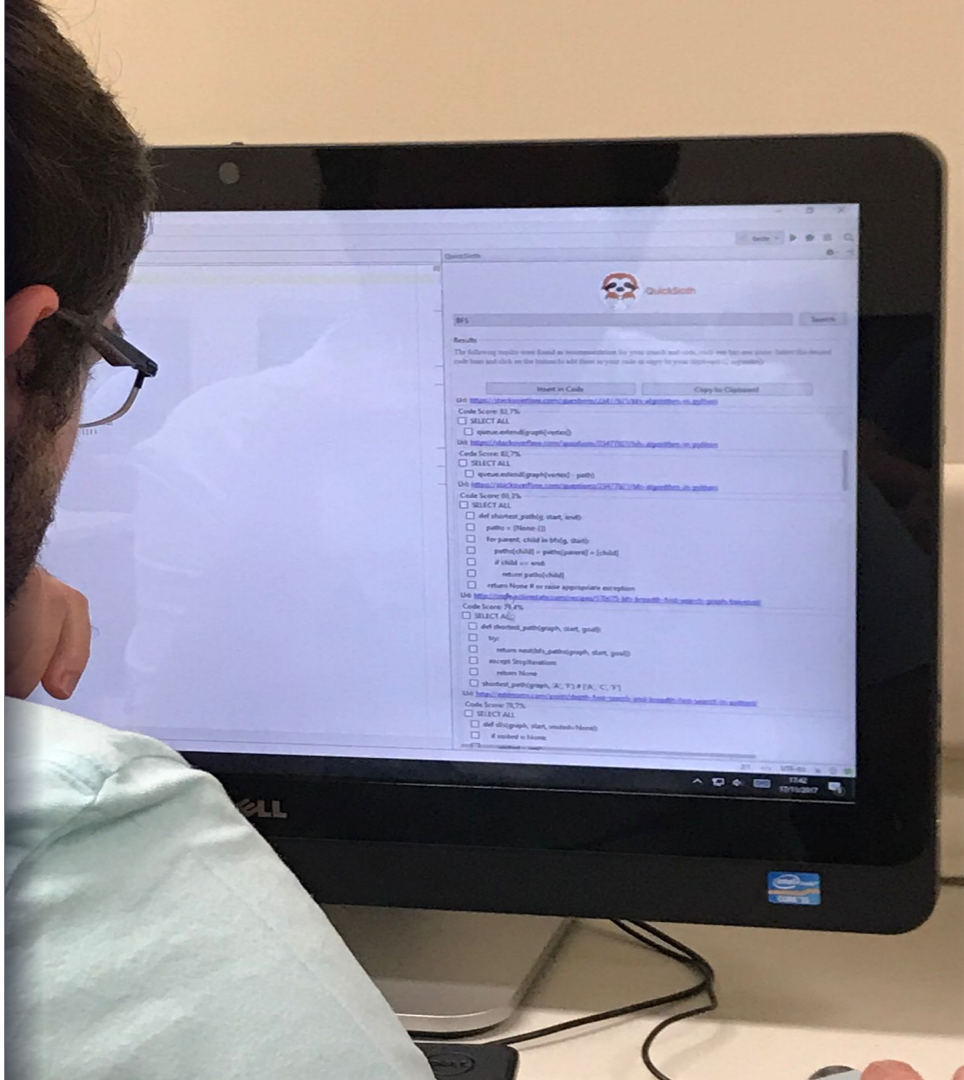
Centro Universitário FEI

QuickSloth

Sistema modular de Recomendação de código-fonte

Ahmad Khatib e Pamela Peixinho
Orientador: Prof. Dr. Guilherme Wachs



Sistemas de Recomendação

- Informações sobre o usuário
- Alta taxa de acerto
- Alta demanda em diversas áreas:
  - Filmes
  - Propaganda
  - Código-fonte (*SCoReS*)

Internet é usada por desenvolvedores em
Média 12 vezes por dia
para buscar códigos

Mais de 19.4M
repositórios de código no Github em 2016

Fontes: (SADOWSKI; STOLEE; ELBAUM, 2015) e https://octoverse.github.com/

Modelar e Implementar um sistema de recomendação de códigos-fonte que pode levar em consideração **métricas** provenientes da área de engenharia de software, assim como processamento de Linguagem Natural. Esse modelo deverá ser estruturado de tal forma que possibilite a **configuração e inclusão de novos módulos** capazes de avaliar códigos-fonte. Além disso, implementar uma interface para os usuários finais que possibilite a redução de tempo de desenvolvimento.

- Processamento de Linguagem Natural (NLP);
- Métricas de Engenharia de Software;
- Abstract Syntax Tree (AST);

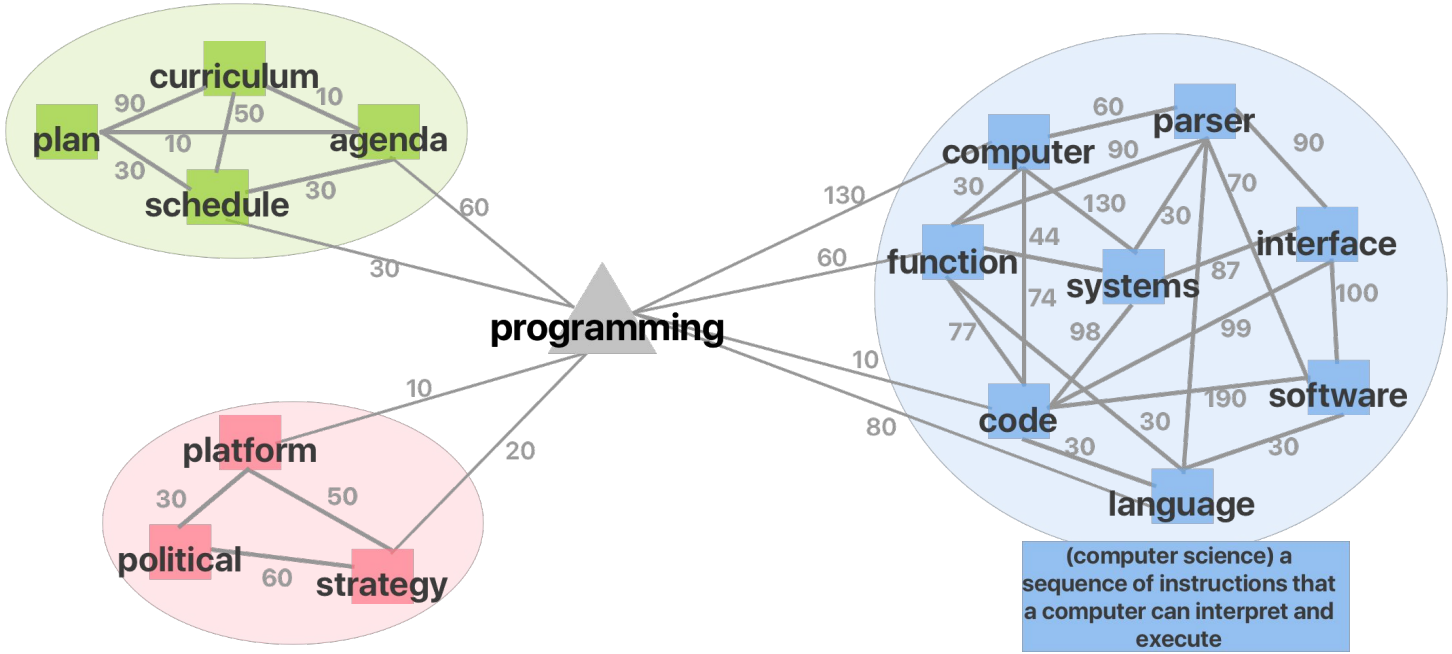
Processamento de Linguagem Natural

- Área multidisciplinar: estatística; linguística; ciência da computação; inteligência artificial;
- Usuários se comunicando com as máquinas através de linguagem natural;

