

## Como rodar o exercício

Na raiz do projeto, digite no console “npm install”.

Para rodar a aplicação, vá até a raiz do projeto e digite no console “node app”;

Para rodar os testes, vá até a raiz do projeto e digite no console “npm test”.

## Introdução

Para realização do exercício, foi utilizado a linguagem javascript interpretada por nodejs e a utilização de 4 dependências, sendo elas Babel, Jest, cvstojson e readline.

## Funcionamento

Primeiramente vamos partir ao arquivo “app.js”. Ele importa os métodos criados e o readline, que será utilizado para criar interface da aplicação. Após sua inicialização, chama a função menu(), que possui uma interface de entrada e saída de dados que aguarda a entrada do usuário e usa ela através de uma estrutura condicional. Para cada valor que o usuário digitar, há uma função que gera a interface dos 4 métodos que temos, ao entrar em uma dessas funções, o mesmo processo ocorre, o programa aguarda a entrada do valor e caso cumpra os requisitos, a função chama o método que então retorna o valor esperado, caso isso não ocorra o programa informa que o dado está inválido e aguarda novamente uma nova entrada.

## Métodos

Com o “app.js” explicado vamos para a parte dos “methods.js”, toda a manipulação dos dados da planilha fora feita utilizando json, para isso foi utilizado a dependência cvstojson que transformou o arquivo.csv e o retornou nesse padrão.

### **getStudentsListJson()**

Falado anteriormente, esse método então, retornava o arquivo em formato csv para json.

### **filterByYear(year)**

Método para atender a busca do primeiro bolsista do ano. Ele então recebe como parâmetro o ano, com isso é realizado uma busca pelo vetor de bolsistas filtrando todos

que se encaixam nesse ano, após isso o método retorna o índice zero desse vetor, ou seja, o primeiro bolsista que ele localizou nessa busca. Também há um tratamento de dados caso seja colocado um dado inválido ou caso não haja bolsistas que tenham aquele ano informado.

### **criptName(name)**

Método que atende a criptografia do nome do bolsista. Primeiramente, foi necessário formatar o nome que veio como parâmetro, retirando acentos e colocado em caixa alta. Em seguida é realizado uma busca no vetor de bolsistas e caso o nome do bolsista possua parte do nome parâmetro, será retornado dentro de um vetor. Depois, é selecionado apenas o primeiro elemento desse vetor e então é feito de criptografia do nome, antes disso é verificado se existe algum bolsista que atendeu o requisito, se não tiver o método retorna undefined. Para a criptografia primeiro o nome é quebrado em letras com o método split e através de um vetor auxiliar com as letras do alfabeto, cada index do vetor do nome vai receber a próxima letra com base no seu valor do index, feito isso, é realizado a troca da primeira letra pela última, antes do retorno há uma verificação se esse nome é maior que 3, se for, é aplicado o método reverse. Após a criptografia pronta é aplicado o método join para juntar o nome novamente.

### **averageYear(year)**

Método que atende realizar a média dos valores pagos aos bolsistas do ano. Primeiramente é filtrando todos os valores do vetor de bolsistas que atendam ao ano parâmetro, isso então é armazenado dentro de um vetor, após isso ele é retornado somando todos os itens, dividindo pelo tamanho do mesmo e fixando 2 casas após o ponto. Há uma verificação caso não tenha nenhum bolsista pertencente ao ano parâmetro ou caso o ano parâmetro seja um dado inválido.

### **showRankingsStudents()**

Método que atende a realização do ranking de maiores e menores valores pagos aos bolsistas. Primeiramente é buscado todos os valores dos estudantes e armazenados em um vetor, após isso foi utilizado o new Set, que retorna um vetor de itens únicos e os primeiros achados. Feito isso é realizado o método sort que organiza os valores em ordem crescente. Com os valores únicos e em ordem percorre-se novamente o vetor de bolsistas e através do find, retorna o primeiro valor que corresponder a cada index do vetor de valores únicos. Por último para separar os maiores dos menores, foi usado o método slice que pegou posições específicas do vetor e então, realizou o retorno.

## Testes

Para realização dos testes foi utilizado o Jest juntamente com o Babel que facilita esse processo. Através do jest foi feito testes unitários referentes aos métodos, onde comparei os resultados obtidos pela aplicação com o filtro do Excel, os mesmos podem ser observados através do comando “npm test”. Quanto a interface, como acredito que seria mais bem representado dessa maneira, fiz um vídeo testando principalmente o comportamento ao colocar dados errados, voltar ao menu, sair da aplicação etc. Esse vídeo está localizado na pasta “parteTeorica”.

## Autoavaliação

Acredito que realizei um bom trabalho com o tempo que tive e com as ferramentas que sabia utilizar, creio que consegui organizar bem a estrutura do código dos métodos de modo que passam clareza do que está ocorrendo, tanto por nomear adequadamente as variáveis, como a organização das etapas dentro de cada método, já na parte da interface creio que consegui deixar ela bem completa com as verificações do dados que o usuário digita assim como achei que ela ficou intuitiva se tratando de uma interface via console. Como melhorias creio que poderia ter organizado a estrutura do código da interface de outra forma, poderia fazer melhor de modo que ficasse mais limpo a sua visualização, assim como poderia ter reaproveitado melhor alguns códigos, principalmente nas opções que a interface dá, pois são muito parecidas. De modo geral gostei muito de realizar o projeto, nunca havia manipulado um arquivo.csv de fato foi muito interessante, tive algumas dificuldades, principalmente na lógica de criptografar o nome, certamente foi a que levei mais tempo, mas no fim, consegui organizar os passos e realizar, também foi um pouco árduo realizar a classificação dos rankings dos bolsistas, já que de primeiro momento desconhecia o new Set, com certeza facilitou muito o processo.