

# ESC: SISTEMA DE EXTRAÇÃO DE CARACTERÍSTICAS DE TAREFAS DE CÓDIGO-FONTE

Caio Igor Costa da Silva, Gabriel Sousa da Costa, Igor Simon Santos, José Carlos Moura, Davi Bernardo Silva  
Instituto Federal de Santa Catarina  
Câmpus Jaraguá do Sul – Rau – Curso Técnico em Desenvolvimento de Sistemas  
e-mail: {caio.ic12, gabriel.sc 19, igor.ss, jose.c2001}@aluno.ifsc.edu.br, {davi.bernardo}@ifsc.edu.br  
Projeto Integrador I - Junho/2023.

**Resumo** - O ensino de programação é desafiador em sua essência. Além disso, cada vez é maior o número de alunos matriculados em cursos da área de computação. O elevado número de tarefas de código-fonte desenvolvidas, torna complexo o acompanhamento do professor. Nesse sentido, o objetivo deste projeto é desenvolver um sistema para extrair características de tarefas de código-fonte, escritas em linguagem C. O método que será utilizado para conduzir este projeto se baseia no modelo de desenvolvimento de software em cascata. Nas primeiras etapas, foram realizados o levantamento de requisitos e o projeto do sistema. Em seguida, serão realizados o desenvolvimento do sistema e os testes de software. Ao fim do projeto, o resultado esperado é um produto de software que consiga apoiar o professor no acompanhamento das tarefas de código-fonte no ensino de programação.

**Palavras-Chave** – Acompanhamento de alunos, Ensino de programação, Extração de características.

## ESC: FEATURE EXTRACTION SYSTEM FROM SOURCE CODE TASKS

**Abstract** – Teaching programming is challenging at its core. In addition, the number of students enrolled in courses in the area of computing is increasing. The high number of developed source code tasks makes teacher monitoring complex. In this sense, the objective of this project is to develop a system to extract features from source code tasks, written in C language. The method that will be used to conduct this project is based on the waterfall software development model. In the first stages, requirements gathering and system design were carried out. Next, system development and software testing will be carried out. At the end of the project, the expected result is a software product that can support the teacher in monitoring source code tasks in teaching programming.

**Keywords** – Student monitoring, Programming teaching, Feature extraction.

## 1. INTRODUÇÃO

Nos últimos anos, devido ao grande avanço da tecnologia e a sua alta presença no cotidiano, principalmente entre os jovens, fez com que houvesse uma grande ascensão do número de alunos ingressando em cursos da área de programação e animados com o mercado de trabalho, que também há uma grande demanda de profissionais dessa área. Estudos demonstram que esses alunos ficam bastante empolgados com o ingresso no novo curso, mas que essa empolgação é erradicada logo nas primeiras disciplinas de programação, por causa da dificuldade apresentada pelas disciplinas [4].

Embora existam diversos meios de aprendizagem dessas disciplinas e muitos conseguem aprender apenas com suporte de pesquisa em livros e na própria *internet*, ainda existem alunos que possuem dificuldades em tópicos iniciais, que são de extrema importância para o andamento do aluno no curso.

Foi pensando nisso, que teve-se a ideia de criar um sistema que auxiliará o método de ensino do professor, com extração de características de código-fonte em tarefas aplicadas pelo professor, tanto em sala de aula como tarefas extra classe. Dessa maneira, será possível investigar em quais áreas da disciplina o aluno está com maior dificuldade.

### A. Justificativa

Com a grande margem de alunos nos cursos de computação cada vez maior, torna-se cada vez mais difícil gerenciar, de forma individual, as tarefas propostas, relacionadas à programação. [6]

Mediante o problema de gerenciamento, o sistema proposto tem o enfoque em identificar, seguindo os problemas propostos pelo professor, no algoritmo, melhorando a eficiência de ensino e meios de correção do professor.

Com isso, a motivação de investigar mais profundamente sobre o tema, é que os professores podem ter mais dinâmica e acompanhar o aluno na sua aprendizagem de programação, tendo esse programa de auxílio.

### B. Objetivos

#### A) Objetivo geral

O principal objetivo deste projeto é desenvolver um sistema para extrair características de tarefas de código-fonte,

escritas em linguagem C. A relação de características extraídas poderá auxiliar o professor na compreensão dos pontos fortes e fracos de cada estudante, sem que o professor precise abrir qualquer código-fonte.

### B) Objetivos específicos

Para atender ao objetivo principal, foram definidos os objetivos específicos descritos a seguir.

- Realizar a análise e projeto do sistema, levando em consideração o planejamento e os diagramas para o desenvolvimento do sistema.
- Implementar um sistema de extração de características de tarefas de código-fonte.
- Fornecer como saída do sistema um resultado que possa auxiliar o professor a identificar possíveis dificuldades que os alunos estejam enfrentando na aprendizagem de programação.

## 2. FUNDAMENTAÇÃO TEÓRICA

### A. Desafios no ensino de programação

A teoria da Aprendizagem Significativa, proposta por David Ausubel, argumenta que a aprendizagem deve ser relevante e significativa para o aluno. Isso significa que os alunos devem ser capazes de relacionar novos conceitos a seus conhecimentos prévios e entender como esses conceitos podem ser aplicados na prática. Para a programação, isso pode significar a necessidade de exemplos e problemas práticos que demonstrem como os conceitos teóricos se aplicam na vida real. [5]

Por fim, a importância da adaptação de métodos de ensino às necessidades individuais dos alunos. Cada aluno é único e tem uma forma diferente de aprender, e os professores precisam estar cientes dessas diferenças e adaptar seus métodos de ensino para atender às necessidades individuais dos alunos.

