

## Resolução comentada de Exercícios na Plataforma URI-Becrowd

Augusto dos Santos Gomes Vaz\*, Caio Ueda Sampaio\*, João Dini de Miranda\*  
Pietro Minghini Morales\*, Thiago Roberto Garcia Albino\*, Renato Bueno‡

Departamento de Computação  
Universidade Federal de São Carlos (UFSCar) - São Carlos, SP - Brasil

**Abstract.** *This article describes the execution of the project entitled ‘URI Platform Exercise Resolution’, started in 2020 and developed by the PET-BCC group at UFSCar. The main objective was to reinforce the undergraduate students in Computer Science courses at UFSCar in learning the C and Python programming languages. The project proposed solutions for C programming exercises available on the URI platform, both at the Beginner level and in topics related to strings. For organization, platforms such as Trello and GitHub were used. Additionally, the project was expanded to include the application of programming in the Python language.*

**Resumo.** *Este artigo descreve a realização do projeto intitulado ‘Resolução de Exercícios da Plataforma URI’, iniciado em 2020, desenvolvido pelo grupo PET-BCC da UFSCar. O objetivo principal foi reforçar o aprendizado das linguagens de programação C e Python dos graduandos de cursos de Computação da UFSCar. O projeto propôs resoluções para os exercícios de programação em C disponíveis na plataforma URI, tanto no nível Iniciante quanto em tópicos relacionados a strings. Para a organização, foram utilizadas plataformas como Trello e GitHub. Além disso, o projeto foi ampliado para incluir programação na linguagem Python.*

### 1. Introdução

O projeto ‘Resolução de Exercícios da Plataforma URI’ surgiu como um projeto realizado pelo grupo do ‘Programa de Educação Tutorial do Bacharelado em Ciência da Computação da UFSCar’ (PET-BCC) [PET-BCC 2009] em 2020. Naquela época, o PET-BCC era coordenado pelo professor Valter Vieira de Camargo, que optou também por registrar o projeto como uma atividade de extensão na Pró-Reitoria de Extensão (ProEx).

No planejamento inicial do projeto, definiu-se que o objetivo principal seria disponibilizar soluções comentadas para os exercícios presentes na plataforma “URI online Judge” [Becrowd 2019], um repositório internacionalmente reconhecido que abriga enunciados de exercícios de programação. A intenção era permitir que interessados no aprendizado da linguagem C, particularmente estudantes na área de computação, pudessem se beneficiar das resoluções desenvolvidas. Com isso, visava-se beneficiar não apenas os alunos do Departamento de Computação (DC) da UFSCar

---

\*Membro discente do Grupo do Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de São Carlos - PET-BCC UFSCar

‡Tutor do Grupo do Programa de Educação Tutorial do curso de Ciências da Computação da Universidade Federal de São Carlos - PET-BCC UFSCar

[Departamento de Computação 2018], mas também estudantes de outros cursos e universidades com as soluções disponibilizadas.

Além do objetivo principal de contribuir para a comunidade interessada em aprofundar-se na programação, o projeto também tinha a finalidade de aumentar a visibilidade do grupo PET-BCC, tanto no âmbito da universidade como da comunidade de estudantes de computação no país.

A atividade envolveu um total de 11 participantes, todos eles membros do PET-BCC. Cada resolução de exercício foi revisada por ao menos mais um membro do projeto, sendo que algumas recebiam a revisão de até 3 membros. Esses exercícios abrangiam desde níveis iniciantes até desafios mais avançados, como os relacionados a *strings*, e inicialmente foram implementados utilizando a linguagem de programação C.

A divulgação do projeto foi realizada por meio de redes sociais, como o Facebook [PET BCC 2020a]. Na Figura 1, é possível observar uma das postagens de divulgação na página oficial do PET-BCC, contendo informações sobre o projeto e instruções de acesso ao repositório.



Figura 1. Divulgação do projeto ‘Resolução de Exercícios da Plataforma URI’ na página do grupo PET-BCC no Facebook no ano de 2020.

## 2. Metodologia

O desenvolvimento do projeto baseou-se na utilização de três plataformas mundialmente reconhecidas: o “URI *online Judge*”, de onde eram selecionados os exercícios a serem resolvidos; o Trello [PET BCC 2020c], que era utilizado para a organização geral do projeto; e o GitHub, empregado para a revisão e análise das soluções dos exercícios pelos alunos. Vale ressaltar que os passos seguidos para a resolução dos exercícios em linguagem de programação C também foram adotados posteriormente para a resolução dos exercícios em Python.

