Apresentação da Ferramenta

Ambas versões podem ser encontradas no github e e hospedado no netlify. (disponibilizar os dois links)

Desenvolvimento do sistema

O sistema Aleph, projetado para a visualização animada de algoritmos de ordenação, teve sua criação inicial durante o projeto de conclusão de curso de Ciência da Computação (UFMA), por Pedro Neto. Desde então, foram implementadas diversas melhorias, incluindo modificações na interface e a adição de novos recursos, como o algoritmo de ordenação insertion sort, os algoritmos de busca binária e busca linear, e um algoritmo de cálculo fatorial utilizando uma função recursiva.

Principais mudanças

A mudança de maior destaque no Aleph foi a antiga rota principal ‘/’ que antes apontava diretamente para a tela de execução do bubble sort, agora aponta para uma ‘tela inicial’, onde é possível escolher uma das seis visualizalções de algoritmos possíveis. Essa tela inicial veio pra substituir a antiga barra de navegação (figura) que continha os algoritmos que estavam no sistema (bubble sort e selection sort)

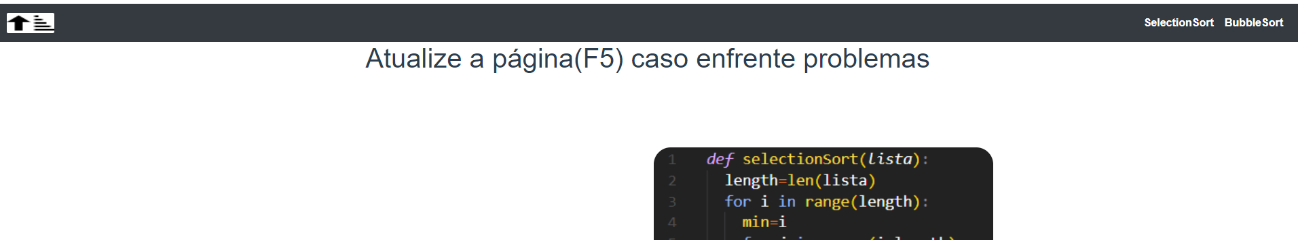


Figura – barra de navegação.

Além disso, algumas mudanças sutis foram feitas na tela de execução: diminuição da fonte do texto: ‘Atualize a página F(5) caso enfrente problemas’, remoção das figuras que representavam os códigos dos algoritmos (figura) por um texto escrito diretamente no arquivo .vue dentro de duas tags: <code> e <pre> usando a biblioteca highlight.js (figura).