### B) Linguagens para o ensino de programação

Dentre as diversas maneiras de ensinar programação, temos a linguagem natural, que é, descrever em uma língua comum um algoritmo ou um processo. Utilizada para detalhar os passos que devem ser executados, que funciona muito bem para ensinar a programação quando associado ao pseudocódigo. Isso acontece porque o pseudocódigo é uma mistura de estruturas e elementos da linguagem comum que fazem parte da programação real, tornando mais claro e legível o algoritmo, facilitando a compreensão do assunto.

### C) Características de código-fonte

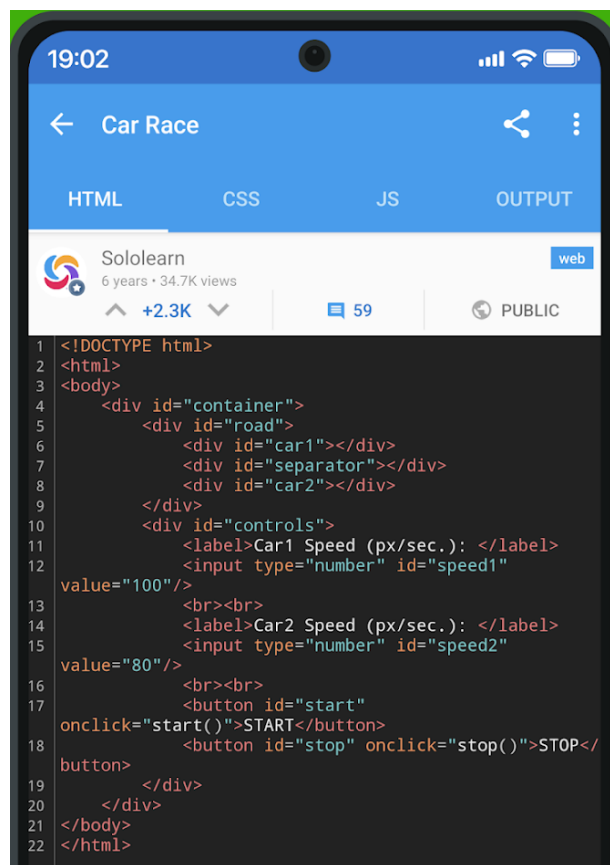
Código-fonte é um conjunto de instruções e declarações escritas em alguma linguagem de programação. As características de um código-fonte, consiste na quantidade de laços de repetição, número de variáveis, tipos das variáveis, as bibliotecas que foram utilizadas para a escrita

do algoritmo e a quantidade de funções secundárias além da função *main*.

## 3. TRABALHOS RELACIONADOS

Todos os exemplos adicionados neste tópico apresentam ideias sobre os programas e suas funções, visto que, os exemplos são voltados para uso pessoal visando apenas o estudante e não o professor. Deste modo, este artigo tem enfoque em utilizar como exemplo esses programas para criar a versão a ser utilizada unicamente pelo professor.

### A. Sololearn



**Figura 1.** Tela de codificação do aplicativo Sololearn

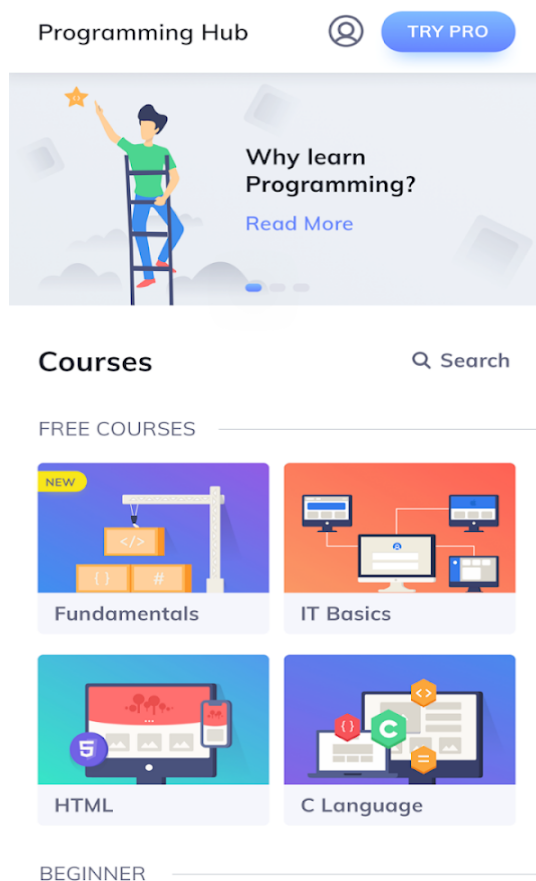
Sololearn é um programa com versões gratuitas e pagas, disponível mobile e web, que introduz a aprendizagem em diversas linguagens de programação, como exemplificado na imagem 3.1, por meio de uma comunidade criada dentro da própria plataforma com atividades propostas por professores.

**Pontos positivos.** É um sistema bastante funcional, entre as linguagens ensinadas estão C, HTML, JavaScript e Python com um blog disponível para os usuários deixarem algumas observações além de um certificado que é gerado após você concluir o curso.

**Pontos negativos.** Mesmo tendo versão gratuita, depois de um período, é cobrado uma tarifa para poder ter acesso total da plataforma, e também não será possível fazer um programa/projeto executável no aplicativo.

**Diferenças.** O projeto tem como parte do planejamento um acesso de forma igualitária, sem diferenciação de versão gratuita ou versão paga.

### B. Programming Hub



**Figura 2 .** Tela inicial do programa Programing Hub

Programming Hub é um aplicativo criado com colaboração de especialistas do Google que oferece um meio, considerado por muitos, como um meio perfeito de aprender, pois terá a oportunidade de aprender a codificar com diversos especialistas.

