

IFSCee

PROJETO INTEGRADOR I

UMA FERRAMENTA WEB EDUCACIONAL PARA AUXILIAR NO ENSINO DA LINGUAGEM C

CLIENTE: LEONARDO PERIN RAUTA

ALUNOS DO ADS DO IFSC

DEVELOPER: GELASIO EBEL JUNIOR

PROFESSORES: RENATO SIMÕES MOREIRA

LEONARDO PERIN RAUTA

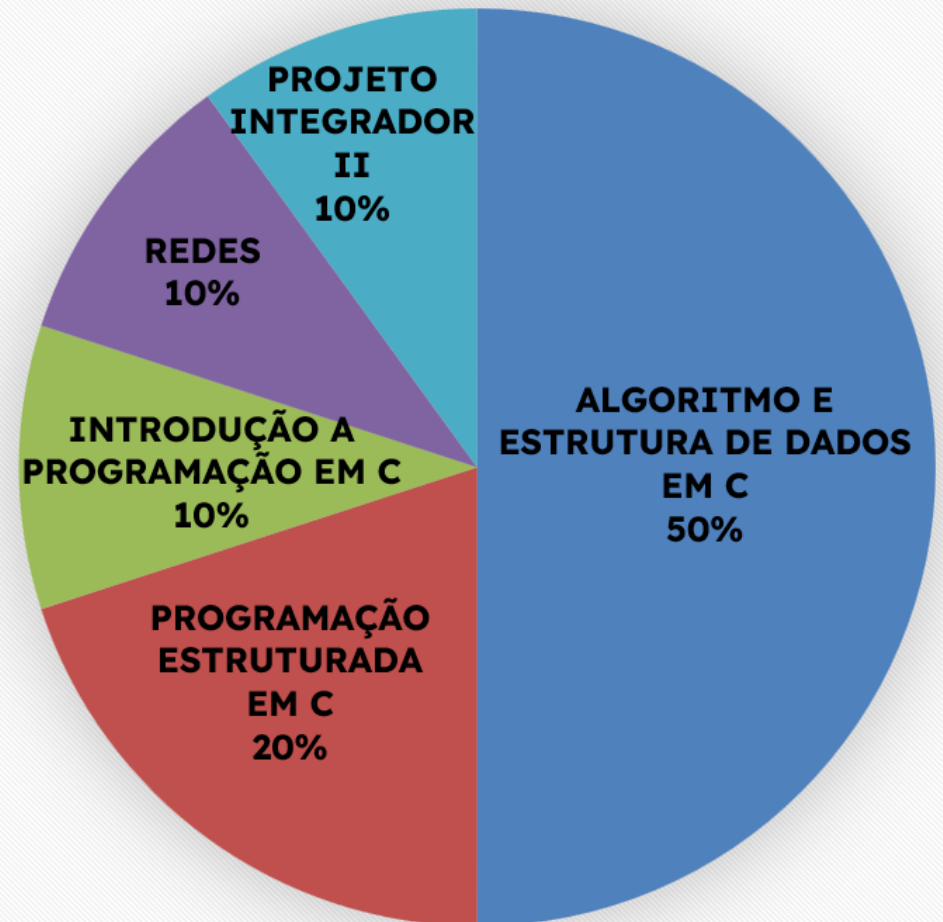
2024 -2

IFSCee

PROBLEMA:

- Aula AED
- Monitoria
- Demanda por suporte em C
 - Lógica
 - Ponteiros
 - Recursividade

DISCIPLINAS ATENDIDAS



Fonte: *Monitoria 2024-2 ADS*

IFSCee

PROBLEMA:

- Variáveis, Arrays, Ponteiros, Structs
- Diferentes tipos de passagem de parâmetros de função
- Erros fora dos limites da matriz
- Identificar ponteiros desalinhados
- Ponteiros de ponteiros

IFSCee

SOLUÇÃO:

Uma ferramenta web interativa
para interpretação e visualização
de algoritmos em C,
sem necessidade de compilação.

