

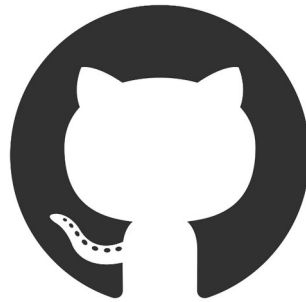
# Definições

## Modelagem GitHub

Disciplina: Fundamentos de Bancos de Dados 2020/2 - UFRGS

Grupo: Nicolle Favero e Matheus Azambuja

Professora: Karin Becker



## Universo de discurso

O universo de discurso é baseado na plataforma de hospedagem de código-fonte GitHub, atualmente gerenciado pela Microsoft Corporation, disponível em <https://github.com/>. Nosso objetivo com o SI é descrever essa plataforma (a parte social da plataforma), e as suas principais funcionalidades que são:

- Atividades entre usuários;
- Ações de usuários em repositórios;
- Estruturação dos repositórios;

## Usuário

Um usuário dentro do universo do GitHub é caracterizado por seu name (opcional), seu nickname (único), seu e-mail (único) e por uma biografia opcional (pequena autodescrição textual do usuário). Um usuário pode ser um contribuidor ou uma organização.

Um usuário do tipo contribuidor pode realizar ações relacionadas a outros repositórios que não o seu, como favoritar e contribuir. Um usuário do tipo organização geralmente pertence a alguma empresa. Organizações não podem realizar ações em repositórios, porém possuem membros (usuários contribuidores ligados de algum modo a essa empresa) que podem.