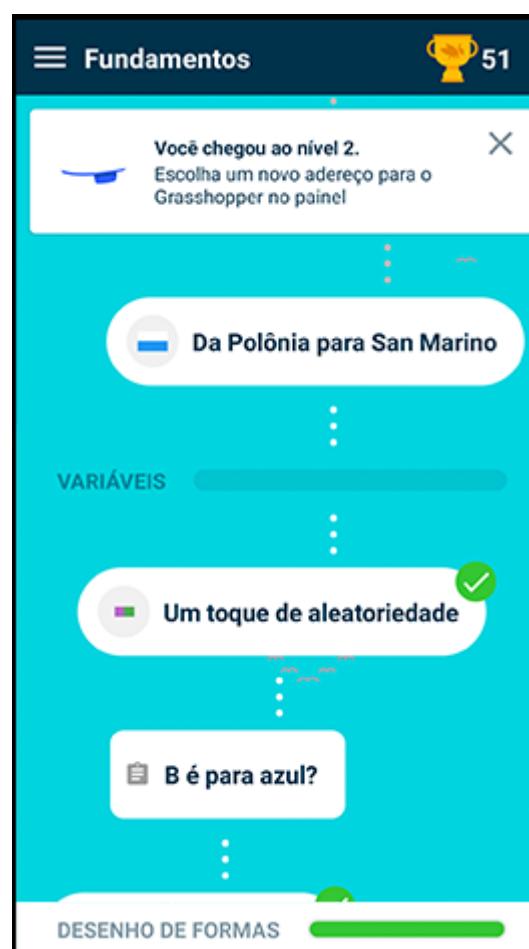
**Pontos positivos.** Colaboração de especialistas para ajudar o usuário a entender determinado assunto e apresenta uma

grande diversidade de áreas de aprendizado.

**Pontos negativos.** Sistema unicamente em inglês; web com desenvolvimento melhor quando comparado com aplicativos baixados e versão gratuita somente para iniciar a programação.

**Diferenças.** Este aplicativo de aprender código criado usando pesquisa e em colaboração com especialistas do Google oferece um meio perfeito para aprender programação. Aprenderá a codificar com especialistas.

### C. Grasshopper



**Figura 3.** Tela de atividades Grasshopper.



Grasshopper é um aplicativo gratuito disponível para Android e IOS, que possui um método mais leve e intuitivo com ensinamentos através de quiz e jogos

**Pontos positivos.** Com esse método mais divertido por meio de jogos e quiz, acaba-se adotando propositalmente um público mais infantil e introduzindo desde novo uma semente na criança para estudar a tecnologia que tanto faz parte do dia a dia delas, além do desenvolvimento de raciocínio lógico para diversas outras áreas. Com a possibilidade de mudar o idioma oferecido no sistema, torna-se um ponto positivo devido ao público infantil, em sua maioria, falar apenas o idioma nativo.

**Pontos negativos:** O aplicativo todo em inglês, necessitando de iniciação básica na língua; sistema de ensino voltado unicamente para Java; feedback negativo dos usuários perante a linha evolutiva da programação; aplicativo voltado somente para iniciantes.

**Diferenças.** Sistema de aprendizagem de programação, não havendo distinção ou foto entre alunos e professores.

#### 4. SOLUÇÃO PROPOSTA

##### A. Requisitos Funcionais

[RF 01] O sistema deve permitir ao professor-usuário realizar o login na tela inicial.

[RF 02] O sistema deve ter a opção de permitir, em todas as telas, o usuário ter a opção de voltar para a tela inicial.

[RF 03] O sistema deve permitir, na opção carregar tarefas, deve permitir ao professor consultar todas as tarefas já cadastradas no sistema.

[RF 04] O sistema apresentará as opções de nomes e configurações. Será solicitado ao professor para inserir as linhas de programação e os arquivos a serem analisados, apresentando os códigos previamente escritos.

[RF 05] O sistema deve permitir que o professor consiga selecionar o tipo de correção que o sistema terá que executar.

[RF 06] O sistema deve permitir ao professor um botão de inserir o arquivo a ser corrigido.

##### B. Requisitos Não-Funcionais

[RNF 01] De prioridade importante e se encaixando na categoria de confiabilidade, o sistema deve ser capaz de tratar exceções e se recuperar de falhas, sem que haja perda de dados. Backup e restauração do banco de dados também se encaixam neste item.

[RNF 02] De prioridade essencial e se encaixando na

categoria de privacidade, as informações pessoais do usuário não devem ser vistas pelos operadores do sistema.

[RNF 03] De prioridade desejável e da categoria organizacional, todos os documentos gerados pelo sistema devem seguir o padrão de relatórios infográficos.

[RNF 04] De prioridade de processamento e da categoria desejável, o sistema deve executar as tarefas dentro de 10 segundos ao máximo.

[RNF 05] Um professor pode adicionar mais de uma disciplina em uma turma se ele tiver cumprido seus pré-requisitos.

[RNF 06] O usuário tem um hardware com no mínimo processador 2GHz, 4GB RAM, Placa Gráfica DirectX 9, Windows 10 versão 19045.3086.

[RNF 07] O leitor do código-fonte deverá ler os padrões EAN-8 e UPC-A em um tempo de até 3s.

[RNF 08] O sistema deve ser compatível com os navegadores Chrome Versão 114.0.5735.134 e FireFox Versão 114.0.1.

##### C. Requisitos de Domínio

[RD 01] Resultado do processamento deve estar próximo ao gabarito oferecido.

[RD 02] Professor só consegue realizar o processamento de uma única turma por vez, cada aluno por vez.

[RD 03] O resultado do processamento deve gerar uma tabela e um gráfico.

[RD 04] A entrada de texto deve ser unificada e padronizada

[RD 05] O sistema será executável e não instalável.

[RD 06] O sistema deve reconhecer a entrada de arquivo-texto e gerará um relatório em formato PDF.

[RD 07] O sistema deve ser desenvolvido em PHP.

##### D. Especificação do Diagrama do Caso de Uso

O diagrama de caso de uso consiste em ator, subordinado e o sistema, o qual realiza a tarefa, entre eles o professor, o aluno, e o sistema. No sistema o aluno e o professor efetuam o login porém apenas o professor pode utilizá-lo, registrando tarefas. O aluno visualiza o feedback e os seus exercícios anteriores. A função do sistema é analisar as características e extrair dados presentes nos códigos seguindo as instruções prévias adotadas pelo professor.



