

4 Programação de Sítios para Internet

Durante o terceiro semestre, comecei um projeto junto ao professor Adriano Bezerra, que consiste na criação de uma aplicação de BMG Canvas com Análise SWOT, este projeto foi feito como uma iniciação científica, porém o conteúdo abordado durante o terceiro semestre do curso principalmente na disciplina de programação de sítios, foi essencial para o desenvolvimento do projeto por completo.

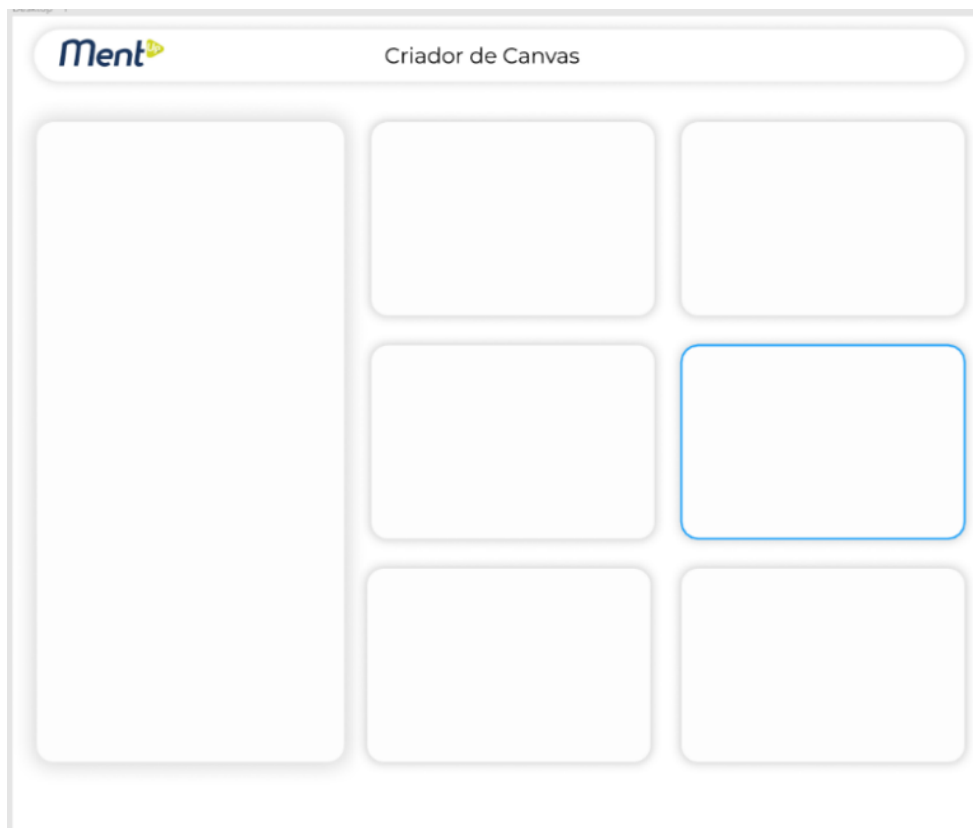
Atualmente o projeto está ainda em fase de desenvolvimento, mas já existem muitas funcionalidades prontas e que foram de grande dificuldade e aprendizado, principalmente de conteúdos que não são ensinados integralmente no curso, muita das vezes tecnologias pouco usadas no mercado porém que possuem muito potencial de crescimento, e por isso foram escolhidas para o desenvolvimento da aplicação, tais como: ReactJS, utilizado para a construção do frontend do site;

Redux, biblioteca utilizada para a criação de todo o fluxo de dados que percorre toda a aplicação, essa biblioteca de código é responsável pelos dados que vem do backend poderem se comunicar com todo o frontend da aplicação de maneira mais automática possível.

Para o início do projeto foi primeiro acordado as tecnologias necessárias para o desenvolvimento do mesmo, assim como todo o layout da aplicação e também a construção do primeiro protótipo. Essa primeira etapa ocorreu durante o primeiro bimestre do terceiro semestre do curso, e durante esse primeiro bimestre enquanto os outros envolvidos se encarregaram de montar o escopo inicial da aplicação, eu fui encarregado de estudar as tecnologias necessárias e conceitos fundamentais para o desenvolvimento web atual.

