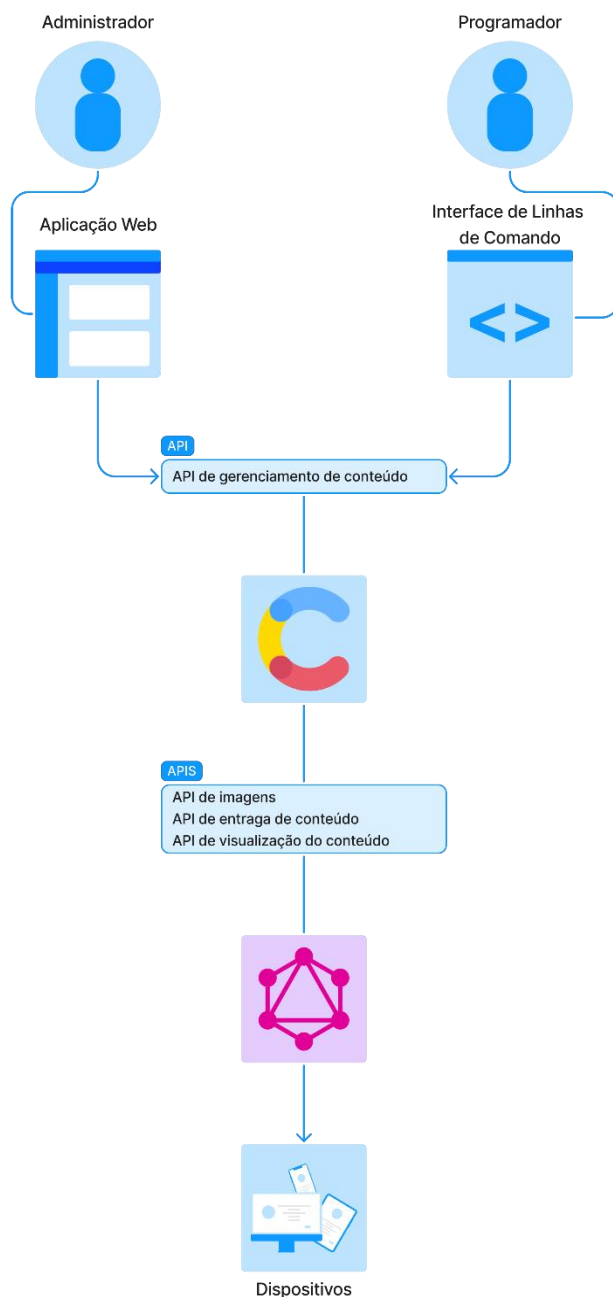


Figura 1 - Processo da aplicação

A ilustração ao lado demonstra a arquitetura da aplicação.

Do lado esquerdo os administradores e/ou editores através da aplicação web poderão fazer o CRUD dos conteúdos, gerir as taxonomias, etc...

Já do lado direito os desenvolvedores através da CLI possuem as mesmas funcionalidades que o administrador mais a funcionalidade de configurar o CMS.

Por fim, através das API's em GraphQL fornecidas pelo Contentful os conteúdos podem ser acessados de qualquer plataforma e dispositivo.

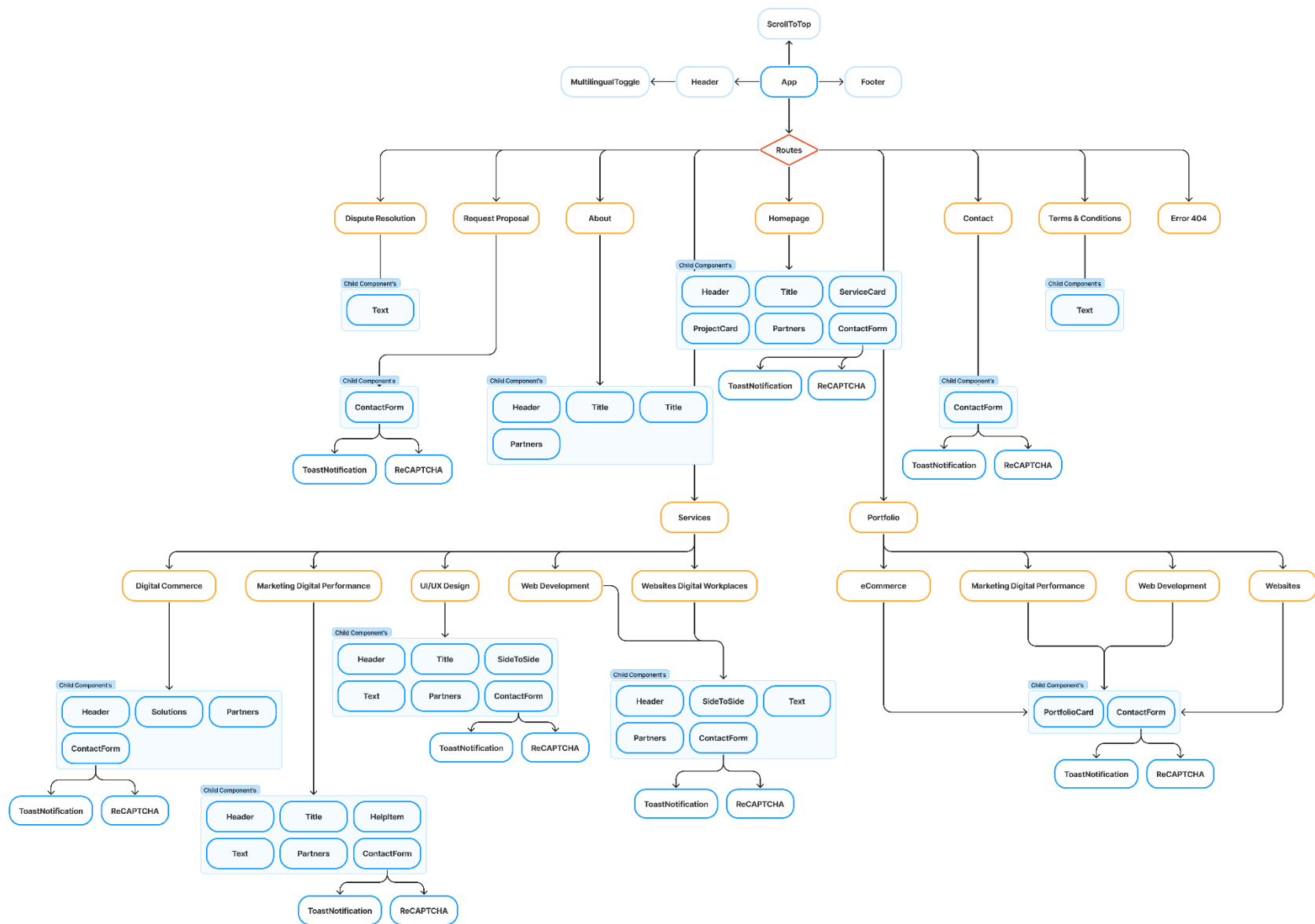


Figura 2 - Árvore de componentes

A ilustração acima é uma visão geral da aplicação, nela podemos observar componentes gerais que englobam toda a aplicação bem como as rotas e as suas dependências e/ou os componentes filhos.

4.2. Diagrama Caso de Uso

O seguinte diagrama de caso de uso demonstra as interações possíveis que um utilizador anônimo pode ter com o sistema.

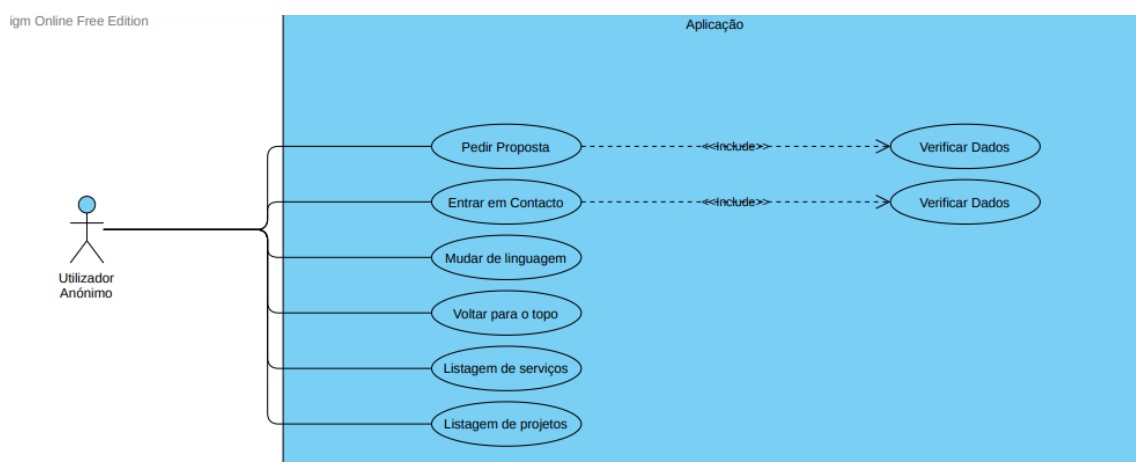


Figura 3 - Diagrama de Caso de Uso

4.3. Diagrama de Sequência

Os seguintes diagramas de sequência abaixo ilustrados são bastante semelhantes ambos descrevem a sequência de ações necessárias até, por exemplo, enviar um formulário de proposta.