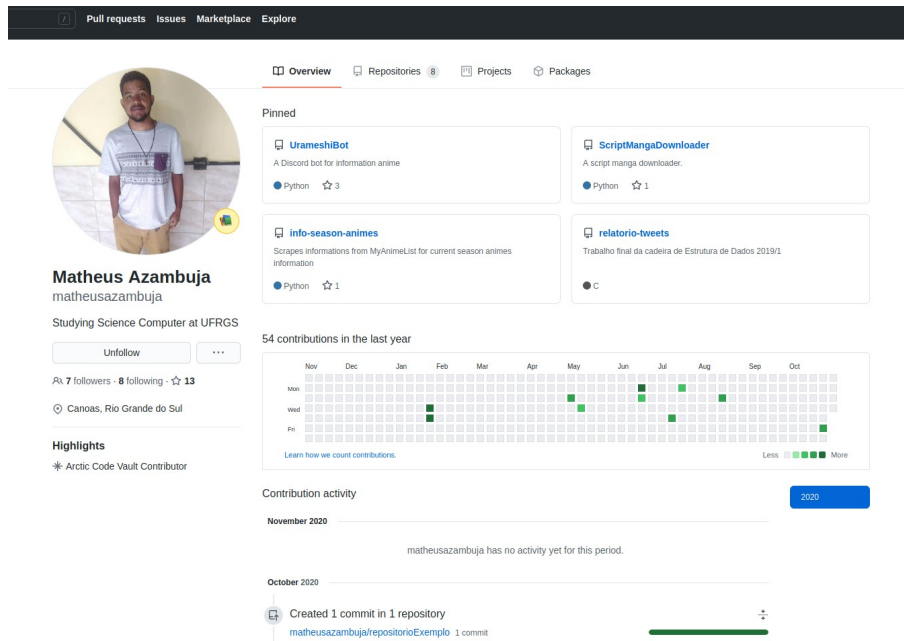


Figura 1: contributor account

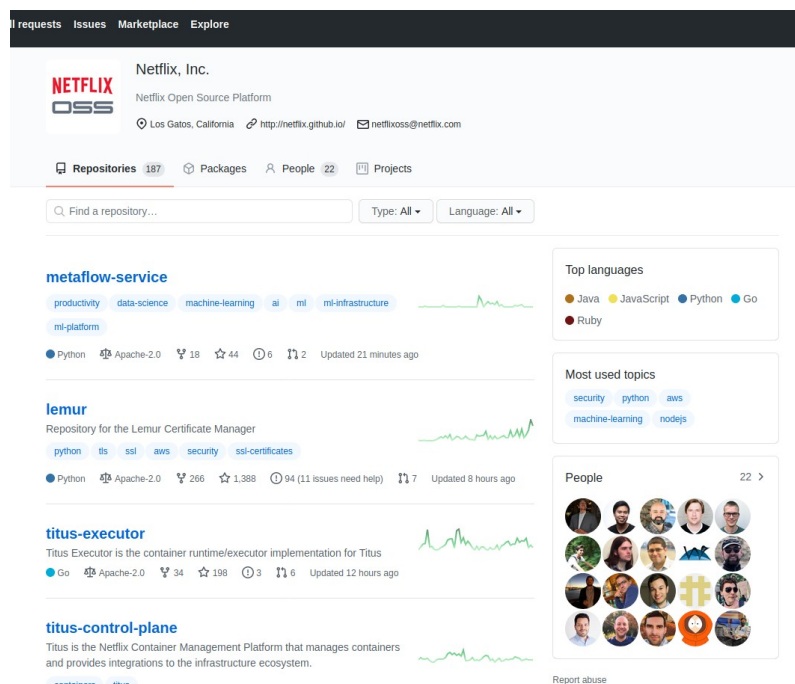


Figura 2: organization account

## Atividades entre Usuários

Um usuário do tipo contribuidor pode seguir outros usuários, desde que sejam outros contribuidores. A data em que foi efetuada essa ação é importante para alimentar uma timeline de notificações sobre as atividades dos contribuidores com quem se tem relacionamento no GitHub. Essa timeline é mostrada na página inicial.

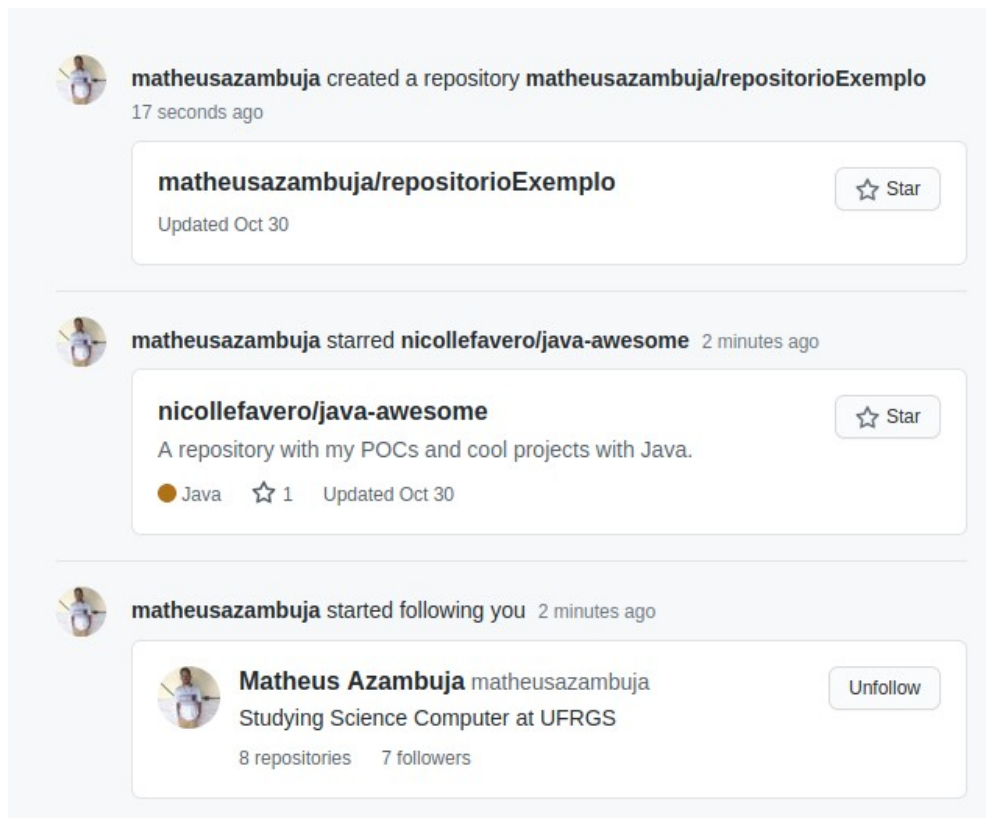


Figura 3: timeline

Um usuário do tipo organização possui membros que devem ser usuários do tipo contribuidor.