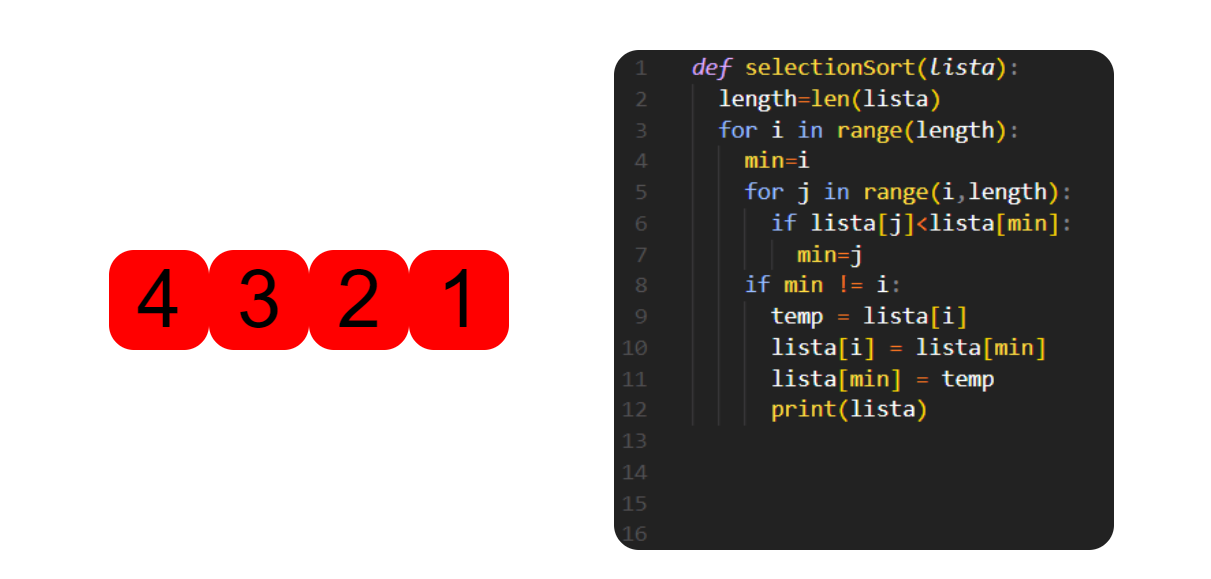


Imagem que representava o código do algoritmo selecionado

Figura – antiga representação do código

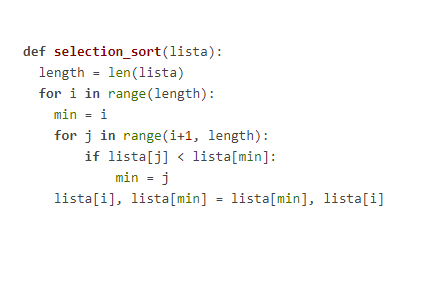


Figura – representação atual do código

Já nos vetores, além da troca de cor, uma sutil borda preta foi adiciona em volta dos índices (figura), com o objetivo de facilitar a visualização principalmente qunado os vetores forem preenchidos por números de dois dígitos

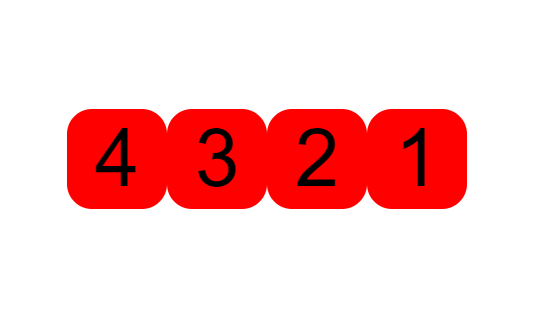
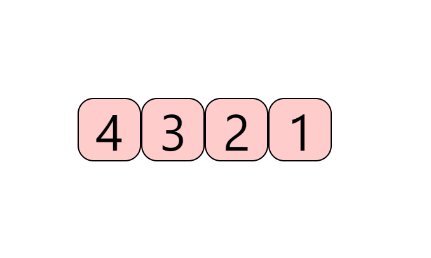
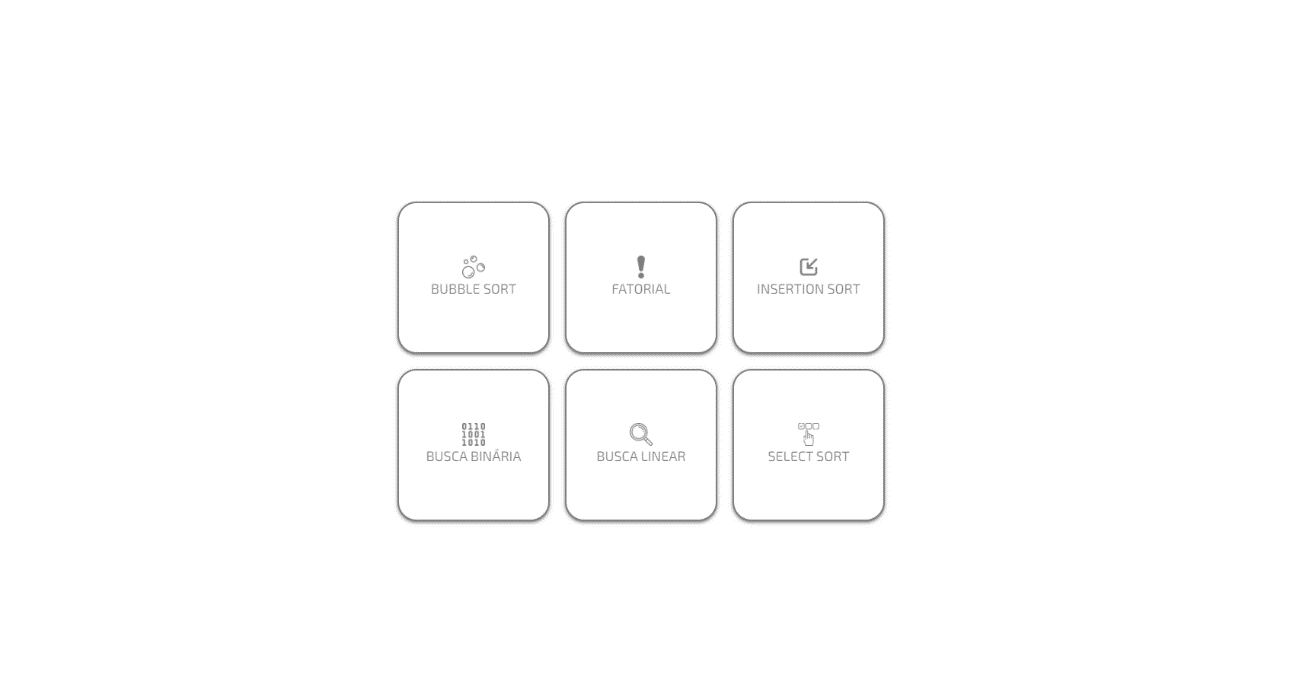
 