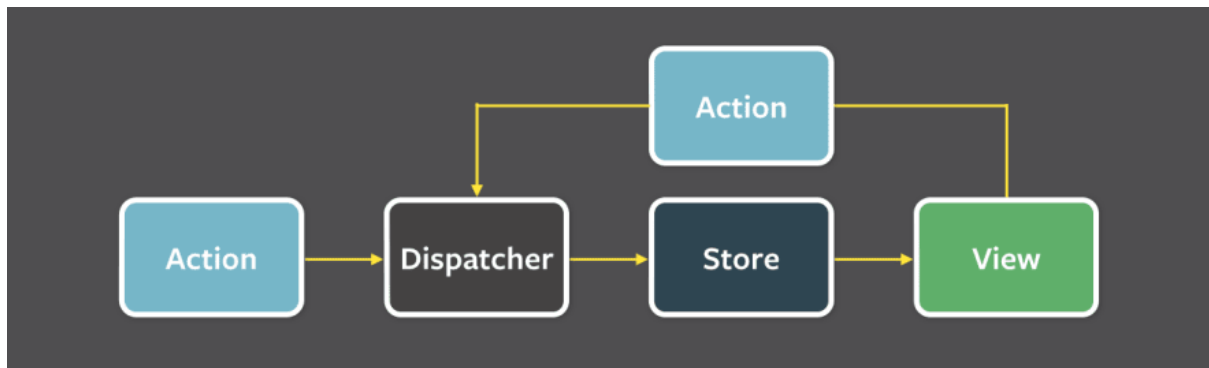
Após essa etapa foi construído o primeiro modelo da aplicação, feita a partir da ferramenta de design de interfaces chamada Figma. Essa primeira versão de layout da aplicação tratou do desenvolvimento da tela de BMG Canvas, umas das seções da aplicação, e focou bastante no que deveria ter na aplicação.

Figura 4.1:Primeiro modelo de tela da aplicação



Após a criação do layout da aplicação, foi realizada a escolha da chamada *stack* de desenvolvimento, que nada mais é que a escolha das tecnologias e linguagens que serão usadas em um projeto. No caso da aplicação de BMG Canvas e Análise SWOT, foi acordado que o backend da aplicação seria feito com PHP e utilizando o framework Laravel, que sem sombra de dúvidas é o framework mais usado para o desenvolvimento WEB com PHP, para o frontend foi escolhido a biblioteca React JS, que foi criada e é mantida atualmente pelo Facebook, e que além da sua simplicidade também é reconhecida pela forma com que trata o desenvolvimento da aplicação baseando-se em componentes, e pela sua construção dinâmica de uma página, onde não é necessário que a página venha totalmente montada do servidor, o que ajuda na performance inicial do site quando aberto pela primeira vez. Como base de comunicação entre o frontend e o backend, foi escolhido o sistema de microsserviços, onde teremos o frontend e o backend como aplicações separadas e conectando-se somente através de uma api rest disponibilizada pelo próprio backend, o que ajuda na modularização da aplicação e da criação de novas aplicações desvinculadas e que podem consumir dessa mesma API. Também dentro do front end da aplicação temos o uso da biblioteca Redux, que é responsável pela construção e manipulação de estados da aplicação, o que envolve uma série de dados que vem da api e que precisam ser disponibilizadas para a aplicação, o Redux gerencia os dados e também quando a página deve carregar e também escolher qual componente carregar para poder atualizar seus dados e mostrar tudo para o cliente.

