



Desenvolvimento de um aplicativo mobile para ensino de programação com a linguagem C

Universidade Federal de Uberlândia

Aluno: Lucas Riccioppo Silva

Orientador: Prof. Dr. André Ricardo Backes



Introdução

Alto índice de reprovação nas matérias de programação

- Alunos sem conhecimento prévio em programação

Primeiro contato com a programação

Tablets e Smartphones

- Alunos normalmente entram no curso já com prática na utilização de smartphones



Objetivo

Desenvolvimento de aplicação que seja de fácil acesso, ágil e elegante, que esteja disponível para quando a pessoa desejar estudar.



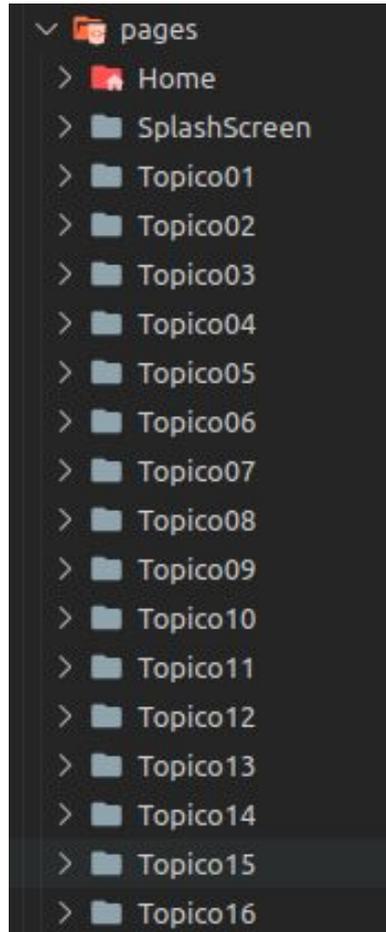
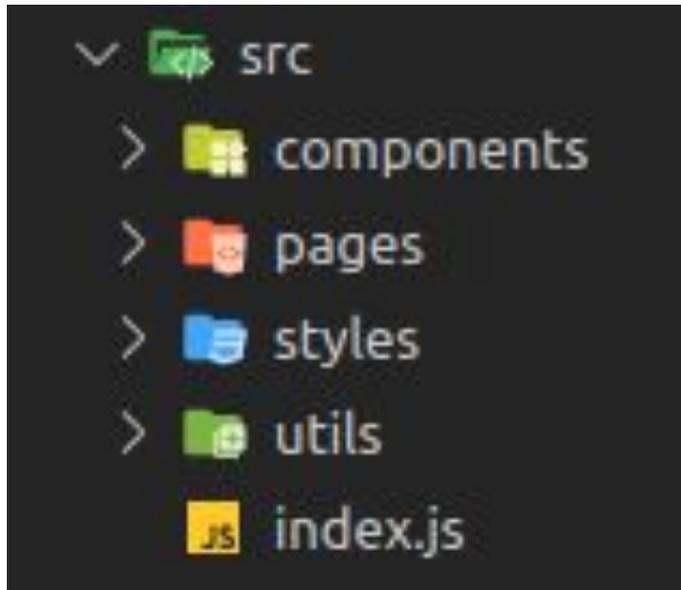
Estado da arte





Ferramentas utilizadas

- Linux
- Visual Studio Code
- JavaScript
- React Native
- Android
- Android Studio
- Git







Desenvolvimento

Introdução

Tipos e variáveis

Leitura e escrita de variáveis

Operadores aritméticos

Operadores relacionais

Operadores lógicos

Comandos condicionais

Comandos de repetição

Array

String

Struct

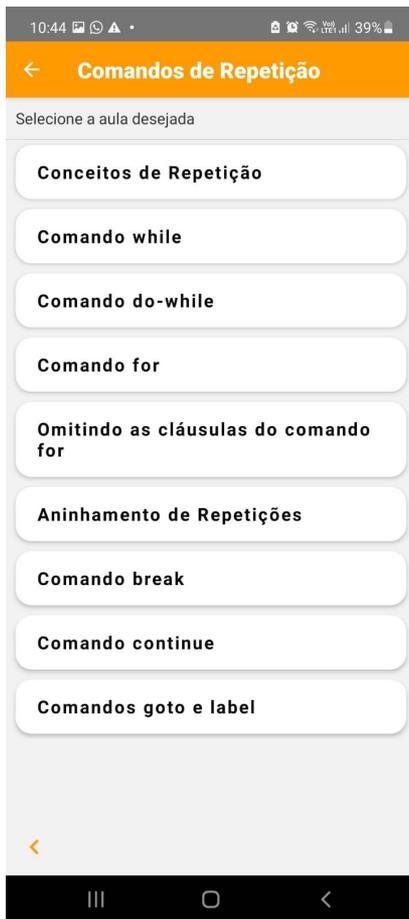
Funções

Funções recursivas

Ponteiros

Alocação dinâmica

Arquivos



Struct

As variáveis vistas até agora podem ser classificadas em:

-Simples: tipos **int**, **float**, **double** e **char**

-Compostas homogêneas: do mesmo tipo definidas por **array**

A linguagem C permite que se crie **novos tipos** a partir dos tipos básicos: um deles é a **struct**, também chamada de estrutura

Uma **estrutura** pode ser vista como um novo tipo de dado formado pela composição de variáveis de outros tipos.

Estruturas são declaradas da seguinte forma:

```
struct nomestruct {  
    tipo1 campo1;  
    tipo2 campo2;  
    ...  
    tipoN campoN;  
};
```

Observação: uma **struct** pode ser declarada em qualquer escopo.



Struct - Exemplo

Em um cadastro de pessoas todas as informações são da mesma pessoa, logo podemos agrupá-las em uma

struct:

```
char nome[50];
```

```
int idade;
```

```
char rua[50];
```

```
int numero;
```

cadastro

Na linguagem C criamos a **struct** cadastro de pessoas da seguinte maneira:

```
struct cadastro {  
    char nome[50];  
    int idade;  
    char [50];  
    int numero;  
};
```



Passagem de Parâmetros

Passagem por valor

Na linguagem C, os parâmetros de uma função são sempre passados por valor.

Isso significa que uma cópia do valor do parâmetro é feita e passada para a função.

Mesmo que esse valor mude dentro da função, nada acontece com o valor de fora da função.

```
void incrementa(int n) {
    n = n + 1;
    printf("Dentro da funcao: x = %d\n", n);
}

int main() {
    int x = 5;
    printf("Antes da funcao: x = %d\n", x);
    incrementa(x);

    printf("Depois da funcao: x = %d\n", x);
    return 0;
}
```

Saída:

```
Antes da funcao: x = 5
Dentro da funcao: x = 6
Depois da funcao: x = 5
```

```
void incrementa(int n) {
    n = n + 1;
    printf("Dentro da funcao: x = %d\n", n);
}

int main() {
    int x = 5;
    printf("Antes da funcao: x = %d\n", x);
    incrementa(x);
    printf("Depois da funcao: x = %d\n", x);
    return 0;
}
```

Saída:

```
Antes da funcao: x = 5
Dentro da funcao: x = 6
Depois da funcao: x = 5
```



Conclusão

Aplicativo desenvolvido

Responsividade

Navegabilidade

Trabalhos futuros