Processamento de Linguagem Natural

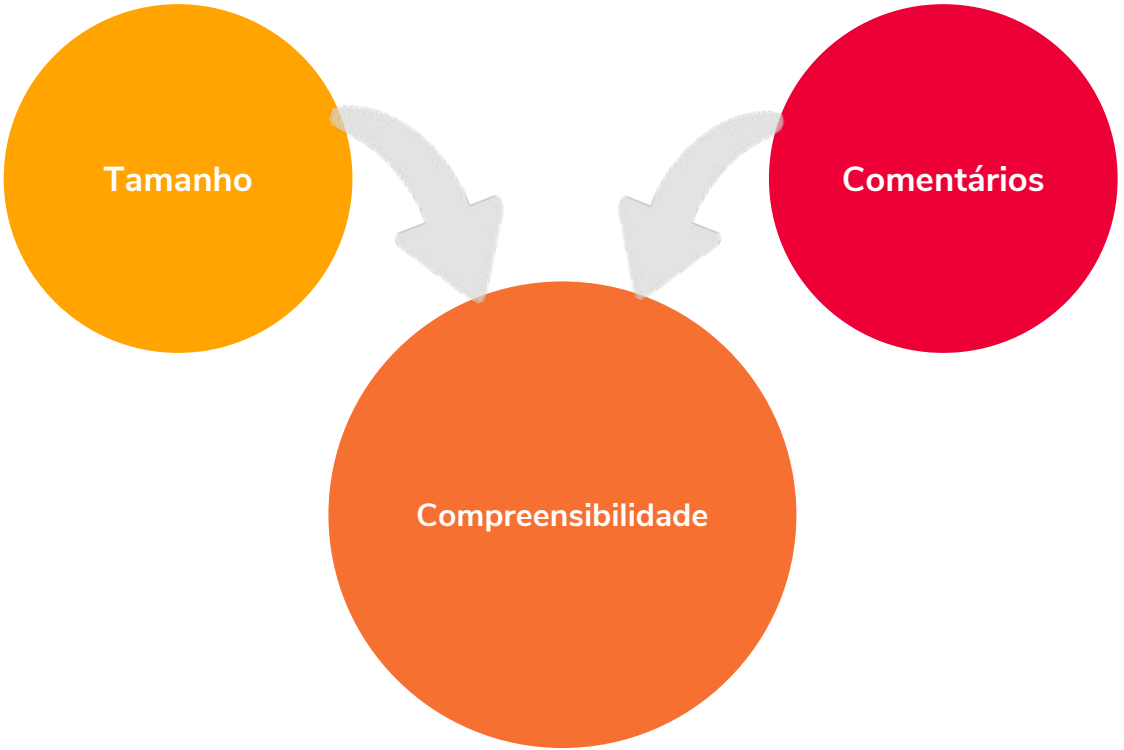
- Contexto;
- Modelagem contextual de palavra;
- *Gap Semântico;*

Processamento de Linguagem Natural



Métricas de Engenharia de Software

Compreensibilidade de Código-fonte



Compreensibilidade de Código-fonte

Muitas linhas de código e nenhuma linha de comentário

```
11  def f(n):
12      a = 0
13      b = 1
14      for i in range(0, n):
15          print(a)
16          temp = a
17          a = b
18          b = temp + b
19      return a
```

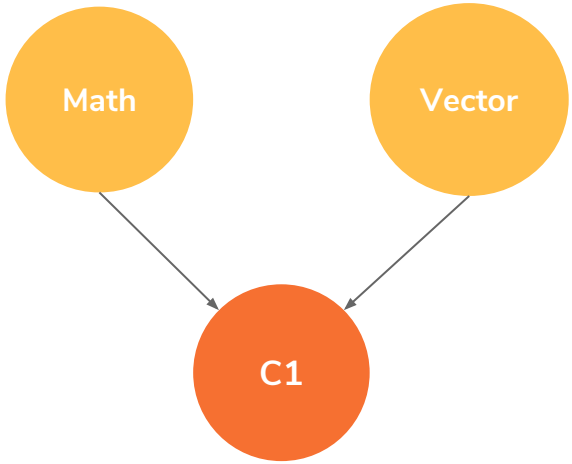
Poucas linhas de código e muitas linhas de comentários

```
2   def fibonacci(n):
3       #1st number in fibonacci sequence
4       if n == 0: return 0
5       #2nd number in fibonacci sequence
6       elif n == 1: return 1
7       #recursive fibonacci call to calculate the n-th value based on the 2 previous values
8       else: return fibonacci(n-1)+fibonacci(n-2)
9
```

