Atividade Prática de Desenvolvimento Curricular - Estágio
Relatório de Estágio

Identificação

Ano Letivo	2019/2020	**Semestre**	2º	**Estágio nº**	4217
Aluno nº	53052				
Nome	Gonçalo Furtado Mateus				
Título do estágio	Knowledge Base react editor (KBRE)				
Empresa	Altitude Software				
Coordenador	Ricardo Torres				
Orientador do DI	Armanda Rodrigues				

Monitorização do Período Introdutório (ver NOTAS**)** **Data:** 27/05/2020 **Local:** Remotamente

Objetivos e resultados esperados *(3000 caracteres)*:

Atualmente a aplicação AgentPortal da suite Altitude encontra-se a ser reformulada (migração de uma aplicação ASP.NET para uma Single Page Application). Esta aplicação, entre várias funcionalidades, possui um editor de Knowledge base[i] (Base de conhecimento).

No geral, este editor permite aos agentes dos contact center inserirem e modificarem documentos com informação especializada do negócio, de modo a que possa ser utilizada nas interações com os clientes, para um melhor atendimento e satisfação dos mesmos. Os documentos que este editor suporta são artigos, textos-rápidos e e-mails. Os textos-rápidos permitem a inserção de vários textos (por norma de dimensão pequena), e permitem a incorporação de um ou mais "Runtime Field" nos textos. Um "Runtime Field" é um texto criado a partir de campos de seleção e quando é inserido fica com uma cor de fundo amarelada (Anexo 2).

Para facilitar a organização, o editor contém várias categorias em que cada categoria pode ter subcategorias e cartões. Os cartões são resumos dos documentos que este editor suporta. O editor também possui um motor de busca e alguns filtros que podem ser aplicados, o que facilita a procura de documentos e categorias (Na secção 1.1 do anexo 1 pode ser observada a estrutura deste editor. Nesta secção é possível observar as categorias com as suas subcategorias e cartões. Também é possível observar o botão para ativar o motor de busca no canto superior direito). Assim, devido à estrutura que o editor tem e devido à facilidade de procura presente neste editor, torna-se bastante mais fácil o atendimento aos clientes. Assim, prevê-se que a satisfação dos clientes, em relação ao atendimento dos agentes dos contact center, aumente.

No seguimento desta transformação insere-se este estágio, onde será desenvolvido o editor de Knowledge base, completamente de raiz, mantendo as funcionalidades do antigo editor, mas com tecnologias mais recentes. Todo o desenvolvimento será feito através de tecnologias client-side, fazendo uso principal da linguagem React[ii], combinada com Typescript[iii]. A linguagem React baseia-se em componentes reutilizáveis, ou seja, cabe ao programador desenvolver várias componentes simples, para que as possa juntar criando componentes mais complexas. Outro benefício desta linguagem é a reutilização de componentes, o que para além de facilitar o desenvolvimento, permite que outros programadores possam reutilizar componentes existentes para outros projetos. Para este projeto, os desenhos (mock-ups) dos vários componentes foram previamente feitos pela equipa de design da Altitude, e foram-me entregues, para servirem de apoio no desenvolvimento dos componentes.

O desenvolvimento do projeto está subdividido em 3 fases:

- Criação dos vários componentes para o editor Knowledge Base em React + Typescript, com base nos desenhos (mock-ups) já feitos. Para isto será necessário fazer uso de técnicas de css[iv];
- Integração de Redux[v], para que os componentes React consigam aceder e guardar informação num local partilhado permitindo às componentes interagir umas com as outras de forma simples;
- Integração das componentes do editor desenvolvido na aplicação do agente portal.

Durante este estágio, é me proposto cumprir a primeira fase do projeto acima descrito. Para isto, será necessário decompor os componentes que me são propostos em componentes mais simples e genéricos e posteriormente criar o componente final a partir destes mais simples, de modo a que o resultado seja igual ao que era pretendido. Para além disto, será necessário desenvolver um relatório organizado em três fases, e juntamente com a entrega do relatório final (3ª fase), efetue uma apresentação de 15 a 20 minutos sobre o trabalho desenvolvido. Este estágio, devido à pandemia de covid-19, iniciou-se remotamente e prevê-se que o período remanescente do mesmo e a apresentação sobre o trabalho desenvolvido, também sejam feitas à distância. Neste estágio, tem-se recorrido à plataforma Teams[vi] para comunicar e realizar as reuniões.

Descrição das atividades realizadas no período introdutório *(3000 caracteres)*:

O primeiro dia de estágio iniciou-se com uma sessão de *onboarding* com todos os estagiários da FCT via teams, conduzida pelos recursos humanos da empresa. Nesta sessão foi nos feita uma breve apresentação sobre a empresa, que incluiu uma descrição de como a mesma estava estruturada em termos de equipas. Logo após esta sessão iniciei as formações, que tiveram a duração de duas semanas.

Após este período, e devido ao estágio estar a ser totalmente remoto, instalei vários programas no meu PC para ter acesso aos recursos da empresa. Foram instalados alguns programas, entre os quais o Visual Studio Code[vii] que tem sido o programa utilizado para programar os componentes e o sourcetree[viii], que serve para interagir com repositórios de git[ix]. Através do sourcetree descarreguei para o meu computador um repositório git com componentes já desenvolvidos na empresa, tornando assim a minha adaptação a linguagem bastante mais fácil.

Com as formações concluídas e com todos os programas necessários já instalados no meu PC, foi combinada uma reunião com as várias pessoas envolvidas neste projeto, na qual me foi descrito, em pormenor, o editor de Knowledge Base e me foram apresentados, pela equipa de design, os desenhos dos vários componentes que eu teria de desenvolver durante o estágio. Em seguida, deu-se início ao desenvolvimento dos vários componentes, sendo que inicialmente e para ter um ponto de partida, o meu coordenador indicou-me quais os componentes que deveria desenvolver primeiro.

O primeiro componente que desenvolvi foi um cartão que contém um resumo de um documento criado no editor. Este cartão inclui um botão que, quando clicado, abre uma página com as informações mais em detalhe do documento.

Em seguida, dei início ao desenvolvimento do menu principal do editor. Este menu tem várias categorias, em que cada categoria, quando clicada, abre as suas subcategorias e cartões associados. Este componente também inclui um motor de busca e alguns filtros que permitem aos agentes pesquisarem e encontrarem documentos facilmente. Como este componente era mais complexo, decidi subdividi-lo em alguns componentes genéricos mais simples (apresentados no anexo 1.2) e em seguida juntá-los, inserindo alguma lógica necessária, de modo a formar o menu pretendido (anexo 1.1). Foi implementado um componente para apresentar as categorias, outro para implementar o motor de busca e por fim um "Container" para englobar todos os componentes anteriormente desenvolvidos (Categorias, Cartões e motor de busca).

Posto isto comecei a desenvolver as várias páginas que permitem a criação e edição dos vários documentos que este editor suporta (artigos, templates e textos rápidos).

Aproximadamente 1 mês depois de ter iniciado o desenvolvimento de componentes para o editor, apresentei, pela primeira vez, o meu trabalho à equipa na reunião 'Sprint Review[x]'.

Dificuldades ou oportunidades encontradas, soluções adotadas, e sua justificação *(6000 caracteres)*:

Neste período introdutório não encontrei muitas dificuldades para além das esperadas. No primeiro componente que desenvolvi, encontrei algumas dificuldades na adaptação à linguagem React + Typescript, porque as formações foram mais à base de React com Javascript. A grande diferença, a nível de implementação, entre Javascript e Typescript

é que quando se usa Typescript necessitamos de definir os tipos e as interfaces para os dados que estamos a usar, tal como é feito em Java. Contudo, como me foi concedido acesso a componentes já desenvolvidos na empresa em React + Typescript consegui inicialmente basear-me nelas para entender melhor como implementar estas diferenças.

Outra dificuldade que encontrei foi no uso de CSS nos componentes. Apesar de já a ter utilizado anteriormente, não tinha muitas bases nesta tecnologia e, inicialmente, foi complicado perceber ao certo como funcionava e como poderia usar para obter o resultado esperado. Porém, na internet existe bastante informação disponível, que me ajudou nesta parte.

A segunda componente que desenvolvi, como referido acima, foi o menu principal do editor. Este menu contém as várias categorias que o utilizador do editor já criou, sendo que é possível que cada categoria, no máximo, possa ter 5 categorias filho. Cada categoria filho também pode ter mais 5 subcategorias, e por aí em diante até um máximo de 5 níveis de categorias. Inicialmente, quando o sistema arranca, apenas devem ser expostas as categorias principais e só quando o utilizador clicar numa dessas categorias serão abertas as categorias filho da mesma, e assim sucessivamente. No sentido inverso, se uma categoria tiver as suas categorias filho abertas e for clicada, o programa deverá fechar tudo o que está abaixo da categoria clicada. Para exemplificar, A,B,C são categorias, sendo que A é uma categoria principal, B uma subcategoria de A e C uma subcategoria de B. Inicialmente, apenas A está aberta, sendo que quando clicada abre B, e B quando clicado abre C. Neste momento, se A for clicada, a categoria B e C terão de ser fechadas. Para o desenvolvimento desta funcionalidade, encontrei algumas dificuldades em organizar que dados deveriam ser associados a cada categoria. Após análise, concluí que cada categoria teria, para além do nome, um identificador (id), uma referência à categoria pai (a indicar o id da categoria pai caso exista, ou -1 caso contrário), um número inteiro que indica o nível da categoria e uma variável booleana a dizer se a categoria está aberta ou não. A variável nível, serve apenas para pôr o ícone correto na categoria, de acordo com o seu valor e, a variável booleana, para indicar em cada categoria, se o ícone a apresentar, é para expandir ou contrair as categorias filho.

Para além disto, no geral, considero que a maior dificuldade esteja a ser o facto de o estágio estar a ser remoto. Apesar de estar a correr bem e os objetivos estarem a ser cumpridos, no meu entendimento penso que se perde alguma dinâmica de trabalho e, por vezes, o nível de produtividade é menor, devido a estar a trabalhar em casa. No entanto, algo de positivo é que, por norma, sempre que tenho uma dúvida maior tenho tido ajuda, principalmente por parte do meu coordenador.

Plano de trabalhos *(3000 caracteres)*:

O estágio teve início no dia 1 de abril e o plano de trabalhos foi definido da seguinte forma:

- Estudo do editor atual e Formação em React/Redux e Typescript – 1ª e 2ª semanas
- Desenvolvimento dos componentes do editor Knowledge Base – 3ª - 17ª semanas

  Lista dos vários componentes a desenvolver para o editor Knowledge Base (Para desenvolver alguns dos componentes listados em seguida, será necessário primeiro criar componentes mais simples e genéricos, e posteriormente, criar o componente final a partir destes):

  1. Cartão resumo – Cartão que contém o resumo de um documento presente no editor (Presente no anexo 1.2).
  2. Menu principal – Menu que incluí as várias categorias, subcategorias, cartões, motor de busca e filtros que este editor incluí (Anexo 1.1).
  3. Página para selecionar que documento se quer criar (Anexo 6).
  4. Página para criar artigos (Anexo 7).
  5. Página para criar textos-rápidos (Anexo 2.1).
  6. Página para responder a e-mails (Anexo 4).
  7. Página para criar e-mails (Anexo 8).
  8. Página para editar texto (Anexo 9).
  9. Página com um resumo do documento criado (Anexo 10).