### *E. Especificação do Diagrama de Classes*

A Figura 5 (Apêndice A) contém o diagrama de Classe, que além dos atributos que elas possuem, podemos visualizar seus métodos e seus relacionamentos entre as entidades presentes no sistema

### *F. Especificação do Diagrama de Banco de Dados*

A Figura 6 (Apêndice C) apresenta o diagrama de banco de dados do sistema. No diagrama de banco de dados é representado três classes criadas entre elas temos A de Usuário, Tarefas e Características.

### *G. Especificação do Protótipos de Tela*

Utilizando a ferramenta Figma foram criados os protótipos de tela, os quais simulam a maneira de como funcionará o software.

Na Figura 7 (Apêndice D) temos a tela de Login com apenas usuário e senha.

Na Figura 8 (Apêndice D) temos a tela inicial, logo após a tela de Login

Na Figura 9 (Apêndice D) temos a tela de carregar tarefas com uma breve visualização dos códigos de cada aluno selecionado.

Na Figura 10 (Apêndice D) temos a tela de resultado, que aparecerá logo após o usuário apertar em processar, onde será mostrada na tela as características de cada tarefa.

## 5. RESULTADOS ESPERADOS

Como resultado, espera-se ter um sistema que funcione como prototipado, que seja uma ferramenta indispensável para professores de unidades curriculares que possuem a linguagem C como parte da ementa institucional. O professor terá mais um modo de auxílio, por meio de uma tabela será possível ver com mais detalhes as características do código-fonte escrito pelo aluno, de uma maneira mais dinâmica poupando mais tempo visto que geralmente ele possui uma demanda de turmas e tarefas extensa.

Com isso, poderá mudar completamente a forma de auxílio pedagógico que atualmente o professor exerce com o aluno em sala de aula.

### *A. Limitações*

A principal limitação do projeto é o aluno não utilizar o sistema, a princípio a ideia era ambos utilizarem porém de fato o aluno não teria acesso a basicamente nada além de visualizar o feedback de suas tarefas, outra ideia era que o aluno enviasse a tarefa pelo sistema porém amadureceu a ideia de que o software seria de uso exclusivo do professor.

Outra restrição foi a de que o sistema seria utilizado para extrair somente características da linguagem C, talvez em uma próxima versão seja compatível com outras linguagens.

## AGRADECIMENTOS

Gostaríamos de agradecer a todas as pessoas que contribuíram direta ou indiretamente na realização desse projeto, pois foi fundamental para o desenvolvimento e conclusão deste.

Em especial, os autores agradecem a Davi Bernardo Silva, pela colaboração, idealização e orientador para este trabalho, além de toda paciência e valiosos insight ao longo do projeto. Seu conhecimento e mentoria foram essenciais para que esse projeto se realizasse e fizéssemos que chegássemos até aqui.

Agradecemos também ao IFSC-Rau por nos disponibilizar toda estrutura adequada essenciais para a conclusão dessa jornada.

## REFERÊNCIAS

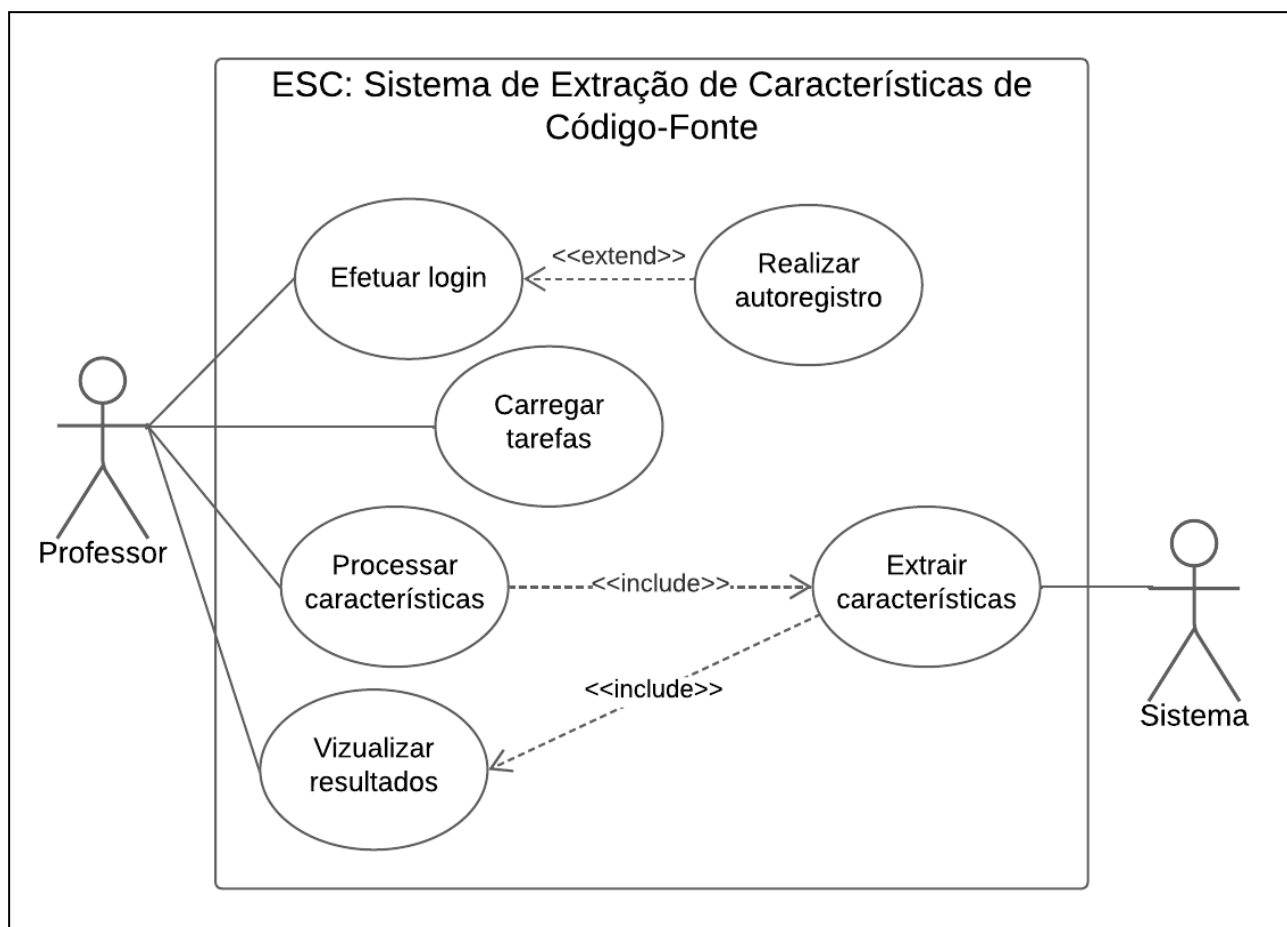
- [1] AUSUBEL, David P. A aprendizagem significativa. São Paulo: Moraes, 1982.
- [2] FOLHA DE SÃO PAULO. Nova leva de cursos tenta suprir o aumento da demanda por profissionais de TI. Disponível em: <<https://encurtador.com.br/hjlzI>>. Acesso em: 9 de maio de 2023.
- [3] GOOGLE LLC. (2022). Grasshopper: Aprenda a programar (Versão 2.1.3). Disponível em: <<https://play.google.com/store/apps/details?id=com.are.a120.grasshopper>> Acesso em: 21 maio. 2023.
- [4] Pup Interactive. (2023). Programming Hub: Aprenda a programar em várias linguagens (Versão 5.0.2). Disponível em: <<https://play.google.com/store/apps/details?id=com.free.it.java>>. Acesso em: 21 maio. 2023.
- [5] SILVA, Davi Bernardo; SILLA, Carlos N. Evaluation of students programming skills on a computer programming course with a hierarchical clustering algorithm. In: 2020 IEEE Frontiers in Education Conference (FIE). IEEE, 2020. p. 1-9.
- [6] SOLOLEARN Inc. (2023). Sololearn: Aprenda a programar, codificar e muito mais (Versão 4.2.1). Disponível em: <<https://play.google.com/store/apps/details?id=com.sololearn>> Acesso em: 21 maio. 2023.