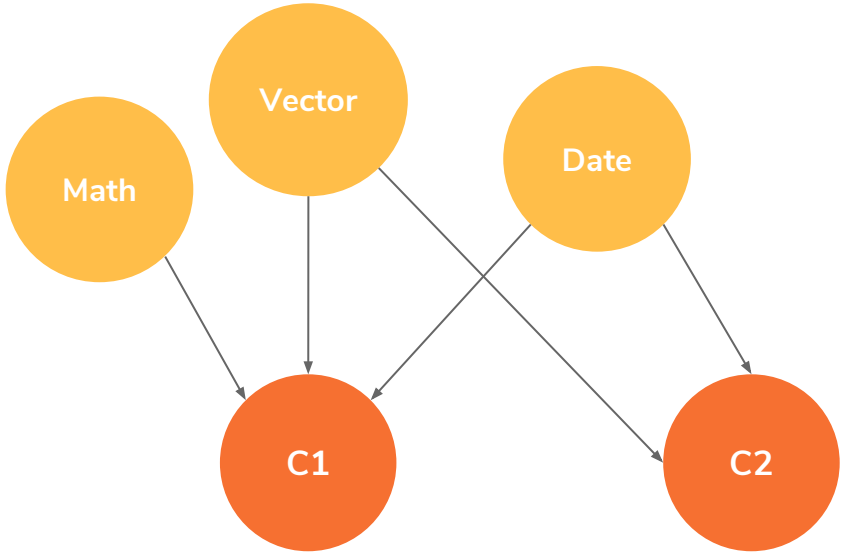
Acoplamento

C1.java

```
1  import java.util.Vector;
2  import java.lang.Math;
3
```

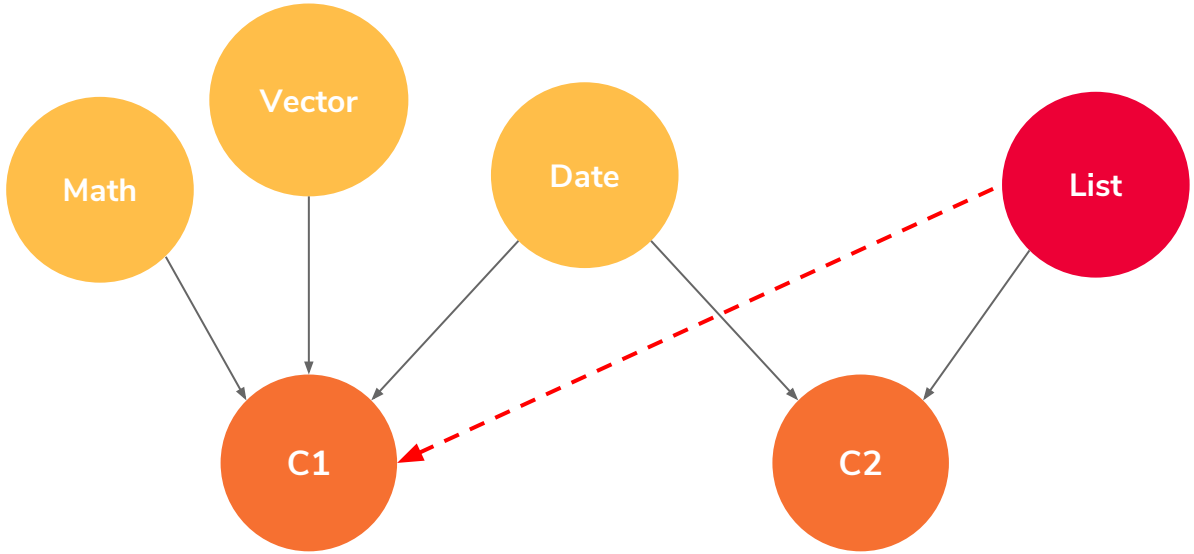


Acoplamento



Fácil integração: C2 utiliza mesma dependência que C1

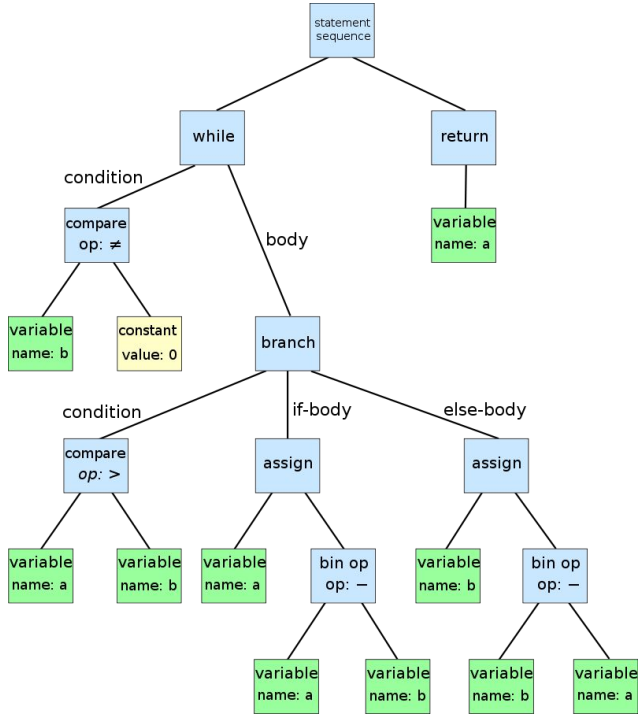