- Apresentação e demonstração, para toda a equipa, do trabalho desenvolvido neste estágio. Esta apresentação, tem como principal objetivo explicar, em detalhe, o código desenvolvido, para tornar mais fácil a futura alteração e modificação a outros programadores – 1 dia
- Relatório final e apresentação final – 17ª - 20ª semanas

Neste momento, o estágio encontra-se no segundo ponto do plano de trabalhos, tendo-se já desenvolvido os 4 primeiros componentes.

Prevê-se que durante este estágio, de forma a validar e avaliar os componentes desenvolvidos, vá apresentando os meus progressos para toda a equipa nas reuniões quinzenais ("Sprint Review"). Nestas reuniões, irei receber feedback por parte da equipa na qual estou inserido e, caso necessário, serão discutidas alterações ou melhorias que possam ser feitas nos componentes já desenvolvidos. Para além disto, prevê-se a realização de alguns "*Code Reviews*" com o meu coordenador, para receber feedback do trabalho desenvolvido e melhorar o código.

Formação recebida e conhecimentos adquiridos *(máx. 3000 caracteres)*:

Devido a pandemia do covid-19 em Portugal, o início do estágio atrasou-se. Contudo, o meu coordenador enviou-me informação sobre as formações previamente, para que fosse possível começar a adiantá-las enquanto o estágio não se iniciava. Assim, as formações foram vistas com calma, o que facilitou a minha adaptação à linguagem React. Estas formações consistiram em dois cursos online da plataforma *Udemy*[xi], um de React/Redux em JavaScript e outro de Typescript. O curso de Typescript não foi muito complicado, pois esta linguagem é semelhante a JavaScript mudando apenas pequenas coisas, entre as quais, os tipos das variáveis e interfaces. A formação em React/Redux foi mais complicada porque foi algo completamente novo para mim. Porém, o curso que me foi fornecido era bastante completo e ajudou-me em grande parte a perceber React/Redux em Javascript. No entanto, as componentes que eu viria a desenvolver seriam em React com Typescript e visto que o curso de React que tinha feito utilizava JavaScript, foi me dado acesso a um repositório git com componentes já desenvolvidas na empresa em React + Typescript, para que pudesse ver exemplos de componentes já desenvolvidas e assim perceber as grandes diferenças que existem quando se utiliza React com Javascript e Typescript.

Neste estágio também tenho aprendido a interagir com repositórios git que era algo que nunca tinha feito anteriormente. Com o git, assegura-se que nenhum do trabalho desenvolvido é perdido. Para isso, tenho trabalhado numa "*branch*" minha onde vou fazendo os vários "*commit*" dos avanços que faço[xii]. Para auxiliar este processo todo, tem-se recorrido ao sourcetree, que é um programa que torna a interação com o git mais simples.

A metodologia de trabalho da equipa em que estou integrado (R&D TEAM) é Agile[xiii]. Esta metodologia baseia-se em reuniões quinzenais chamadas sprints ("Sprint Review") em que são apresentados os desenvolvimentos nos projetos, sendo depois apresentado feedback por parte da restante equipa. Para além desta reunião quinzenal, existem reuniões scrum[xiv] diárias, onde os vários elementos da equipa falam brevemente sobre o que fizeram no dia anterior e irão fazer no presente dia. Como o estágio se encontra a ser totalmente remoto, estas reuniões foram importantes para a minha integração na equipa, pois permitiram dentro do possível ter um contacto mais próximo com a mesma e manter a motivação.

Para além das reuniões, de quinze em quinze dias, são definidas tarefas para cada pessoa. Estas tarefas são os vários objetivos que cada pessoa se propõe a cumprir nos próximos quinze dias. Para isto, recorre-se à plataforma JIRA[xv], onde cada pessoa tem as suas respetivas tarefas e à medida que as inicia, altera o seu estado para "em progresso" e quando finaliza, coloca-as como "finalizada"[xvi]. O objetivo é que passados os quinze dias, todas as tarefas estejam concluídas. Este sistema ajuda a que tenhamos objetivos fixos e trabalhemos para os conseguir cumprir.

Monitorização Intercalar (ver NOTAS**)** **Data:** 03/07/2020 **Local:** Remotamente

Descrição técnica preliminar *(máx. 4000 caracteres)*

Como já referido anteriormente o objetivo principal deste estágio é o desenvolvimento de componentes React para serem usadas no novo editor de Knowledge Base da aplicação AgentPortal da suite altitude. O trabalho foi subdividido em várias componentes mais simples, para que depois se pudessem juntá-las e utilizá-las em conjunto. Inicialmente, os desenhos (mock-ups) dos componentes foram realizados pela equipa de design e competia-me a mim reproduzi-los em componentes React de acordo com as funcionalidades pretendidas. Estes desenhos foram-me apresentados numa plataforma designada Invision[xvii], na qual é possível ver algum CSS, como o tamanho de letra, a cor de fundo, entre outros (como é demonstrado no anexo 11), de modo a auxiliar no desenvolvimento dos componentes.

O desenvolvimento das componentes tem sido feito em Visual Studio Code, sendo o código testado através de um StoryBook[xviii]. Um StoryBook é, de um modo geral, um ficheiro com várias subpastas que contêm alguns componentes já desenvolvidos pela empresa, o que torna o desenvolvimento dos mesmos mais simples, devido à sua organização e eficiência.

As tecnologias usadas até ao momento foram React + Typescript fazendo-se uso recorrente a componentes React da biblioteca Material-UI[xix]. Esta biblioteca possui componentes já desenvolvidos como campos de textos, campos de seleção entre outros, tornando o desenvolvimento dos componentes mais ágil e fácil. Para além desta biblioteca, fez-se uso de um componente do repositório NPM[xx] que permitiu apresentar os destinatários dos emails com o layout pretendido[xxi] (como é mostrado no anexo 4). Outra tecnologia que se fez uso neste projeto, foi o CKEditor[xxii], que será falado mais ao detalhe em baixo.

O git tem sido uma tecnologia com uso recorrente, pois assegura que haja um backup do trabalho desenvolvido. Para o uso do git, tem se feito uso a uma plataforma chamada sourcetree. Esta plataforma funciona como uma interface para interagir com repositórios do git, tornando esta interação muito simples, prática e rápida.

O estágio teve um avanço bastante mais rápido do que inicialmente foi previsto, tendo o objetivo inicial ficado cumprido ao fim de 13 semanas. Devido a isso, acrescentou-se um terceiro ponto em relação ao plano inicial que é a integração de Redux nos componentes já desenvolvidos. Até ao final do estágio, o objetivo irá passar por fazer esta integração nos vários componentes desenvolvidos, permitindo assim concentrar o estado de um programa/componentes num só lugar em vez de estar espalhado pelos vários componentes. A utilização do Redux, irá também tornar mais fácil, a interação entre os vários componentes.

Atualização do plano de trabalhos *(máx. 3000 caracteres)*:

Como os objetivos inicialmente definidos para este estágio foram cumpridos antes do tempo, houve uma modificação no plano de trabalhos, acrescentando-se o 3º ponto em relação ao plano inicial:

- Estudo do editor atual e Formação em React/Redux e Typescript – 1ª e 2ª semanas
- Desenvolvimento dos componentes do editor Knowledge Base – 3ª - 13ª semanas

  Lista dos vários componentes a desenvolver para o editor Knowledge Base (Para desenvolver alguns dos componentes listados em seguida, será necessário primeiro criar componentes mais simples e genéricos, e posteriormente, criar o componente final a partir destes):

  1. Cartão resumo – Cartão que contém o resumo de um documento presente no editor (Presente no anexo 1.2).
  2. Menu principal – Menu que incluí as várias categorias, subcategorias, cartões, motor de busca e filtros que este editor incluí (Anexo 1.1).
  3. Página para selecionar que documento se quer criar (Anexo 6).
  4. Página para criar artigos (Anexo 7).
  5. Página para criar textos-rápidos (Anexo 2.1).
  6. Página para responder a e-mails (Anexo 4).
  7. Página para criar e-mails (Anexo 8).
  8. Página para editar texto (Anexo 9).
  9. Página com um resumo do documento criado (Anexo 10).

- Integração de Redux nos componentes já desenvolvidos – 14ª - 19ª semanas
- Apresentação e demonstração, para toda a equipa, do trabalho desenvolvido neste estágio. Esta apresentação, tem como principal objetivo explicar, em detalhe, o código desenvolvido, para tornar mais fácil a futura alteração e modificação a outros programadores – 1 dia
- Relatório final e apresentação final – 17ª - 20ª semanas

Neste momento, está-se a iniciar o terceiro ponto do plano de trabalhos.

Desvios ao plano inicial, dificuldades endereçadas e sua justificação *(máx. 3000 caracteres)*:

Neste estágio até ao momento não houve nenhum desvio do plano inicial, tendo sido tudo executado como pretendido. Para avaliação dos resultados dos componentes que tenho estado a desenvolver, têm sido feitos alguns "*Code Reviews*" com o meu coordenador, de modo a receber feedback do trabalho já desenvolvido e melhorar o código. Para além disso, tenho apresentado regularmente os meus avanços para a equipa toda, nas reuniões quinzenais chamadas sprints ("Sprint Review"), e quando necessário, são discutidas alterações ou melhorias, que possam ser feitas nos componentes que foram desenvolvidos.

Durante a implementação dos componentes foram surgindo algumas dificuldades. A primeira grande dificuldade encontrada foi durante o desenvolvimento do componente para criar textos rápidos ("Quick Texts") no editor (Anexo 2). Este componente oferece uma caixa de texto onde o utilizador pode inserir texto (como mostrado no anexo 2.2) e, se pretender, pode clicar no botão "Runtime Field". Ao clicar neste botão, é aberta uma caixa de diálogo que contém campos de seleção. De acordo com o que o utilizador selecionar, é criado o "Runtime Field" na posição onde o cursor do rato se encontra, ficando o "Runtime Field" com uma cor de fundo amarela (como é mostrado no anexo 2.3). Estando o "Runtime Field" inserido é possível apagá-lo e editá-lo.

O primeiro problema ao fazer este componente, foi saber em que posição se encontrava o cursor do rato. Para isto, tive de recorrer a alguma documentação na internet, onde encontrei alguns exemplos que me ajudaram. Este problema foi ultrapassado recorrendo a uma função que o React fornece chamada "Refs[xxiii]". Esta função permite obter uma referência de um elemento criado. A partir desta referência, consegue-se obter muito mais informação sobre o elemento em questão, como a posição do cursor do rato. No meu caso, criei uma referência para o elemento que permite a inserção de texto, através da função "*React.createRef<HTMLTextAreaElement>()*".

Com esta primeira dificuldade superada, já foi possível dividir o texto em duas partes e inserir um "Runtime Field" no local pretendido. Contudo, como não existem caixas de texto redimensionáveis em largura, de acordo com o seu conteúdo, surgiu outra adversidade, pois quando se edita o texto que está a esquerda do "Runtime Field" (no caso do anexo 2.2 a palavra "Welcome"), tudo o que está para a frente do mesmo tem de se mover de forma uniforme. Por exemplo, se fosse apagado texto da palavra "Welcome" tudo o que está para a frente da mesma iria recuar proporcionalmente. Este problema foi superado através do uso de uma " " auxiliar, com visibilidade ocultada ("*hidden*[xxiv]"), que permitiu em cada momento calcular o tamanho que a palavra à esquerda do "Runtime Field" estava a ocupar. Tendo este cálculo feito, tornou-se mais simples desenvolver este componente.