## 2.1. URI *online Judge*

Inicialmente, os exercícios eram selecionados na plataforma “URI *online Judge*”, que atualmente é conhecida como Beecrowd [Beecrowd 2019]. Essa plataforma é reconhecida internacionalmente pela comunidade de computação e oferece uma ampla variedade de problemas e exercícios de programação em diversas linguagens de programação. O seu objetivo é auxiliar estudantes e programadores a aprimorar suas habilidades em programação e algoritmos.

A plataforma categoriza os exercícios em várias áreas, incluindo matemática, estrutura de dados, *strings*, algoritmos, entre outras. Cada problema é associado a um número de identificação exclusivo e pode ter múltiplas entradas e saídas para auxiliar os usuários a compreender como se espera que a resolução seja implementada.

O processo de utilização da plataforma é simples. Após a criação de uma conta e realização do *login*, os usuários podem escolher o exercício a ser resolvido e submeter o código de resposta na linguagem escolhida após a conclusão. Após a submissão, a URI realiza uma série de testes para verificar a presença de erros. Se algum teste falhar, é fornecido um *feedback* indicando as discrepâncias entre as saídas esperadas e as obtidas.

Além disso, a plataforma oferece a possibilidade de comparar o progresso individual com o de outros estudantes. Cada exercício resolvido resulta em um aumento na quantidade de pontos e níveis da conta do usuário.

A princípio, o projeto focou na resolução dos exercícios na linguagem C. Esta linguagem foi criada nos anos 1970 por Dennis Ritchie e apresenta um nível de abstração mais baixo, o que induz seus programadores a obterem mais conhecimento sobre o funcionamento dos computadores e de suas estruturas, e é considerada “uma das linguagens mais versáteis disponíveis, sendo relevante até os dias de hoje”[Balreira 2022]. Além disso, esta é a primeira linguagem de programação usada pelos cursos de computação oferecidos pelo DC. Desta maneira, a resolução de exercícios em C seria a que melhor ajudaria os estudantes do departamento com suas resoluções, e por isso foi definida como prioridade.

Posteriormente, foi observada também a necessidade da resolução dos exercícios do URI em Python. A linguagem se apresenta como uma alternativa mais acessível para iniciantes: “Entre as demais linguagens consideradas mais didáticas, observa-se que no decorrer dos últimos anos, uma predominância na escolha do Python como linguagem inicial para cursos de nível superior e cursos de formação de programadores em geral”[Forbellone and Eberspächer 2023]. A linguagem Python é utilizada atualmente na grande maioria das disciplinas de programação oferecidas pelo DC aos demais cursos da universidade. Além disso, Python também é usado para várias outras áreas mais avançadas na computação, como aprendizado de máquina, ciência de dados e computação científica. Desta maneira, para suprir a demanda dos alunos, o projeto iniciou resoluções dos exercícios também em Python.

Além da utilização dessas linguagens nas disciplinas ofertadas aos alunos de graduação, o “Stack Overflow”[Stack Overflow 2020] - uma das maiores e mais populares plataformas de perguntas e respostas relacionadas à programação e ao desenvolvimento de *software* - classificou as principais linguagens de programação de 2020, onde tanto a linguagem C quanto a linguagem Python estavam entre as mais utilizadas.

Na fase de seleção dos exercícios a serem resolvidos, o grupo coletou todos os exercícios que existiam até aquele momento na plataforma nas categorias iniciante e ‘string’. Ao todo, foram selecionados 288 exercícios da categoria ‘iniciantes’ e 60 da categoria *string* para a linguagem C. Posteriormente, foram selecionados 225 exercícios da categoria ‘iniciantes’ e 117 da categoria ‘string’ para a linguagem Python.

## 2.2. Trello

O Trello é uma plataforma de gerenciamento de projetos e tarefas *online* que utiliza o método de organização visual conhecido como quadro Kanban [Kniberg and Skarin 2010], projetado para ajudar indivíduos e equipes a acompanhar e colaborar em projetos de forma eficiente.