Figura 4.2: Figura que mostra o fluxo de funcionamento do redux



Fonte: claudiojunior.me, 2001

Acima está um grafo que mostra como funciona a biblioteca de gerenciamento de estados Redux, primeiro temos o objeto de action que é responsável por armazenar quais são as ações que podem ocorrer dentro do front end da aplicação, depois temos o dispatcher, que funciona como o disparador da ação para que possa ser executado uma ação, depois temos a Store, que nesse caso serve para guardar o código que deve ser executado quando uma action for chamada, ou seja, o dispatcher verifica qual action foi acionada, pesquisa o que deve fazer essa action, isso dentro da store, e depois retorna o estado novo para a view. Todo esse fluxo serve para evitar que a aplicação tenha que se recarregar completamente quando um dado novo chega ao frontend ou então uma ação do usuário que precise carregar apenas algum componente, o que faz com que a página não recarregue sempre, aumentando assim a satisfação do cliente, assim melhorando o experiência do usuário dentro da aplicação.

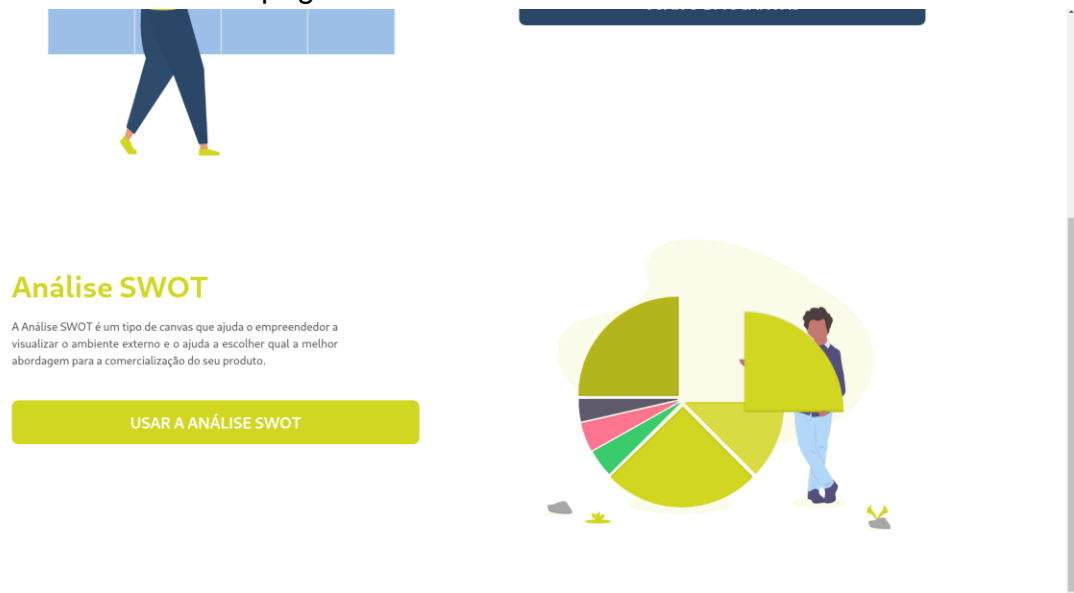
Figura 4.3: Página inicial da aplicação

Canvas e SWOT



Acima está a implementação da chamada home da aplicação, que é nada mais nada menos que a primeira página da aplicação, onde o usuário tem o seu primeiro contato com a mesma. Essa pagina foi feita utilizando, além das tecnologias já citadas, o framework css BulmaCSS, um framework que vem crescendo muito no mercado atualmente, e que está se baseando nas tendências de design mais atuais do mercado, por isso foi escolhida para ser a base da estilização da aplicação, pois contém um design elegante e que presa muito pelo minimalismo, tema muito abordado ao longo dos últimos anos.