A Segunda grande dificuldade encontrada, foi a inserção do CKEditor nos componentes que necessitavam do mesmo. O CKEditor é um editor de texto WYSIWYG ("*What You See Is What You Get*") implementado em JavaScript, sendo que nos meus componentes utilizei-o para inserir uma barra de edição ("*Toolbar*") que permite que quando um utilizador insere texto possa definir os vários tipos de letra, tamanho, etc… à semelhança do Word, mas mais simples (como é mostrado no anexo 3).

Para a utilização do CKEditor, é necessário descarregar um pacote e inseri-lo na pasta pública do projeto. Contudo, este projeto está a ser desenvolvido num StoryBook da empresa que não tem uma pasta pública, algo necessário neste caso. Foram tentadas várias abordagens, sobre onde esta pasta poderia ser colocada, mas chegou-se à conclusão que não seria possível usar o CKEditor desta forma. Após análise, decidiu-se que, ao invés de termos um pacote com tudo o que o CKEditor necessita para trabalhar, iríamos buscar os recursos necessários através da definição de um URL. O único problema desta abordagem é que, em situações em que não haja ligação à internet, os componentes que usam o CKEditor podem apresentar alguns defeitos. Contudo, como neste caso, os componentes que estão a ser desenvolvidos são para uma página web, este problema perdia a importância, pois irá sempre existir internet, quando os componentes forem inicializados.

Avaliação Final **Data:** 31/07/2020 **Local:** Remotamente

Descrição final do trabalho efetuado:

Objetivos *(máx. 3000 caracteres)*

O objetivo principal deste estágio foi o desenvolvimento de vários componentes para serem utilizados mais tarde na aplicação AgentPortal da suite Altitude. Para isso, foram-me concedidos os desenhos (mock-ups) dos mesmos pela equipa de design e competiu-me a mim reproduzi-los em React + Typescript com as funcionalidades desejadas. Para o desenvolvimento dos componentes mais complexos, o objetivo passou por dividi-los em componentes mais simples (como é mostrado no anexo 1) e depois através destes, construir os componentes finais com as funcionalidades pretendidas. A razão pela qual se optou por construir os componentes desta forma foi porque, assim, tornamos as componentes mais fáceis de serem reutilizadas. Ou seja, como estes componentes estão implementadas de uma forma genérica, se alguém, mais tarde, precisar de utilizar um componente semelhante ou igual, não necessitar de o desenvolver de raiz. Por fim, para desenvolver os componentes, foi necessário recorrer a técnicas de CSS para que os componentes ficassem com o aspeto que inicialmente tinha sido pensado pela equipa de design e recorrer a algumas frameworks, como o Material-UI e o CKEditor.

Como a parte do desenvolvimento de componentes teve um avanço mais rápido do que inicialmente tinha sido previsto, acrescentou-se a este estágio o objetivo de inserir Redux nos componentes já desenvolvidos. O Redux permite concentrar o estado de um programa/componentes num só lugar, em vez de estar espalhado pelos diversos componentes, o que acaba por facilitar a interação entre os vários componentes.

Após a inserção de Redux nos componentes, e devido ao tempo que ainda faltava para acabar o estágio, foi me pedido que desenvolvesse um componente para ser utilizado na página de administração do editor de Knowledge base. Este componente tem como objetivo apresentar as várias categorias que o editor tem e permitir editar, apagar, mover e criar categorias. Ou seja, este componente permite criar categorias, alterar os nomes das categorias, remover categorias e alterar o local onde cada categoria se encontra. Para além disto, este componente permite ordenar as categorias por nome e, inclui um motor de busca e alguns filtros para facilitar a procura de categorias.

Descrição técnica e seu contexto *(máx. 10000 caracteres)*

Na altitude, a aplicação AgentPortal encontra-se a ser reformulada. Uma das funcionalidades desta aplicação é o editor de Knowledge Base. Este editor permite aos agentes dos contact center inserirem e modificarem documentos com informações especializadas nos vários negócios. Assim, a interação com os clientes torna-se mais fácil e rápida pois o acesso aos vários documentos (artigos, emails e textos rápidos) encontra-se apresentado de forma simples e acessível. Com isto, é possível ter um melhor atendimento aos clientes, aumentando a satisfação dos mesmos.

No seguimento desta transformação insere-se este estágio, onde me foi proposto que desenvolvesse os componentes, completamente de raiz, para serem utilizados neste novo editor de Knowledge base, mantendo as funcionalidades do antigo editor, mas com tecnologias mais recentes. Primeiramente, foram-me apresentados os desenhos (mock-ups) dos vários componentes e, o objetivo do estágio, passou por reproduzir esses desenhos em componentes React, com todas as funcionalidades pretendidas.

Foram desenvolvidos componentes para listar as várias categorias, em que cada categoria pode ter cartões (que contêm o resumo de um documento) e subcategorias (como mostrado no anexo 1). Este componente também possui um motor de busca que, de acordo com o texto inserido, apresenta todas as categorias que contenham esse texto, facilitando a procura aos agentes dos contact center. Para além disto, foram desenvolvidos componentes que permitem criar e editar os vários tipos de documentos que este editor suporta. Devido ao tempo extra que houve neste estágio, foram desenvolvidos mais alguns componentes para permitir aos agentes criarem, moverem, editarem e apagarem categorias (como é mostrado no anexo 12), e foi inserido Redux nos componentes que tinham sido desenvolvidos. A inserção de Redux, leva a que toda a informação e lógica dos componentes seja centralizada num só local, em vez de estar espalhada internamente pelos diversos componentes. Com Redux inserido, é mais fácil testar e encontrar erros na aplicação, pois é simples observar quando, onde e como o estado dos componentes se está a alterar.

Para o desenvolvimento dos vários componentes que me foram propostos, inicialmente, tive de realizar uma análise aos desenhos que me foram concedidos, com o objetivo de conseguir efetuar uma divisão dos componentes

mais complexos, em componentes mais simples. Em seguida, deu-se início ao desenvolvimento destes componentes mais simples, sendo que estes foram implementados de uma forma genérica. Estes componentes têm como objetivo serem totalmente configuráveis, com o propósito de poderem ser utilizados não só no contexto em que estão a ser desenvolvidos, mas em muitos outros. Posteriormente, e depois de os componentes mais simples já estarem criados, deu-se início à construção dos componentes, finais através da utilização dos componentes mais simples, e adicionando alguma lógica quando necessário, de modo a que o resultado seja igual ao que era pretendido.

Outro desafio que surgiu durante este estágio foi que formato usar para guardar as várias categorias, de modo a ser simples e eficiente. Inicialmente, pensou-se em ter um vetor de objetos, em que cada objeto tinha as informações de uma categoria (nome, data, etc.), um identificador único e uma referência à categoria pai caso tivesse, ou -1 caso contrário. Contudo, com o decorrer do estágio, chegou-se à conclusão de que, apesar de esta solução funcionar, não era a mais adequada pois dificultava a inserção e remoção de categorias, algo que era necessário nos componentes da parte de administração do editor de Knowledge Base que foram desenvolvidos (anexo 12). De forma a solucionar este problema, optou-se por guardar as categorias num esquema em árvore, em que cada categoria tem as suas informações, bem como um vetor de categorias filho. Posteriormente, cada filho pode inclusive ter filhos ou não (como é demostrada no anexo 5).

Com este novo esquema, evitou-se guardar variáveis desnecessárias (como a referência à categoria pai) e simplificou a implementação dos componentes ao tornar mais simples editar, modificar e apagar as informações de cada categoria. Devido a esta mudança de formato, foi necessário alterar alguns componentes já desenvolvidos (como o menu principal do editor (anexo 1)), para que estes componentes passassem a funcionar com este novo formato.

Todo o desenvolvimento dos componentes durante este estágio fez-se em Visual Studio Code, tendo o código sido testado através de um StoryBook. Um StoryBook é, de um modo geral, um ficheiro com várias subpastas que contêm alguns componentes já desenvolvidos pela empresa, e torna o desenvolvimento das mesmas mais simples devido à sua organização e eficiência. A principal linguagem de programação usada neste estágio foi React combinado com Typescript. No início do estágio a adaptação a esta linguagem foi complicada, pois é bastante diferente das que estava habituado. Contudo, à medida que fui desenvolvendo os componentes, esta linguagem tornou-se bastante mais simples e acessível.

Para o desenvolvimento deste projeto, fez-se uso recorrente a componentes React da biblioteca “Material-UI”. Esta biblioteca foi bastante útil durante o desenvolvimento dos componentes, pois possui vários componentes já desenvolvidos como campos de textos e campos de seleção. Fez-se uso também de um componente do repositório NPM, para apresentar os destinatários dos emails com o layout pretendido (como é mostrado no anexo 4). Neste projeto também se fez uso do CKEditor, para inserir uma barra de edição (como é mostrado no anexo 3), que permitiu ao utilizador inserir texto com vários tipos de letra, tamanho, etc.

Por último, o git foi uma ferramenta que se recorreu constantemente, para assegurar que o trabalho desenvolvido nunca fosse perdido. Para o uso do git, fez-se uso de uma plataforma chamada sourcetree. Esta plataforma funciona como uma interface para interagir com repositórios do git, tornando a interação muito simples, prática e rápida.

Estado de desenvolvimento final das tarefas do Plano de Trabalhos *(máx. 3000 caracteres)*:

Os objetivos que inicialmente foram definidos para este estágio foram cumpridos antes do tempo, tendo-se no plano de trabalhos intercalar, acrescentado a tarefa de integrar Redux nos componentes que tinha desenvolvido. Esta nova tarefa, também teve um avanço rápido e, como quando foi concluída, ainda faltavam 3 semanas para o final do estágio, foi me proposto que desenvolvesse mais um componente para ser utilizado na parte administrativa do editor de Knowledge Base. No fim do estágio, quer o objetivo principal do estágio, quer os objetivos extra que surgiram, foram cumpridos com sucesso. (No anexo 13, encontra-se um cronograma do plano de trabalhos final)

A duração das várias tarefas, durante este estágio, foram as seguintes:

- Estudo do editor atual e Formação em React/Redux e Typescript – 1ª e 2ª semanas
- Desenvolvimento dos componentes do editor Knowledge Base – 3ª - 13ª semanas

  Lista dos vários componentes a desenvolver para o editor Knowledge Base (Para desenvolver alguns dos componentes listados em seguida, será necessário primeiro criar componentes mais simples e genéricos, e posteriormente, criar o componente final a partir destes):

  1. Cartão resumo – Cartão que contém o resumo de um documento presente no editor (Presente no anexo 1.2).

2. Menu principal – Menu que incluí as várias categorias, subcategorias, cartões, motor de busca e filtros que este editor incluí (Anexo 1.1).
3. Página para selecionar que documento se quer criar (Anexo 6).
4. Página para criar artigos (Anexo 7).
5. Página para criar textos-rápidos (Anexo 2.1).
6. Página para responder a e-mails (Anexo 4).
7. Página para criar e-mails (Anexo 8).
8. Página para editar texto (Anexo 9).
9. Página com um resumo do documento criado (Anexo 10).

- Integração de Redux nos componentes já desenvolvidos – 14ª - 16ª semanas
- Desenvolvimento de um componente para a parte de administração do editor de Knowledge Base – 17ª - 19ª semanas

  O componente desenvolvido para a parte de administração do editor encontra-se apresentado no anexo 12.1. Para o desenvolvimento deste componente criou-se primeiro, os seguintes componentes mais simples:
  1. Componente para apresentar uma categoria.
  2. Diálogo para permitir a edição.
  3. Diálogo para permitir criar categorias.
  4. Diálogo para permitir apagar categorias.
  5. Diálogo para permitir mover o local onde as categorias estão.
  6. Motor de busca.
  7. Componente para permitir ordenar as categorias.