IFSCee

OBJETIVO: PROJETO INTEGRADOR I

DESENVOLVER

UMA PROVA DE CONCEITO

- Interpretar algoritmos simples em C
- Visualizar o estado da memória

IFSCee

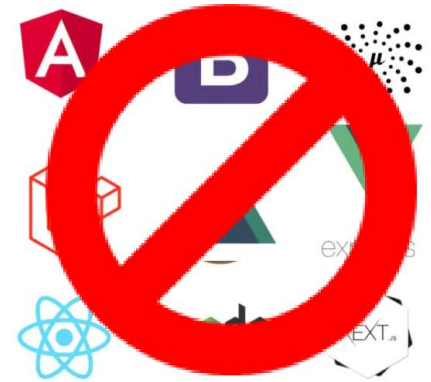
REQUISITOS:

- Inserção de código
- Navegação passo a passo
- Representação de dados na memória
- Mostrar a memória não inicializada

IFSCee

TECNOLOGIAS:

- HTML5 e CSS3
- JavaScript
- Sem dependência de frameworks
- Servidor web simples



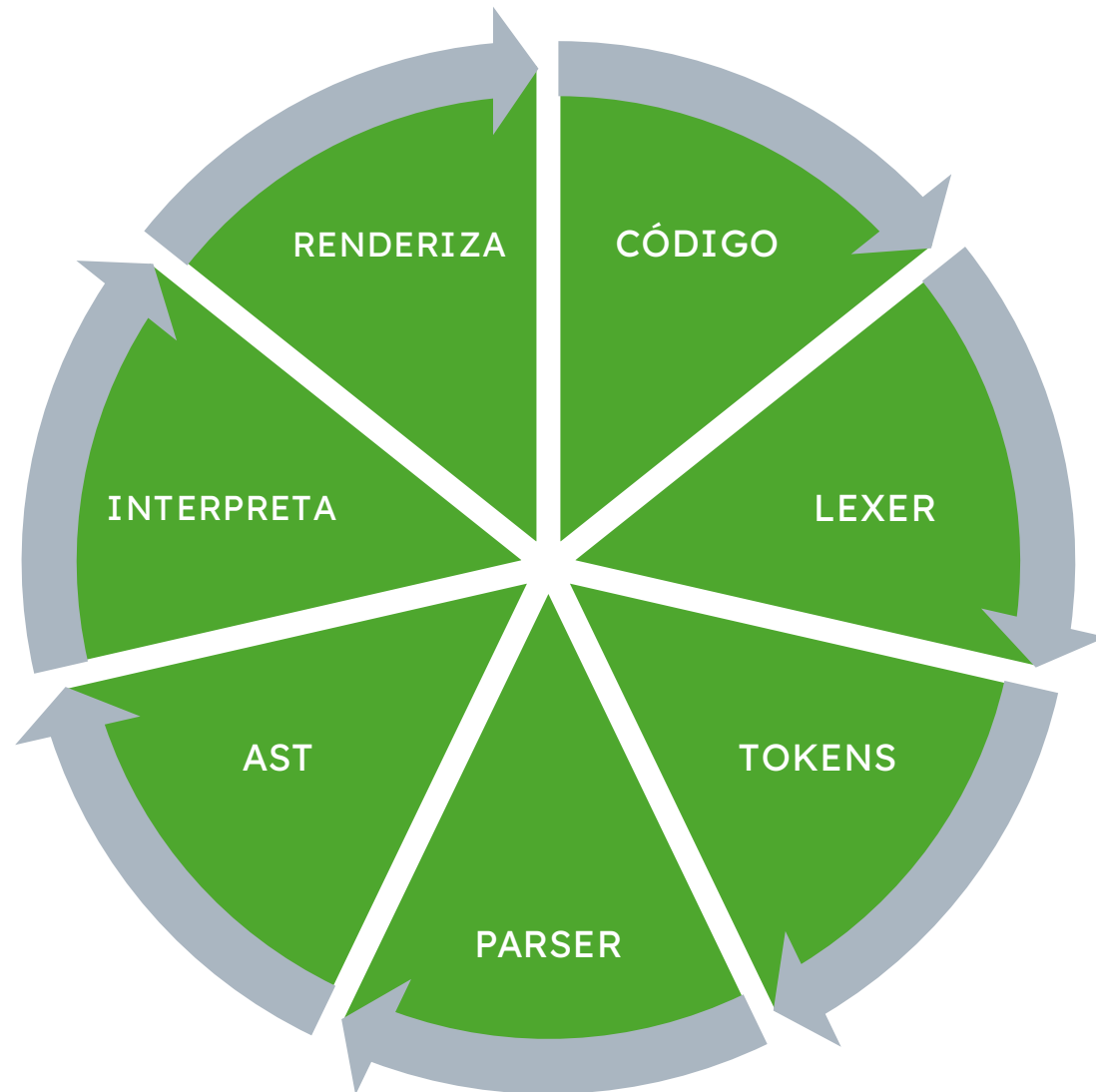
IFSCee

ABORDAGEM INTERPRETAÇÃO:

- Nossa própria “máquina virtual” e conjunto de instruções (JavaScript).
- Nosso próprio analisador léxico de C.
- Nosso próprio analisador sintático de com “caminhada” recursiva.