## Ações de Usuários em Repositórios

O GitHub é uma plataforma onde é possível realizar contribuições em repositórios de desconhecidos. Portanto, um usuário pode auxiliar na manutenção, correção de bugs de diversos repositórios. Um usuário pode possuir vários repositórios que devem ter um nome e uma data de criação, porém apenas usuário contribuidores podem:

- Favoritar repositórios: a data em que essa ação ocorreu deve ser registrada.

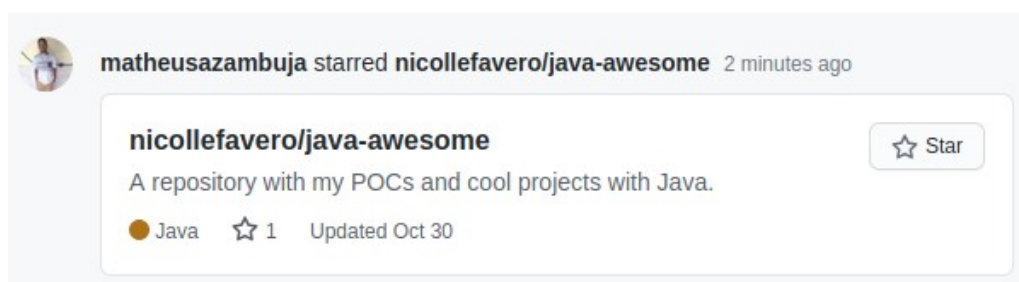


Figura 4: starred notification

- Abrir e comentar issues em repositórios. Na modelagem, representará um usuário que descobriu um problema ou bug em um repositório e relatou por meio dessa ação. Os issues tem um número de identificação (único dentro dos issues de um repositório), uma descrição do problema encontrado no repositório e sua data de abertura. Quando um issue for comentado, a data dessa ação deve ser registrada. No Github, os issues contêm um status que indica se ainda estão abertos ou se já foram fechados, porém na nossa aplicação simplificaremos os issues de modo que fiquem sempre abertos e os contribuidores podem comentar neles a qualquer momento.