- Apresentação e demonstração, para toda a equipa, do trabalho desenvolvido neste estágio. Esta apresentação, tem como principal objetivo explicar, em detalhe, o código desenvolvido, para tornar mais fácil a futura alteração e modificação a outros programadores – 1 dia
- Relatório final e apresentação final – 17ª - 20ª semanas

Resultados obtidos *(máx. 3000 caracteres)*

O objetivo principal deste estágio passou pelo desenvolvimento de componentes React. Para o desenvolvimento dos componentes mais complexos, o objetivo passou por dividi-los em componentes mais simples e, através destes, construir os componentes finais com as funcionalidades pretendidas. Assim, no início do desenvolvimento de cada componente, era sempre feito um estudo com o objetivo de extrair estes componentes mais simples.

As reuniões quinzenais nas quais mostrava os avanços que o meu projeto tinha tido, tornaram-se bastante úteis para descobrir erros e pormenores que poderiam ser melhorados neste projeto. Nestas reuniões, através da partilha de ecrã, demonstrava os avanços que o projeto tinha tido e, caso alguém presente na reunião, encontrasse erros ou algo que pudesse ser melhorado era me dito, com o objetivo, de no final, o resultado ser o melhor possível. Assim, para além dos testes que fiz, tinha a opinião da restante equipa, o que era importante para validar os resultados obtidos (a nível visual). Para além disso, foram feitos alguns Code Reviews por parte do meu coordenador, de modo a melhorar pormenores relacionados com o código e, assim, validar os resultados obtidos (neste caso, a nível técnico).

Depois de os componentes estarem todos desenvolvidos, e devido ao tempo extra, iniciou-se a inserção de Redux nos componentes. Para isto, tive uma ajuda inicial por parte do meu coordenador, que se veio a revelar bastante útil, pois a partir dessa ajuda, consegui inserir facilmente Redux no resto dos componentes.

Durante este estágio, foram surgindo vários problemas em que a resolução não foi imediata, mas devido ao avanço rápido do estágio, sobrou bastante tempo para se conseguir arranjar soluções para todos os problemas. Um exemplo disto, foi no componente para criar textos rápidos (anexo 2). Este componente, devido à grande complexidade que apresentava, acabou por ficar com um código bastante extenso e complexo, o que iria dificultar a posterior modificação, por parte de outro programador. Como consequência disto, foi feita uma análise a alternativas de implementação que tornassem este componente mais simples, concluindo-se que era possível obter uma solução mais simples e menos complexa através do uso do CKEditor. Contudo, para se fazer uso desta framework, foi necessário fazer algumas alterações na implementação da mesma, pois, neste caso, pretendia-se que a Toolbar apresentada pelo

CKEditor fosse bastante simples. (As únicas funcionalidades que a Toolbar apresentada pelo CKEditor, neste caso, necessitava de ter, era um botão para andar para trás e outro para andar para a frente.)

Neste estágio, o resultado de todos os componentes desenvolvidos foi o pretendido, não tendo sido necessário fazer alterações aos planos iniciais. Para além disto, devido ao tempo extra, foi possível testar mais ao pormenor todos os componentes desenvolvidos e o Redux inserido.

Formação recebida, conhecimentos adquiridos *(máx. 3000 caracteres)*:

O estágio deu-se início com 2 formações online da plataforma Udemy, uma de React/Redux e outra de Typescript. Estas formações, principalmente a de React/Redux foram uma forte ajuda para este estágio, visto esta ser uma linguagem totalmente nova para mim.

Outra vertente que foi totalmente nova para mim neste estágio foi a utilização do git. De modo a salvaguardar todo o trabalho desenvolvido, foram feitas sistematicamente várias submissões ao git através da aplicação sourcetree, permitindo assim que saísse deste estágio com conhecimentos mais aprofundados sobre isto.

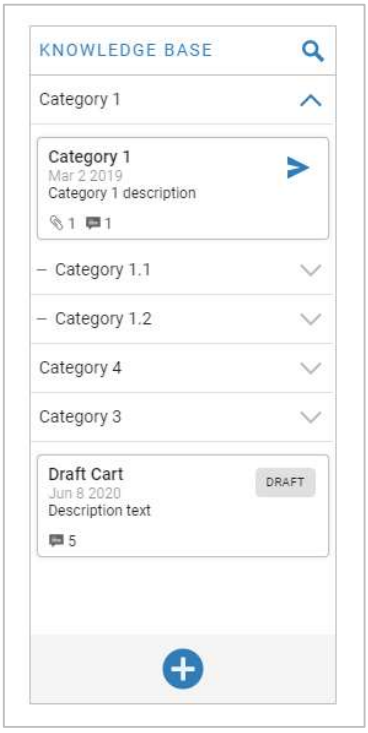
A metodologia de trabalho da equipa onde estive inserido é Scrum. Esta é uma metodologia que está organizada em ciclos ("Sprints"), em que cada ciclo, na altitude, tem a duração de 2 semanas. No início de cada "Sprint", é realizada uma reunião ("Sprint Planning") onde são atribuídas tarefas a cada elemento da equipa. O objetivo é que no final de cada "Sprint", todas as tarefas estejam finalizadas. No meu caso, eu não ia a estas reuniões pois, no início de cada "Sprint", definia com o meu coordenador, quais as tarefas que me propunha a cumprir durante a "Sprint". Devido à metodologia de trabalho Scrum, houve reuniões diárias ("Daily Scrum") onde cada elemento da equipa comunica à restante equipa o que fez no dia anterior e o que irá fazer no dia atual. Estas reuniões devido ao estágio ter sido totalmente remoto, foram bastante importantes para conhecer e ter um contacto mais próximo com equipa onde estive inserido. Por fim, a metodologia de trabalho Scrum leva a que existam mais duas reuniões durante cada "Sprint", a "Sprint Review" e a "Sprint Retrospective". A primeira, servia para mostrar à restante equipa os desenvolvimentos que os projetos estavam a ter. A segunda, era realizada no último dia de cada "Sprint", e tinha como objetivo refletir sobre o que correu bem e mal, na "Sprint" que estava a acabar. Adoptar esta metodologia ajudou-me bastante a ter objetivos fixos e a trabalhar em prol de os conseguir cumprir. Para além disto, devido a estas reuniões, foi possível desenvolver as minhas SoftSkills em apresentações orais. (No anexo 14, encontra-se um esquema sobre a metodologia de trabalho Scrum)

Neste estágio não encontrei muitas dificuldades para além das que já foram descritas. Contudo, estas dificuldades conseguiram ser ultrapassadas através da ajuda do meu coordenador, de outros membros da equipa e de alguma pesquisa.

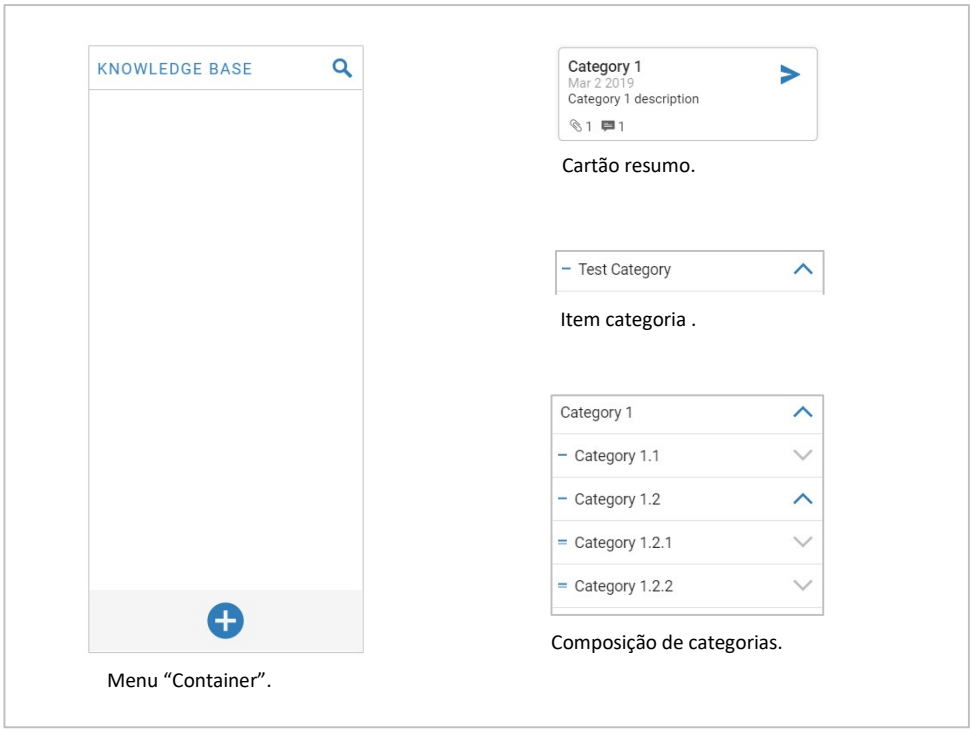
No geral, com este estágio adquiri conhecimentos mais aprofundados sobre algumas línguas de programação e ferramentas que fiz uso recorrente. Ainda que este estágio tenha sido totalmente remoto, penso que consegui adquirir algumas SoftSkills em trabalho de grupo, devido às reuniões descritas anteriormente e às chamadas que tive com o meu coordenador e com outros membros da equipa para discutir problemas e por vezes conseguir ter alguma ajuda.

Anexos ao documento:

Anexo 1 – Menu principal

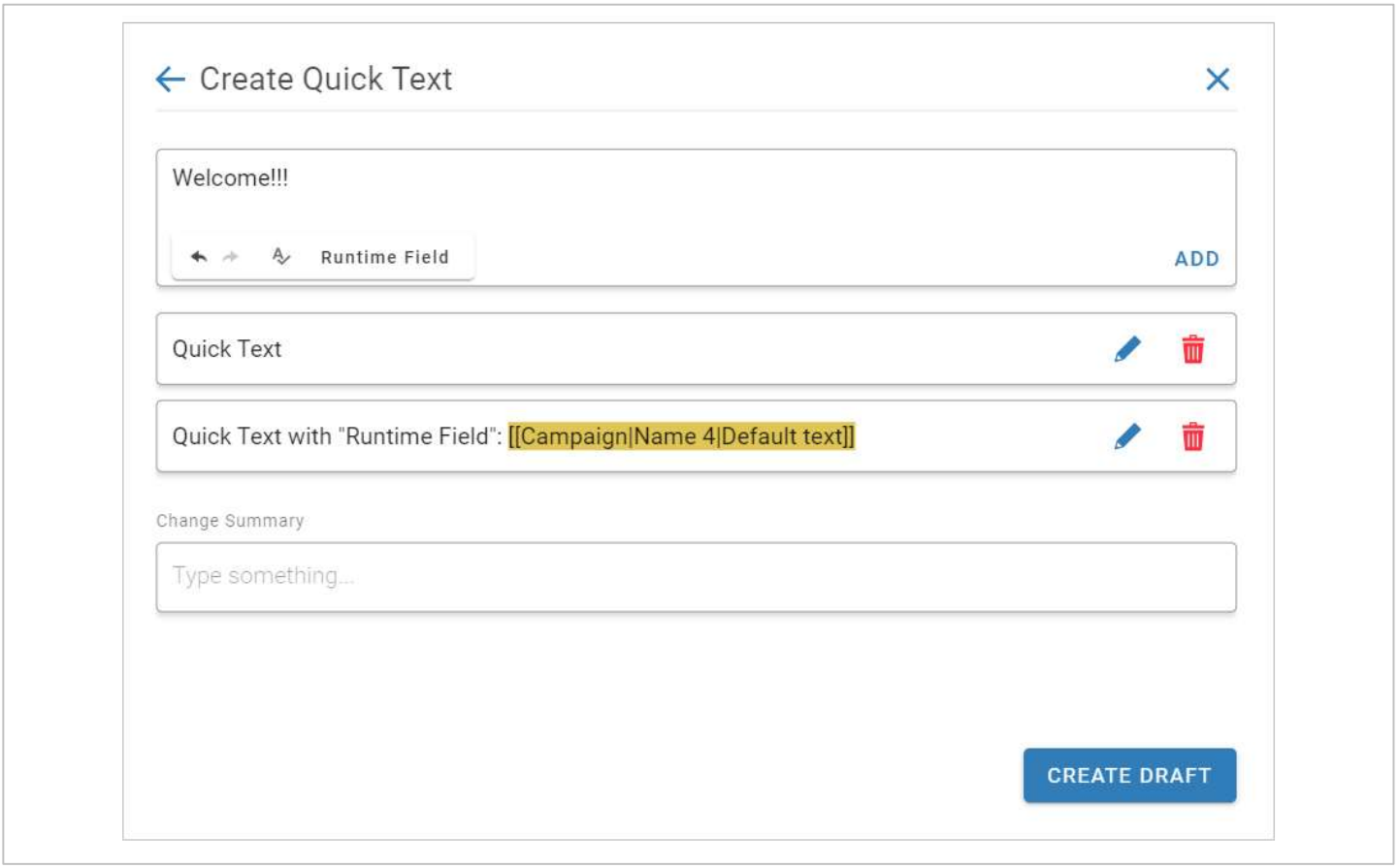


1.1 – Menu principal do editor Knowledge Base.



