

Conversor de notas escolares

Função do conversor de notas escolares é basicamente converter notas escolares do formato americano, conhecido por ser letras, para o sistema de notas brasileiro que vai de números de 0 a 10.

Conversor de notas escolares é um sistema capaz de ler um arquivo texto escrito com diversas instruções predefinidas e entregar o resultado da computação gerada dessas instruções.

Sistema implementado na linguagem de programação javascript utilizando do subconjunto typescript. Back-end desenvolvido utilizando a Node.js e front-end desenvolvido utilizando React.

Requisitos

Para a execução do back-end do sistema será necessário a instalação do Node.js e seu gerenciador de pacotes, Yarn.

Para a execução do front-end será necessário um navegador web atualizado.

Node

Instalação do Node no Windows

Acesse o site oficial (<https://nodejs.org/pt-br/download/>) e baixe instalador.

Instalação do Node no Ubuntu

Você pode instalar Node e npm com o comando apt install, no terminal linux.

```
$ sudo apt install nodejs
```

```
$ sudo apt install npm
```

Outros sistemas operacionais

Você pode encontrar mais informações sobre a instalação no site oficial (<https://nodejs.org/pt-br/download/>).

Yarn

Depois de instalar o node, este projeto vai precisar também do yarn, então execute o seguinte comando no terminal.

```
$ npm install -g yarn
```

Instalação

Para executar o sistema será necessário instalar todos os pacotes dependentes usando os seguintes comandos no terminal.

Instalar pacotes do back-end. Na pasta do back-end (/server) execute:

```
/server$ yarn install
```

Instalar pacotes do front-end. Na pasta do front-end (/web) execute:

```
/web$ yarn install
```

O back-end da implementação estará rodando no endereço: (<http://localhost:3333>).

O front-end da implementação estará rodando no endereço: (<http://localhost:3000>).

Execução

Após a instalação, para rodar o sistema execute seguintes comandos no terminal.

Executar back-end. Na pasta do back-end (/server) execute:

```
/server$ yarn start
```

Executar front-end. Na pasta do front-end (/web) execute:

```
/web$ yarn start
```

Utilização

Imagens da execução

Com todos os requisitos devidamente instalados assim como o próprio sistema, após a execução, o resultado deve ser parecido com as seguintes imagens.

Back-end rodando:

```
fernando@fernando-pc:~/Documents/Projetos/Dell_IT_Academy/server$ yarn start
yarn run v1.22.5
$ tsnd --transpile-only --ignore-watch node_modules --respawn src/server.ts
ts-node-dev ver. 1.0.0-pre.62 (using ts-node ver. 8.10.2, typescript ver. 4.0.2)
```

Front-end rodando:

```
Compiled successfully!

You can now view web in the browser.


Local:      http://localhost:3000
On Your Network: http://192.168.0.107:3000

Note that the development build is not optimized.
To create a production build, use yarn build.
```

Interface web no navegador:

Conversor de notas escolares

- 1- Clique ou arraste o arquivo na dropzone.
- 2- Clique em Adicionar para fazer o upload do arquivo.
- 3- Clique em Converter para realizar a conversão do arquivo.
- 4- Clique em Atualizar para adicionar outro arquivo.