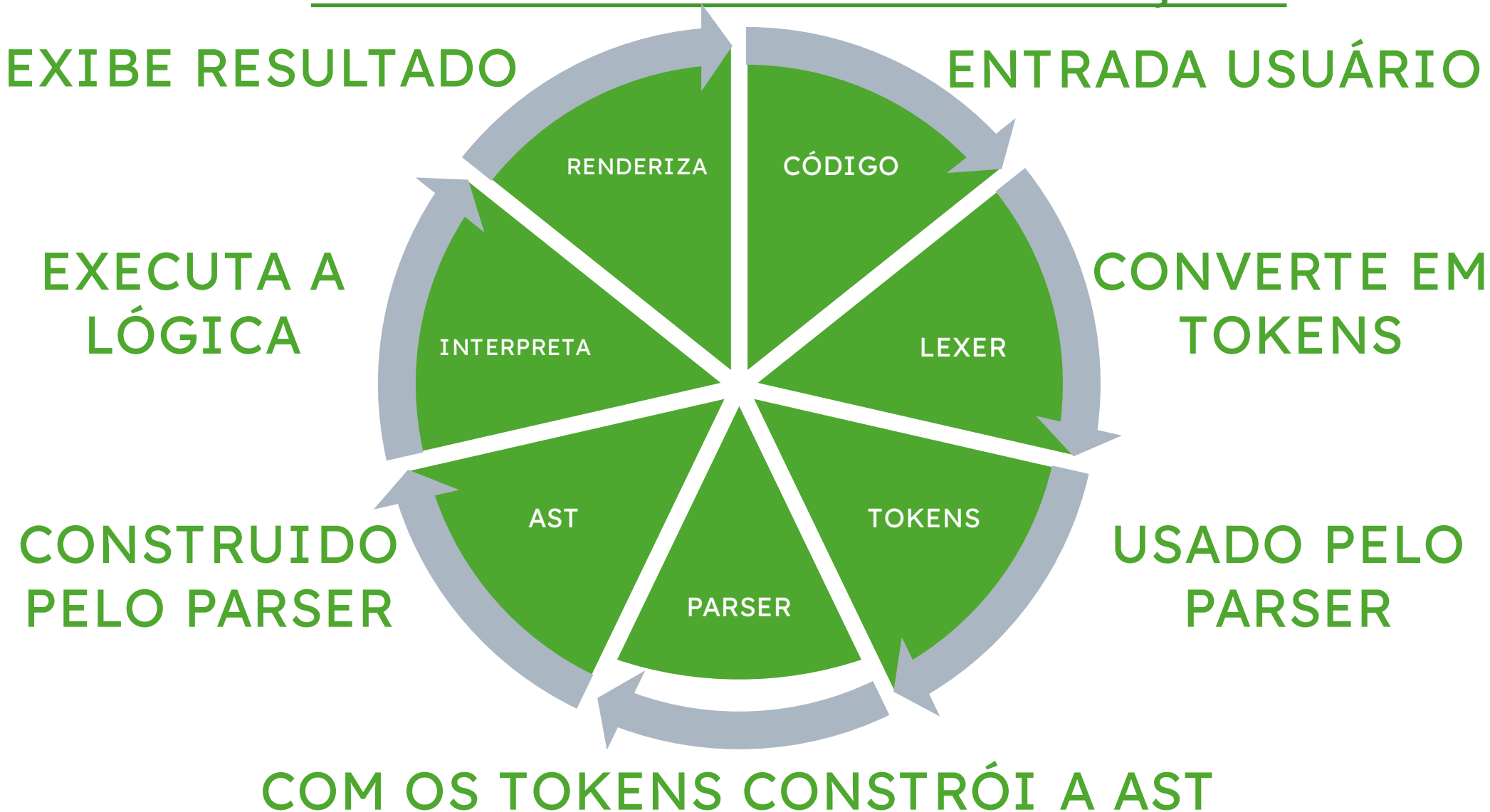
IFSCee

ABORDAGEM INTERPRETAÇÃO:

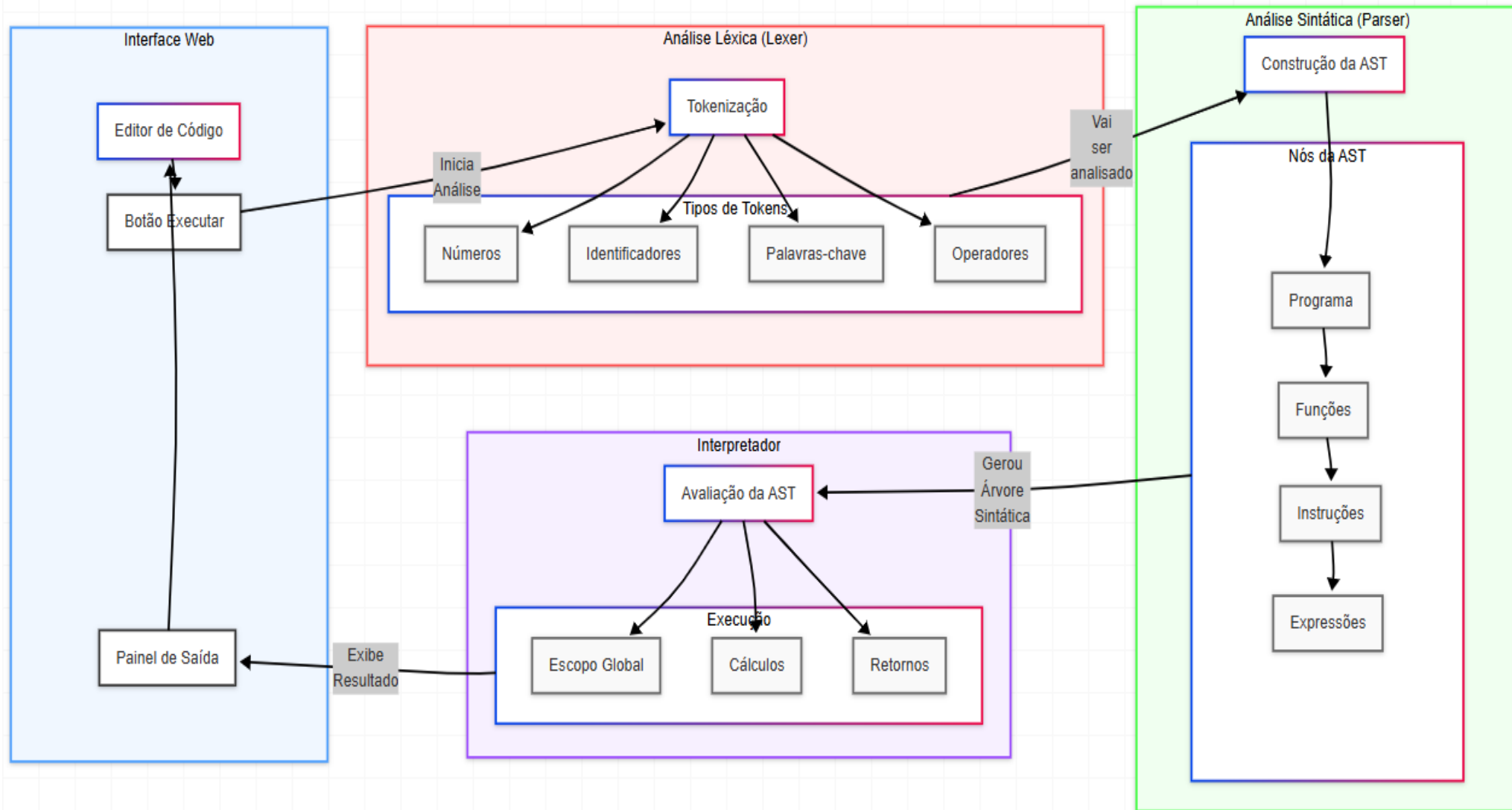


IFSCee

ABORDAGEM INTERPRETAÇÃO:



ABORDAGEM INTERPRETAÇÃO:



IFSCee

ABORDAGEM INTERPRETAÇÃO:

Memória:

- Stack
- Heap
- Dados

Instruções:

- Mover
- Empurar
- Pular
- Remover
- Chamadas
- Contador
- Alocar
- Operadores

IFSCee

Primeira Tentativa:

C Code Editor

```
// Example code with loops and conditions
int sum = 0;
int i = 1;

// For loop example
for(i = 1; i <= 5; i = i + 1) {
    sum = sum + i;
    printf("Current sum is %d\n", sum);
}

printf("Final sum is %d\n", sum);

// While loop example
int count = 0;
while(count < 3) {
```

Compile

Step Back

Step Forward

Reset

Memory View

Program Output

IFSCee

IDEIA/PROTÓTIPO:

IFSCee

Visualize e entenda código C passo a passo

Editor de Código

```
#include <stdio.h>

int calcular_factorial(int n) {
    if (n <= 1) {
        return 1;
    } else {
        return n * calcular_factorial(n - 1);
    }
}

int main() {
    int result = calculate_factorial(5);
    printf("Fatorial de 5 é %d\n", result);
    return 0;
}
```