A principal característica do Trello é a organização de tarefas em “quadros”, que funcionam como painéis virtuais onde os usuários podem criar listas verticais para representar etapas ou estágios do projeto. Dentro de cada lista, é possível criar “cartões” que representam tarefas individuais. Esses cartões podem ser movidos entre as listas à medida que o trabalho progride, refletindo visualmente o fluxo de trabalho.

No Trello, todos os exercícios estão categorizados em grupos que indicam o status atual, incluindo os exercícios que devem ser resolvidos, os que estão com resolução em andamento por algum membro, os que requerem correção, os que estão sendo corrigidos por algum membro e os que já foram incorporados à *branch* principal do repositório do projeto no GitHub, como demonstrado na Figura 2.

Nesta plataforma, o estudante seleciona, das listas “Para fazer - Iniciante” e “Para fazer - Strings” os exercícios que deseja resolver, movendo o cartão para a lista “Fazendo”. Após desenvolver a resolução do exercício, o aluno deve mover o cartão do seu respectivo exercício para o quadro “Corrigindo”, onde os outros membros do projeto devem analisar a solução. Estando tudo correto, o cartão é movido para o quadro “Merged”, que indica que o exercício foi executado e avaliado pelos membros do projeto.

## 2.3. GitHub

GitHub é uma plataforma de hospedagem de código-fonte, que oferece serviços de controle de versão usando o sistema de controle de versão distribuído Git. A plataforma permite que desenvolvedores e equipes de desenvolvimento colaborem em projetos de *software* e trabalhos semelhantes.

Na execução deste projeto, para a utilização do GitHub é necessário inicialmente realizar um *fork* do repositório principal [PET BCC 2020b]. Isso corresponde à criação de uma cópia do repositório de código-fonte existente, permitindo assim que o desenvolvimento prossiga de forma independente, sem prejudicar outras pessoas que estejam trabalhando no mesmo repositório. Após a conclusão da etapa de produção do algoritmo, que corresponde à etapa “Fazendo” no Trello, o membro deve enviar os exercícios realizados para o seu repositório e criar uma *pull request*. Esta *pull request* é, na verdade, uma solicitação para que as alterações feitas no *fork* sejam incorporadas ao repositório original. Em outras palavras, essa fase representa a etapa “Corrigindo” no Trello.

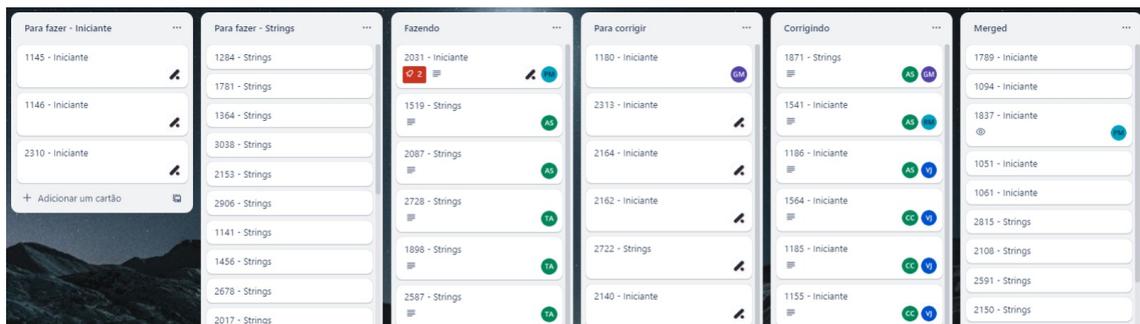
Para garantir que a contribuição do aluno seja atribuída ao repositório principal, é fundamental que o mesmo siga os guias de contribuição e padronização. O primeiro

guia diz respeito aos passos que devem ser seguidos para solicitar a ”pull request” no repositório principal. Esses passos incluem a verificação da funcionalidade do código no URI, a padronização da nomenclatura do código, a explicação do código de acordo com as diretrizes de padronização, a organização do código em uma pasta com o nome correspondente ao exercício e, finalmente, a realização da *pull request*. A Figura 3 mostra detalhadamente os passos do “Guia de Contribuição”.

O “Guia de Padronização” está relacionado à maneira como a estrutura do código-fonte deve ser organizada. Isso inclui desde a definição do problema fornecida pelo URI até a estruturação da resolução, como a modularização de funções e a formatação do número de casas decimais que um valor deve conter. As Figuras 4 e 5 apresentam um guia passo a passo para este tipo de orientação.