figura – vetores(antes) figura- vetores(depois)

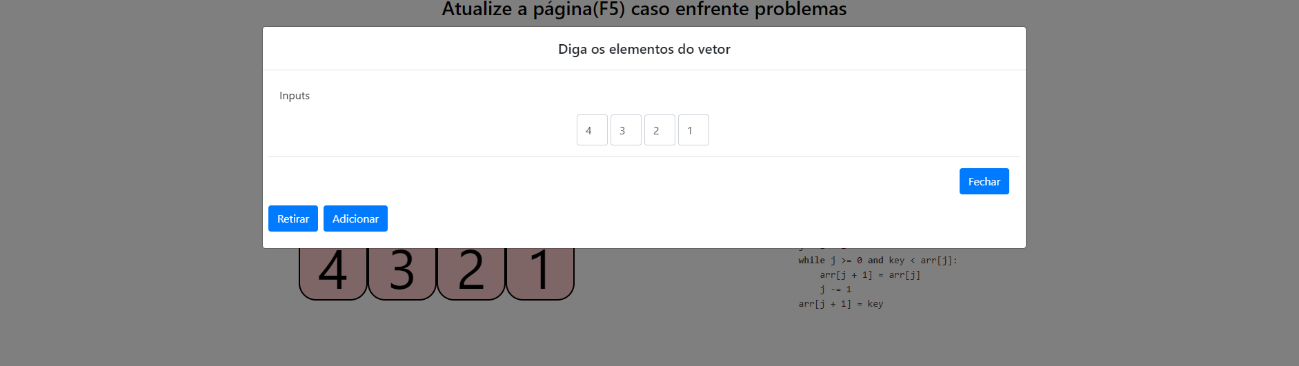
Tela inicial do sistema

A figura representa a tela inicial do sistema. A rota base da aplicação foi modificada para uma página inicial, na qual o usuário pode escolher qual algoritmo deseja visualizar. Os seis algoritmos disponíveis são exibidos em três colunas, distribuídos em duas linhas.

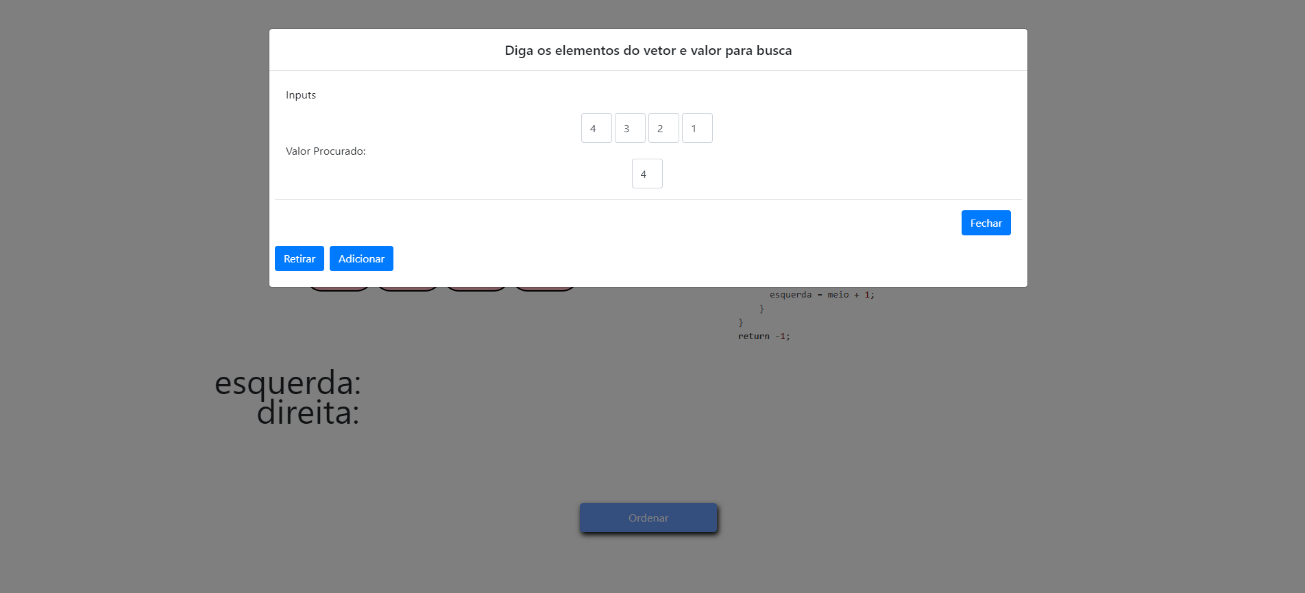
(figura) - Tela inicial da aplicação.