O utilizador introduz os dados necessários no formulário que são então validados, se algum dos dados estiver incorreto o utilizador é informado de tal, caso todos os dados estejam corretos o administrador e/ou o departamento responsável é então notificado, como também é enviado um e-mail *auto-reply* de confirmação para o e-mail submetido e posteriormente é informado o utilizador de tal.

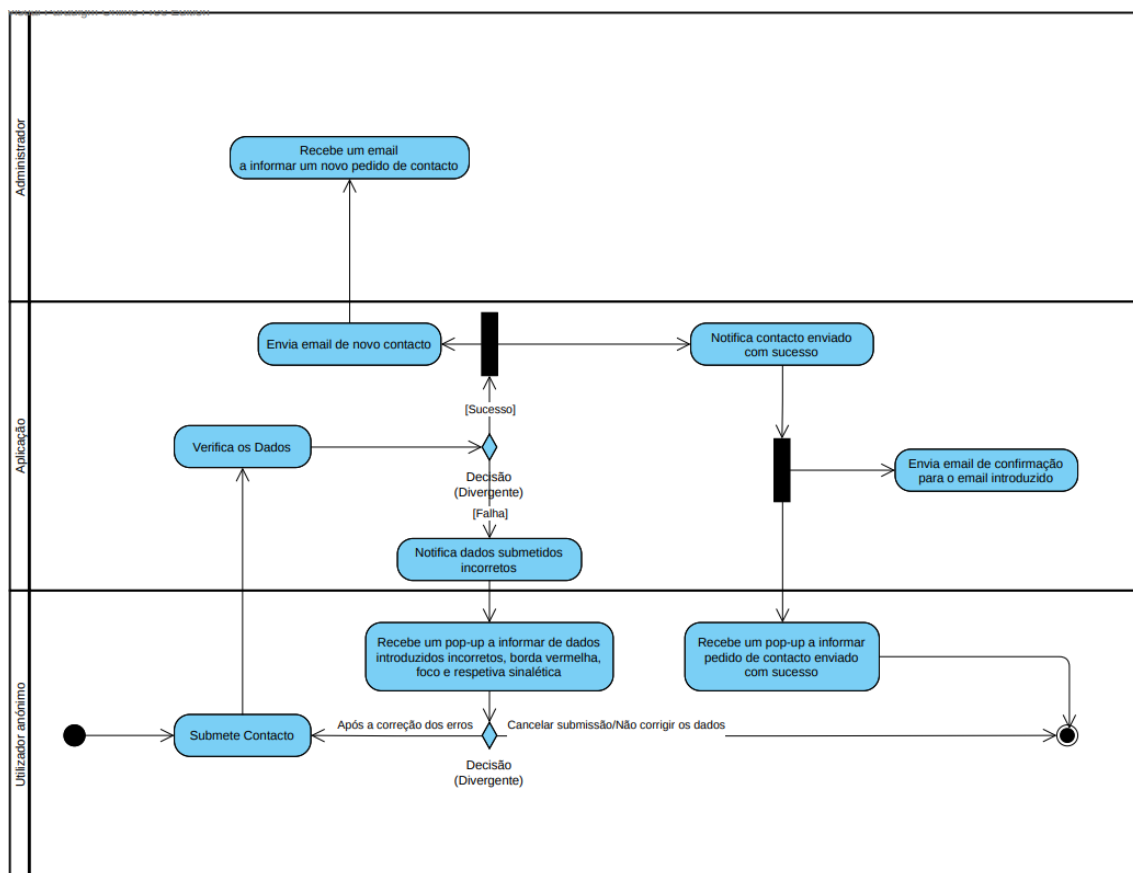


Figura 4 - Diagrama de Sequência de Contacto

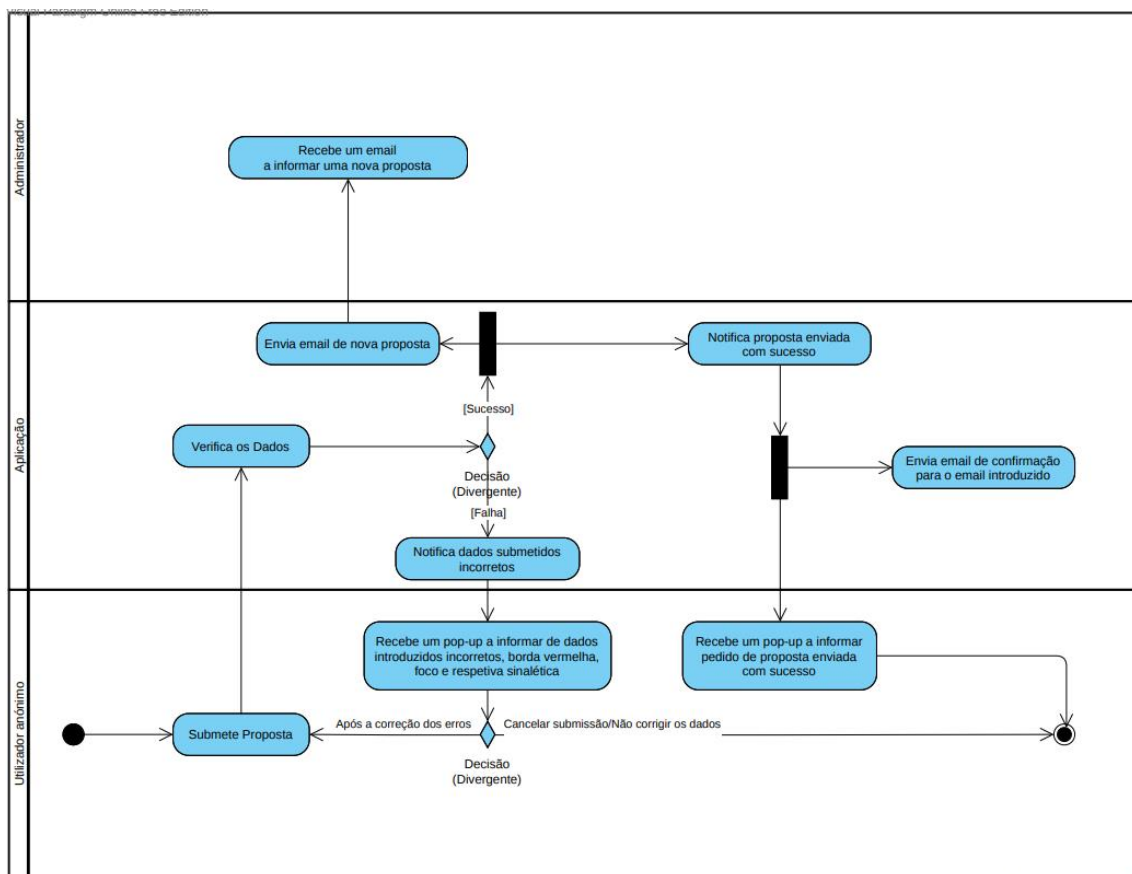


Figura 5 - Diagrama de Sequência de Proposta

4.4. Protótipos de Baixa-Fidelidade

As ilustrações abaixo demonstram as primeiras ideias colocadas em papel, onde foram desenhados alguns componentes gerais, que irão sofrer alterações quando forem transformados em protótipos mais fiéis.