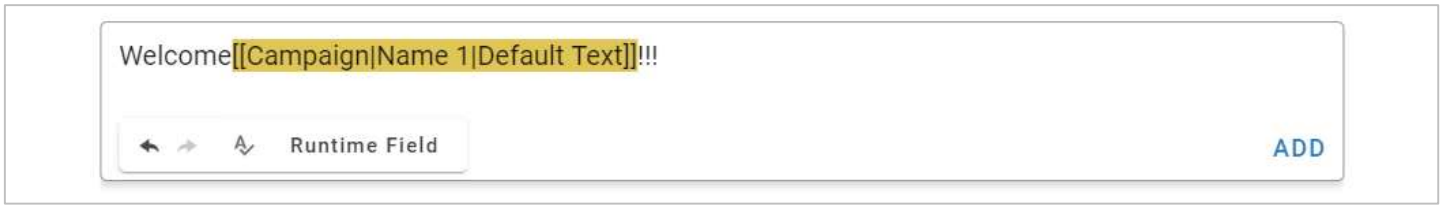
1.2 – Componentes mais simples para auxiliar na construção do menu principal do editor .

Anexo 2 – Componente para criação de textos rápidos



2.1 – Componente para criação de textos rápidos.

2.2 – Caixa para criar/editar textos rápidos com cursor posicionado entre “Welcome” e “!!!”.

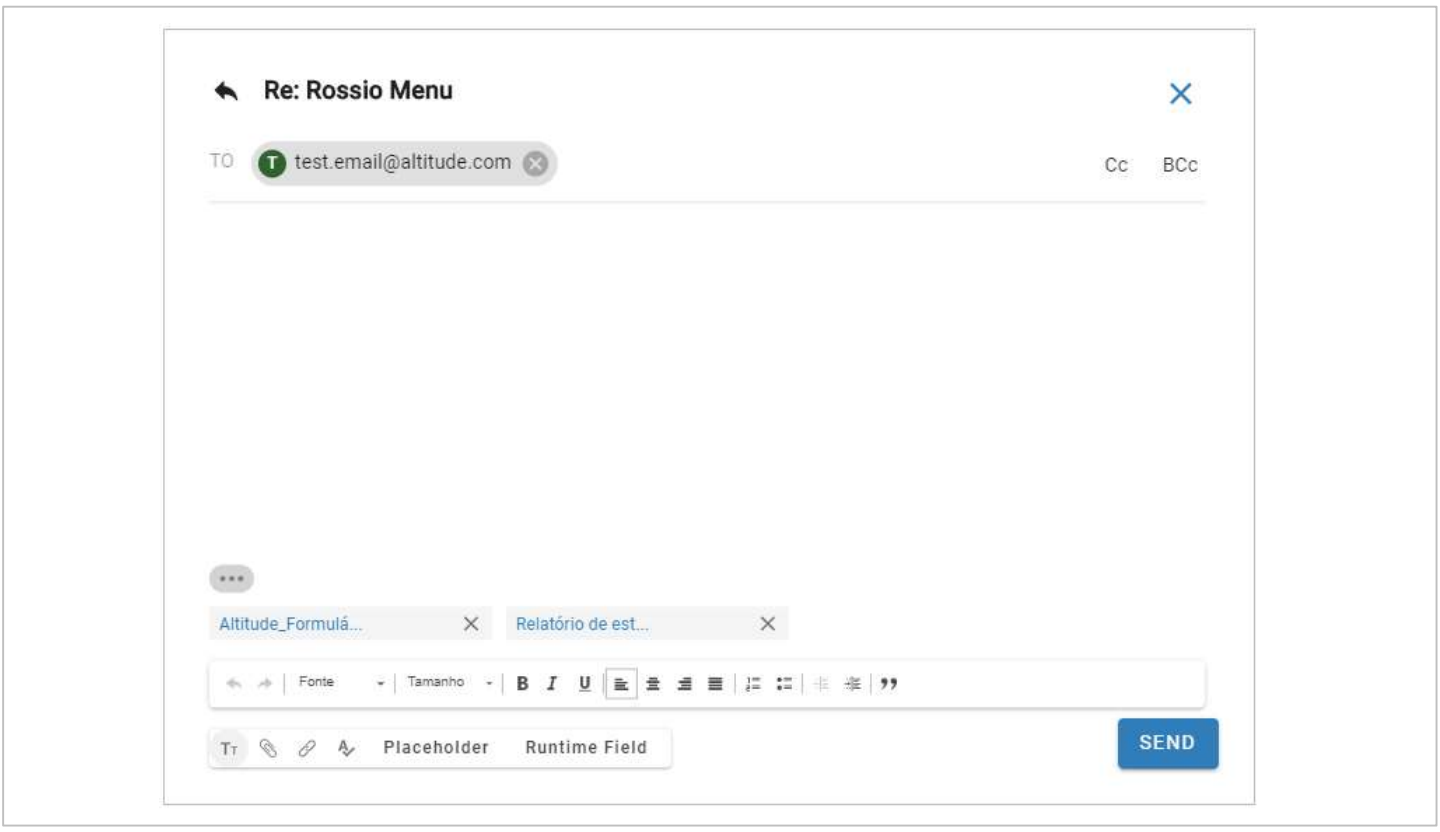


2.3 – Caixa para criar/editar textos rápidos, com um “Runtime Field” inserido no local onde o cursor se encontra em 2.1.

Anexo 3 – “*Toolbar*” para edição de texto

3 – “*Toolbar*” para edição de texto através do uso do CKEditor.

Anexo 4 – Página para responder a e-mails



4 – Página para responder a e-mails.

Anexo 5 – Objetos usados para guardar as categorias e os seus cartões

```
let data: {
  name: string;
  id: string;
  description: string;
  card?: {
    title: string;
    date: string;
    description: string;
    messagesNum: number;
    attachments: number;
    submit: KBCardTypes;
  }[];
  campaign: string;
  date: string;
  children: Children[];
};
```

Tipo do objeto principal para guardar as informações de cada categoria.

```
type Children = {
  name: string;
  id: string;
  description: string;
  card?: {
    title: string;
    date: string;
    description: string;
    messagesNum: number;
    attachments: number;
    submit: KBCardTypes;
  }[];
  children: Children[];
};
```

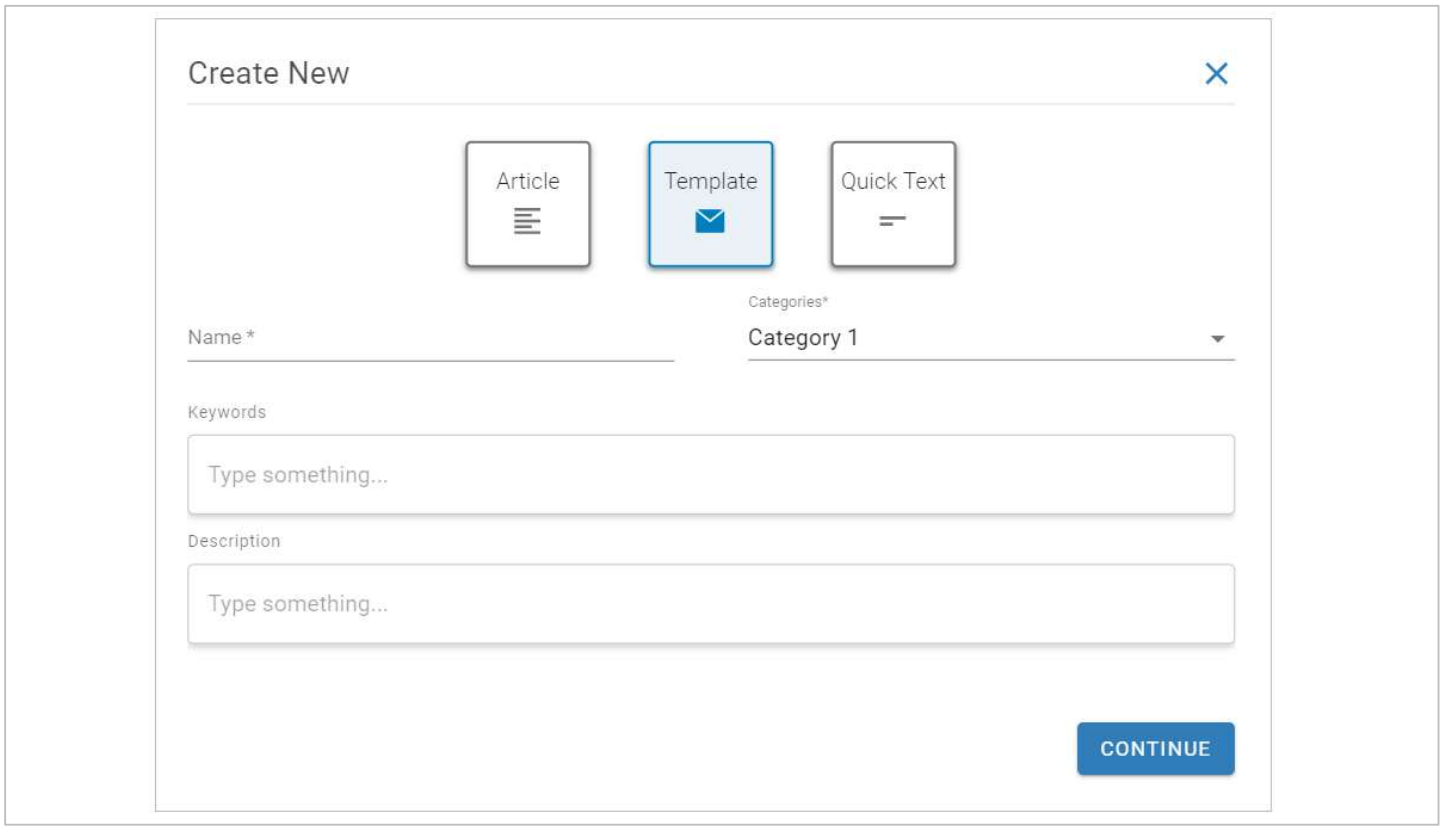
Tipo do objeto filho para guardar as informações de cada categoria.