Acoplamento



Difícil integração (custoso): C2 possui uma dependência diferente de C1

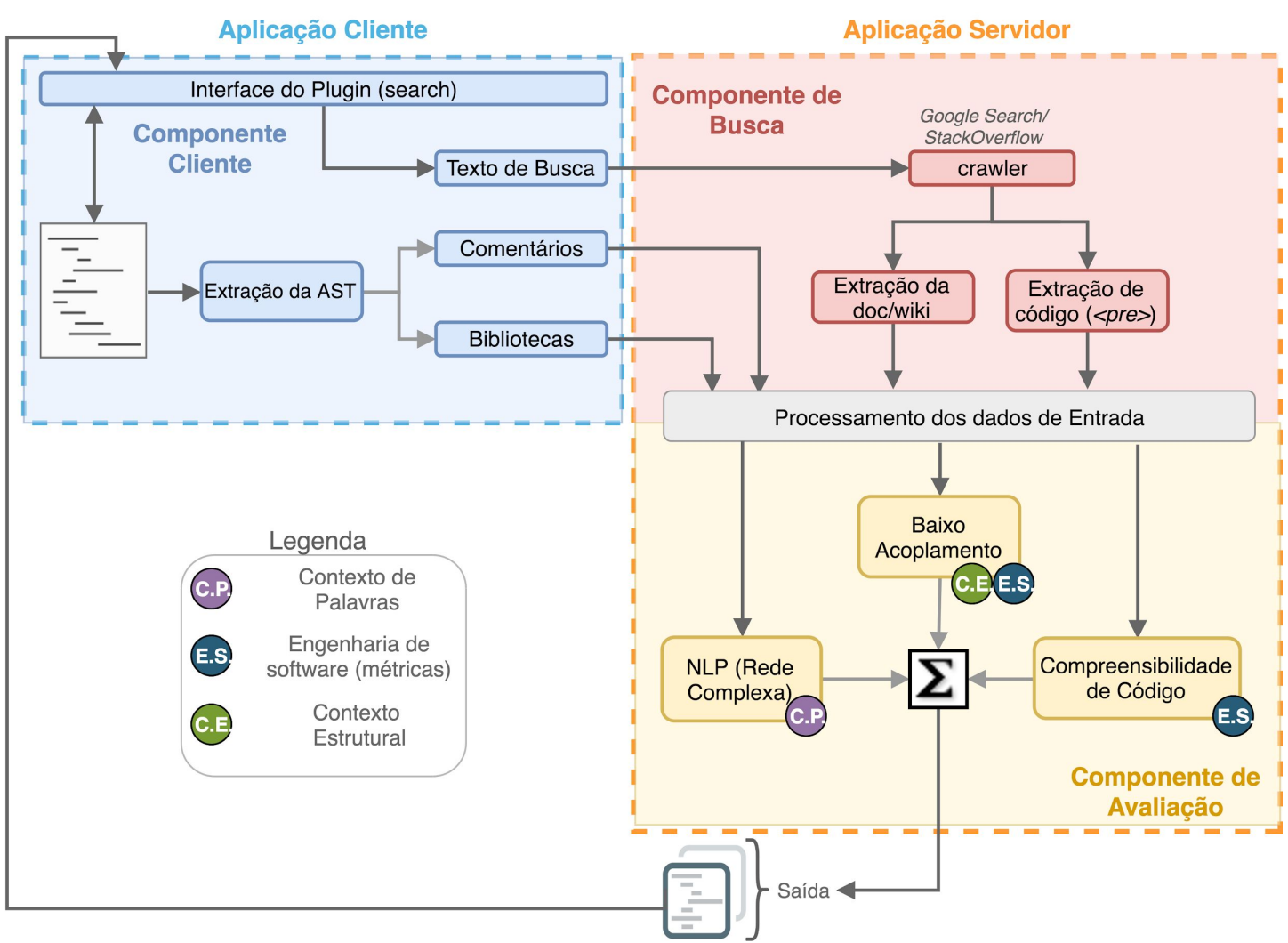
Abstract Syntax Tree

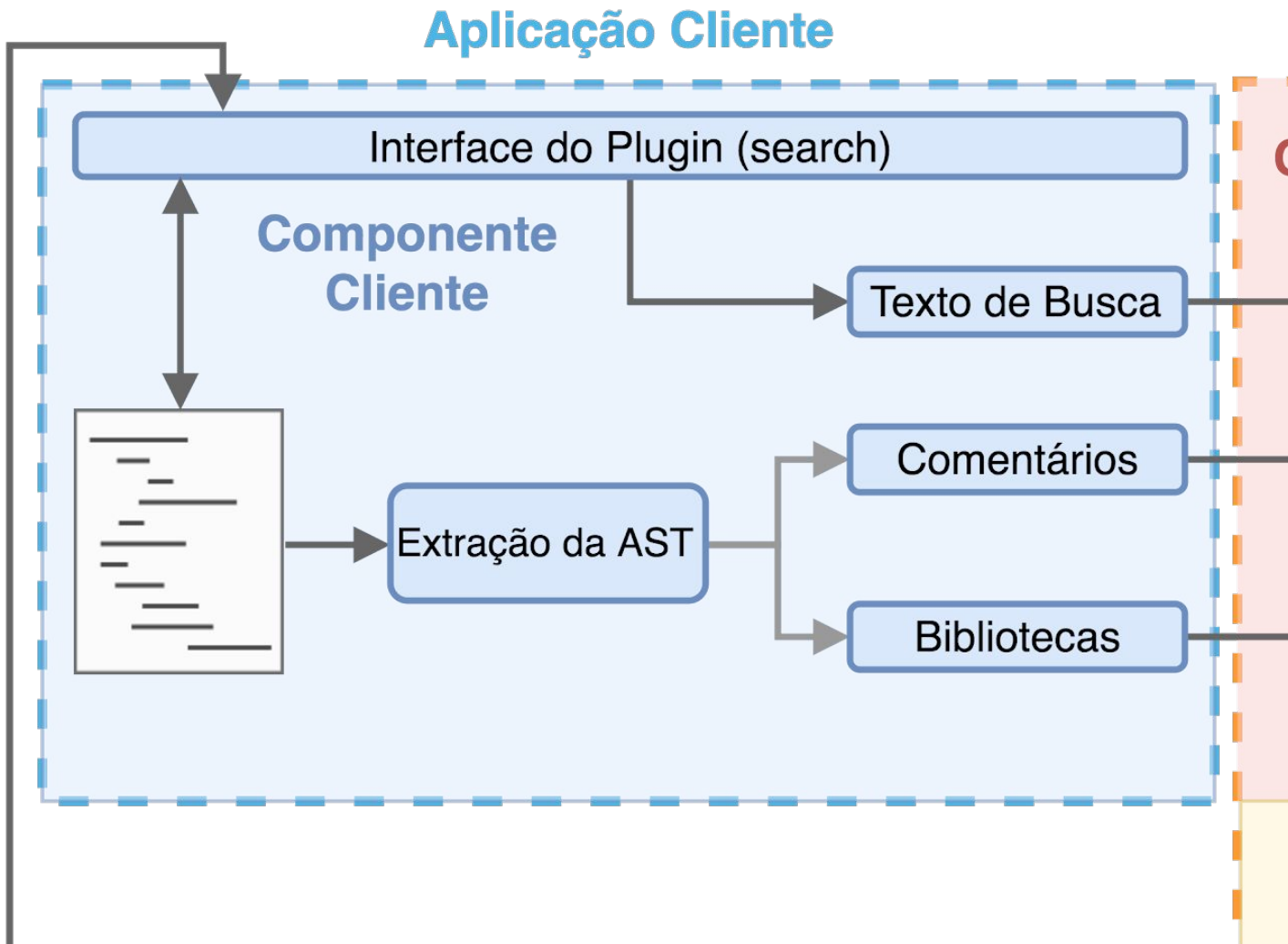
- Representação em árvore da estrutura sintática de um arquivo de código-fonte