Tela de execução

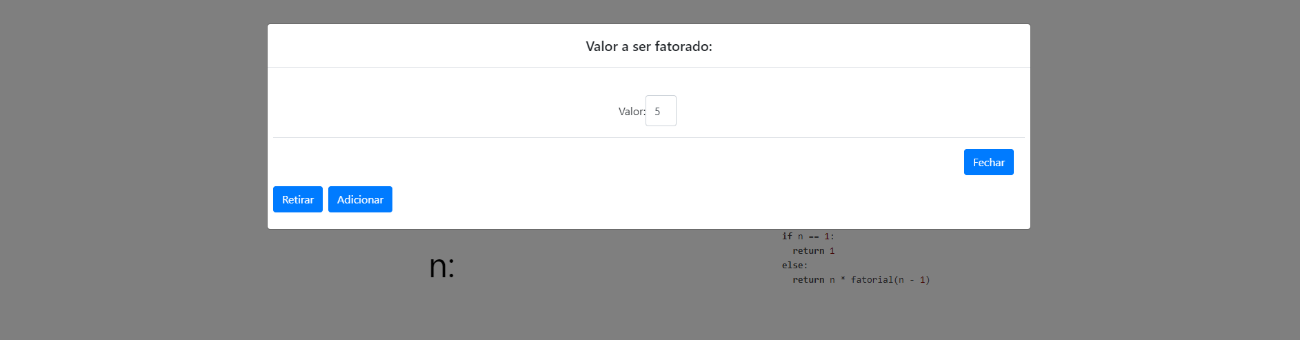
As figuras representam a tela de execução logo após a escolha do algoritmo desejado. Nesta fase, um modal é exibido, solicitando os dados que serão utilizados na animação. Se o usuário selecionar algoritmos de ordenação, como Insertion Sort, Bubble Sort ou Selection Sort, o modal será composto apenas por vetores, permitindo a alteração do tamanho até 7 índices e a escolha do valor correspondente para cada índice. Por outro lado, se o algoritmo escolhido for alguma técnica de busca, como Selection Sort e Busca Fatorial, o modal apresentará uma opção adicional: o valor procurado, mantendo as demais opções inalteradas. Por fim, se a escolha for o algoritmo de fatorial, aparecerá uma única opção de input: o valor a ser fatorado. Para acessar a View, basta fechar o Modal, o que fará com que o vetor do modal seja enviado para a respectiva view.



(figura) – Modal insertion sort.



(figura) – Modal busca binária.