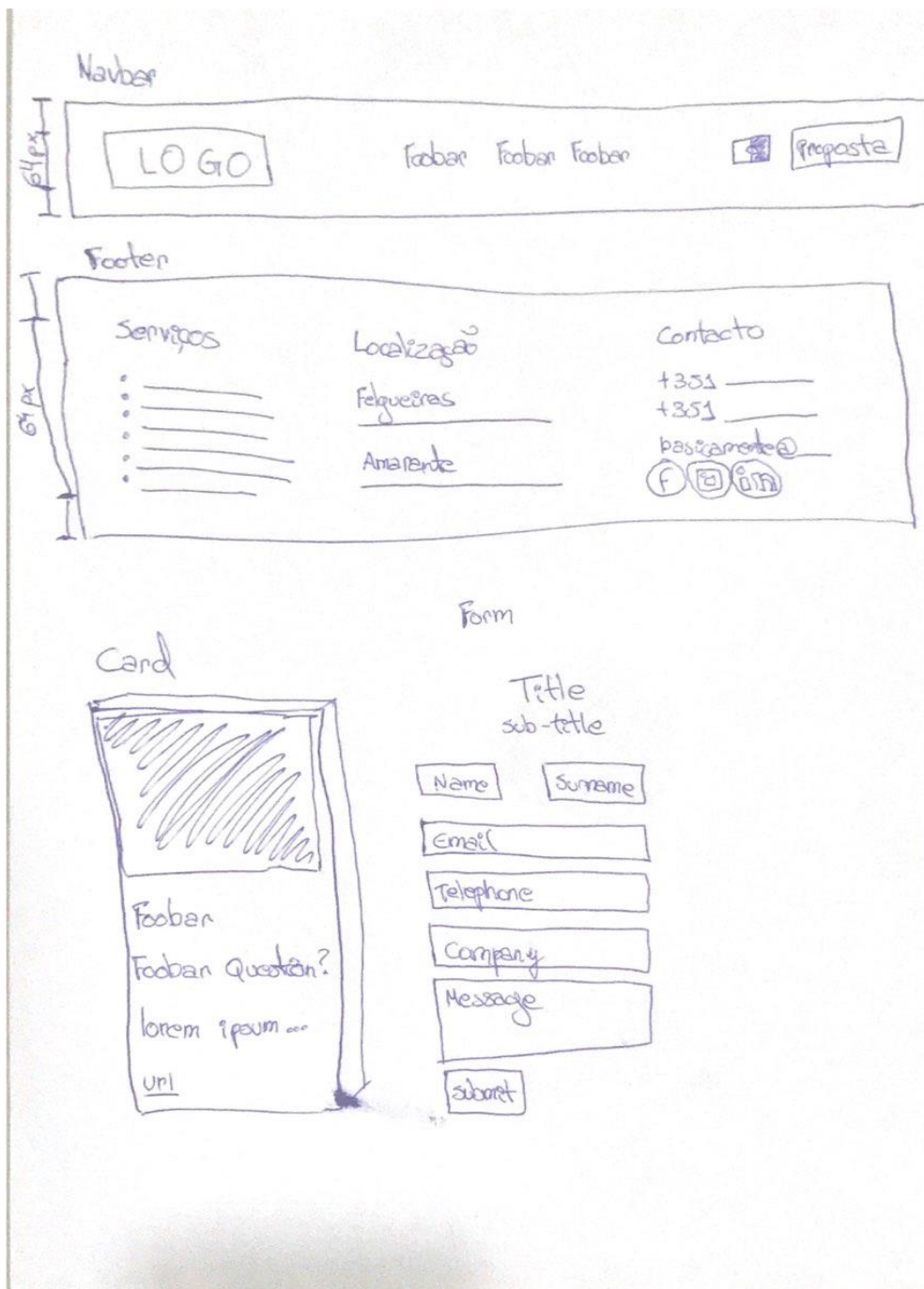
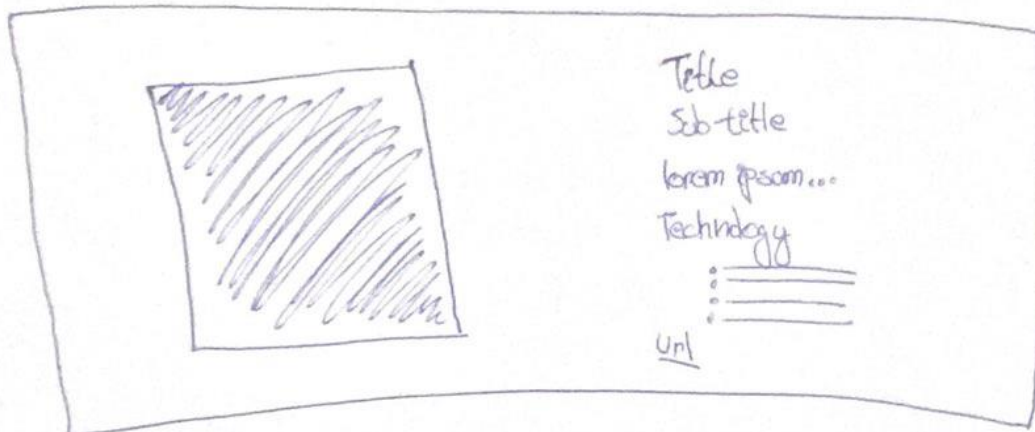


Figura 6 - Protótipo inicial de componentes gerais

Portfolio



Contato



Figura 7 - Protótipo inicial de alguns componentes de determinadas páginas

4.5. Protótipos de Alta-Fidelidade

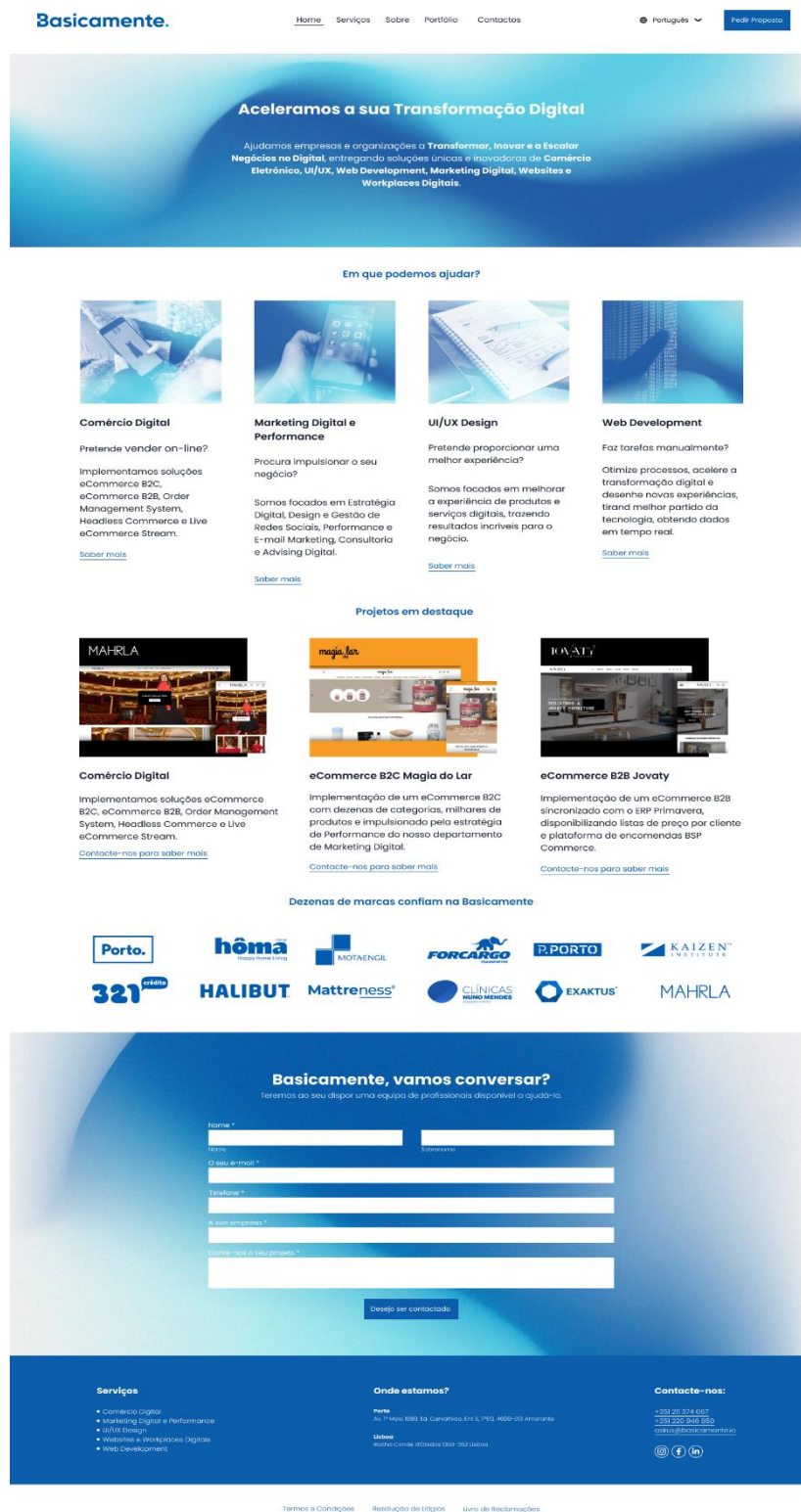


Figura 8 - Homepage (Versão Desktop)



Figura 9 - Homepage (Versão Mobile)

Peça-nos agora uma proposta e saiba como o podemos ajudar a ter sucesso.

Ao trabalhar com a Basicamente contará com:

- Um parceiro de negócio;
- Um parceiro de negócio;
- Um parceiro de negócio;
- Um parceiro de negócio;

O seu nome: *

O seu contacto: *

O seu e-mail: *

A sua empresa: *

O seu budget disponível: *

Escolha uma opção ▼

Escolha uma opção

5.000€ a 5.000€

5.000€ a 15.000€

15.000€ a 50.000€

> 50.000€

Descreva de que forma o podemos ajudar: *

[Pedir Proposta](#)

Serviços

- Comércio Digital
- Marketing Digital e Performance
- UI/UX Design
- Websites e Workplaces Digitais
- Web Development

Onde estamos?