**INSTITUTO FEDERAL**  
Santa Catarina  
Câmpus Jaraguá do Sul – Rau

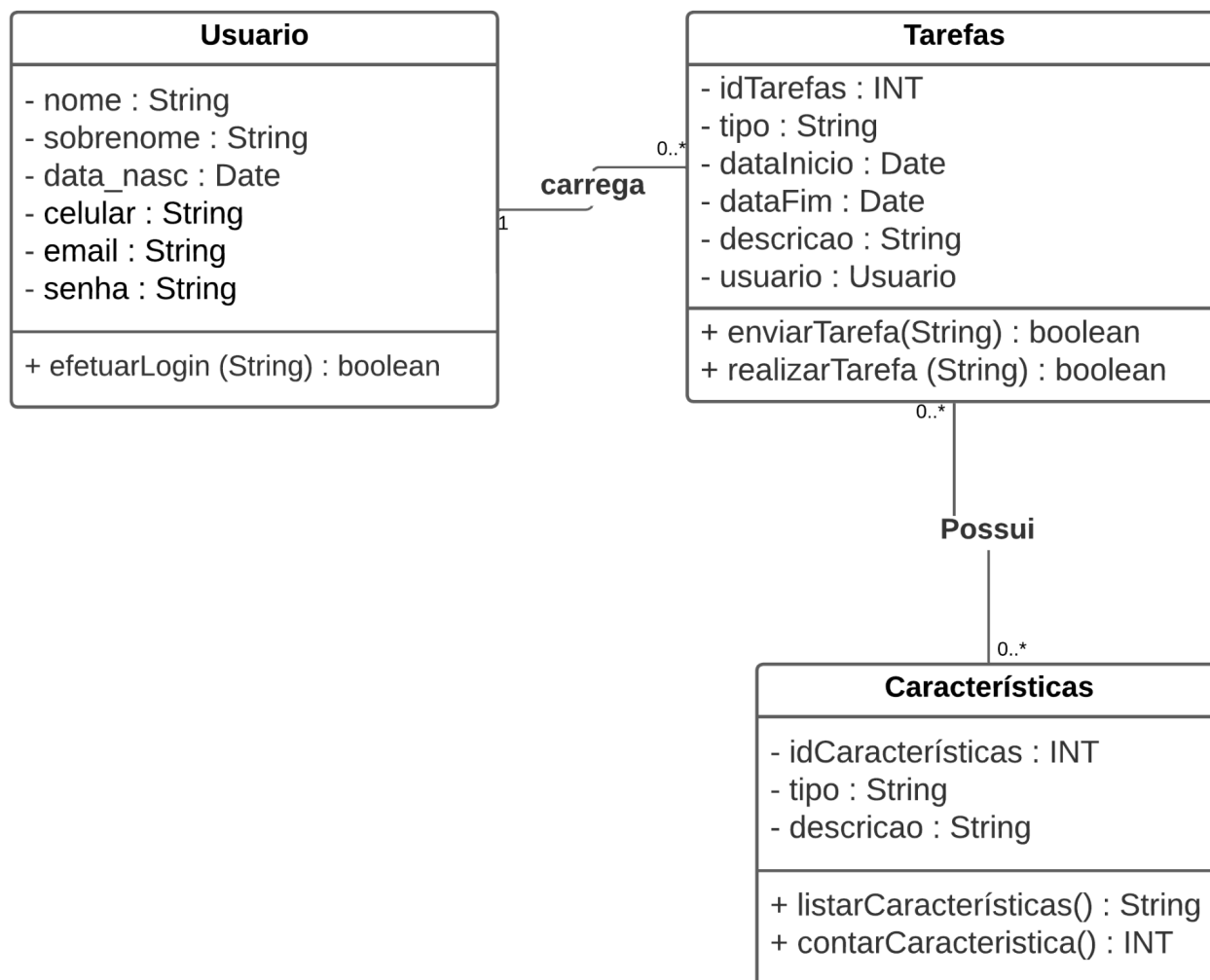
Ministério da Educação  
Secretaria de Educação Profissional e Tecnológica  
**INSTITUTO FEDERAL DE SANTA CATARINA**