Seguindo esses guias e executando a solução adequada para o problema, os demais membros do grupo avaliam a resolução e podem aprovar ou solicitar alterações nos arquivos produzidos por outro membro do projeto.

Na Figura 6, é possível observar um fluxograma que resume os passos adotados para a resolução dos exercícios, desde a seleção, até a aprovação e posteriormente a junção no repositório do projeto.



**Figura 2. Exemplo de utilização da plataforma Trello.**

Olá!

Para fazer sua contribuição no repositório do URI, siga os seguintes passos:

1. **Teste se seu código realmente funciona no URI.**

Parece básico, mas temos o costume de esquecer de testar se nosso código realmente é aceito pelo URI. Certifique-se que ele funcione no problema referente sem nenhum erro.

2. **Nomeie seu código como [número do problema].c**

Como nosso repositório tem o foco na linguagem C e para que fique mais fácil de encontrá-lo, nomeie-o desta forma. Ex: 1018.c

3. **Faça a explicação do código seguindo o [guia de padronização](#) e utilizando [Markdown](#)**

Como referência de explicação, utilize nossos exercícios já feitos, disponíveis em nosso repositório. Lembre-se de terminar com os links de contato do PET BCC, conforme o guia e os exercícios disponíveis.

Algumas referências que podem ajudar com o Markdown estão disponíveis [aqui](#), além de editores de Markdown online, como o [Dillinger](#) e o [Editor.md](#)

---

4. **Salve sua explicação como README.md**

Assim, sua explicação ficará disponível de forma mais acessível, logo no começo da pasta.

5. **Salve ambos código.c e README.md em uma pasta, nomeada como o número do seu exercício**

Exemplo: a pasta 1018 contém o 1018.c e o README.md referente a ele

6. **Faça seu pull request em nosso repositório.**

Pronto! Logo um membro da equipe do PET-BCC UFSCar avaliará sua contribuição.

Agradecemos o interesse em contribuir com nosso projeto!

Figura 3. Guia de Contribuição

Guia do README

## Problema:

Cópia da parte lógica do problema do site do URI, que descreve o problema em língua natural.

Caso a entrada/saída seja vital na explicação do problema, pode-se citá-la como "conforme a entrada/saída exemplificada no exercício [exemplo da entrada/saída] (...)". Caso contrário, deve ser omitida.

Problema Completo: link\_do\_problema

Figura 4. Guia de Padronização - parte 1.

## Resolução:

A resolução deve começar com uma explicação resumida do enunciado do problema, caso a interpretação não seja objetiva, e da lógica de resolução. Exemplo: "O exercício pede para calcular se 3 retas formariam um triângulo, para resolvê-lo, deve-se criar ler os valores e calcular com a fórmula (...)"

Definir e explicar o uso de bibliotecas não triviais do C, como `<math.h>` e semelhantes.

Declaração de variáveis: preferencialmente com as variáveis que serão utilizadas na resolução do problema, com nomes intuitivos. Exemplo: "Utilizamos uma variável inteira para a soma, uma variável de ponto flutuante para o valor (...)"

```
int soma;
float valor;
```

Lógica do problema com trechos do código. Buscar explicar todos os trechos do código completo com passo a passo, incluindo a saída do problema formatada para o URI.

Para trechos de explicação que referenciam variáveis, funções, bibliotecas e tipos deve-se usar `variável`, como nos exemplos: "Para leitura da entrada, utiliza-se a função `scanf`..."

"Utilizamos `%.3lf` ao invés de apenas `%lf` (que representa um valor `double`) para delimitarmos..."