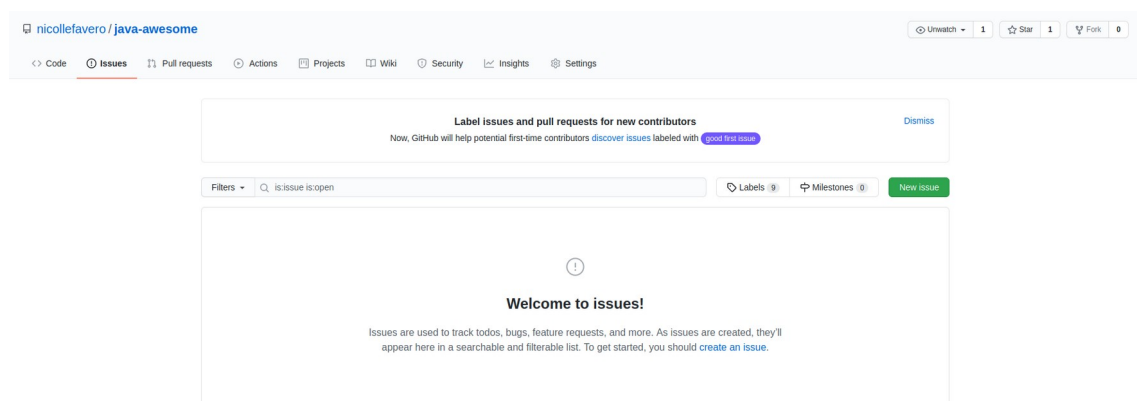


Figura 5: issues

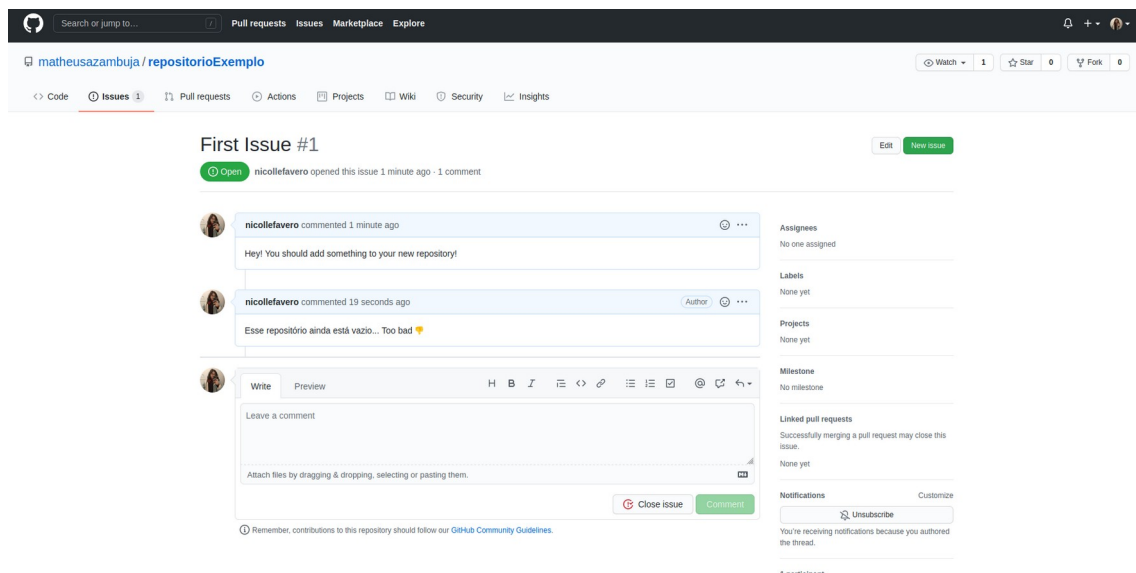


Figura 6: opened issue with comment

- Contribuir em repositórios através de commits. Um contribuidor pode commitar vários Items em vários repositórios. No nosso UdD, simplificamos que um commit poder ter apenas um Item. Cada commit deve possuir um hashcode único (dentro do repositório), um comentário, a data em que o commit foi feito e a modificação efetuada no item (usado para fazer o versionamento do item). Um contribuidor pode commitar o mesmo arquivo mais de uma vez, gerando uma nova versão desse arquivo. Normalmente, o ato de contribuir para um repositório pode acontecer de dois jeitos: em um jeito, o um usuário que tem permissão igual ao do criador do repositório, e no outro o usuário faz fork (cópia) do repositório que ele gostaria de sugerir uma modificação e envia um pull request para os usuários daquele repositório. No nosso universo de discurso, simplificamos de modo que apenas o primeiro modo seja possível. Ou seja, não incluímos na modelagem o fork e o PR.

A data das duas primeiras ações é registrada para fins de notificação na timeline.

## Estruturação dos Repositórios

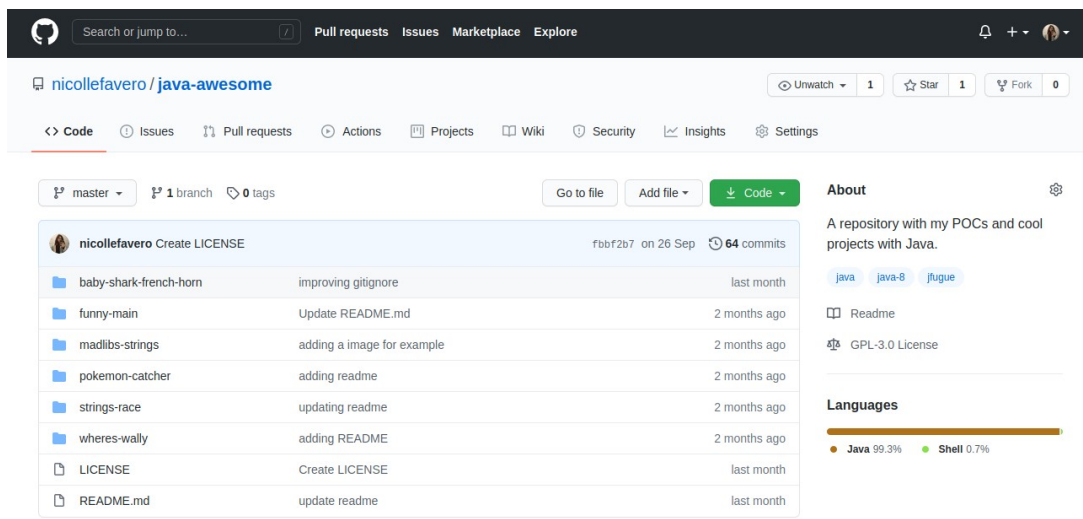


Figura 7: repository

Um repositório tem nome e data de criação e pode ter uma descrição (opcional). O nome de um repositório deve ser único dentre os repositórios de um usuário. Além dos seus relacionamentos com usuários, um repositório pode fazer parte de inúmeros tópicos que

são caracterizados por um nome. Um repositório também pode ter uma licença previamente definida pelo GitHub.