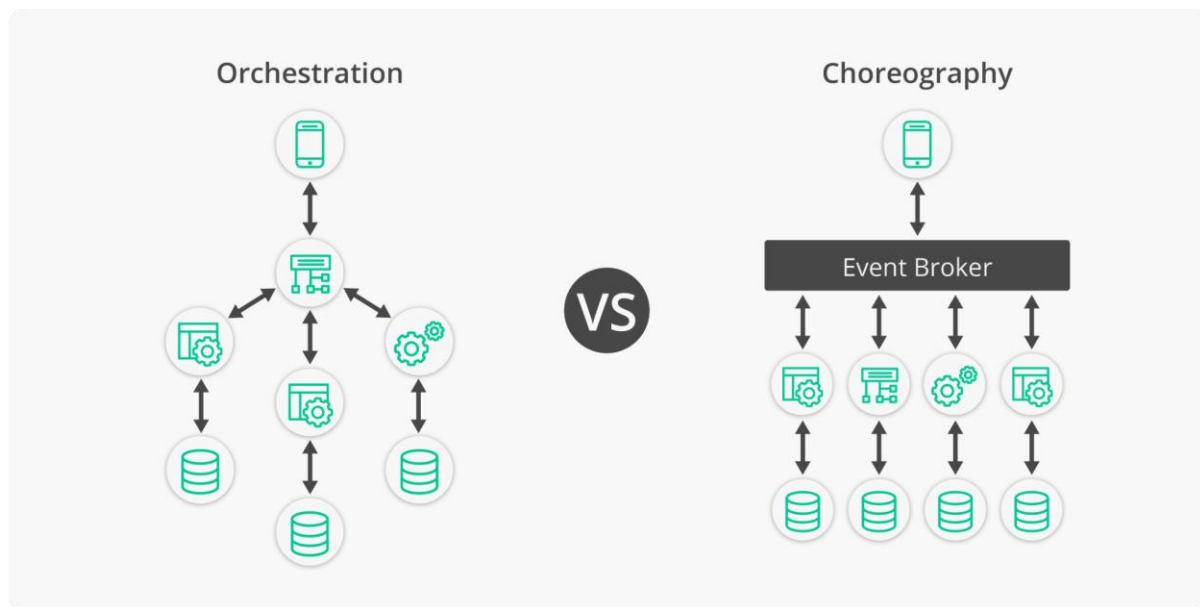
Figura 4.4: Parte inferior da página de home



Fonte: Aplicação desenvolvida, 2020

Para construir a aplicação de forma mais atual e seguindo os padrões desse último ano, o grupo optou por desenvolver a aplicação utilizando a estrutura de microsserviços. Por conta disso resolvemos fazer duas aplicações totalmente separadas, que contém de ligação entre si, apenas uma api, que não prende nenhuma aplicação à outra necessariamente, essa estrutura além de ser muito utilizada devido às novas tecnologias que surgiram no mercado, também é usada atualmente pela facilidade de dar manutenção no código das aplicações, pois não tem um código muito extenso e também não dificulta o entendimento do mesmo.

Figura 4.5: Fluxo de microsserviços



Fonte: <https://solace.com/blog/microservices-choreography-vs-orchestration/>

Acima podemos ver porque a estrutura de microsserviços acaba sendo muito utilizada, no lado esquerdo do esquema está um modelo chamado de monolito, onde tudo está em ligado a uma aplicação, e na esquerda está uma estrutura de microsserviços, podemos ver que no primeiro esquema, tudo está vinculado a apenas uma aplicação de fato, é que ela conecta o celular do usuário, agora vejamos a série de problemas que podem ocorrer, se essa aplicação final travar, todo o sistema para de funcionar, e como ela está conectada ao mesmo tempo com todas as outras aplicações, se alguma dessas aplicações tiver uma falha, toda a aplicação acaba sendo prejudicada, isso se a aplicação não parar de funcionar completamente. No segundo modelo, este baseado em microsserviços, vemos que toda a aplicação é dividida em pequenos módulos e que cada módulo conecta a um banco de dados próprio, e que para que essas aplicações se comuniquem com o usuário é usado um event broker, que serve para capturar possíveis falhas e redirecionar o usuário para outra aplicação caso alguma esteja com mau funcionamento, esse tipo de estrutura já consegue resolver todos os problemas citados acima em relação ao primeiro modelo, como por exemplo: se a aplicação de configurações (simbolizada pelas duas engrenagens) parar de funcionar, por não estar diretamente ligada com outra aplicação, o sistema consegue funcionar e somente essa aplicação será finalizada para poder resolver o problema.