APÊNDICE A - DIAGRAMA DE CASOS DE USO



**Figura 4.** Diagrama de Casos de Uso do sistema

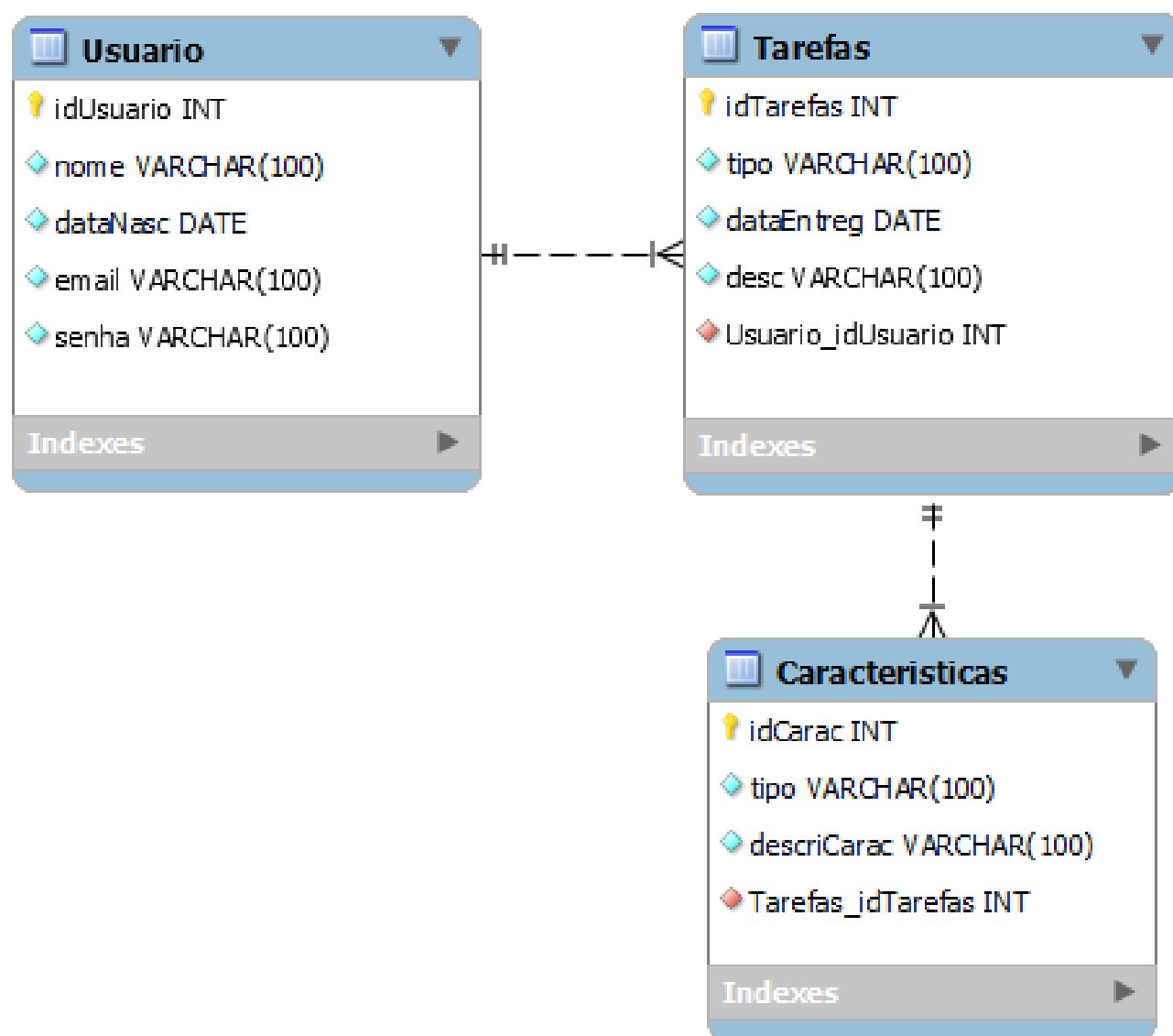
APÊNDICE B – DIAGRAMA DE CLASSES



**Figura 5.** Diagrama de Classes do sistema

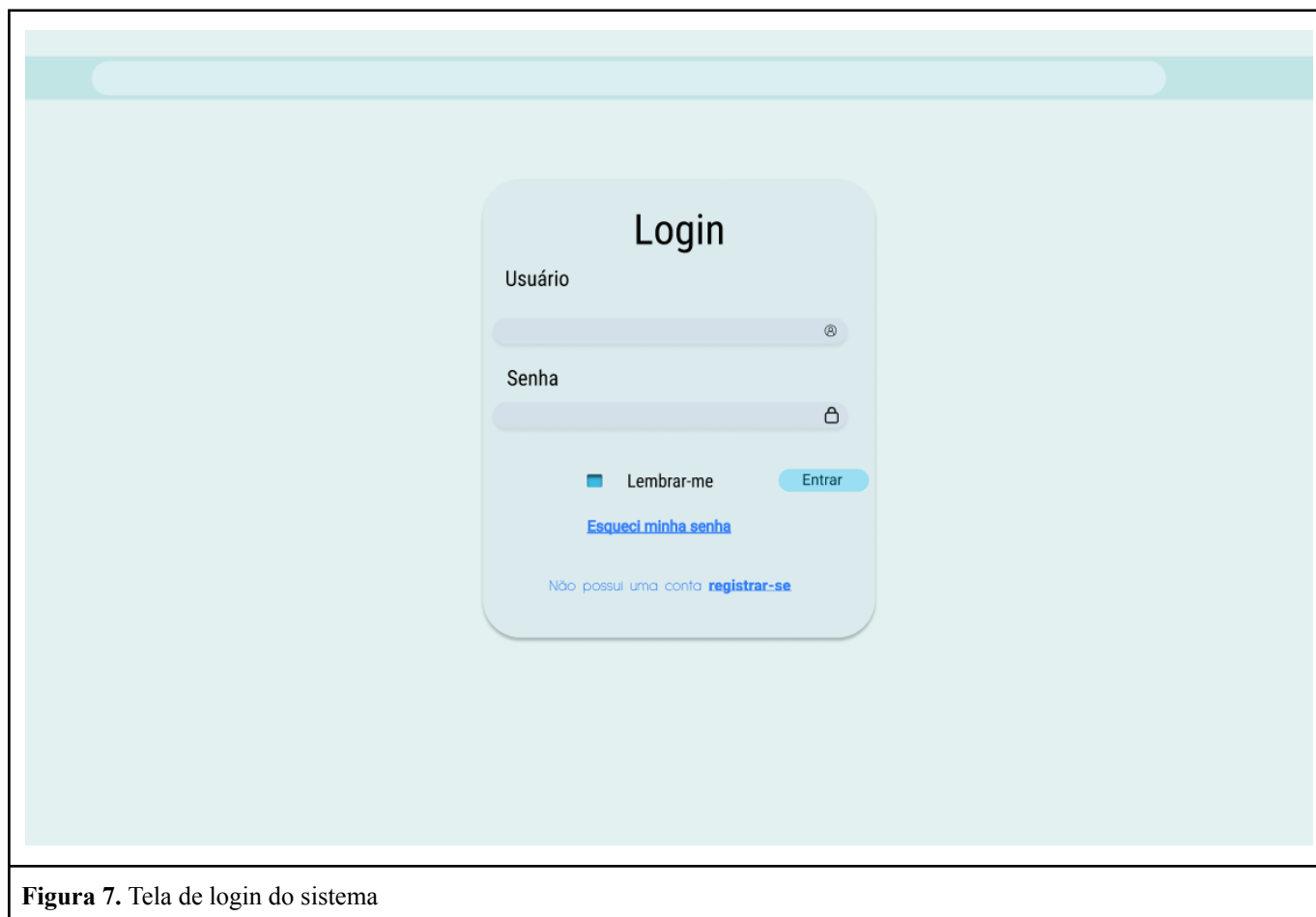


APÊNDICE C - DIAGRAMA DE BANCO DE DADOS

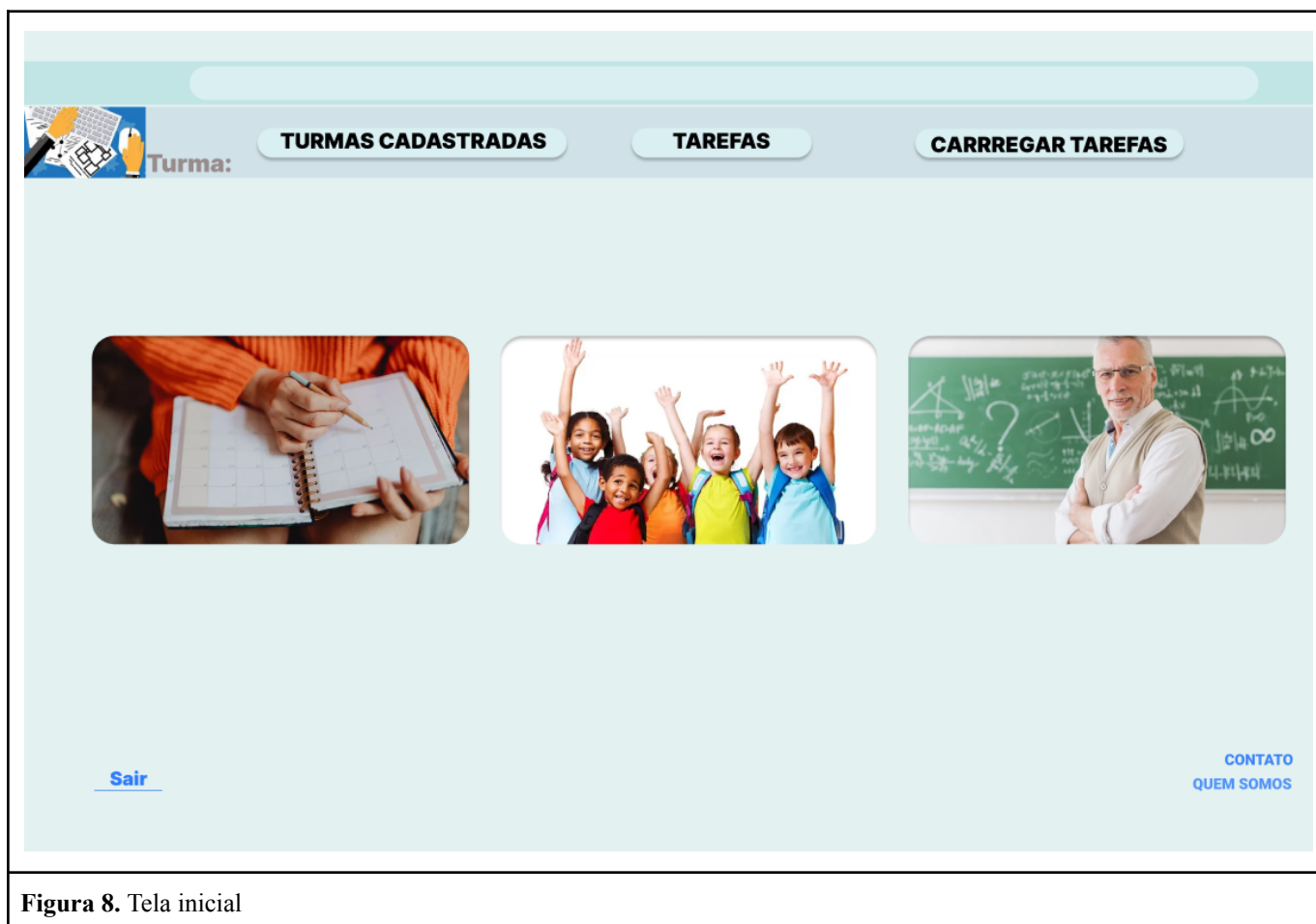


**Figura 6.** Diagrama de Banco de Dados do sistema

APÊNDICE D - PROTÓTIPOS DE TELA