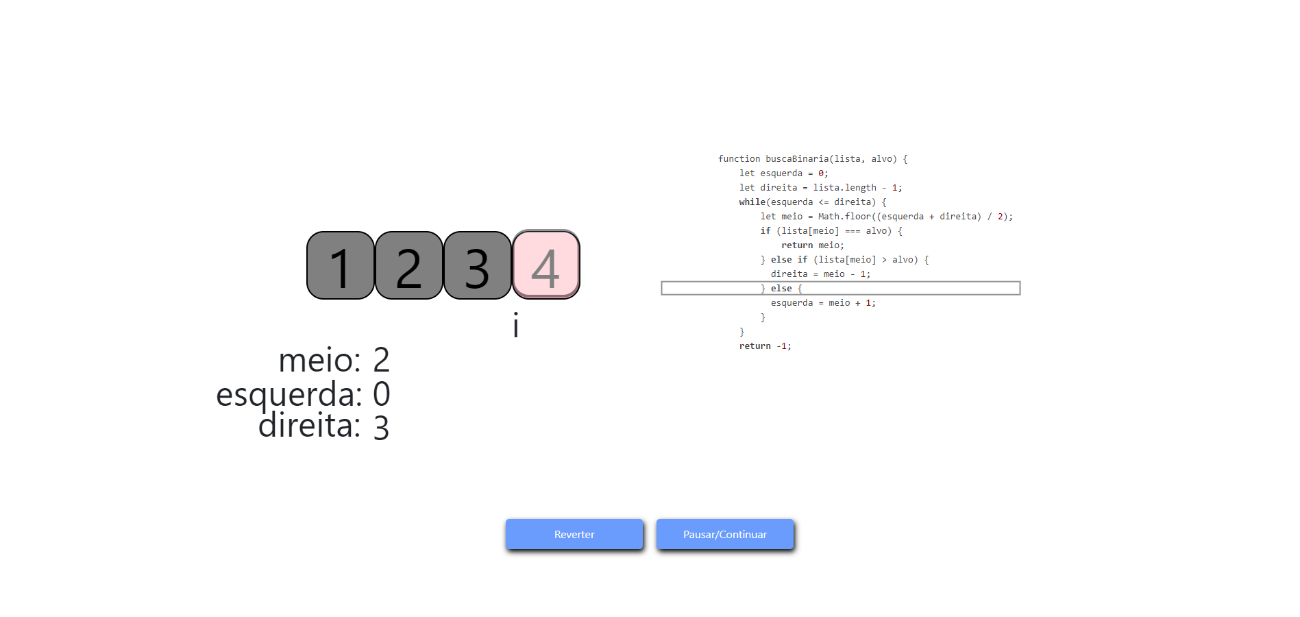
(figura) – Modal fatorial.

Animação

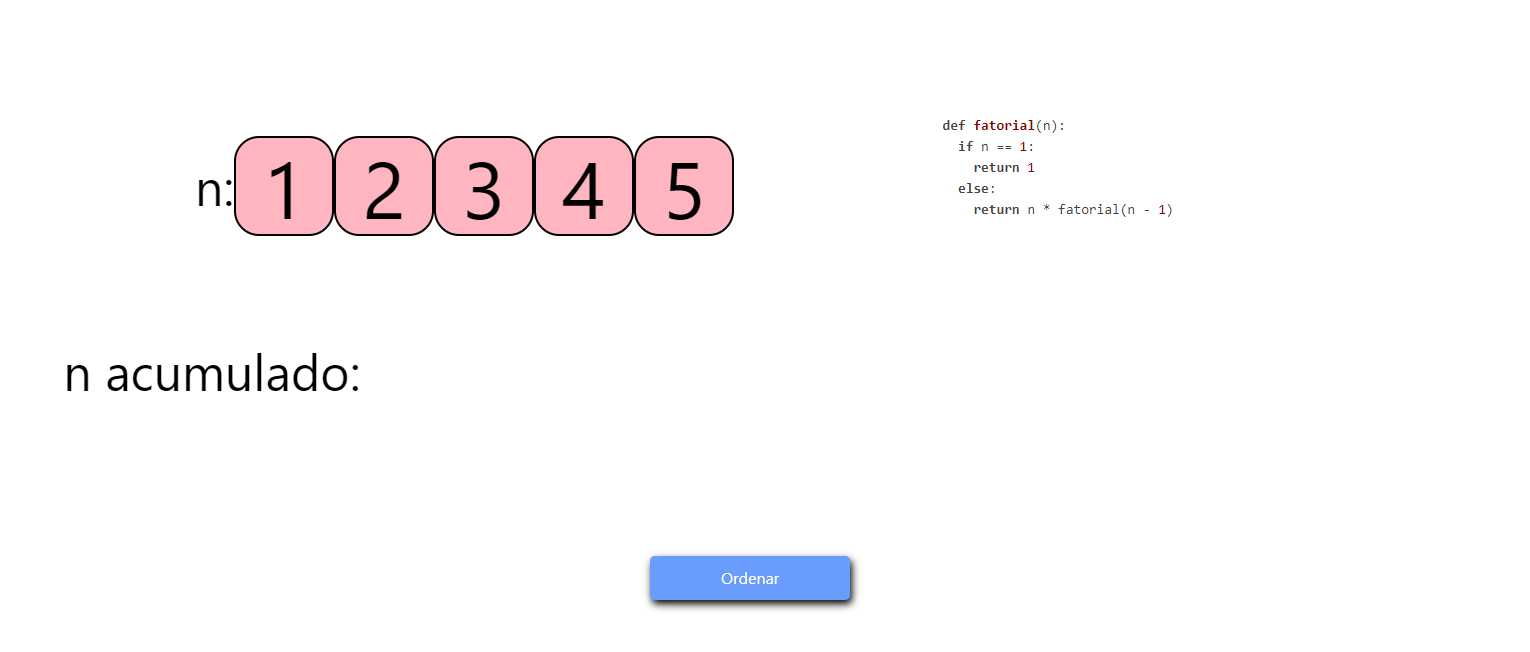
Após o modal ser fechado, as figuras mostram a tela de execução de um algoritmo de cada tipo: Insertion sort(ordenação), Fatorial(recursão) e Busca Binária(busca),. Além disso, no canto superior direito está indicando os links para acessar o Selection Sort ou o Bubble Sort. Ao clicar no botão “Ordenar”, o mesmo irá desaparecer e se iniciará a animação