Porto
Av. 1º Maio 1080, Ed. Carvalhidos, Ent 3, 7ºEQ, 4600-013 Amarante

Lisboa
Rua da Conde d'Oleiros 1350-352 Lisboa

Contacte-nos:

+351 211 374 667
+351 220 046 850
askus@basicamente.io



Figura 10 - Página de Pedir Proposta (Versão Desktop)

Peça-nos agora uma proposta e saiba como o podemos ajudar a ter sucesso.

Ao trabalhar com a Basicamente contará com:

- Um parceiro de negócio;
- Um parceiro de negócio;
- Um parceiro de negócio;
- Um parceiro de negócio;

O seu nome: *

O seu contacto: *

O seu e-mail: *

A sua empresa: *

O seu budget disponível: *

Escolha uma opção
Escolha uma opção
500€ a 5.000€
5.000€ a 15.000€
15.000€ a 50.000€
> 50.000€

Descreva de que forma o podemos ajudar *

Pedir Proposta

Serviços

- Comércio Digital
- Marketing Digital e Performance
- UI/UX Design
- Websites e Workplaces Digitais
- Web Development

Onde estamos?

Porto
Av. 1.º Maio 1080, Ed. Carvalho, Ent 3, 7.ºEQ, 4600-013 Amarante

Lisboa
Rua Conde d'Óbidos 1350-352 Lisboa

Contacte-nos:

+351 211 374 667
+351 220 946 850
askus@basicamente.io



Figura 11 - Página de Pedir Proposta (Versão Mobile)

As ilustrações acima demonstram a página inicial do *website* e a página de pedido de proposta em versão *desktop* e *mobile*, com uma maior aproximação da realidade, ou seja, na fase de desenvolvimento as diferenças serão mínimas ou impercetíveis à exceção de alterações significativas estipuladas futuramente.

4.6. Estrutura do Projeto

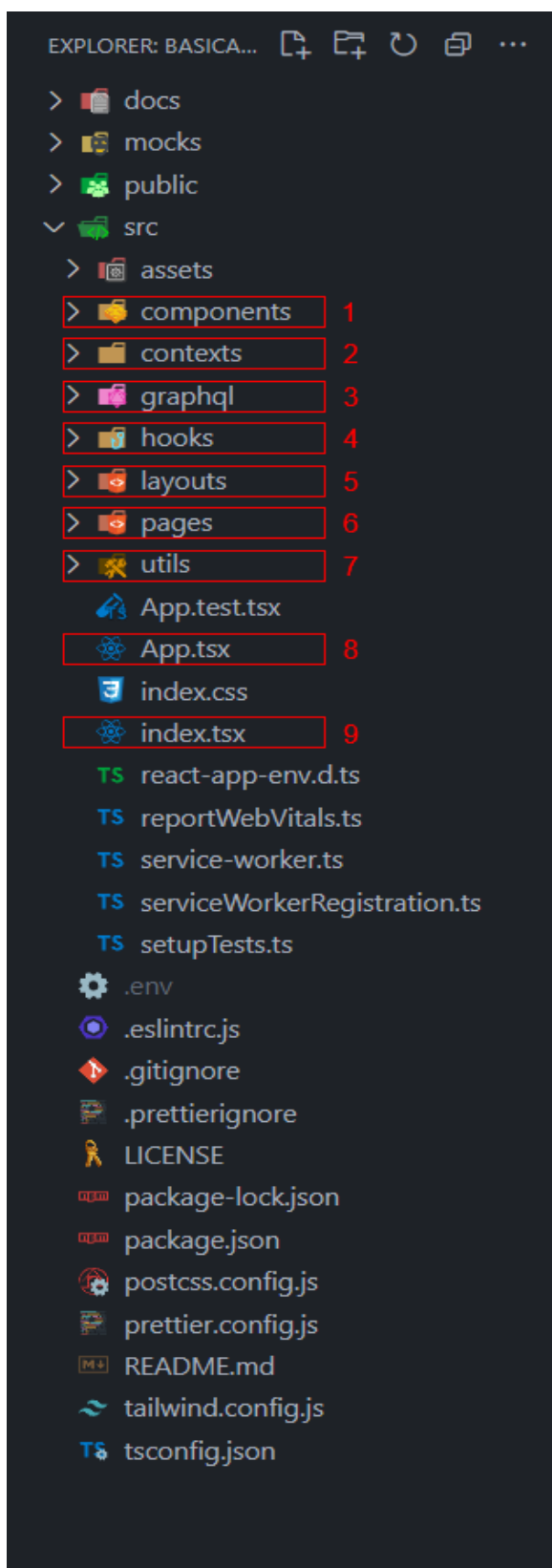


Figura 12 - Estrutura do projeto

1. Local onde os componentes reutilizáveis ficam armazenados, por ex. Botão, Navbar, Tabela de Dados, etc...
2. Local que contém os contextos da aplicação que possibilitam passar dados pela árvore de componentes sem recorrer a *prop drilling*
3. Local onde são armazenadas todas as queries GraphQL da aplicação
4. *Hooks* personalizados que auxiliam em determinadas tarefas
5. Local onde se encontra os *layouts* da aplicação
6. Páginas constituídas pelos seus respetivos componentes para se formarem
7. Local onde armazena pequenas lógicas e procedimentos úteis na aplicação em geral, por ex. devolver a data num determinado formato, transformar um *node* na respetiva tag HTML
8. Ficheiro onde está definido os *layouts* a utilizar com as respetivas rotas
9. Ficheiro *entry point* da aplicação e respetivo cliente Apollo

4.7. Cliente Apollo

De forma a usufruir da API em GraphQL do Contentful era necessário escolher uma biblioteca que suportasse tal feito como tal foi escolhida a biblioteca mais popular e abrangente o Apollo Client.

A screenshot of a code editor with a dark background and light-colored text. The code is written in TypeScript and shows the configuration of the Apollo Client. It includes the creation of an HttpLink with a URI and environment variables, an onError function for handling GraphQL and network errors, and the ApolloClient instance with the links and cache. Finally, it shows the ReactDOM.render call with the ApolloProvider and the application component.

```
1  const httpLink = new HttpLink({
2    uri: `https://graphql.contentful.com/content/v1/spaces/
3      ${process.env.REACT_APP_SPACE_ID}?access_token=
4      ${process.env.REACT_APP_ACCESS_TOKEN}`,
5  });
6
7  const errorLink = onError(({ graphqlErrors, networkError }) => {
8    if (graphqlErrors) graphqlErrors.forEach(({ message, locations, path }) =>
9      console.error(`[GraphQL error]: Message: ${message}, Location: ${locations}, Path: ${path}`));
10   if (networkError)
11     console.error(`[Network error]: ${networkError}`);
12 });
13
14 const client = new ApolloClient({
15   link: from([errorLink, httpLink]),
16   cache: new InMemoryCache(),
17   connectToDevTools: process.env.NODE_ENV === "development" ? true : false,
18 });
19
20 const root = ReactDOM.createRoot(
21   document.getElementById("root") as HTMLElement,
22 );
23
24 root.render(
25   <React.StrictMode>
26     <ApolloProvider client={client}>
27       <App />
28     </ApolloProvider>
29   </React.StrictMode>,
30 );
```