Adicione o arquivo texto

Adicionar

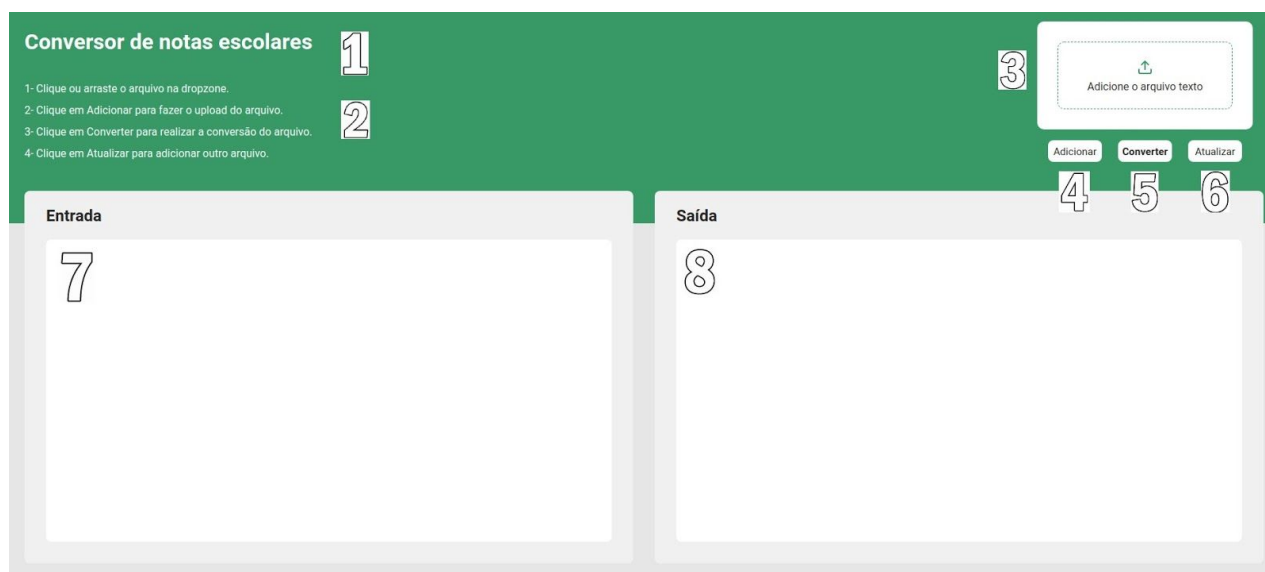
Converter

Atualizar

Entrada

Saída

Descrição do conteúdo da página web:



1- Título da aplicação

2- Descrições de uso

3- Dropbox para inserção do arquivo texto

4- Botão para fazer o upload do arquivo

5- Botão para realizar a conversão do arquivo

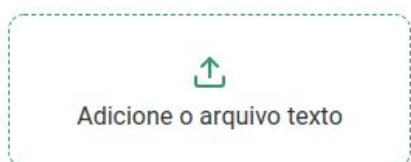
6- Botão para atualizar a página, para uma nova utilização

7- Área onde o conteúdo do arquivo texto é mostrado

8- Área onde o resultado da conversão do arquivo texto é mostrado

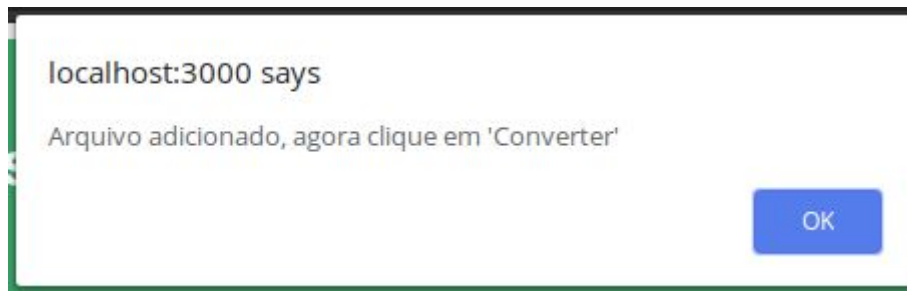
Dropbox

Para a inserção do arquivo texto é usado uma dropbox area, podendo assim arrastar ou clicar na dropbox para anexar o arquivo.



Imagens do dropbox, antes e depois do arquivo ser anexado.

Alerta



Após o usuário clicar em adicionar, um alerta é mostrado na tela indicando que o upload do arquivo foi executado.

Resultado após clicar no botão converter

Entrada (mostra o conteúdo do arquivo):

Entrada

Logica Matematica Media C

Engenharia de Software Prova1 A

Engenharia de Software Prova2 B

Banco de Dados Media B

Teoria da Computacao Prova1 F

Teoria da Computacao Prova2 D

qual a média em pontuacao brasileira de Logica Matematica?

voce foi aprovado em Engenharia de Software ?

qual a media da disciplina de Teoria da computacao?

voce foi aprovado em todas as disciplinas?

quantos creditos você cursou neste semestre?

quantos creditos você concluiu?