Para conceitos-chave da lógica do problema, inserir um link de explicação no texto, dando preferência para explicações em português. Exemplo: "(...) calcularemos a [área do triângulo](#)"

Para conceitos base e anteriores ao problema, deixar links de explicação no final da resolução. Evitar conceitos que já sejam presumíveis para o nível do exercício, como declaração de variáveis em exercícios de nível médio/avançado. Exemplo:

Para aprender um pouco mais sobre variáveis: [Variáveis](#)

Caso tenha alguma dúvida sobre este problema ou sobre a resolução, entre em contato com o PET-BCC pelo nosso [Facebook](#), [Instagram](#) ou nos mande um e-mail com o assunto "URI" para [petbcc.ufscar@gmail.com](mailto:petbcc.ufscar@gmail.com)

### Guia do código

- Sem comentários, todo o código deve ser explicado no README
- Atentar-se com o fluxo de leitura do código, respeitando a indentação
- Busque deixar seu código o mais modularizado possível, para facilitar o entendimento no README. Exemplo: definir uma função para ordenação de listas (caso necessária) fora da função main.
- **TESTA NO URI ANTES DE ENVIAR PELO AMOR DE ALAN TURING**

Figura 5. Guia de Padronização - parte 2.

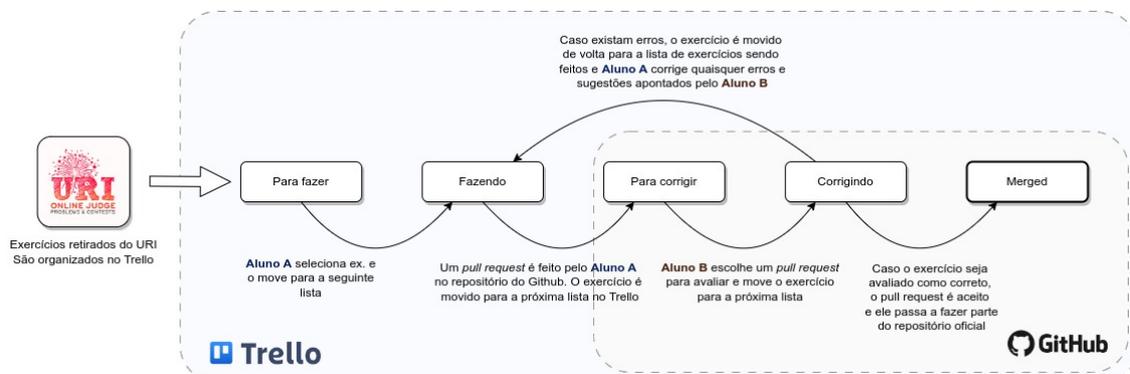


Figura 6. Fluxograma: resolução de exercícios URI.

### 3. Resultado e Impacto

Uma das bases fundamentais da criação deste projeto foi o fato de que nos cursos de Computação da UFSCar, os discentes, nas disciplinas iniciais, como “Construção de Algoritmos e Programação” e “Algoritmos e Estrutura de Dados 1” [UFSCar 2018], têm o seu primeiro contato com uma linguagem de programação, que no caso é a Linguagem C. O mesmo ocorre com os discentes de outros cursos da universidade que cursam disciplinas de programação ofertadas pelo DC, como “Programação e Algoritmo 1 e 2”, em que os alunos tem seu primeiro contato com a linguagem de programação Python.

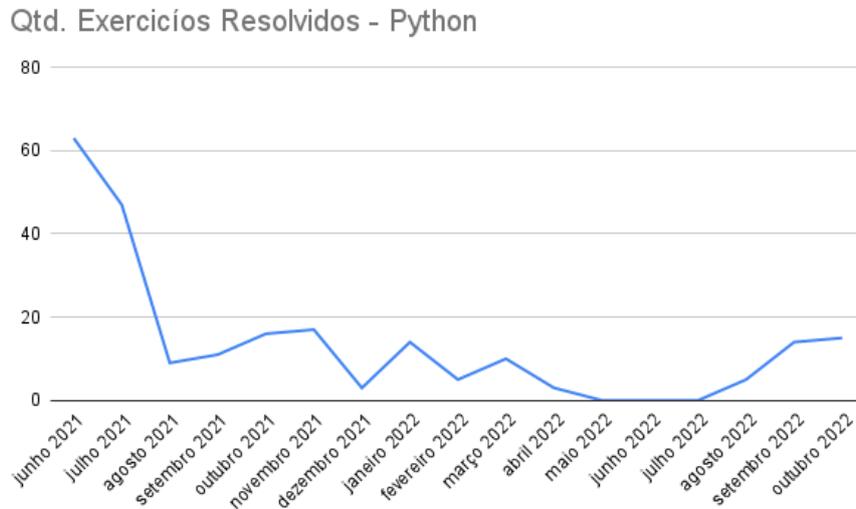
Dessa forma, a criação do repositório surgiu com o propósito de auxiliar, inicialmente, aqueles que iniciantes na linguagem de programação C, com foco nos calouros e até mesmo nos veteranos do DC. Posteriormente, o projeto optou por também atender àqueles que estão iniciando na linguagem de programação Python. O repositório disponibiliza soluções que abrangem desde exercícios mais simples até exercícios mais complexos, com o objetivo de contribuir para uma compreensão abrangente dessas linguagens.