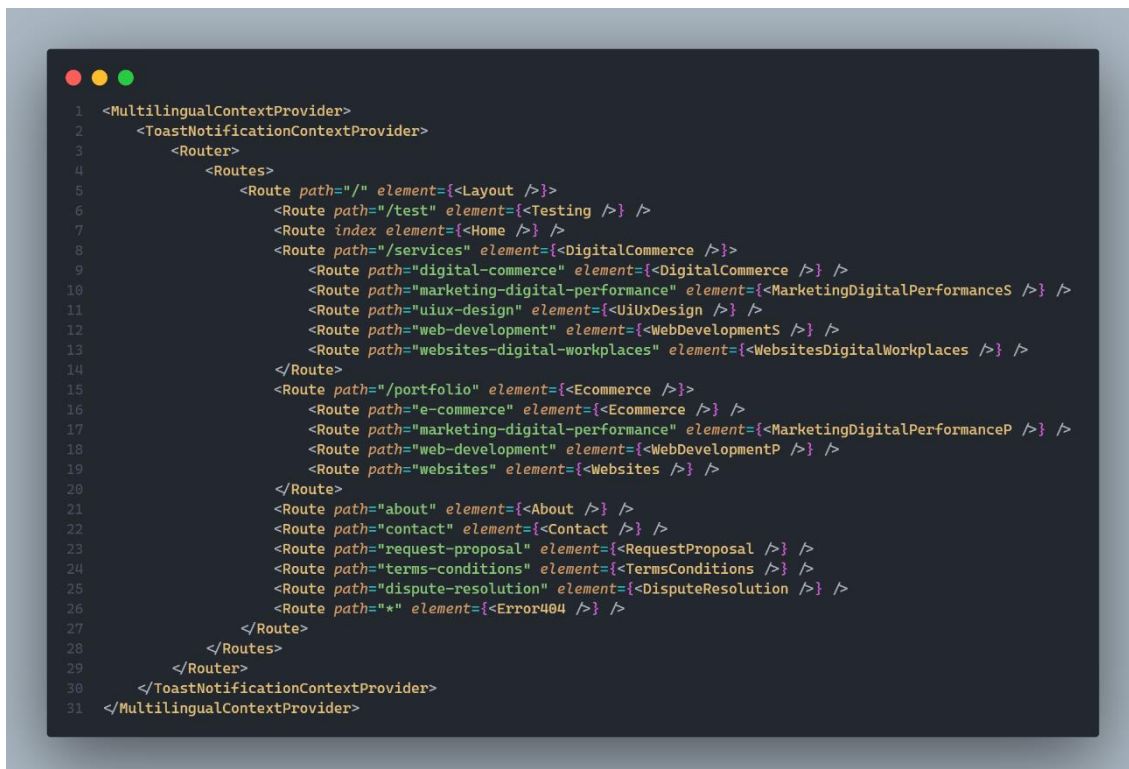
Figura 13 - Criação do cliente Apollo (ficheiro index.tsx)

Após a leitura da documentação oficial, visualização de tutorias e pequenos testes isolados em menos de 2 dias foi fácil a implementação e compressão do funcionamento na aplicação.

A ilustração acima demonstra a criação do cliente Apollo na aplicação com as chaves fornecidas pelo serviço do Contentful que irá permitir a aplicação comunicar e gerenciar dados remotos através da API de Conteúdos do Contentful com GraphQL.

4.8. Definição de Rotas

Na seguinte ilustração é possível observar todas as rotas criadas, através do React Router Dom uma coleção de componentes de navegação que compõem declarativamente a aplicação, tornando assim possível associar cada página ao seu respetivo caminho.



```
1 <MultilingualContextProvider>
2   <ToastNotificationContextProvider>
3     <Router>
4       <Routes>
5         <Route path="/" element={<Layout />} />
6         <Route path="/test" element={<Testing />} />
7         <Route index element={<Home />} />
8         <Route path="/services" element={<DigitalCommerce />} />
9           <Route path="digital-commerce" element={<DigitalCommerce />} />
10          <Route path="marketing-digital-performance" element={<MarketingDigitalPerformanceS />} />
11          <Route path="uiux-design" element={<UiUxDesign />} />
12          <Route path="web-development" element={<WebDevelopmentS />} />
13          <Route path="websites-digital-workplaces" element={<WebsitesDigitalWorkplaces />} />
14        </Route>
15        <Route path="/portfolio" element={<Ecommerce />} />
16          <Route path="e-commerce" element={<Ecommerce />} />
17          <Route path="marketing-digital-performance" element={<MarketingDigitalPerformanceP />} />
18          <Route path="web-development" element={<WebDevelopmentP />} />
19          <Route path="websites" element={<Websites />} />
20        </Route>
21        <Route path="about" element={<About />} />
22        <Route path="contact" element={<Contact />} />
23        <Route path="request-proposal" element={<RequestProposal />} />
24        <Route path="terms-conditions" element={<TermsConditions />} />
25        <Route path="dispute-resolution" element={<DisputeResolution />} />
26        <Route path="*" element={<Error404 />} />
27      </Routes>
28    </Router>
29  </ToastNotificationContextProvider>
30 </MultilingualContextProvider>
```

Figura 14 - Criação das rotas (ficheiro app.tsx)

Na linha 8 até a linha 14 (páginas relacionadas aos serviços) e na linha 15 até a linha 20 (páginas relacionadas ao portfólio) podemos observar as únicas *nested routes* da aplicação.

4.9. Multilingue

```
1  export type MultilingualContextType = {
2    language: string;
3    setLanguage: React.Dispatch<React.SetStateAction<string>>;
4    isPortuguese: () => boolean;
5    isEnglish: () => boolean;
6  };
7
8  export type MultilingualContextProviderType = {
9    children: React.ReactNode;
10 };
11
12 export const MultilingualContext = React.createContext<MultilingualContextType | null>(null);
13
14 export const MultilingualContextProvider = ({ children }: MultilingualContextProviderType) => {
15   const [language, setLanguage] = React.useState<string>(
16     localStorage.getItem("BasicamenteLang") || "pt-PT",
17   );
18
19   const isPortuguese = (): boolean => {
20     return language === "pt-PT";
21   };
22
23   const isEnglish = (): boolean => {
24     return language === "en-US";
25   };
26
27   return (
28     <MultilingualContext.Provider value={{ language, setLanguage, isPortuguese, isEnglish }}>
29       {children}
30     </MultilingualContext.Provider>
31   );
32 };
```

Figura 15 - Código fonte do contexto multilingue

Na ilustração n.º 15 demonstra o código fonte do contexto multilingue que torna possível em toda a aplicação saber o valor da língua selecionada auxiliando assim as consultas a devolver os dados na linguagem correta como também os dados estáticos.

4.10. Fetch dos dados em conformidade com a linguagem

```
1 const { language } = React.useContext(MultilingualContext) as MultilingualContextType;  
2 const { loading, error, data } = useQuery<FetchDataInt>(query, { variables: { language } });
```

Figura 16 – Uso do hook `useQuery` em conformidade com o valor multilingue atual

Como é possível observar na linha 1 da ilustração acima, utilizando a API de contexto nativa do React é nos armazenado o valor multilingue atual em uma constante.

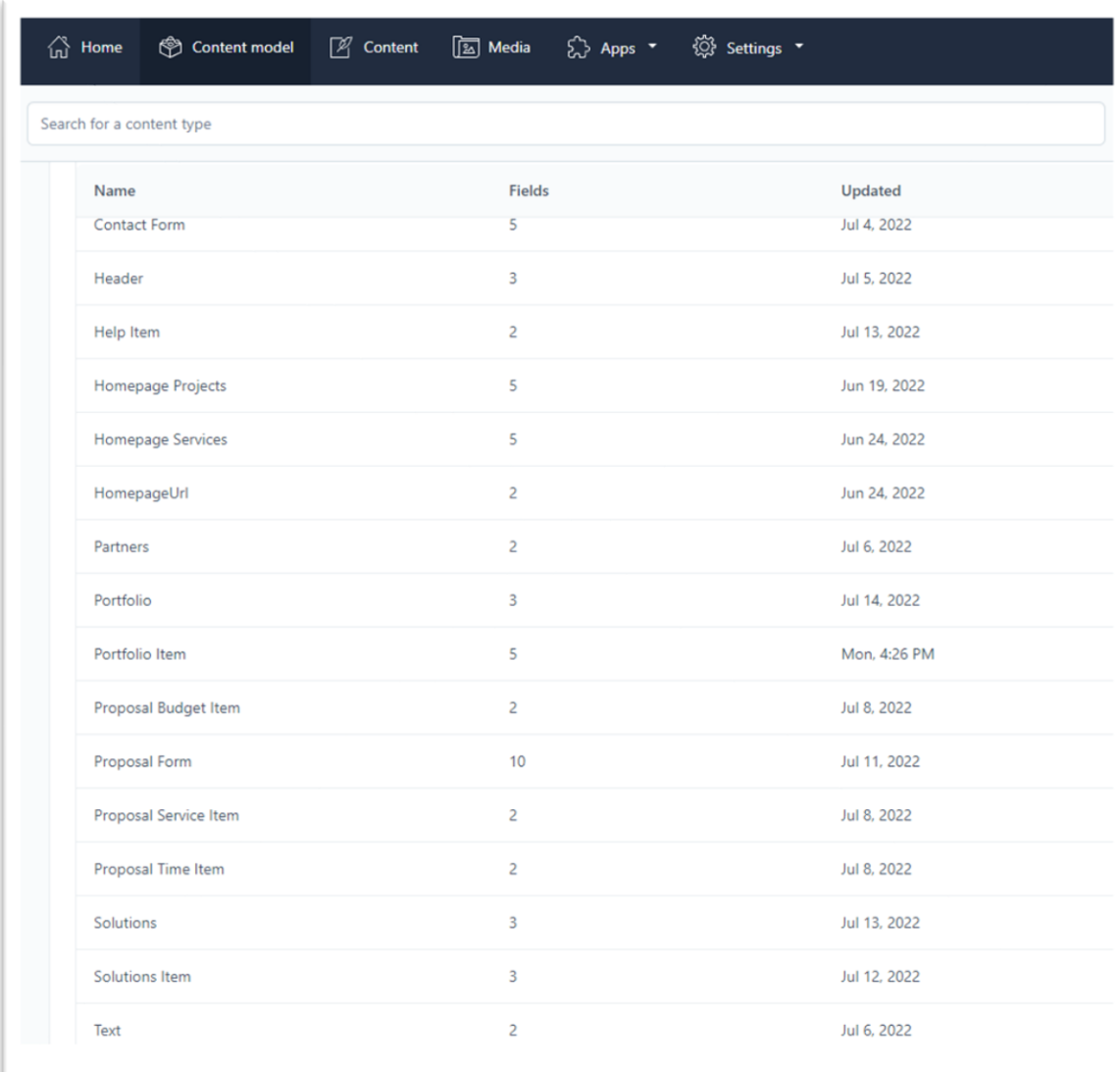
Logo após através do Apollo é possível usar o principal *hook* `useQuery` que permite executar as *queries* na aplicação, o mesmo recebe por parâmetro a respetiva *query* que é de carácter obrigatório e como parâmetro opcional o valor atual do contexto multilingue.

```
1 export const HOMEPAGE_HEADER = gql`  
2   query getHomepageHeaderContent($language: String) {  
3     header(id: "1U1rbuhNVpBRyFw1AMnGKZ", locale: $language) {  
4       title  
5       text {  
6         json  
7       }  
8       image {  
9         title  
10        description  
11        url  
12        sys {  
13          id  
14        }  
15      }  
16    }  
17  }  
18 `;
```