Saída (mostra o resultado da conversão do arquivo):

Saída

A media em Logica matematica é 5.95
Sim Engenharia de software media 8.725
Teoria da computacao media 2.7
Não reprovei em Teoria da computacao
No semestre cursei 19 creditos
Conclui apenas 16 creditos

Código do Back-end

Constantes, Interfaces e Variáveis:

```
const grades = ['a', 'b', 'c', 'd', 'f'];

const subjects = [
  { name: 'logica matematica', credits: 4 },
  { name: 'engenharia de software', credits: 6 },
  { name: 'teoria da computacao', credits: 3 },
  { name: 'banco de dados', credits: 6 },
  { name: 'arquitetura de software', credits: 4 }
]

var studentAverages = <subject[]>[];
var studentTests = <test[]>[];
var studentAveragesTests = <subject[]>[];
```

```
interface subject {
  name: string,
  average: number
}

interface test {
  subjectName: string,
  testNumber: number,
  grade: number
}
```

A constante grades armazena as notas possíveis no sistema de notas escolares americano.

A constante `subjects` armazena as disciplinas possíveis e seu respectivo número de créditos.

A interface `subject` é utilizada para definir as propriedades de uma disciplina que são o nome e a média.

A interface `test` é utilizada para definir as propriedades de uma prova que são o nome da disciplina, se é a primeira ou segunda prova e a nota.

A variável `studentAverages` armazena as disciplinas encontradas no arquivo texto que contém a média em formato americano (letras) e suas respectivas médias convertidas.

A variável `studentTests` armazena todas as provas encontradas no arquivo texto.

A variável `studentAveragesTests` armazena as disciplinas encontradas no arquivo texto que contém a prova 1 e a prova 2, sendo possível calcular a média armazenando a mesma.

```
let inputArray: string[] = [];  
let outputArray: string[] = [];  
  
r1.on('line', function(line) {  
    inputArray.push(line);  
  
    const newLine = classifyLine(removeAccents(line));  
    outputArray.push(newLine);  
});
```

A variável `inputArray` é um vetor de textos que receberá cada linha do arquivo para apresentar o conteúdo original no front-end.

A variável `outputArray` é um vetor de textos que receberá a linha convertida para apresentar no front-end.

A constante `line` é a linha do arquivo.

Funções

Funções encontradas em: (server/src/Controllers/FileController.ts)

removeAccents

Responsável por remover os acentos das palavras encontradas nas linhas do arquivo.

```
function removeAccents (line: string) {  
}
```

gradeLetterToNumber

Converte as notas americanas, que são letras, em notas brasileiras que são números de 0 a 10.

```
function gradeLetterToNumber(grade: string | undefined) { ...  
}
```

classifyLine

Classifica a linha recebida em três opções: linhas de pergunta, linhas que não são de pergunta, mas que trazem uma nota em formato americano ou uma linha fora dos padrões necessários para a conversão.

```
function classifyLine(line: string) { ...  
}
```

classifyNonQuestion

Classifica as linhas que não são perguntas em duas opções, as linhas que apresentam uma média e as que apresentam uma prova.

```
function classifyNonQuestion(line: string) { ...  
}
```

mediaLine

Armazena a disciplina apresentada com sua respectiva média.

```
function mediaLine(line: string) { ...  
}
```


testLine

Classifica as linhas que contém a nota de provas entre as linhas de prova 1 e de prova 2.

```
function testLine(line: string) { ...  
}
```

testLine1

Armazena a prova 1 apresentada com sua respectiva nota.

```
function test1Line(line: string) { ...  
}
```

testLine2

Armazena a prova 2 apresentada com sua respectiva nota.

```
function test2Line(line: string) { ...  
}
```

alreadyHasTest

Verifica se a prova já foi armazenada.

```
function alreadyHasTest(studentTest: test) { ...  
}
```

alreadyHasTest1

Verifica se uma prova 1 já foi armazenada, só assim permitir o armazenamento da prova 2.

```
function alreadyHasTest1(subjectName: string) { ...  
}
```

classifyQuestion

Classifica as linhas de perguntas pelo seu início. Podendo existir linhas que começam as tais palavras: qual, você, quantos e para.

```
function classifyQuestion(line: string) { ...  
}
```

whichLine

Classifica as linhas que começam com 'qual' entre as que querem descobrir a média de uma disciplina ou a nota de uma prova.

```
function whichLine(line: string) { ...  
}
```

whichAverageLine

Classifica as linhas entre as três possíveis opções:

1. qual a média em pontuação brasileira de (*nome da disciplina*)?
2. qual a média da disciplina de (*nome da disciplina*)?
3. qual a média em (*nome da disciplina*)?

```
function whichAverageLine(line: string) { ...  
}
```

getAverage e getAverage2

Retornam a média da disciplina passada como parâmetro.

```
> function getAverage(subjectName: string) { ...  
}  
  
> function getAverage2(subjectName: string) { ...  
}
```

whichGradeLine

Classifica as linhas entre as duas possíveis opções:

1. qual a nota da (*prova1* ou *prova2*) em (*nome da disciplina*)?
2. qual a nota em pontuação brasileira preciso tirar em (*nome da disciplina*) para passar na disciplina?

```
function whichGradeLine(line: string) { ...  
}
```

getAnotherTestGrade

Retorna qual a nota que aluno obteve na prova não apresentada, baseado no cálculo da média e da nota da prova apresentada.

```
> function getAnotherTestGrade(subjectName: string, testNumber: number) {...
```

getGradeToBeApproved

Retorna qual a nota o aluno deve obter na segunda prova para ser aprovado, baseado no cálculo da média e da nota da primeira prova.

```
function getGradeToBeApproved(subjectName: string) {...
```

didYouLine

Classifica a pergunta em duas diferentes alternativas: se quer verificar se o aluno está aprovado em uma determinada disciplina ou se o aluno foi aprovado em todas as disciplinas.

```
function didYouLine(line: string) {...
```

checkSubjectApproved

Verifica se o aluno foi aprovado na disciplina passada no parâmetro.

```
function checkSubjectApproved(subjectName: string) {...
```

checkSubjectApproved

Verifica se o aluno foi aprovado na disciplina passada no parâmetro.

```
function checkApprovedAll(line: string) {...
```

howManyLine

Verifica se a linha pergunta quantos créditos o aluno cursou ou quantos créditos o aluno concluiu.

```
function howManyLine(line: string) {...
```

checkStudiedCredits

Reúne todas as disciplinas mencionadas no arquivo texto e retorna quantos créditos o aluno cursou.

```
function checkStudiedCredits() {...
```

checkConcludedCredits

Reúne todas as disciplinas mencionadas no arquivo texto e retorna o total de créditos que o aluno concluiu.

```
function checkConcludedCredits() {...
```

toGetAverageLine

Verifica se a linha é igual a do exemplo abaixo:

para ficar com média (*média*) em (*nome da disciplina*) qual deve ser a nota da prova2?

```
function toGetAverageLine(line: string) {...
```

getGradeToReachAverage

Calcula, baseado na média e na primeira prova da disciplina apresentada, quanto o aluno deve obter na segunda prova para alcançar a média também apresentada.

```
function getGradeToReachAverage(subjectName: string, averageToReach: number) {
```

Observações importantes

Conversão de notas

Para realizar a conversão das notas americanas para as notas brasileiras é necessário utilizar a tabela abaixo:

Notas EUA	Notas Brasil
A	9 – 10
B	7 – 8,9
C	5 – 6,9
D	3 – 4,9
F	0 – 2,9

Não ficou muito claro para mim qual a maneira que eu deveria converter essas notas, com o menor valor das notas brasileiras ou com o maior valor.

Por tanto na função abaixo foi convertida a nota americana para a média das notas brasileiras.

```
function gradeLetterToNumber(grade: string | undefined) {  
  
    switch (grade){  
  
        case grades[0]:  
  
            return (9+10)/2;  
  
        case grades[1]:  
  
            return (7+8.9)/2;  
  
        case grades[2]:  
  
            return (5+6.9)/2;  
  
        case grades[3]:  
  
            return (3+4.9)/2;  
  
        case grades[4]:  
  
            return (0+2.9)/2;  
  
    }  
}
```

Número de créditos cursados

Todas as disciplinas conhecidas pelo programa estão na tabela abaixo.

Disciplina	Créditos
Lógica matemática	4
Engenharia de software	6
Teoria da computação	3
Banco de dados	6
Arquitetura de software	4

No primeiro exemplo de teste foi apresentado no arquivo texto o total de 4 disciplinas, sendo elas: Lógica matemática, Engenharia de software, Banco de dados e Teoria da computação. A disciplina de Arquitetura de software não foi apresentada nesta execução.