No decorrer do projeto foram solucionados 288 problemas da categoria ‘iniciante’ e 60 exercícios da categoria ‘strings’ em linguagem de programação C, todos eles revisados, disponíveis no repositório original. O gráfico da Figura 7 apresenta a quantidade de exercícios solucionados ao longo dos meses do projeto. É interessante destacar que o mês de agosto de 2020 registrou o maior número de contribuições. Além disso, os meses de novembro e dezembro foram dedicados exclusivamente aos exercícios relacionados a strings.

Quanto aos exercícios de Python, foram solucionados um total de 187 problemas da categoria ‘iniciante’ e 35 exercícios da categoria de ‘strings’, todos revisados e disponibilizados no repositório original. O gráfico da Figura 8 apresenta a quantidade de exercícios solucionados ao longo dos meses do projeto.



Figura 7. Resolução de Exercícios na linguagem C durante o projeto.



**Figura 8. Resolução de Exercícios na linguagem Python durante o projeto.**

#### 4. Conclusão

Pode-se afirmar que a atividade transcorreu da maneira como foi planejada, com os membros se organizando semanalmente para resolver os exercícios e disponibilizá-los para qualquer interessado. Houve a elaboração de padrões para a descrição das resoluções e para a padronização das soluções dos exercícios. A atividade começou com a resolução de exercícios mais simples e progrediu em níveis de complexidade, de modo que o repositório de resoluções possa beneficiar tanto os alunos do DC quanto os alunos de outras áreas que tenham interesse em soluções de exercícios de programação em C e Python. Além disso, houve um notável crescimento na formação dos alunos da equipe.

Com isso, é possível concluir que o desenvolvimento do projeto foi plenamente satisfatório. Além do desenvolvimento notável dos membros do projeto, é evidente a contribuição do repositório para o desenvolvimento dos demais discentes dos cursos de computação da UFSCar.

Por último, vale ressaltar que o projeto ainda está em desenvolvimento. O objetivo é que os alunos do PET que participam do projeto terminem os exercícios de *Python* até o final do ano de 2023.

#### Referências

- Balreira, D. G. (2022). *Programação Didática com Linguagem C*. Paco Editorial, 1 edition.
- Beecrowd (2019). Site Oficial do Beecrowd. Disponível em: <https://www.beecrowd.com.br/judge/en/login>. Acessado em: 20 set. 2023.
- Departamento de Computação (2018). Departamento de Computação. Disponível em: <https://site.dc.ufscar.br/>. Acessado em: 20 set. 2023.
- Forbellone, A. L. V. and Eberspächer, H. F. (2023). *Lógica de programação: a construção de algoritmos e estruturas de dados com aplicações em Python*. Grupo A, 1 edition.

- Kniberg, H. and Skarin, M. (2010). *Kanban and Scrum - Making the Most of Both*. Enterprise software development series. C4Media Incorporated.
- PET-BCC (2009). Site do PET-BCC. Disponível em: <https://petbcc.ufscar.br/>. Acessado em: 20 set. 2023.
- PET BCC (2020a). Perfil do Facebook do PET-BCC. Disponível em: <https://www.facebook.com/petbcc>. Acessado em: 20 set. 2023.
- PET BCC (2020b). Repositório do projeto Resolução de Exercícios da Plataforma URI/Beecrowd. Disponível em: <https://github.com/petbccufscar/beecrowd/>. Acessado em: 20 set. 2023.
- PET BCC (2020c). Trello Board. Disponível em: <https://trello.com/b/4PozH52B/beecrowd>. Acessado em: 20 set. 2023.
- Stack Overflow (2020). Stack Overflow Developer Survey 2020. Disponível em: <https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages>. Acessado em: 30 set. 2023.
- UFSCar ((2018)). Bacharelado em Ciência da Computação - Projeto Pedagógico.