(figura) – Execução insertion sort.



(figura) – Execução busca binária.



(figura) – Execução fatorial.

A barra de que identifica em qual parte do código está sendo executada, valores de algumas variáveis e animações da troca do vetor são visíveis durante a animação. É possível pausar a animação clicando no botão “Pausar/Continuar” além de fazer a animação desordenar o vetor clicando no botão “Reverter”, essas duas funções em específico foram mantidas.

Conclusão do Capítulo

Neste capítulo, destacam-se os pontos essenciais do desenvolvimento inicial do projeto, apresentando o trabalho inicial realizado pelo aluno Pedro e as mudanças estéticas implementadas para aprimorar o funcionamento do sistema. Além disso, são introduzidas as novas visualizações dos algoritmos adicionados, com uma breve explicação de seu funcionamento.

O primeiro algoritmo animado a ser introduzido ao projeto foi o insertion sort, escolhido por sua similaridade com os algoritmos de busca já implementados. Dessa forma, o autor desta monografia enfrentou menos dificuldades na criação da animação, dedicando mais tempo ao estudo das ferramentas utilizadas no projeto e à compreensão das funcionalidades de cada variável, tag e trecho de código. Para isso, foram realizados diversos testes no código fonte, reproduzidos em um servidor local e registrados como comentários no próprio código.

A partir dessas experiências, o desenvolvimento dos demais algoritmos de busca e do algoritmo fatorial foi realizado, utilizando principalmente conceitos básicos de programação

Trabalhos Relacionados

Uma das principais fontes de inspiração para o desenvolvimento dos novos algoritmos adicionados ao Aleph foram retiradas dos modelos já existentes em sua primeira versão. Além disso, é comum encontrar outros sistemas que buscam ensinar algoritmos de forma semelhante ao Aleph, os quais também serviram como exemplo para a criação de algoritmos totalmente diferentes dos já implementados anteriormente.

Data Structure Visualizations

O sistema denominado simplesmente como Data Struct Visualizations (visualizações de estrutura de dados), desenvolvido pelo autor David Galles, professor associado da USFCA, apresenta uma variedade de algoritmos que visam ensinar por meio da visualização dos elementos que compõem determinado código. Essa ferramenta de visualização é escrita em JavaScript usando o elemento canvas HTML5 e pode ser executada em praticamente qualquer navegador moderno.

A (figura) mostra a tela inicial do Data Structure Visualizations, na qual é possível ver uma coluna à esquerda relacionada a informações sobre o desenvolvimento, contato, área de sugestões e até mesmo uma seção com instruções para criar sua própria animação, embora exija um conhecimento avançado. Já na (figura2), é apresentada a animação de diversos tipos de algoritmos de ordenação.

Com base na (figura2), o Data Structure Visualizations utiliza a mesma estrutura de interface para todos os outros algoritmos, onde a parte superior, dedicada ao input ou seleção do usuário, varia de acordo com o algoritmo selecionado, enquanto a parte inferior permanece constante, sendo responsável apenas por pausar, prosseguir, avançar ou retroceder e alterar a velocidade da animação, que, por sua vez, muda completamente dependendo do algoritmo que está sendo visualizado.