Figura 17 - Consulta GraphQL com valor da linguagem como parâmetro

Através da *query* acima ilustrada a API de conteúdos do Contetful irá devolver os respetivos dados na linguagem filtrada no parâmetro *locale*.

4.11. Definição dos modelos de conteúdos



Name	Fields	Updated
Contact Form	5	Jul 4, 2022
Header	3	Jul 5, 2022
Help Item	2	Jul 13, 2022
Homepage Projects	5	Jun 19, 2022
Homepage Services	5	Jun 24, 2022
HomepageUrl	2	Jun 24, 2022
Partners	2	Jul 6, 2022
Portfolio	3	Jul 14, 2022
Portfolio Item	5	Mon, 4:26 PM
Proposal Budget Item	2	Jul 8, 2022
Proposal Form	10	Jul 11, 2022
Proposal Service Item	2	Jul 8, 2022
Proposal Time Item	2	Jul 8, 2022
Solutions	3	Jul 13, 2022
Solutions Item	3	Jul 12, 2022
Text	2	Jul 6, 2022

Figura 18 - Modelos de conteúdo (Contentful Dashboard)

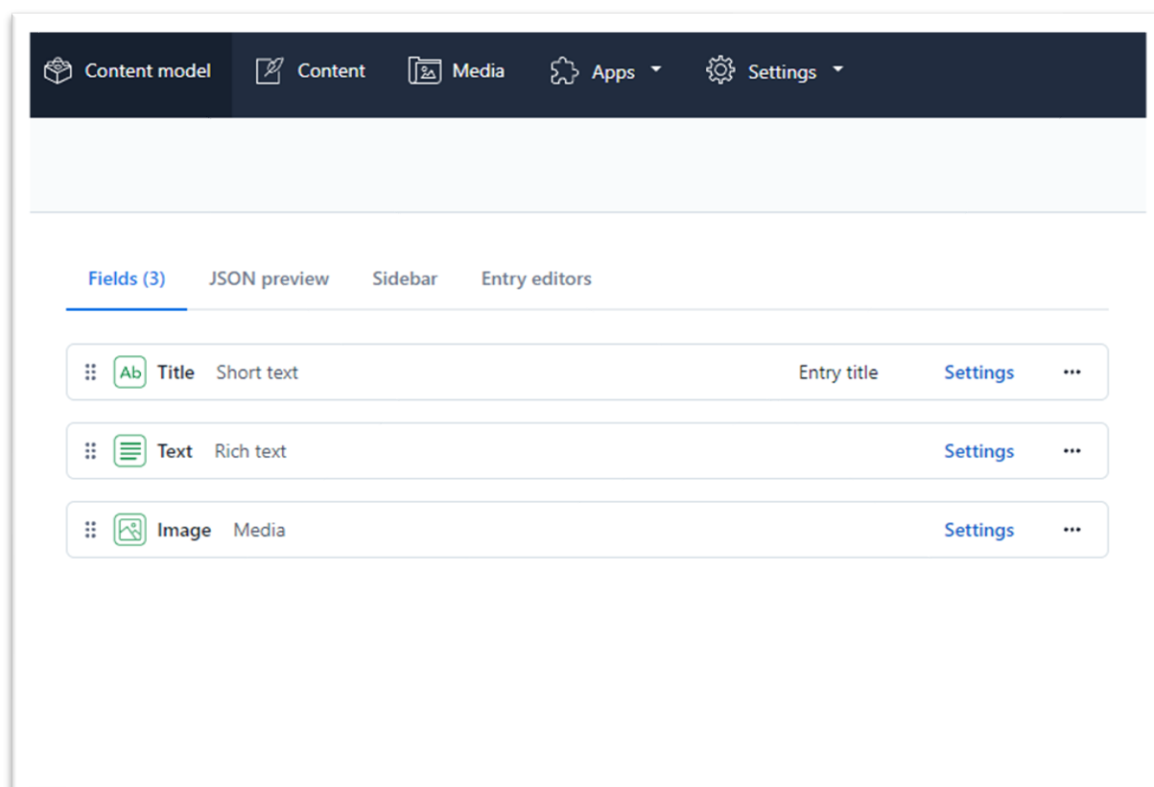


Figura 19 - Campos do modelo de conteúdo (Contentful Dashboard)

Através da plataforma Contentful os administradores têm a possibilidade de criar modelos de conteúdo, modelos esses que possuem determinados campos de um determinado tipo com determinadas regras e restrições.

Facilitando assim a inserção dos conteúdos por parte de utilizadores do tipo editor.

4.12. Envio de e-mails

Na seguinte ilustração é possível observar o formulário de contacto da aplicação criado e com os respetivos dados validados através da biblioteca de formulários React Hook Form.


É ainda possível observar a implementação do serviço reCAPTCHA da GOOGLE para uma melhor a segurança contra ataques automáticos.

The image shows a contact form on a blue background. The title is "Basicamente, vamos conversar?". Below it, a subtitle says "Teremos ao seu dispor uma equipa de profissionais disponível a ajudá-lo.". The form has several input fields: "Nome *" with the value "Sergio", "Sobrenome *" with "Felix", "E-mail *" with "sergiofelixdev@gmail.com", "Telefone *" with "916275619", and "A sua empresa *" with "Teste". There is a text area for "Conte-nos como podemos ajudá-lo *" containing the text "Isto é apenas um teste!". At the bottom left is a reCAPTCHA widget showing a green checkmark and the text "Não sou um robô". At the bottom right is a blue button labeled "Desejo ser contactado".

Figura 20 - Formulário de contato (com os dados preenchidos e validados)

Após o botão “Desejo ser contactado” ser premido os dados preenchidos no formulário são limpos e o reCAPTCHA volta ao estado de não verificado.

Edit Service

 Gmail

Personal Service | 500 emails per day

Name *

Gmail


Service ID *

service_sxocxgb

Gmail Connect

Connected as sergiofelixdev@gmail.com

Disconnect

 Allow "Send email on your behalf" permission during connection.
Both Gmail and Google Apps accounts are supported.

☒ Send test email to verify configuration

Cancel

Update Service

Figura 21 - Serviço SMTP utilizado (EmailJS Dashboard)

Após isso o serviço EmailJS através do SMTP e do *template* anteriormente criados e configurados na *dashboard* do EmailJS consoante a linguagem atual da aplicação irá enviar um e-mail para o administrador com os respetivos dados introduzidos no formulário como também enviar um email de resposta automática de confirmação para o e-mail introduzido no formulário.

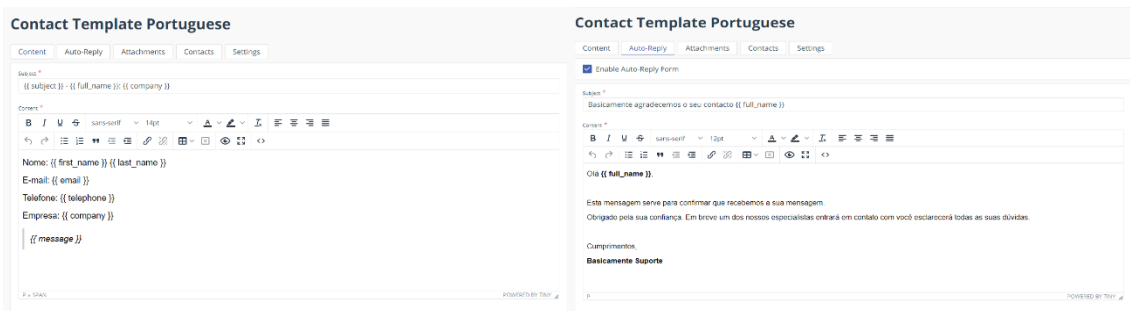


Figura 22 - E-mail template e e-mail Aauto-Reply Template (Dashboard do EmailJS)

Após todo processo anterior uma notificação no canto superior direito na aplicação é mostrada ao utilizador anónimo a informar para confirmar o seu e-mail para mais detalhes como é possível observar na imagem ilustrada abaixo.