Visualizar Execução Limpar Código

Visualização

Estado da Memória

n = 5
result = 120

Pilha de Chamadas

calculate_factorial(n=5)
calculate_factorial(n=4)
calculate_factorial(n=3)

Controles

Primeiro Anterior Próximo Último

Passo 3 de 6

Saída do Programa

Fatorial de 5 é 120

IFSCee

O QUE FALTA AINDA:

VISUALIZAÇÃO DA MEMÓRIA

- Visualização o Heap, Stack
- Mostrar alocação de memória (animação)
- Rastrear ponteiros e seus endereços
- Visualizar arrays, strings, types na memória
- Destacar memória não inicializada

IFSCee

O QUE FALTA AINDA:

INTERFACE E USABILIDADE

- Mensagens de erro
- Documentar exemplos e limitações
- Integrar lista com algoritmos comuns
- Testar com material didático
- Estilizar o Layout

IFSCee

O QUE FALTA AINDA:

PÓS GRADUAÇÃO/MESTRADO

- Bibliotecas padrão (stdio.h, stdlib.h)
- SQLite para algoritmos comuns, salvos.
- Suporte a:
 - Structs
 - Enum
 - FILE*
 - #define, #include

IFSCee

IDEIA ORIGINAL:

<https://pythontutor.com/c.html#mode=edit>

Online C compiler, visual debugger, and AI tutor - the only tool that lets you visually debug your C code step-by-step (also debug [Python](#), [JavaScript](#), [Java](#), and [C++](#) code)

Here is a demo. **Scroll down** to compile and run your own code!

C (C17 + GNU extensions)

```

1 #include <stdio.h>
2
3 int main() {
4     int x[] = {10, 20, 30};
5     int* p = &x[1]; // pointer into middle
6     char* fruit[3] = {"apples",
7                       "bananas",
8                       "cherries"};
9
10    printf("I have %d %s\n", *p, fruit[1]);
11    return 0;
12 }
```

[Edit Code & Get AI Help](#)

→ line that just executed
→ next line to execute

<< First < Prev Next > Last >>

Done running (6 steps)

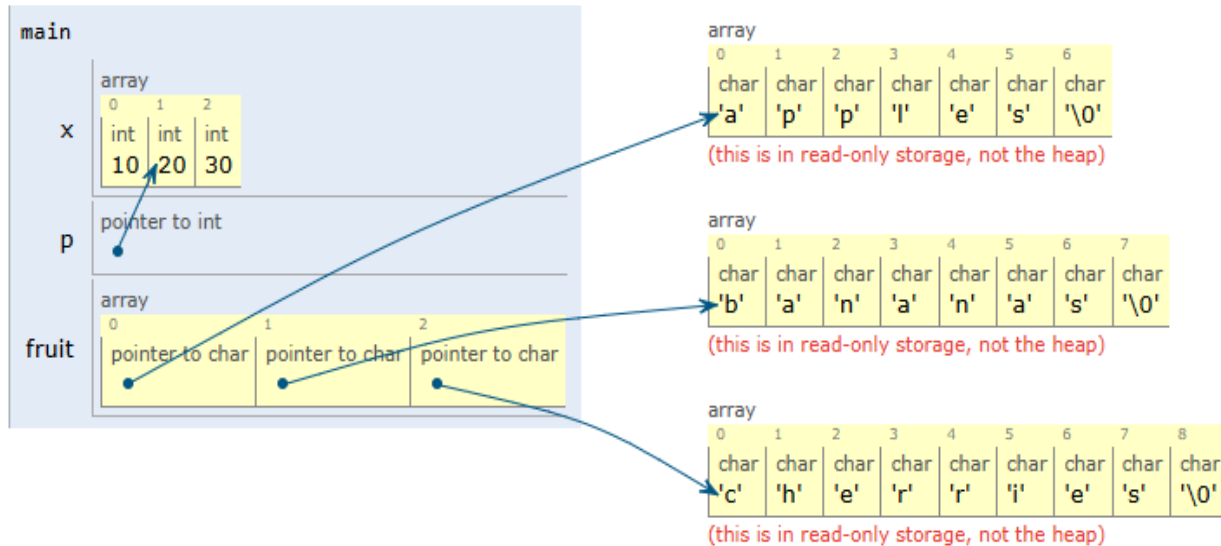
Visualized with pythontutor.com

Print output (drag lower right corner to resize)

I have 20 bananas

Stack

Heap



C/C++ details: none [default view]

IFSCee

REFERÊNCIAS:

<https://pythontutor.com/articles/c-cpp-visualizer.html>

<https://pt.wikipedia.org/wiki/JavaScript>

<https://en.wikipedia.org/wiki/HTML>

<https://en.wikipedia.org/wiki/HTML5>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

<https://www.ibm.com/docs/en/wdfrhcw/1.4.0?topic=commands-debugging-c-c-programs>

<https://github.com/lotabout/write-a-C-interpretor>

<https://github.com/descent/write-a-C-interpretor>

<https://github.com/Pconst167/c-interpretor>

<https://learn.microsoft.com/pt-br/cpp/c-language/organization-of-the-c-language-reference?view=msvc-170>

<https://compilers.iecc.com/crenshaw/>

DELAMARO, M. E. Como construir um Compilador: utilizando ferramentas Java. São Paulo: Novatec, 2004. 262 p.

KERNIGHAN, B. W.; RITCHIE, D. The C Programming Language: ANSI C Version. 2. ed. Upper Saddle River: Prentice Hall, 1988. 274 p.

IFSCee

IDEIA ORIGINAL/OBJETIVO:

<https://pythontutor.com/c.html#mode=edit>

Online C compiler, visual debugger, and AI tutor - the only tool that lets you visually debug your C code step-by-step (also debug [Python](#), [JavaScript](#), [Java](#), and [C++](#) code)

Here is a demo. **Scroll down** to compile and run your own code!

C (C17 + GNU extensions)

```

1 #include <stdio.h>
2
3 int main() {
4     int x[] = {10, 20, 30};
5     int* p = &x[1]; // pointer into middle
6     char* fruit[3] = {"apples",
7                       "bananas",
8                       "cherries"};
9
10    printf("I have %d %s\n", *p, fruit[1]);
11    return 0;
12 }
```

[Edit Code & Get AI Help](#)

→ line that just executed
→ next line to execute

<< First < Prev Next > Last >>

Done running (6 steps)

Visualized with pythontutor.com

Print output (drag lower right corner to resize)

I have 20 bananas

Stack

Heap

main

array		
0	1	2
int	int	int
10	20	30

p
pointer to int

fruit

array		
0	1	2
pointer to char	pointer to char	pointer to char

array

0	1	2	3	4	5	6
char	char	char	char	char	char	char
'a'	'p'	'p'	'l'	'e'	's'	'\0'

(this is in read-only storage, not the heap)

array

0	1	2	3	4	5	6	7
char	char	char	char	char	char	char	char
'b'	'a'	'n'	'a'	'n'	'a'	's'	'\0'

(this is in read-only storage, not the heap)

array

0	1	2	3	4	5	6	7	8
char	char	char	char	char	char	char	char	char
'c'	'h'	'e'	'r'	'r'	'i'	'e'	's'	'\0'

(this is in read-only storage, not the heap)

C/C++ details: none [default view]

IFSCee

QUESTÕES?

Código:

<https://github.com/gelasioebel/ifscee>

Proof of Concept:

<https://gelasioebel.github.io/ifscee/>

IFSCee

- FIM -

IFSCee

Uma Ferramenta Web Educacional para
Auxiliar no Ensino da Linguagem C

PROJETO INTEGRADOR I

Professores: Renato Simoes Moreira

Leonardo Perin Rauta

Cliente:

Leonardo Perin Rauta.

Alunos do ADS do IFSC Campus Gaspar

GELASIO EBEL JUNIOR

GASPAR

2024

Desenvolvimento de uma Ferramenta Web Educacional para o Ensino da Linguagem C:

Relato de Experiência

Introdução

A experiência como monitor do curso de Análise e Desenvolvimento de Sistemas (ADS) durante o período de 2024-2, bem como a percepção das dificuldades enfrentadas por meus colegas das classes de programação, particularmente na linguagem C, me convenceram a desafiar-me em querer criar algo similar com ao site: <https://pythontutor.com/c.html#mode=edit>, que apesar de muito útil, é em inglês e possui algumas limitações como, número reduzido de passo de execução, requer que o código seja compilável, entre outras, o que acaba sendo em contrapartida ao seu objetivo principal de ensinar quem está iniciando a programar.

Contextualização

Durante todo o meu percurso acadêmico, pude notar dificuldades minhas e de meus colegas na programação em C, mas principalmente durante o período de monitoria que prestei, que se estendeu de novembro de 2024 a fevereiro de 2025, foi que pude ter uma visão mais aguçada das necessidades dos estudantes. Durante este período, foram realizados atendimentos em duas modalidades: presencial, às quartas-feiras, das 16:00 às 18:30, e remoto, aos sábados, das 13:30 às 16:00. Esta flexibilidade no formato dos atendimentos permitiu alcançar diferentes perfis de estudantes, desde os que tem mais dificuldades de lógica aos que simplesmente estão engessados em suas soluções para a resolução de problemas, revelando assim diferentes padrões de dificuldades importantes no processo de aprendizagem da programação.