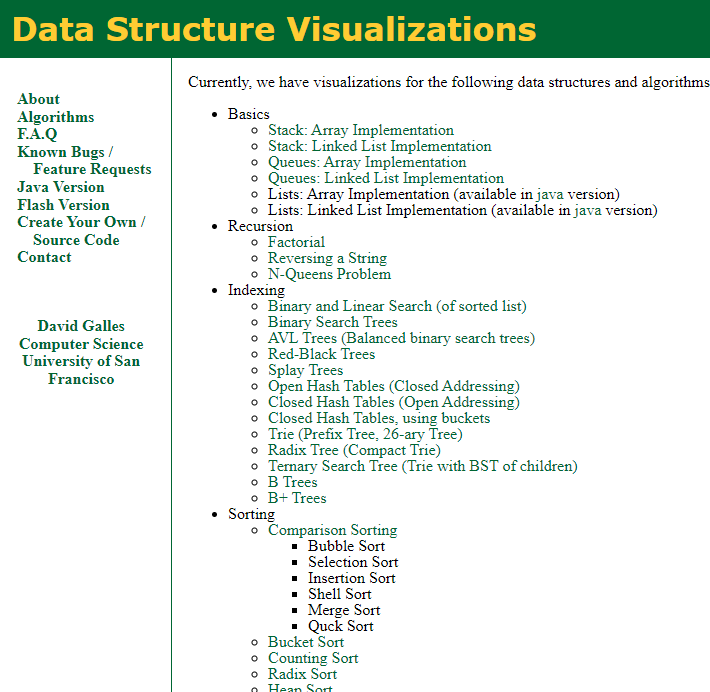


Figura – home do DSV

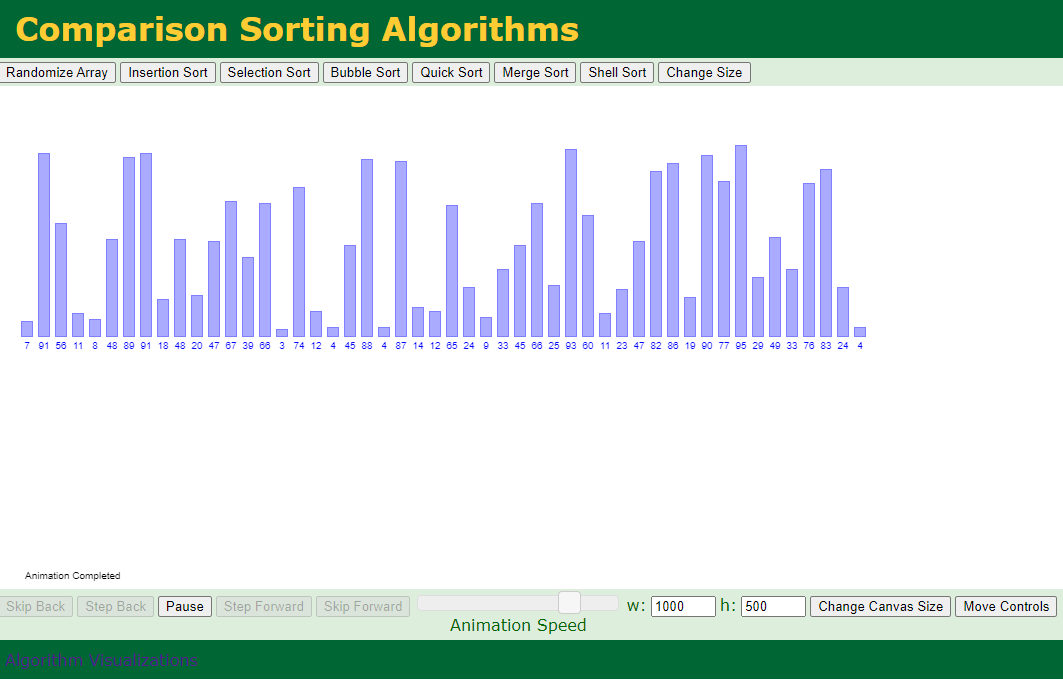


Figura 2 – tela de execução dos algoritmos de ordenação

Comparações com o Aleph

O sistema Data Structe Visualization possui alguns recursos a mais que o Aleph como a maior variedade de algoritmos e alguns recursos a mais na questão de visualização da animação, sendo possível mudar a velocidade e avançar muitas vezes. Porém, se tratando de algoritmos de ordenação, os valores do sistema em questão já estão pré definidos, sendo impossível alterar de acordo com o que queremos ver, como é mostrado na (figura 2).