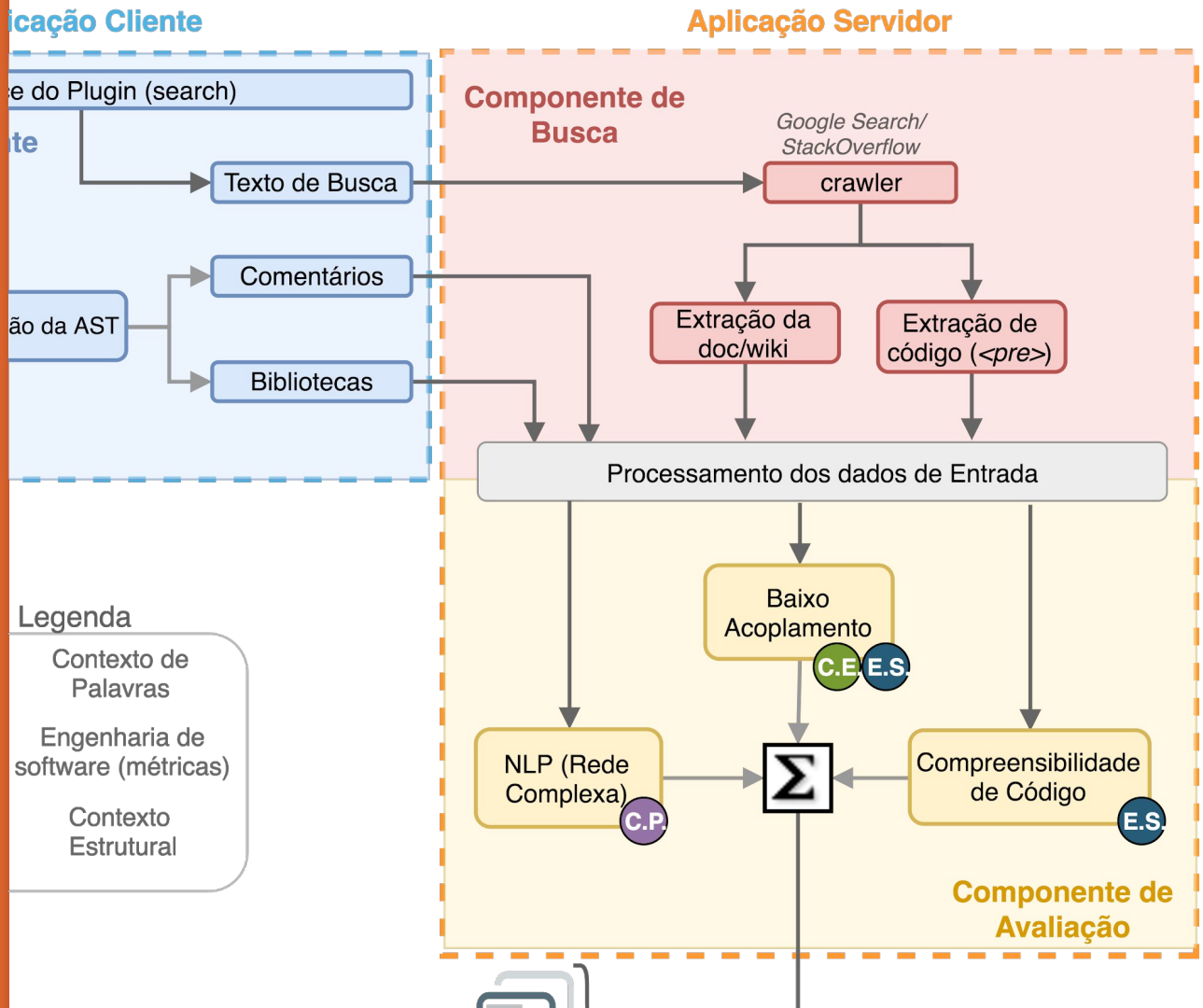


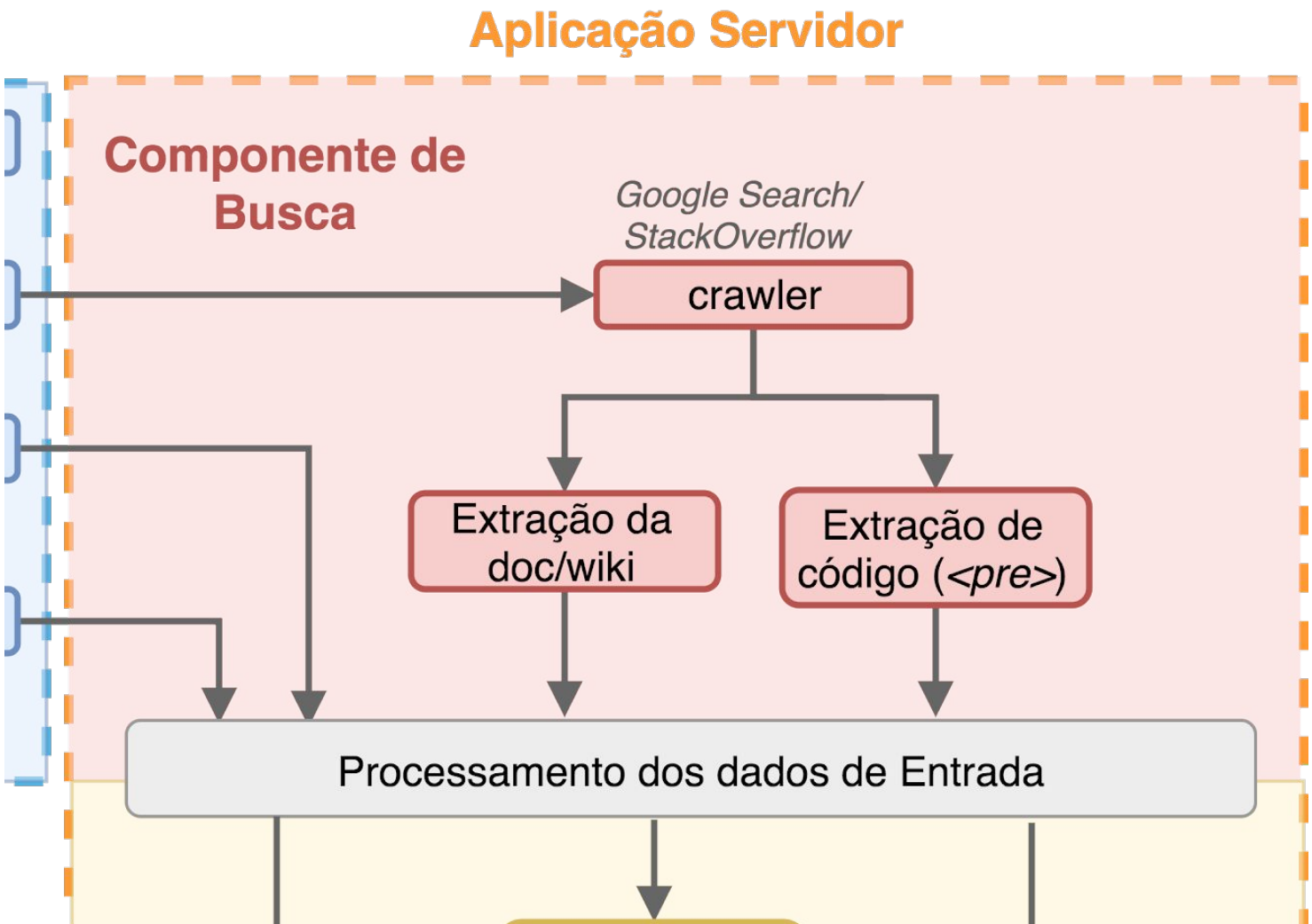

```
while b ≠ 0
   if a > b
      a := a - b
   else
      b := b - a
return a
```