Figura 23 - Formulário de contato submetido (pop-up aberta)

5. Gestão e acompanhamento do projeto

5.1. Planeamento

No início foi criado um cronograma semanal de março até agosto com uma aproximação do tempo que seria necessário para concluir as atividades estipuladas a desenvolver.

		2022																					
		Março		Abril				Maio					Junho					Julho				Agosto	
	Atividades	1	2	1	2	3	4	1	2	3	4	5	1	2	3	4	5	1	2	3	4	1	2
	Elicitação e análise de requisitos	1	2																				
	Especificação de requisitos			1	2																		
	Validação dos requisitos				1																		
	Documentos de requisitos					1	2																
	Prototipagem							1	2	3	4	5											
	Desenvolvimento												1	2	3	4	5	6	7	8			
	Testes Integrados																			1	2	3	
	Projeção p/ Produção																					1	2
	Outros																						1

Figura 24 - Cronograma de Desenvolvimento do Projeto Inicial

Durante o período de abril houve alguns atrasos mínimos cerca de uma semana na questão de especificação, validação e documentação dos requisitos que posteriormente afetou futuras tarefas nomeadamente os testes de software.

O mesmo aconteceu durante o desenvolvimento dos protótipos de alta-fidelidade mais concretamente no início, houve algumas dificuldades com a ferramenta de design Figma devido a necessidade de aprender os básicos, porém como de se tratar de uma ferramenta simples e intuitiva a curva de aprendizagem foi rápida.

Já durante o processo de desenvolvimento da aplicação como se era de esperar, com a envolvimento de tecnologias nunca anteriormente usadas, foi onde houve um maior tempo despendido com a necessidade de efetuar uma pesquisa aprofundada acerca da configuração, implementação, melhor abordagem relativamente a estrutura e melhores práticas das tecnologias.

		2022																					
		Março		Abril				Maio					Junho					Julho				Agosto	
Atividades		1	2	1	2	3	4	1	2	3	4	5	1	2	3	4	5	1	2	3	4	1	2
	Elicitação e análise de requisitos	1	2																				
	Especificação de requisitos			1	2																		
	Validação dos requisitos				1	2																	
	Documentos de requisitos					1	2	3															
	Prototipagem								1	2	3	4	5										
	Desenvolvimento													1	2	3	4	5	7	8	9		
	Testes Integrados																						
	Projeção p/ Produção																					1	
	Outros																						1

Figura 25 - Cronograma de Desenvolvimento do Projeto Final

Com todos estes contratempos a ilustração acima é a representação mais aproximada da realidade, do tempo despendido nas atividades a desenvolver.

5.2. Acompanhamento do projeto

Ao longo do projeto foram realizadas reuniões periódicas tanto com o orientador de estágio como também com o supervisor da empresa, de forma a validar e retificar não só o progresso do projeto como também o progresso do estágio em si.

Existindo a necessidade de esclarecimento de dúvidas ou resolução de problemas rápidos havia uma troca de e-mails para tal feito, caso contrário se fosse algo relacionado com a empresa houve visitas periódicas a empresa as segundas-feiras.

5.3. Ferramentas de gestão

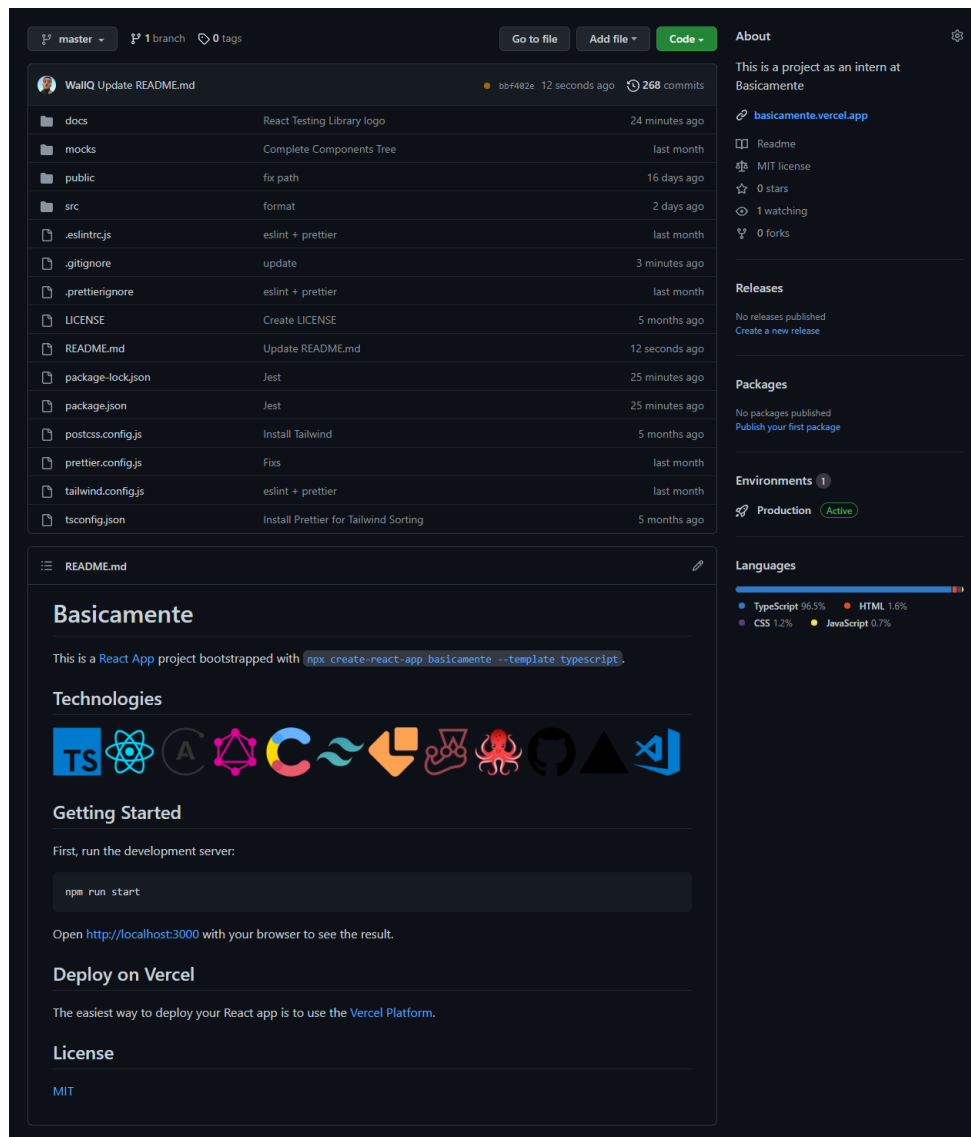


Figura 26 - Repositório do projeto no GitHub

Como restrição de negócio o código-fonte do projeto foi hospedado na popular plataforma GitHub, facilitando assim a gestão e o controlo de versões através do GIT.

6. Resultados

6.1. Cenários de validação

São apresentados de seguida os cenários das principais funcionalidades implementadas.

6.1.1. Pedir Proposta

Utilizador

Anônimo

Condições

Sem condições

Cenário

- Após a escolha da opção pedir proposta
- No formulário “Pedir Proposta”, preencher os campos com a respetiva informação do pedido
- Enviar pedido de proposta

Veredito

Verificou-se que após o envio de pedido de proposta foi mostrada uma mensagem indicando que o pedido foi submetido com sucesso e após alguns segundos chegou um e-mail com a informação que foi recebido um pedido de proposta e aguardar por resposta por parte da empresa.

6.1.2. Entrar em Contacto

Utilizador

Anônimo

Condições

Sem condições

Cenário

- Após a escolha da opção contacto
- No formulário “Entrar em Contacto”, preencher os campos com a respetiva informação do contacto
- Enviar contacto

Veredito

Verificou-se que após o envio do contacto foi mostrada uma mensagem indicando que o pedido de contacto foi submetido sucesso e após alguns segundos chegou um e-mail com a informação que foi recebido um contacto e aguardar por resposta por parte da empresa.

6.1.3. Mudar de Linguagem

Utilizador

Anônimo

Condições

Sem condições

Cenário

- Após a escolha da linguagem
- A página é recarregada
- Novos conteúdos são exibidos

Veredito

Verificou-se que após a escolha da linguagem a página foi recarregada mostrando uns *placeholders* no lugar dos conteúdos enquanto os novos não eram exibidos.