Ferramenta de visualização de Algoritmos

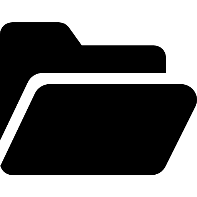
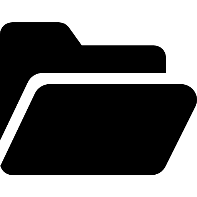
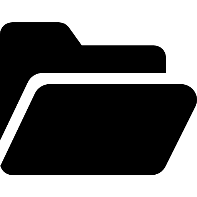
Descrição Geral

A sistema Aleph foi inicialmente desenvolvido com o propósito de ser um site capaz de auxiliar no aprendizado de algoritmos de ordenação. Com a inclusão de novas visualizações o propósito passa a ser de um sistema que seja capaz de ajudar alunos a aprenderem mais algoritmos além dos clássicos da ordenação.

Essa ferramenta dispensa um backend, pois já possui os dados necessários ou permite que o usuário os forneça. Ela não acessa bancos de dados, não faz autenticação nem cadastros. Por isso, não se enquadra no padrão MVC. Sua única função é renderizar animações e componentes web. As regras de negócio estão intimamente relacionadas aos elementos visuais. (referenciar TCC Pedro)

Para dar continuidade no desenvolvimento do Aleph, foi necessário conhecimento em 3 tecnologias que compõem o sistema: Vue, bootstrap e GSAP. Além dessas tecnologias, foi adicionado ao projeto a biblioteca Highlight.js, que é uma biblioteca JavaScript que permite destacar a sintaxe de código-fonte em páginas da web de forma simples e eficiente. Seu uso foi evidente na substituição das imagens que representavam os códigos dos algoritmos.

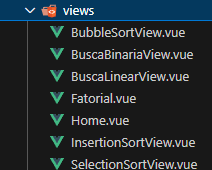
A (Figura) mostra um diagrama de pacotes usado para organizar diagramas de classes complexos. O sistema é dividido em três módulos para agrupar funções semelhantes. Essa escolha se baseia na estrutura típica do Vue CLI, que cria três pastas principais: views, router e components, facilitando a compreensão e a organização do projeto. (referenciar TCC Pedro)

Views Components Routes

Figura – Diagrama de pacotes

O módulo das views (figura) engloba as telas do sistema, como Selection Sort e Bubble Sort, os demais são ‘cópias’ dessas duas views que iniciaram o projeto, sofrendo. Em termos simples, as views são semelhantes aos arquivos .html padrão no desenvolvimento web, pois são os elementos retornados diretamente de uma requisição HTTP.



O módulo das rotas permaneceu o mesmo (figura1), apenas as novas rotas (home, insertiosort, busca binária, busca sequencial e fatorial) (figura2) foram adicionadas.



Figura1 - pasta router

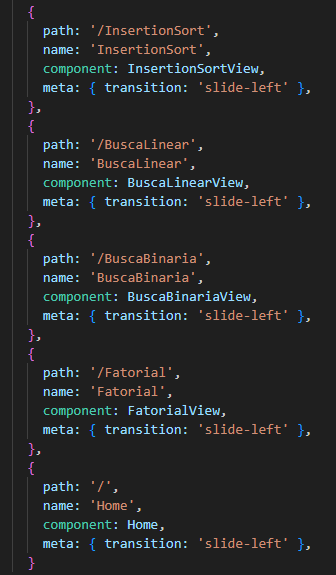


Figura 2 - novas rotas

Cada página de um website, especialmente ao considerar frameworks no Front-end como Vue e React, abraça o conceito de componentização de maneira ainda mais pronunciada. Na essência, a componentização é um dos fundamentos primordiais do Vue. Este conceito implica que uma aplicação é composta por múltiplos elementos autônomos capazes de interagir entre si.

Na primeira versão do projeto, os módulos representam os elementos que serão reutilizados pela view, nesse caso, o projeto fez a escolha da utilização do modal, que é a área responsável por mostrar as opções de escolha ao usuário.