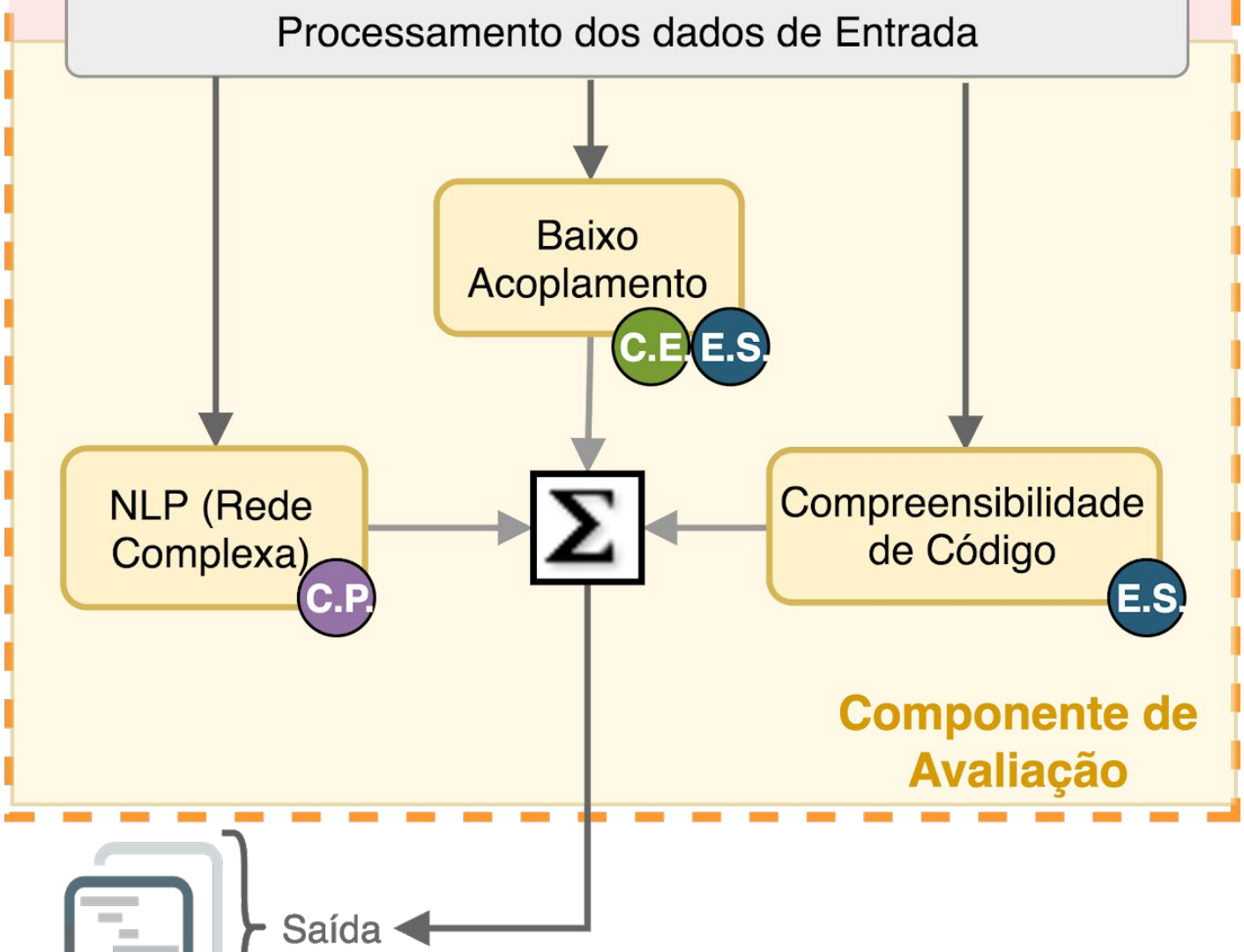
Fonte: http://mnielsen.github.io/notes/parentheses/index.html



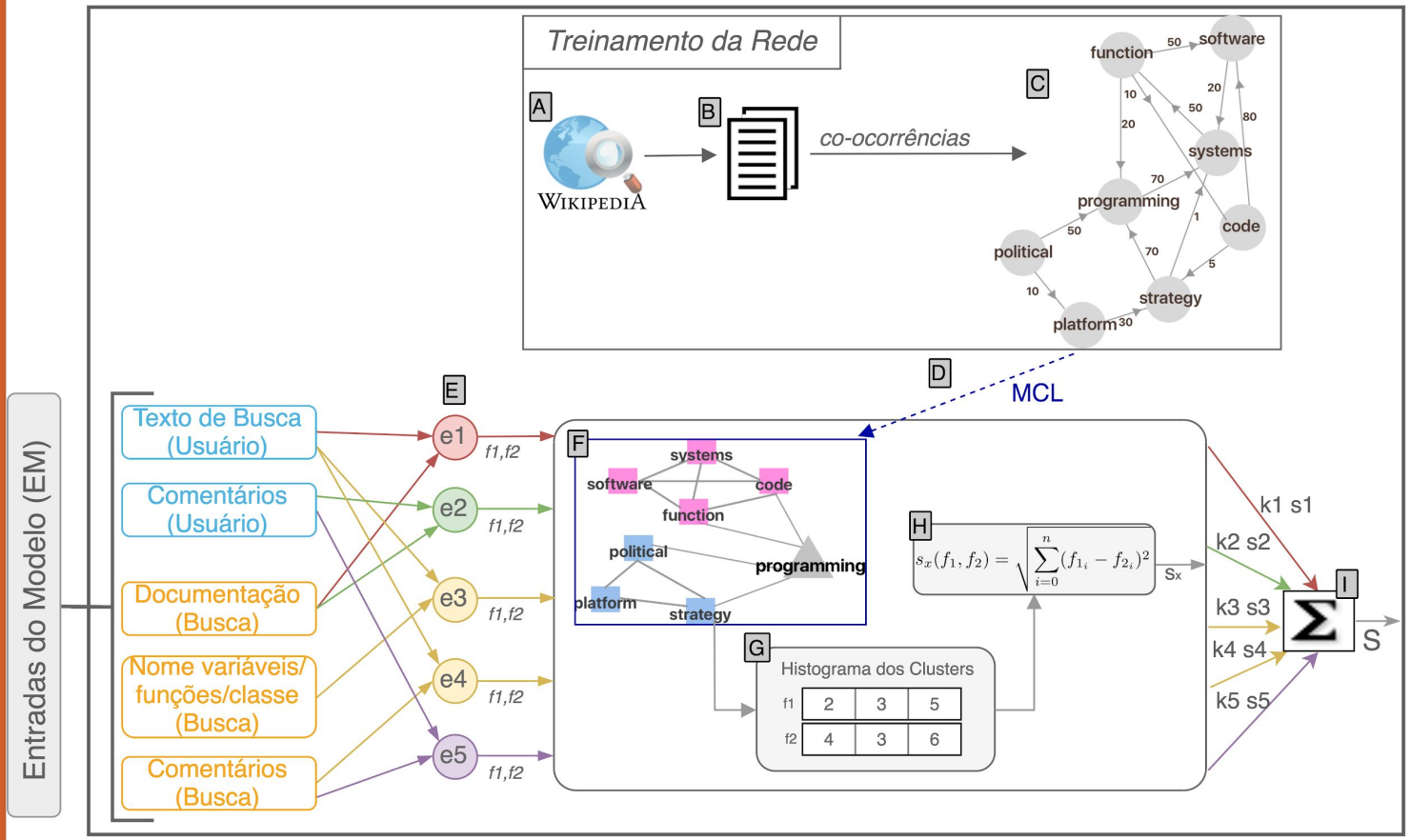



icação Cliente
Aplicação Servidor
e do Plugin (search)
te
Texto de Busca
Comentários
ão da AST
Bibliotecas
Componente de Busca
Google Search/ StackOverflow
crawler
Extração da doc/wiki
Extração de código (<pre>)
Processamento dos dados de Entrada
Baixo Acoplamento
C.E.
E.S.
NLP (Rede Complexa)
C.P.
Σ
Compreensibilidade de Código
E.S.
Componente de Avaliação
Legenda
Contexto de Palavras
Engenharia de software (métricas)
Contexto Estrutural