Exemplo 1:

Entradas (de Teste)

```
Logica Matematica Media C
Engenharia de Software Prova1 A
Engenharia de Software Prova2 B
Banco de Dados Media B
Teoria da Computacao Prova1 F
Teoria da Computacao Prova2 D
qual a média em pontuacao brasileira de Logica Matematica?
voce foi aprovado em Engenharia de Software ?
qual a media da disciplina de Teoria da computacao?
voce foi aprovado em todas as disciplinas?
quantos creditos você cursou neste semestre?
quantos creditos você concluiu?
```

Na pergunta '**quantos créditos você cursou neste semestre?**' o exemplo de saída mostra o total de 23 créditos.

Na pergunta '**quantos créditos você concluiu?**' o exemplo de saída mostra o total de 20 créditos.

Saídas (do Teste)

```
A media em Logica matematica foi 5
Sim Engenharia de Software media 9.4
Teoria da computacao media 2.4
Não reprovei em Teoria da computacao
No semestre cursei 23 creditos
Conclui apenas 20 creditos
Nem ideia do que isto significa!
```

A resposta da primeira pergunta leva a entender que o programa buscou o total de créditos das disciplinas na tabela de disciplinas e não apenas os créditos das disciplinas apresentadas no arquivo texto. Já que ele apresenta o total de 23 créditos, contando com os 4 créditos da disciplina de Arquitetura de software, não apresentada no arquivo texto.

A resposta da segunda pergunta mostra que o aluno concluiu 20 créditos, levando em consideração a sua reprovação em Teoria da computação, foram descontados 3 créditos do total de 23 créditos apresentados na pergunta antecessora.

A analisando as duas perguntas e suas respostas levam ao seguinte questionamento:

Como o programa concluiu que o aluno foi aprovado em Arquitetura de software sendo que a disciplina não foi mencionada no arquivo texto, assim não sendo apresentada nenhuma média ou prova da mesma.

Levando em conta esse questionamento o programa desenvolvido por mim, responde de forma diferente à apresentada no exemplo.

As perguntas '**quantos créditos você cursou neste semestre?**' e '**quantos créditos você concluiu?**' são respondidas levando em consideração apenas as disciplinas presentes no arquivo texto, como nas imagens abaixo.

Entrada	Saída
Logica Matematica Media C ————— 4	A media em Logica matematica é 5.95
Engenharia de Software Prova1 A ————— 6	Sim Engenharia de software media 8.725
Engenharia de Software Prova2 B ————— 6	Teoria da computacao media 2.7
Banco de Dados Media B ————— 6	Não reprovei em Teoria da computacao
Teoria da Computacao Prova1 F ————— 3	No semestre cursei 19 creditos
Teoria da Computacao Prova2 D ————— 3	Conclui apenas 16 creditos
qual a média em pontuacao brasileira de Logica Matematica?	
voce foi aprovado em Engenharia de Software ?	
qual a media da disciplina de Teoria da computacao?	
voce foi aprovado em todas as disciplinas?	
quantos creditos você cursou neste semestre?	
quantos creditos você concluiu?	
	Créditos cursados 19

Entrada	Saída
Logica Matematica Media C ————— 4	A media em Logica matematica é 5.95
Engenharia de Software Prova1 A ———— 6	Sim Engenharia de software media 8.725
Engenharia de Software Prova2 B ———— 6	Teoria da computacao media 2.7
Banco de Dados Media B ————— 6	Não reprovei em Teoria da computacao
Teoria da Computacao Prova1 F ———— 3	No semestre cursei 19 creditos
Teoria da Computacao Prova2 D ———— 3	Conclui apenas 16 creditos
qual a média em pontuacao brasileira de Logica Matematica?	
voce foi aprovado em Engenharia de Software ?	
qual a media da disciplina de Teoria da computacao?	
voce foi aprovado em todas as disciplinas?	
quantos creditos você cursou neste semestre?	
quantos creditos você concluiu?	Créditos concluidos 16

Se você chegou até aqui

Muito obrigado pela atenção

Sobre o autor

Fernando Costa Heisser

Linkedin: Fernando Heisser

fernando.heisserch@gmail.com

GitHub: FernandoHeisser