6.1.4. [Voltar para o topo](#)

Utilizador

Anônimo

Condições

Para o botão ficar visível é necessário existir um offset no eixo do Y de 300 pixels.

Cenário

- Efetuar o *scroll* na página
- Após um certo ponto o botão fica visível
- Clicar no botão

Veredito

Verificou-se que após clicar no botão a página moveu-se de forma suave novamente para o topo.

6.1.5. [Listagem de Serviços](#)

Utilizador

Anônimo

Condições

Sem condições

Cenário

- Após a escolha do tipo de serviço
- São exibidos os respetivos detalhes acerca do serviço

Veredito

Verificou-se que ao seleccionar um respetivo tipo de serviço os detalhes foram exibidos com sucesso contendo anteriormente um *placeholder* antes de serem carregados.

6.1.6. Listagem de Projetos

Utilizador

Anônimo

Condições

Sem condições

Cenário

- Após a escolha do tipo de projeto
- São exibidos os projetos anteriormente desenvolvidos

Veredito

Verificou-se que ao selecionar um respetivo tipo de projeto os mesmos foram exibidos com sucesso contendo anteriormente um *placeholder* antes de serem carregados.

6.1.7. Introdução de conteúdos

Utilizador

Administrador e/ou editor

Condições

Estar registado e autenticado na plataforma Contentful

Cenário

- Após a escolha do tipo de conteúdo
- No formulário, preencher os campos com a respetiva informação do tipo de conteúdo
- Submeter o conteúdo novo

Veredito

Verificou-se que ao introduzir um novo conteúdo dependendo do papel do utilizador o mesmo fica sujeito para aprovação e só após o mesmo ser aprovado ir então ser possível visualizá-lo no *website*.

6.2. Análise de resultados

Após a execução dos cenários descritos no ponto anterior, pode-se concluir que todos os objetivos foram cumpridos.

Com a seguinte base criada possuindo as funcionalidades mais urgentes, no futuro a solução poderá, no entanto, ter novas funcionalidades como por exemplo uma *dashboard* dos clientes atuais onde irão poder com mais facilidade entrar em contacto, agendar reuniões e/ou manutenções.

Porém, antes do desenvolvimento de novas funcionalidades será necessário desenvolver os testes da aplicação em falta, de forma a evitar que ocorram problemas no futuro quando o código for alterado.

Apesar de a aplicação segundo as métricas da ferramenta Lighthouse da Google possuir uma pontuação geral de 98 como mostra a imagem abaixo, a aplicação com um código mais refatorizado poderá ter ainda melhores resultados.

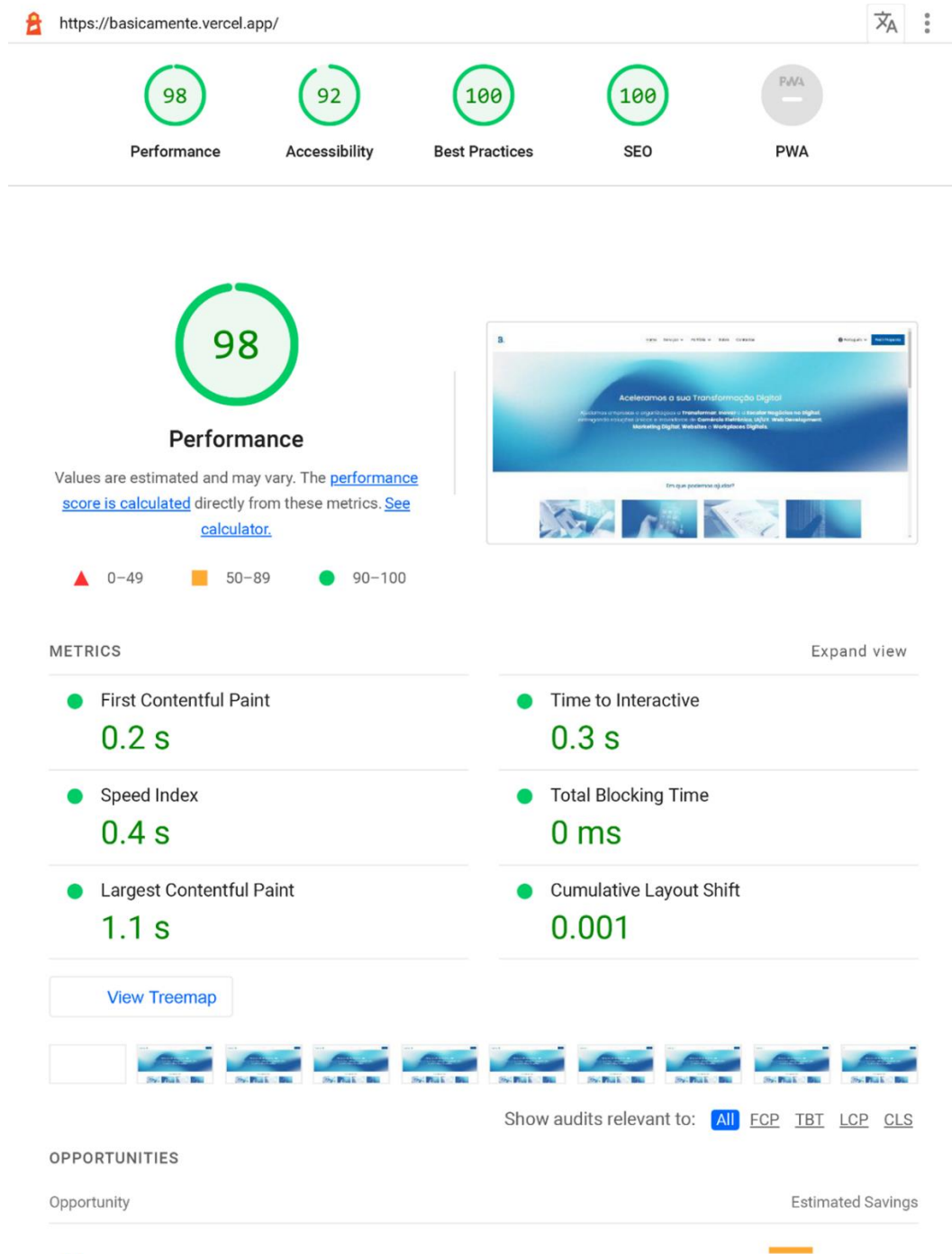


Figura 27 - Resultados do Lighthouse

7. Conclusões

Ao longo deste estágio curricular foi-me permitida a aprendizagem e o novo conhecimento de novas temáticas, aplicadas num contexto real de trabalho.

Foi possível o melhoramento dos conhecimentos tanto teóricos como práticos já existentes e a compreensão dos princípios técnicos de uma empresa, princípios esses que são uma mais-valia para a minha formação e para o sucesso durante este estágio curricular. Esses princípios passaram desde o trabalho em equipa, à seriedade e responsabilidade com que se trabalha para obter bons resultados, às regras da empresa e da própria equipa e aos métodos de trabalho.

O projeto na qual fui destacado contribuiu bastante para a minha evolução a nível de desenvolvimento *front-end* da web, uma área na qual não tinha muita experiência e conforto, mas com a ajuda dos colegas de trabalho e de autoaprendizagem, permitiu-me adquirir conhecimentos nesta nova área e evoluir na mesma, ao nível de num próximo desafio semelhante ultrapassar o mesmo sem grande dificuldade.

Acredito que a aplicação irá ser uma forte base para a empresa, através dela irá ser possível sem grandes complicações desenvolver novas funcionalidades, como por exemplo anteriormente referido uma *dashboard* para atuais clientes, dessa forma sinto-me realizado por uma parte do que poderá vir a se tornar a aplicação ter sido realizada por mim.

Por fim, posso concluir que este estágio curricular efetuado na Basicamente, correspondeu às expectativas e foi uma mais-valia para o meu futuro tanto a nível profissional como curricular, tendo aprendido bastante não só como aluno, mas também como trabalhador.

8. Bibliografia e Webgrafia

React Docs - reactjs.org/docs

React Docs Beta - beta.reactjs.org

Tailwind CSS Docs - tailwindcss.com/docs

TypeScript Docs - typescriptlang.org/docs

Contentful Docs - contentful.com/developers/docs/references/graphql

GraphQL Docs - graphql.org/learn

Apollo Docs - apollographql.com/docs

EmailJS - emailjs.com/docs

StackOverflow - stackoverflow.com

Contentful Youtube - youtube.com/c/Contentfulinc

PedroTech YouTube - youtube.com/c/PedroTechnologies

Midudev YouTube - youtube.com/c/midudev

Ben Awad YouTube - youtube.com/c/BenAwad97

9. Anexos



Basicamente.pdf



Components
Tree.pdf



Lighthouse Scoring
calculator.pdf



Lighthouse Report
Viewer.pdf