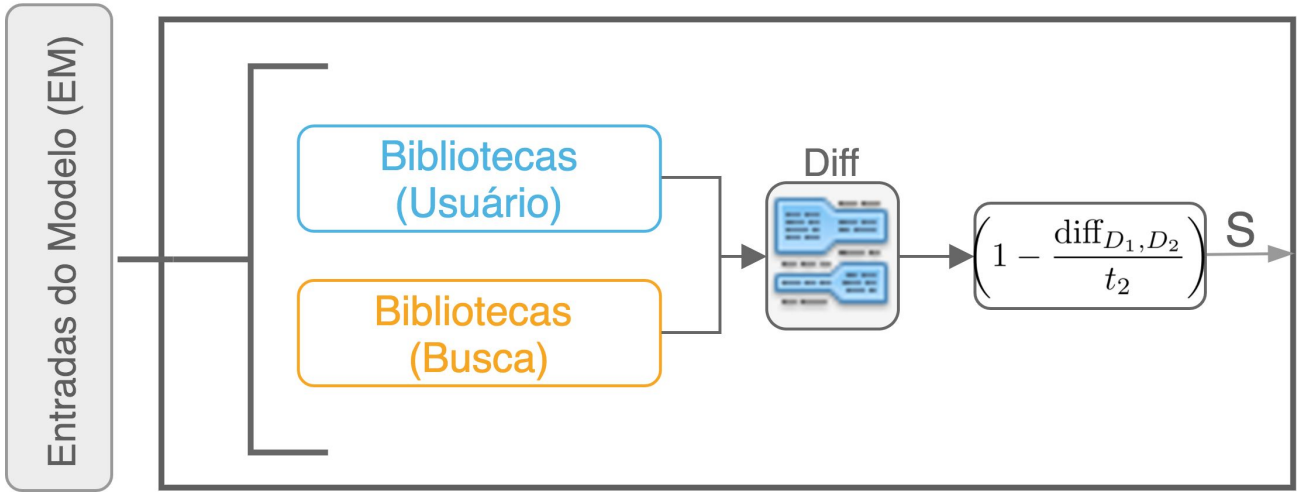

Aplicação Servidor
Componente de Busca
Google Search/ StackOverflow
crawler
Extração da doc/wiki
Extração de código (<pre>)
Processamento dos dados de Entrada