5.1 – Tipo dos objetos para guardar as categorias e as suas informações.

```
data = {
  id: '1',
  name: 'Category 1',
  campaign: 'Service 1',
  card: [{
    title: 'Category 1',
    date: 'Mar 2 2019',
    description: 'Category 1 description',
    messagesNum: 1,
    attachments: 1,
    submit: KBCardTypes.SENDCARD
  }],
  description: '',
  date: '5 Feb 2019 14:59:16 GMT',
  children: [
    {
      id: '3',
      name: 'Category 1.2',
      description: '',
      children: [
        {
          id: '4',
          name: 'Category 1.2.1',
          description: '',
          children: [
            {
              id: '6',
              name: 'Category 1.2.1.2',
              description: '',
              children: []
            }
          ]
        },
      ]
    }
  ]
};
```

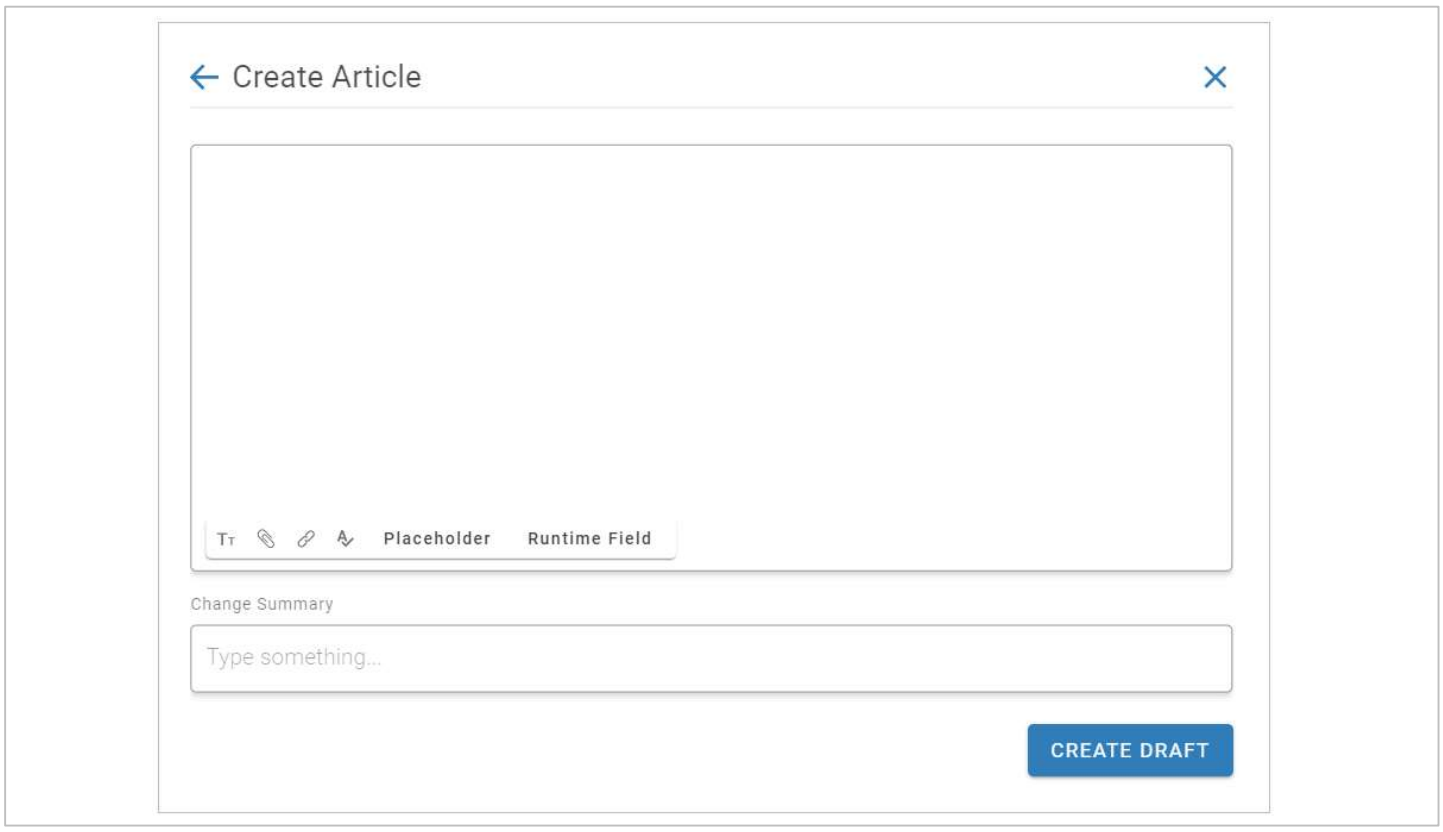
5.2 – Simulação de uma listagem de categorias com o formato definido em 5.1.

Anexo 6 – Página para selecionar que documento se quer criar.



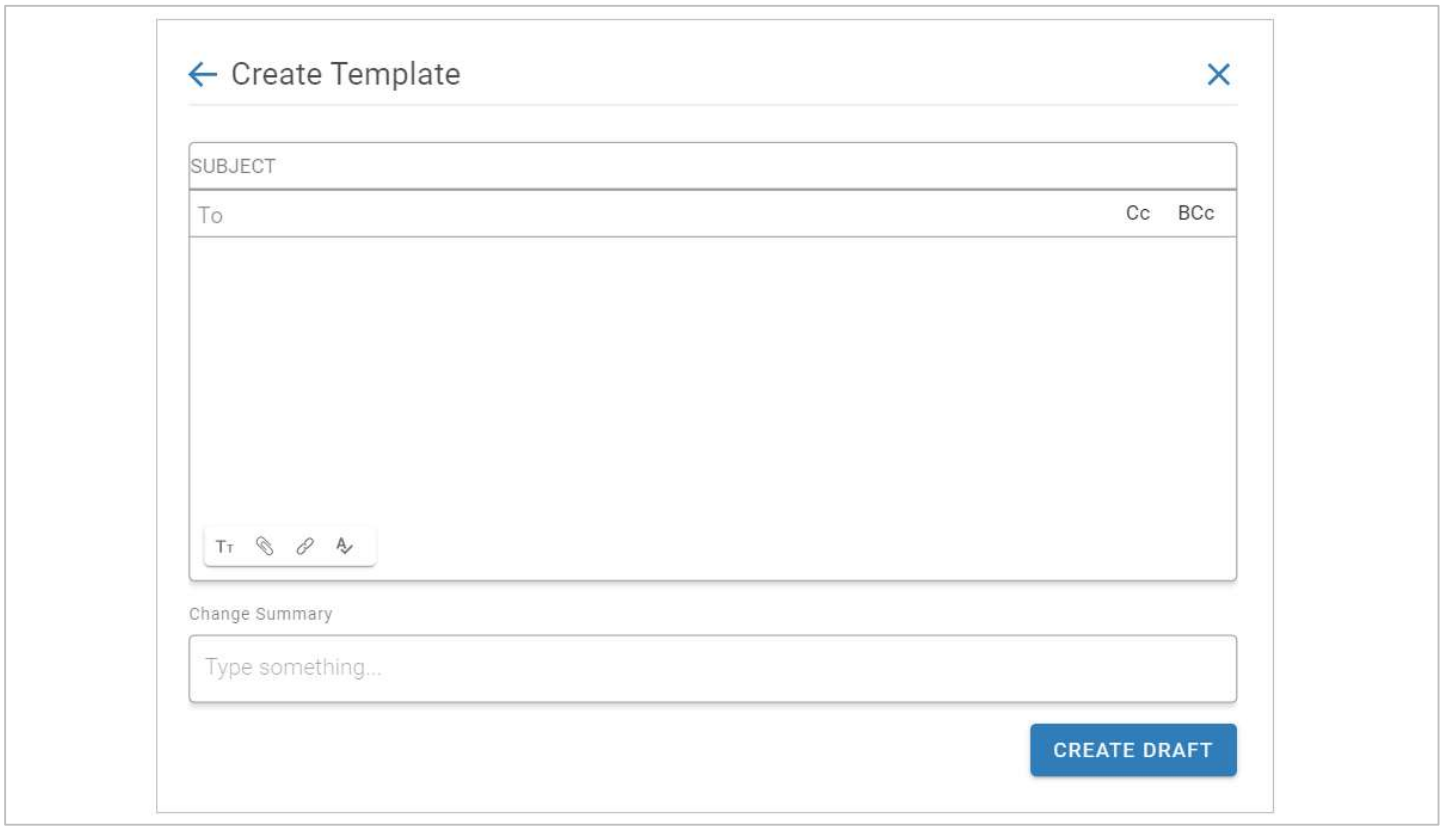
6 – Página para selecionar que documento se quer criar.

Anexo 7 – Página para criar artigos.



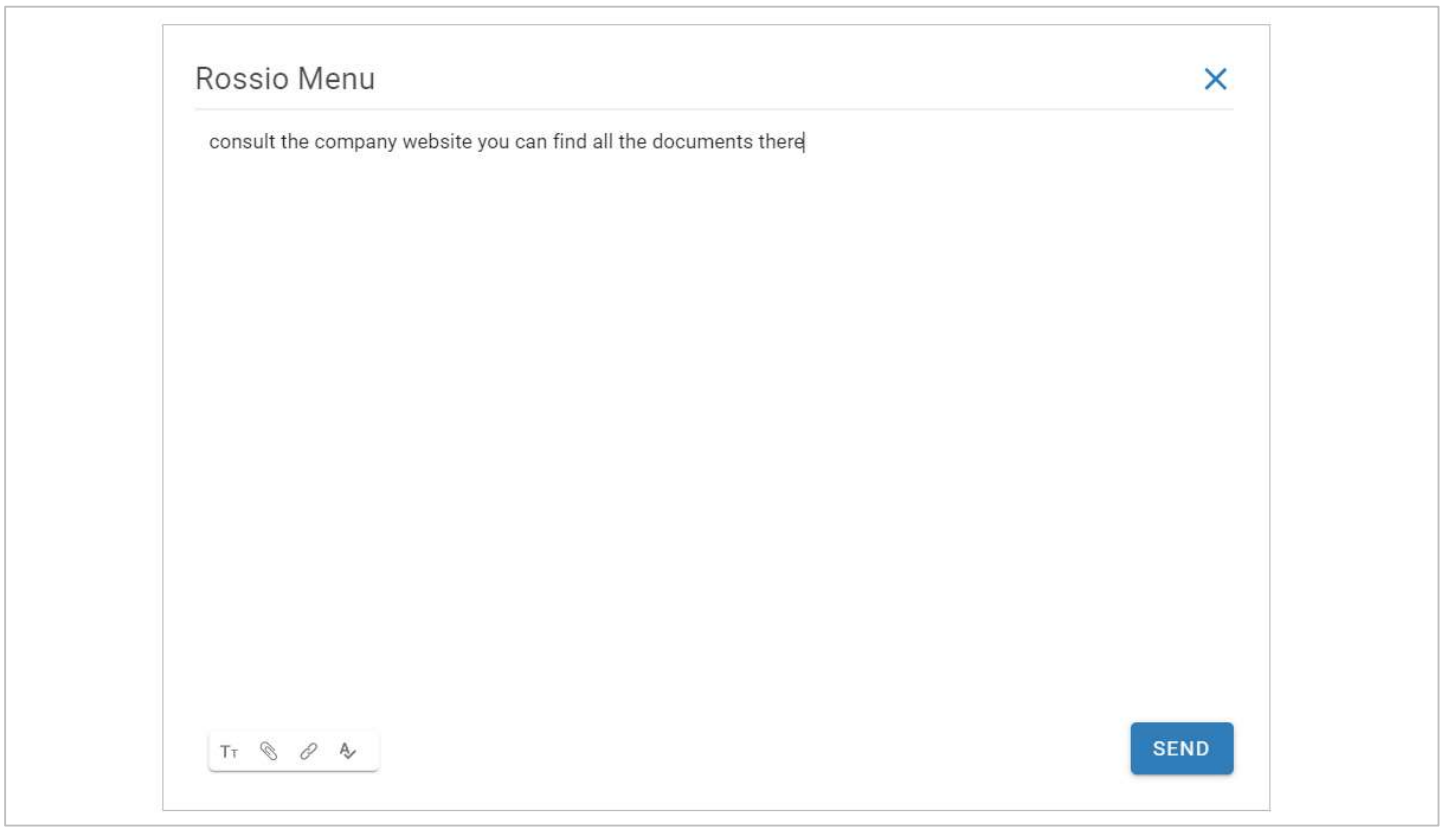
7 – Página para criar artigos.