Justificativa

A justificativa do desenvolvimento do IFSCee surgiu da observação direta destes desafios enfrentados pelos estudantes, exemplos não limitados às dificuldades na compreensão de conceitos da programação, problemas na visualização do fluxo de execução dos programas, confusão no entendimento do gerenciamento de memória, problemas de lógica na implementação de estruturas de dados um pouco mais robustas e dificuldades na mentalização do que está ocorrendo na execução do código.

Objetivos

Objetivo Geral

A proposta é desenvolver uma ferramenta web educacional interativa que auxilie no processo de ensino-aprendizagem da linguagem C, facilitando a visualização e compreensão dos conceitos fundamentais de programação.

Objetivos Específicos

Para alcançar tal objetivo e após muitas tentativas frustradas estabeleci limitar o escopo e focar em alguns pontos chaves ainda não alcançados:

- Implementar um interpretador da linguagem C em JavaScript.
- Desenvolver uma interface intuitiva para visualização do código.
- Criar representações visuais do estado da memória durante a execução.
- Permitir a navegação passo a passo no código.
- Facilitar a identificação de erros comuns de programação.

ABORDAGEM INTERPRETAÇÃO:



Fundamentos

O ensino de programação é uma área de suma importância no contexto do curso de ADS do IFSC. A linguagem C, em particular, oferece uma vasta compreensão reutilizável para os estudantes entenderem os princípios básicos da programação. Sua sintaxe rigorosa e necessidade de gerenciamento de memória, C desafia os alunos a desenvolver um entendimento profundo de conceitos computacionais importantes e a pensar de maneira estruturada.

Ensino de Programação

A meu ver o maior desafio da programação é a abstração necessária para compreender conceitos complexos, bem como à natureza técnica da sintaxe da linguagem de programação C, e o quanto ela cria barreiras para os estudantes iniciantes ou sem experiência anterior em programar.

A Linguagem C

A linguagem C, devido à sua proximidade com o hardware e sua influência no desenvolvimento de inúmeras outras linguagens, mesmo com todas essas barreiras, acredito ser uma boa escolha para o ensino de programação. Seu uso permite aos estudantes compreenderem os fundamentos de como um computador funciona, desenvolver pensamento algorítmico, entender gerenciamento de memória e aprender estruturas de dados.

Ferramentas Educacionais

Acredito que o uso de ferramentas educacionais no ensino de programação pode reduzir a curva de aprendizado inicial, proporcionar feedback imediato, facilitar a visualização de conceitos abstratos, permitir experimentação sem riscos e aumentar o engajamento dos estudantes.

Metodologia

Caracterização da Pesquisa

A metodologia empregada no desenvolvimento do IFSCee foi:

Pesquisa sobre a Linguagem C, e de como compiladores e linguagens interpretadas funcionam.

Após tentativas sem sucesso de desenvolver um protótipo e ver que não ia dar conta do recado comecei a estudar o que torna uma linguagem interpretada, seus detalhes e como ela é ao final de tudo interpretável.

Depois de muito estudo e tomar conhecimento do básico, aí sim pude começar a saber qual rumo tomar.

Desenvolvimento da Ferramenta

O processo a ser tomado para o desenvolvimento da ferramenta ao meu ver deve seguir as seguintes etapas: análise de requisitos, prototipação, implementação do interpretador, desenvolvimento da interface, testes e ajustes, e documentação.

Desenvolvimento do Projeto

Arquitetura da Solução

Após inúmeras tentativas de desenvolver uma prova de conceito e fazer uma pesquisa mais robusta de como uma linguagem de programação é criada e especificada, a arquitetura do IFSCee foi estruturada em componentes distintos:

- Interface do usuário em HTML5 e CSS3.
- Campo para edição de código.
- Botões de navegação da execução.
- Tokenização das linhas (Lexer)
- Análise Sintática
- Construção da AST (Abstract Syntax Tree)
- Interpretação do AST em JavaScript.
- Renderização do Visualizador de memória.

IMPLEMENTAÇÃO

Interface do Usuário

A interface precisa ser projetada considerando usabilidade, clareza na apresentação e feedback visual.