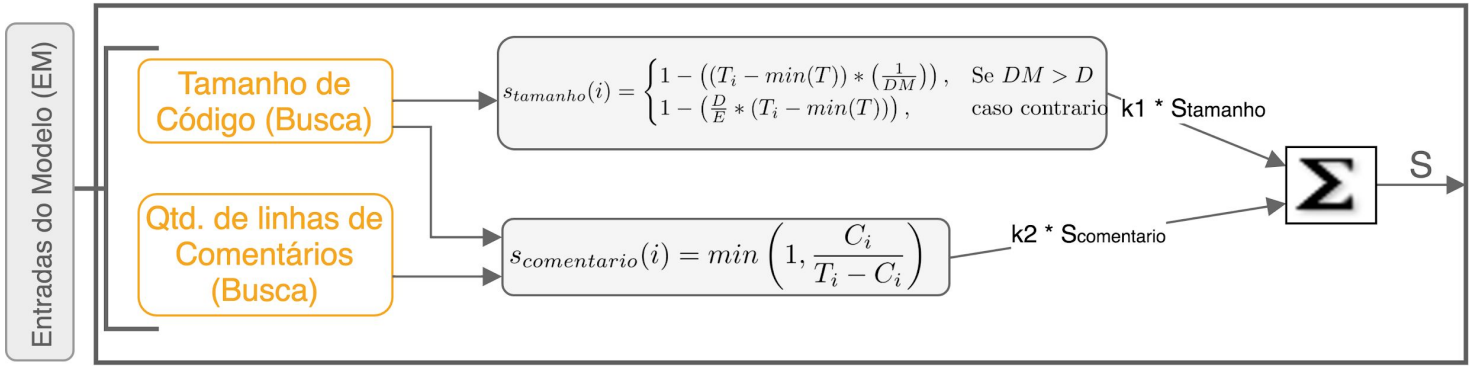
Módulo NLP

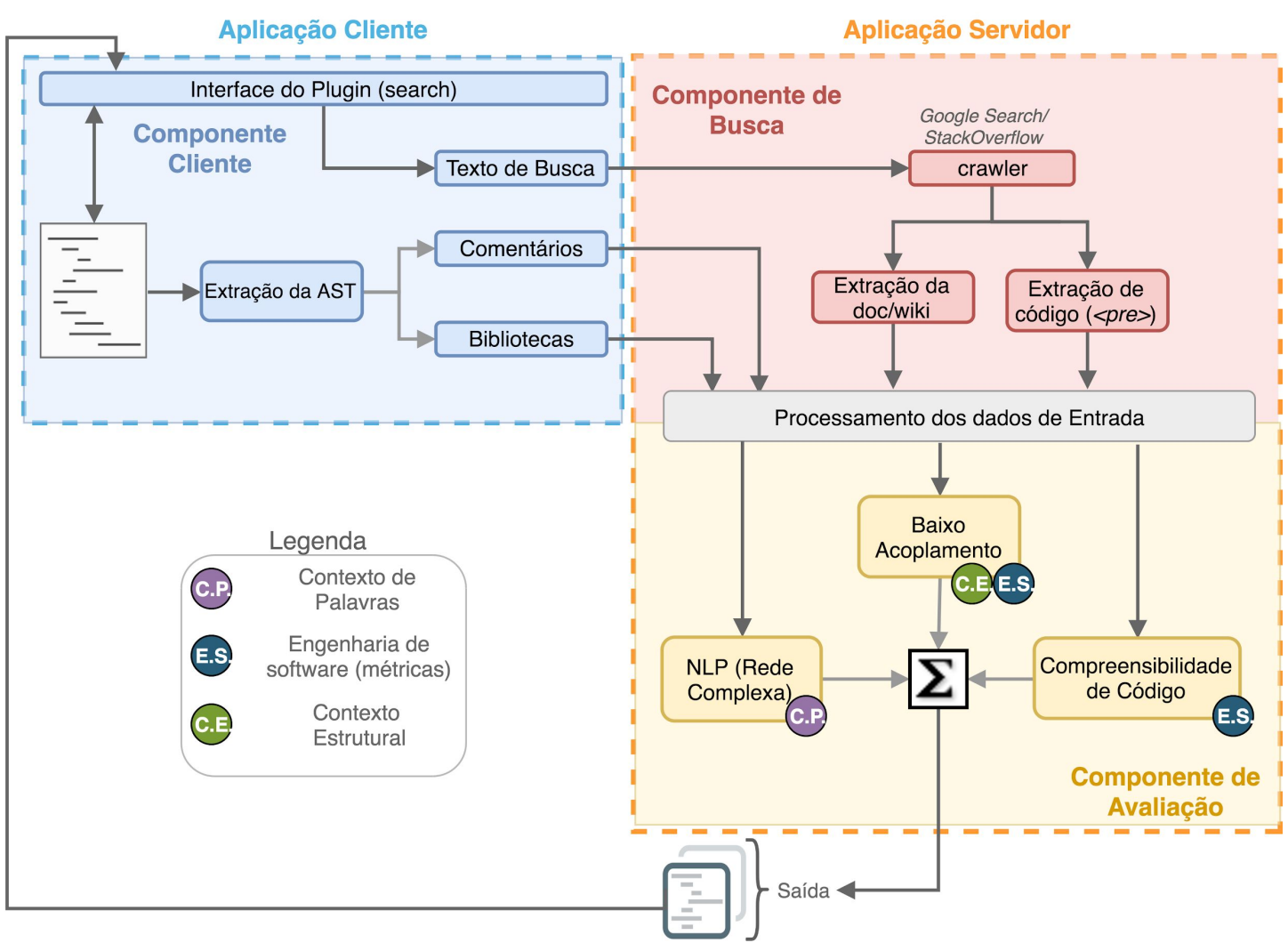


Módulo Baixo Acoplamento



Módulo Compreensibilidade de Código

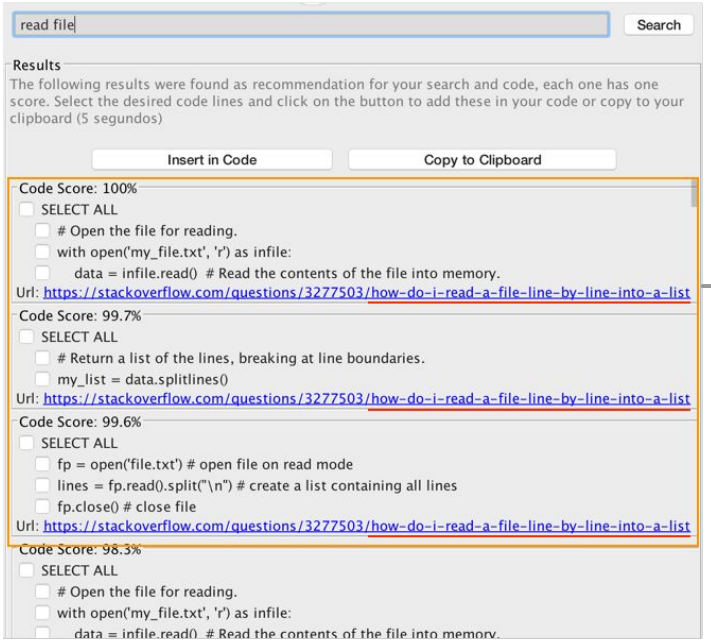




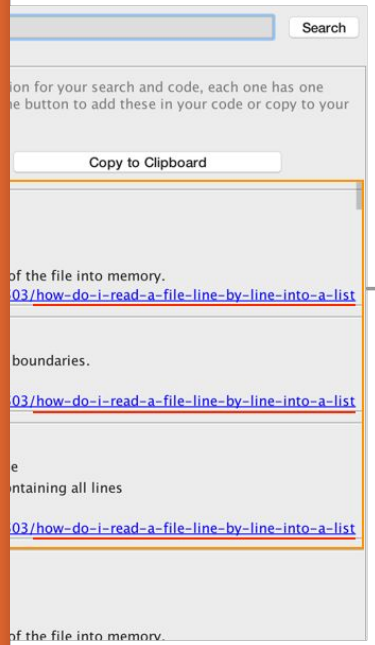
- Resultados das Recomendações
  - Comparação com Google
- Teste com Usuários

Resultados das Recomendações

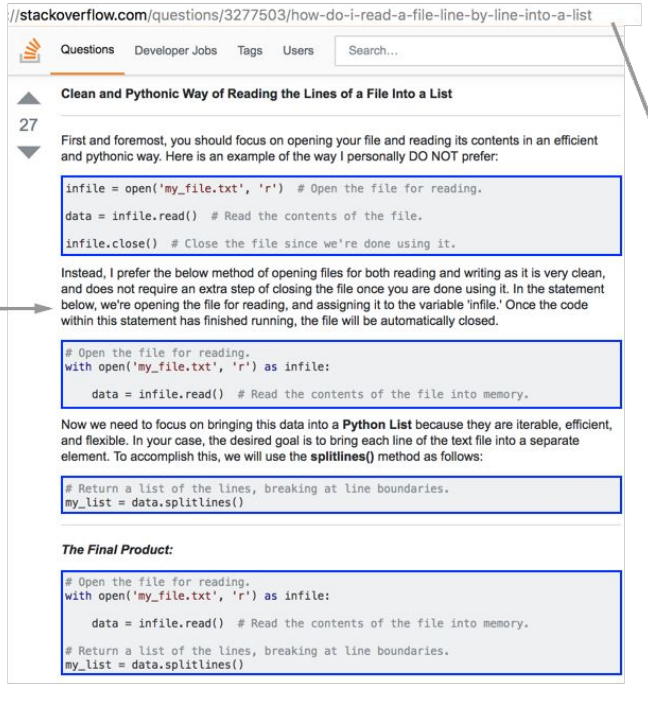
- Comparação com Google ("*read file*")



Resultados das Recomendações



9º resposta no StackOverflow



8º link do google

Resultados das Recomendações

8º link do google

Teste com Usuários

- Objetivo:
  - Validar efetividade
  - Validar aceitação
- Participantes:
  - Alunos do Curso de Computação
- 3 tarefas
- Questionário pós-teste

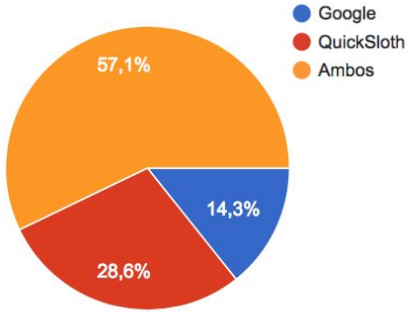
Teste com Usuários

- Resultados Efetivos:

	QuickSloth	Google
Tarefa 1	80%	75%
Tarefa 2	100%	50%

Teste com Usuários

- Resultados de Aceitação
  - Escolha da Ferramenta na Tarefa 3



QuickSloth	Google
85,7%	71,4%

Teste com Usuários

- Conclusão dos Resultados:
  - Recomendações atenderam as necessidades dos usuários;
  - A maioria dos usuários consideraram o sistema fácil de usar;
  - Usuários tiveram a impressão de que a aplicação economizou seu tempo de desenvolvimento;
  - Desenvolvedores inexperientes tiveram dificuldades em utilizar o QuickSloth.

Conclusão

- **Resultados Esperados:**

  Evitar que o usuário saia do ambiente de desenvolvimento para encontrar o códigos-fonte;

  Facilidade de incorporar códigos sugeridos pelo sistema;

  Redução de tempo na busca por códigos-fonte;

  Diminuição do gap semântico através do uso das Redes Complexas;

  Sistema fácil de configurar e incluir conceitos (módulos).

Conclusão

- Evitar que o usuário saia do ambiente de desenvolvimento para encontrar o códigos-fonte;
- Facilidade de incorporar códigos sugeridos pelo sistema;
- *Redução de tempo na busca por códigos-fonte;*
- Diminuição do gap semântico através do uso das Redes Complexas;
- Sistema fácil de configurar e incluir conceitos (módulos);
- * Publicação do QuickSloth no Repositório Oficial da JetBrains.

Aplicação

QuickSloth

Obrigado