Anexo 8 – Página para criar e-mails.



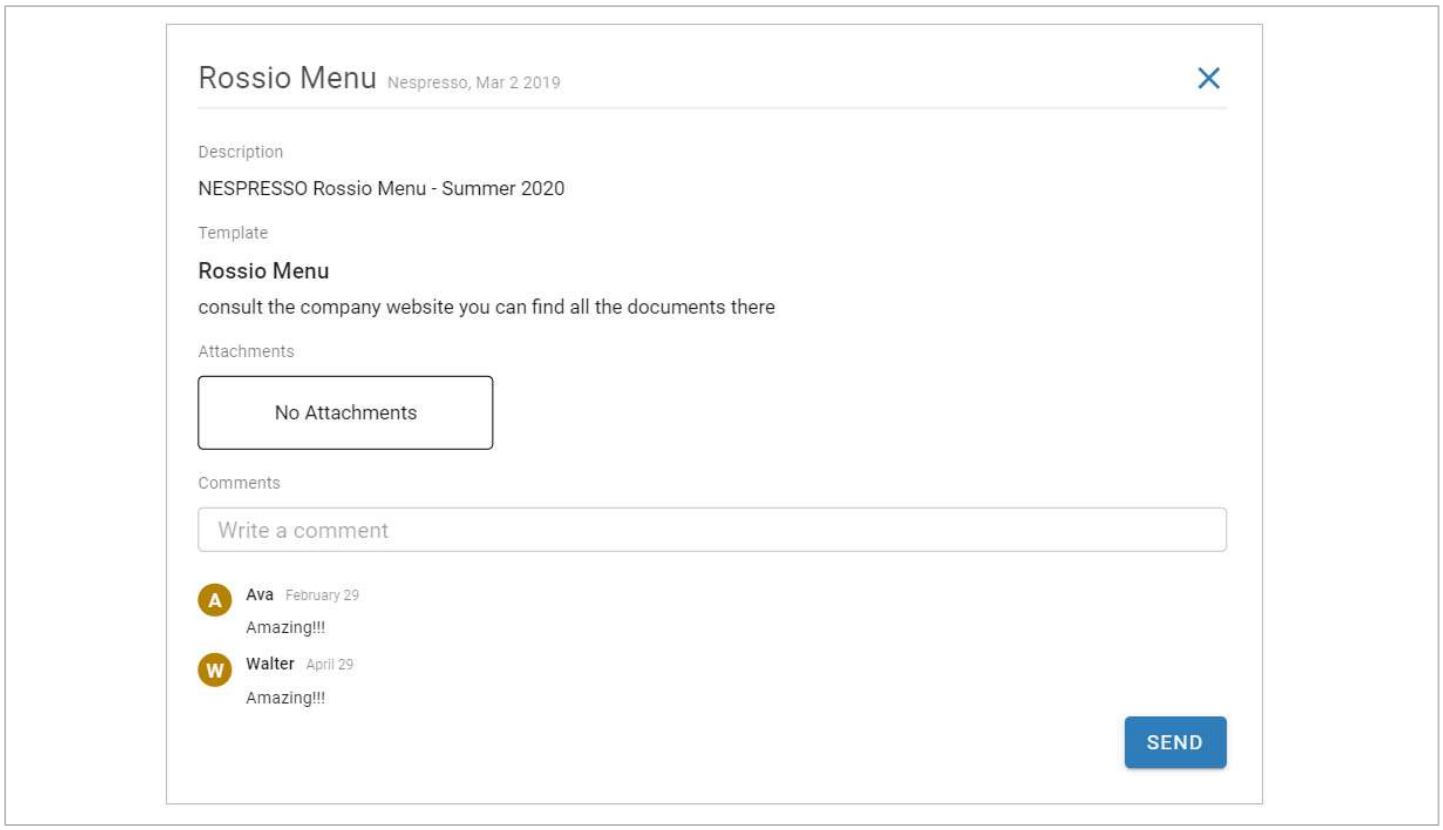
8 – Página para criar e-mails

Anexo 9 – Página para editar texto.



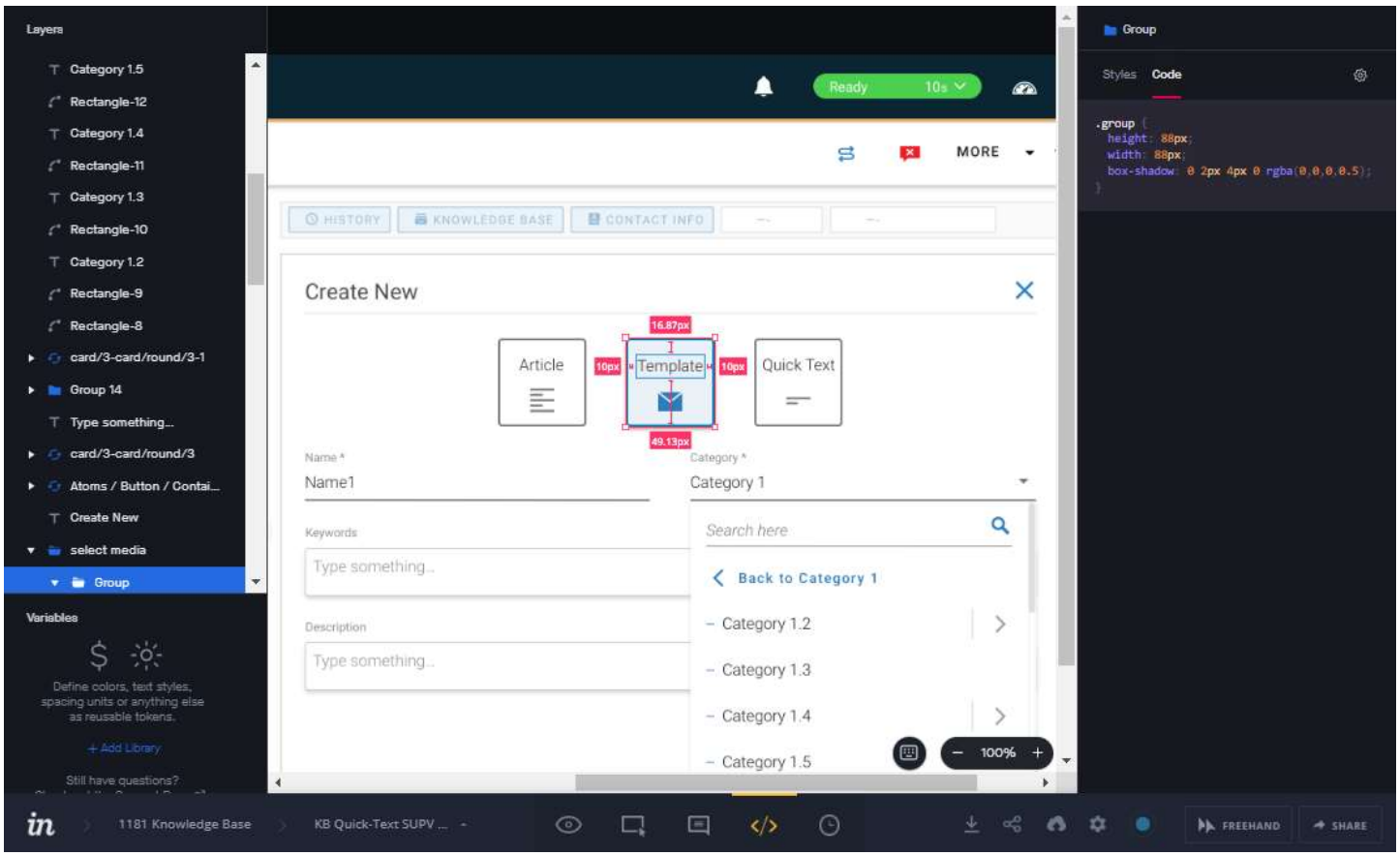
9 – Página para editar texto.

Anexo 10 – Página com o resumo do documento que vai ser criado.



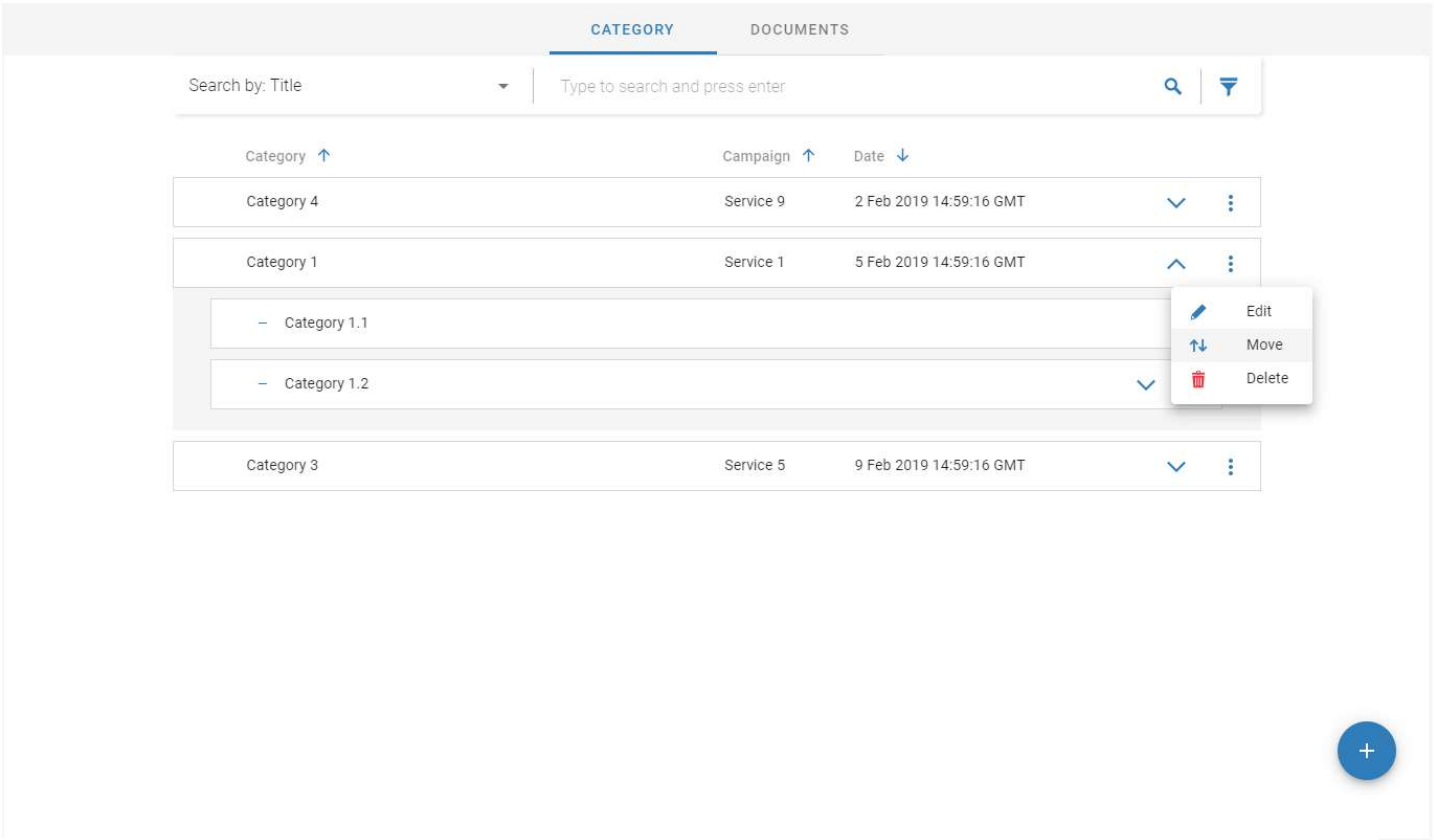
10 – Página com o resumo do documento que vai ser criado.