Análise Léxica

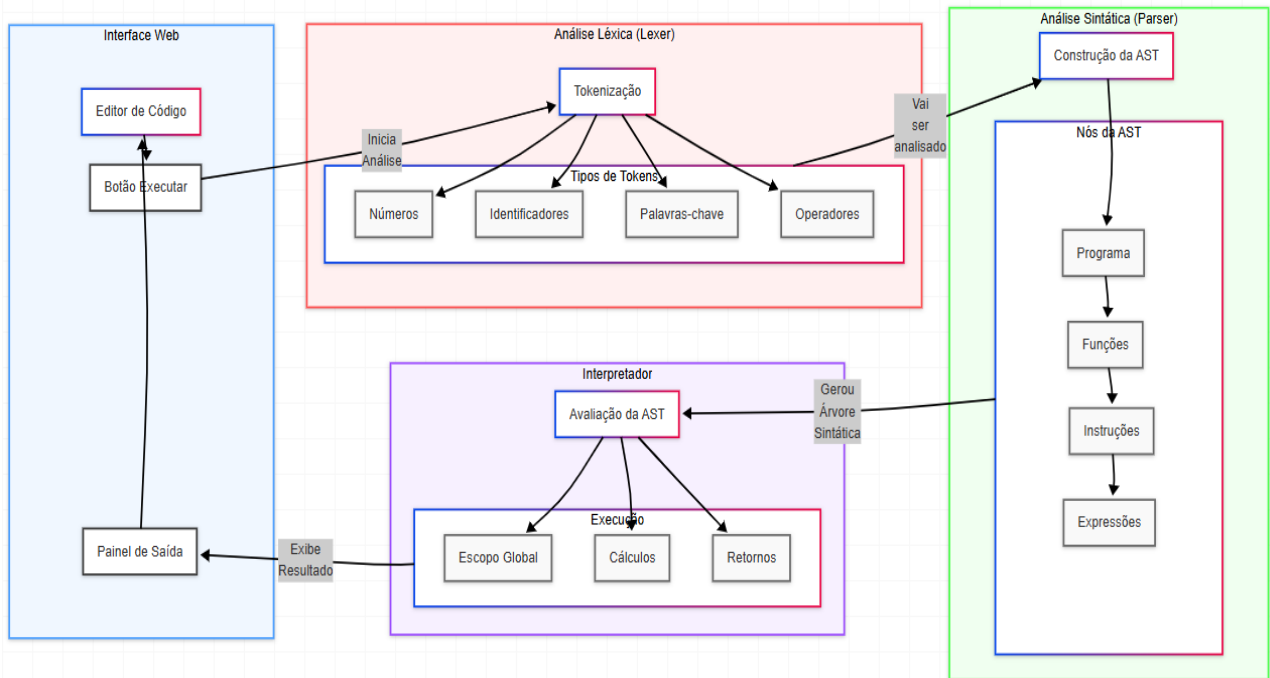
O analisador léxico precisa reconhecer palavras-chave da linguagem C, identificadores, números e literais, operadores e delimitadores.

Análise Sintática

O parser sintático precisa implementar a construção da AST, validação de estrutura, análise de escopo e verificação de tipos.

Interpretador

O interpretador precisa incluir funcionalidades para execução passo a passo, gerenciamento da visualização da memória, avaliação de expressões, controle de fluxo e chamadas de função.



Limitações Atuais

- Foram identificadas as seguintes limitações:
- suporte restrito a bibliotecas padrão,
- necessidade de aprimoramentos na visualização,
- ausência de muitos recursos
- carência de exemplos práticos adicionais

Trabalhos Futuros

- Visualização da Memória
 - Visualização o Heap, Stack
 - Mostrar alocação de memória (animação)
 - Rastrear ponteiros e seus endereços
 - Visualizar arrays, strings, types na memória
 - Destacar memória não inicializada
- Interface e Usabilidade
 - Mensagens de erro
 - Documentar exemplos e limitações
 - Integrar lista com algoritmos comuns
 - Testar com material didático
 - Estilizar o Layout.

Considerações Finais

O desenvolvimento do IFSCee não é tarefa fácil e apesar de visar a melhoria no ensino da linguagem C, pude ver que minha abordagem sem planejamento dificultou o processo, principalmente por não fazer uso de frameworks que poupariam tempo, mas visando ser um projeto que pretendo levar para uma pós graduação ou um mestrado, eu o vejo como uma oportunidade de aprendizado e crescimento. A integração de algoritmos comuns e a realização de testes com material didático devem ser o objetivo fundamentais para justificar a existência da ferramenta. Tornando assim o IFSCee uma ferramenta acessível e prática para estudantes. O projeto tem potencial como ferramenta educacional para o ensino da linguagem C. Mas este desenvolvedor ainda tem muito desafio a serem superados.