Figura 8: repository's description, topics and license

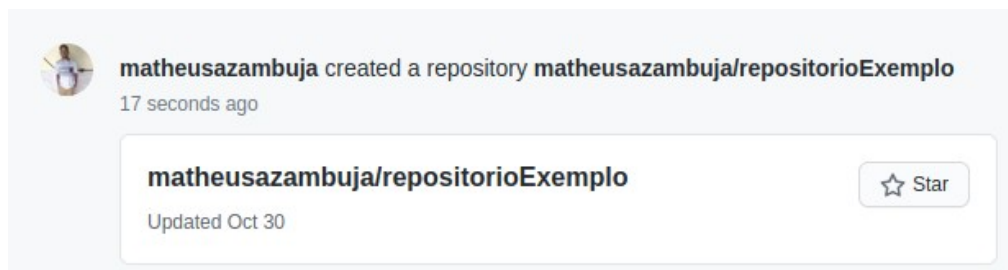
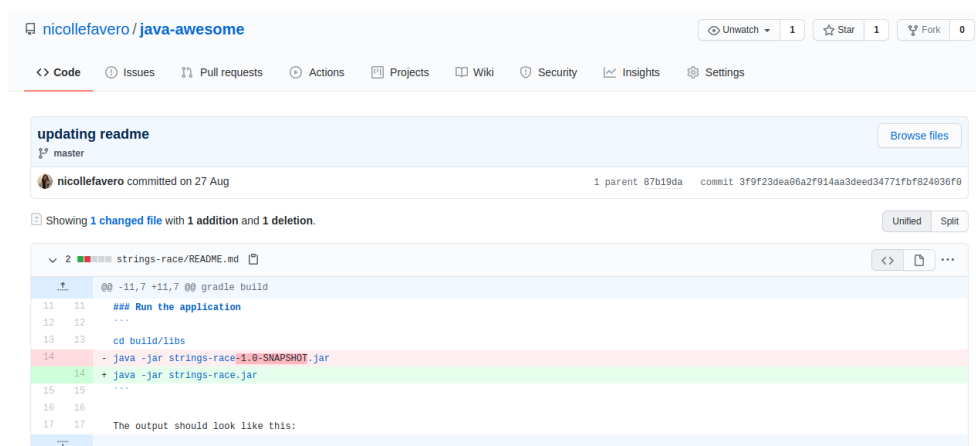


Figura 9: repository's created notification

Ainda, em um repositório podem ser commitados itens por usuários. Esses itens podem ser do tipo pasta ou do tipo arquivos. Se um item for do tipo pasta, ele terá um nome e deverá conter algum outro item (só é possível commitar pastas se elas tiverem arquivos dentro ou outras pastas que tenham arquivos dentro). Na nossa modelagem, não é possível que dois ou mais itens tenham o mesmo nome em um repositório.



Se esse item for do tipo arquivo, ele terá um nome, um conteúdo e uma terminação (a terminação indica que tipo de arquivo ele é).

|                          |            |  |                                          |
|--------------------------|------------|--|------------------------------------------|
| nicollefaver0 main class |            |  | 9b6f39c on 7 Aug <a href="#">History</a> |
| ..                       |            |  |                                          |
| Racer.java               | wip        |  | 3 months ago                             |
| StringBufferRacer.java   | wip        |  | 3 months ago                             |
| StringBulderRacer.java   | wip        |  | 3 months ago                             |
| StringRacer.java         | wip        |  | 3 months ago                             |
| StringsRaceApp.java      | main class |  | 3 months ago                             |

*Figura 11: file type java*

Um item do tipo arquivo pode estar classificado a uma ou nenhuma linguagem de programação previamente definida. Essa informação é armazenada pelo GitHub para calcular a porcentagem de linguagens de programação que o repositório tem.



*Figura 12: repository's programming languages*