Anexo 11 – Plataforma onde os desenhos (mock-ups) dos componentes foram feitos



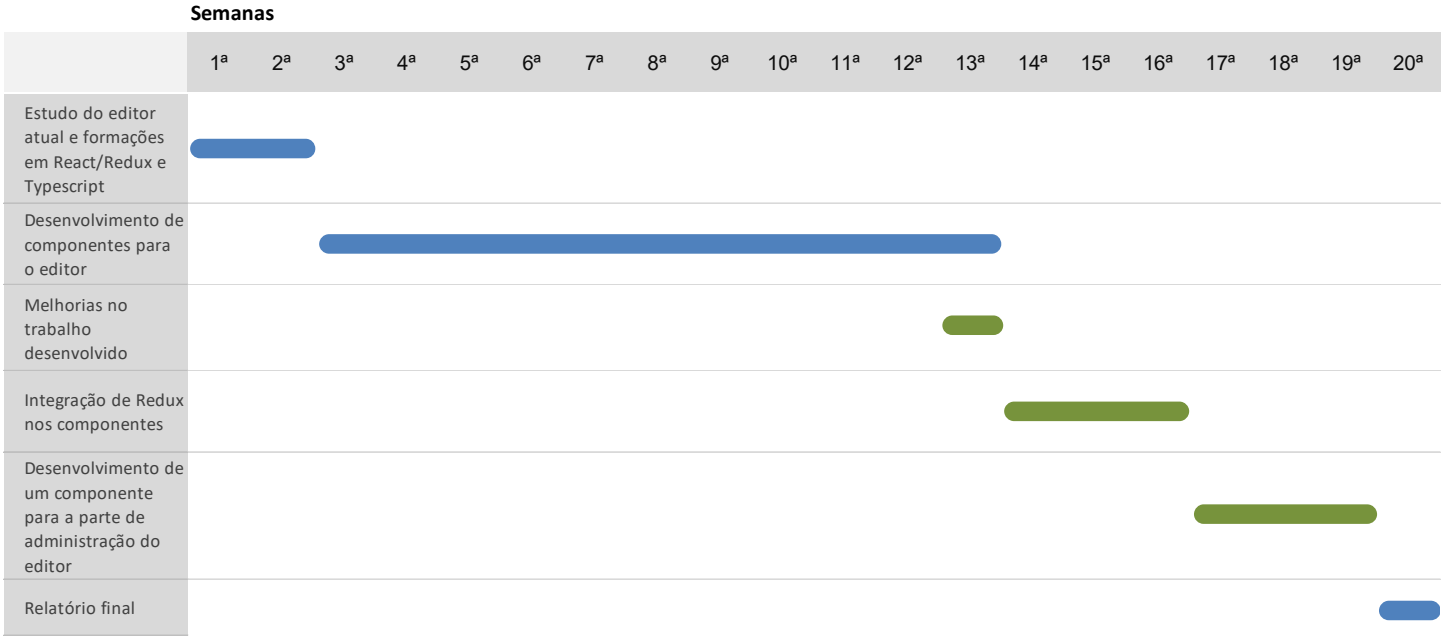
11 – Captura de ecrã da plataforma Invision onde os desenhos (mock-ups) dos componentes foram feitos.

Anexo 12 – Componente para a parte de administração do editor Knowledge Base



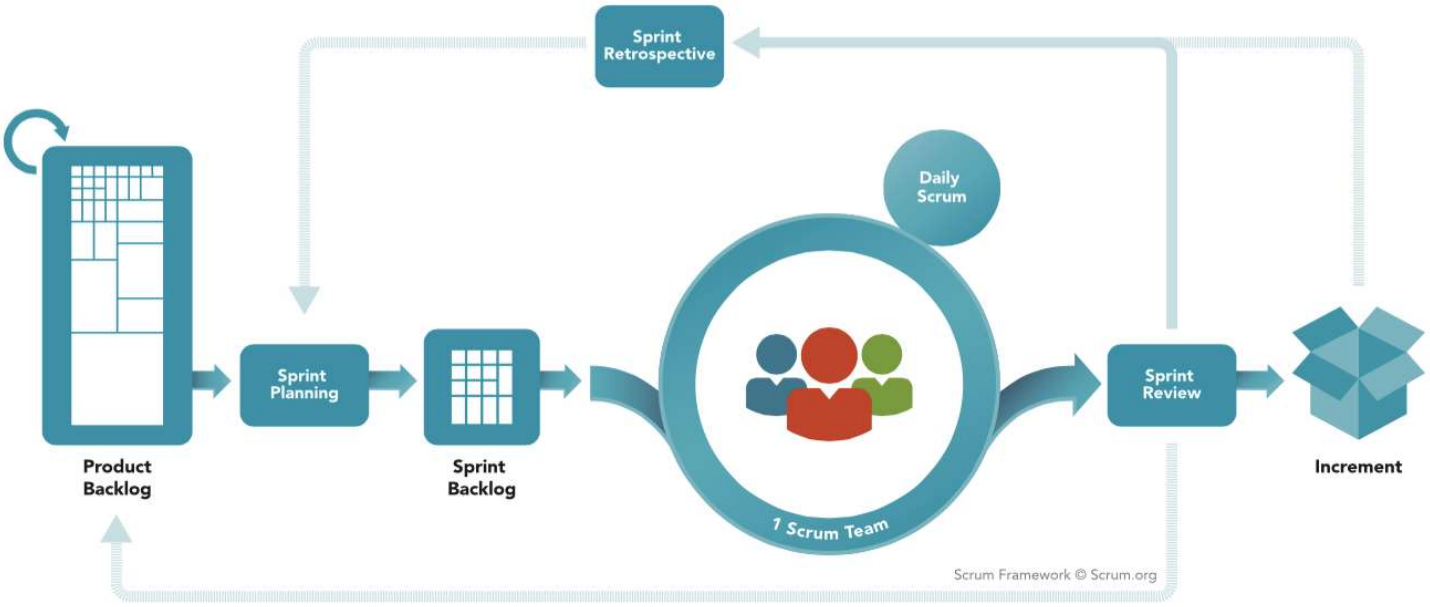
12.1 – Componente para a parte de administração do editor Knowledge Base que permite inserir, editar, apagar e mover categorias.

Anexo 13 – Cronograma do plano de trabalhos final



13.1 – Cronograma do plano de trabalhos final. A azul, encontram-se as tarefas que inicialmente me foram propostas para este estágio, e a verde, as tarefas que foram acrescentadas ao plano de trabalhos, devido ao tempo extra que houve neste estágio.

Anexo 14 – Metodologia Scrum



14.1 – Diagrama que mostra como a metodologia de trabalho Scrum está organizada.

Referências:

[i] "Knowledge Base". https://whatis.techtarget.com/definition/knowledge-base (em inglês). Consultado a 10 de Setembro de 2020.

[ii] "React". https://reactjs.org/ (em inglês). Consultado a 10 de agosto de 2020.

[iii] "Typescript". https://www.typescriptlang.org/ (em inglês). Consultado a 10 de agosto de 2020.

[iv] "Css". https://www.w3schools.com/css/ (em inglês). Consultado a 10 de agosto de 2020.

[v] "An Introduction to Redux". *Smashing Magazine,* https://www.smashingmagazine.com/2016/06/an-introduction-to-redux/ (em inglês). Consultado a 10 de agosto de 2020.

[vi] "Teams". https://www.microsoft.com/pt-pt/microsoft-365/microsoft-teams/group-chat-software. Consultado a 10 de agosto de 2020.

[vii] "Visual Studio Code". https://code.visualstudio.com/ (em inglês). Consultado a 10 de agosto de 2020.

[viii] "Sourcetree". https://www.sourcetreeapp.com/ (em inglês). Consultado a 10 de agosto de 2020.

[ix] "Git". https://git-scm.com/ (em inglês). Consultado a 10 de agosto de 2020.

[x] "What is a Sprint Review". https://www.scrum.org/resources/what-is-a-sprint-review (em inglês). Consultado a 10 de agosto de 2020.

[xi] "Udemy". https://www.udemy.com/. Consultado a 10 de agosto de 2020.

[xii] "How does git work?". https://developer.ibm.com/technologies/web-development/tutorials/d-learn-workings-git/ (em inglês). Consultado a 10 de agosto de 2020.

[xiii] "O que é o Ágil? Uma nova definição formal". https://www.linkedin.com/pulse/o-que-%C3%A9-%C3%A1gil-uma-nova-defini%C3%A7%C3%A3o-formal-andr%C3%A9-gomes. Consultado a 10 de agosto de 2020.

[xiv] "What is a daily scrum". https://www.scrum.org/resources/what-is-a-daily-scrum (em inglês). Consultado a 10 de agosto de 2020.

[xv] "JIRA". https://www.atlassian.com/software/jira (em inglês). Consultado a 10 de agosto de 2020.

[xvi] "How to Track Progress with Agile Management: The task Board". https://www.dummies.com/careers/project-management/how-to-track-progress-with-agile-management-the-task-board/ (em inglês). Consultado a 10 de agosto de 2020.

[xvii] "Invision". https://www.invisionapp.com/ (em inglês). Consultado a 10 de agosto de 2020.

[xviii] "StoryBook". https://storybook.js.org/ (em inglês). Consultado a 10 de agosto de 2020.

[xix] "Material-UI". https://material-ui.com/pt/. Consultado a 10 de agosto de 2020.

[xx] "NPM". https://www.npmjs.com/ (em inglês). Consultado a 10 de agosto de 2020.

[xxi] Componente npm usado - "Material-ui-chip-input". https://www.npmjs.com/package/material-ui-chip-input (em inglês). Consultado a 10 de agosto de 2020.

[xxii] "CKEditor". https://ckeditor.com/ckeditor-4/ (em inglês). Consultado a 10 de agosto de 2020.

[xxiii] "Refs". https://reactjs.org/docs/refs-and-the-dom.html (em inglês). Consultado a 10 de agosto de 2020.

[xxiv] "CSS visibility Property". https://www.w3schools.com/cssref/pr_class_visibility.asp (em inglês). Consultado a 10 de agosto de 2020.