**Figura 7.** Tela de login do sistema



**Figura 8.** Tela inicial



INT, FLOAT, DOUBLE, CHAR, IF, ELSE

\* Use vírgulas para separar os termos

Escolher Arquivo

Nenhum arquivo escolhido

\* Você pode adicionar mais de um arquivo

Lista de arquivos carregados

L1EX01\_pedro.c

L1EX01\_maria.c

L1EX01\_joana.c

L1EX01\_tereza.c

L1EX01\_patricia.c

L1EX01\_paulo.c

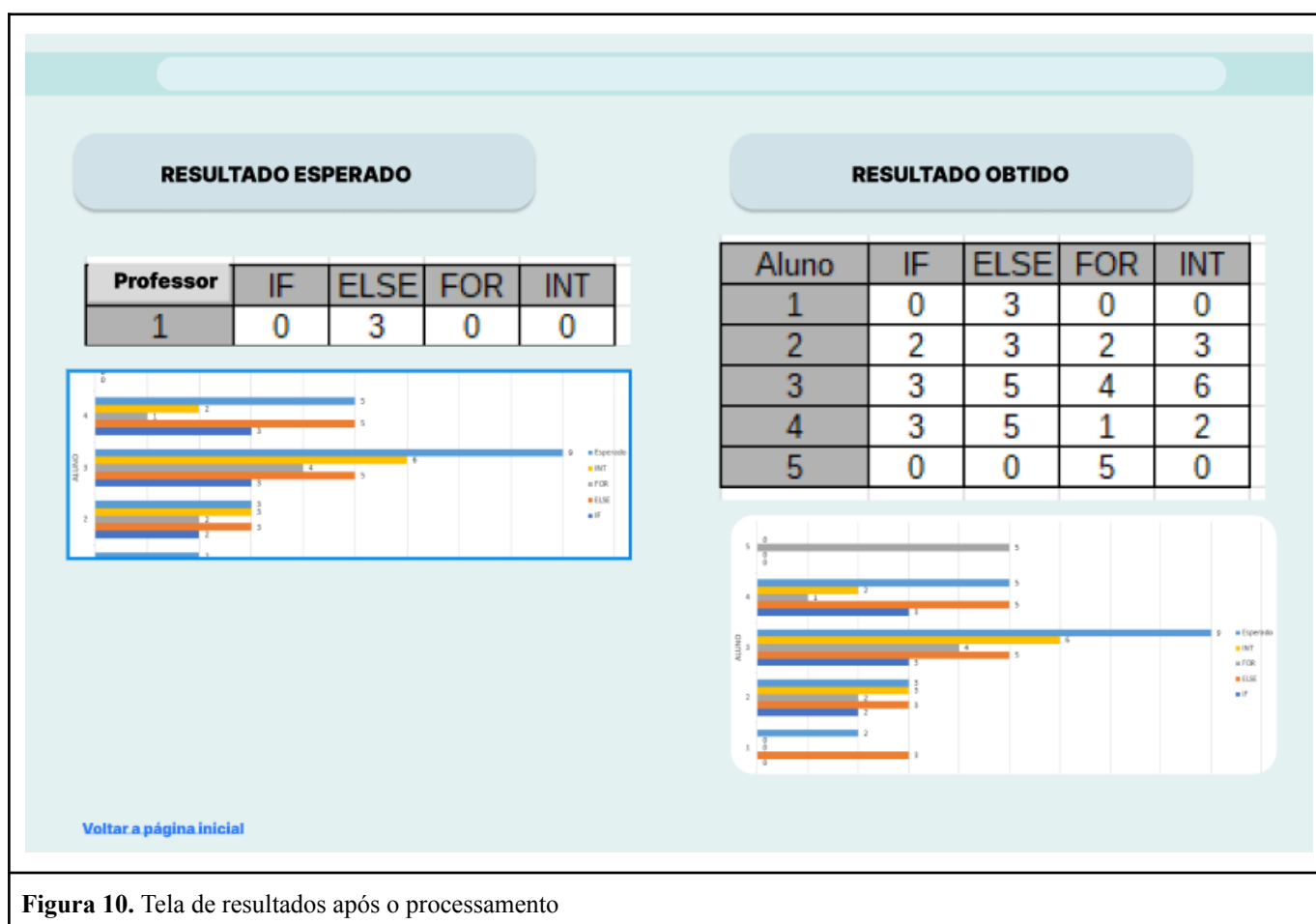
[Voltar](#)

Tarefa de código-fonte selecionada

```
#include <stdio.h>
int main (void)
{
    int cont = 0;
    while (cont < 10)
    {
        if (cont == 5) continue;
        printf("%d ", cont);
        cont=cont+1;
    }
    printf("fim\n");
    return 0;
}
```

[Processar](#)

**Figura 9.** Tela de carregar tarefas, com breve visualização do código.



**Figura 10.** Tela de resultados após o processamento