Visualização de Dados
TP2 – Visualizações Interativas de Dados

Gabriel Pereira de Oliveira

Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

gabrielpoliveira@dcc.ufmg.br

1. Introdução

Com a expansão da Internet e o acesso a serviços de banda larga, pessoas comuns têm à disposição uma gama de aplicativos e serviços online. Assim, grandes volumes de dados são gerados diariamente, podendo ser processados com os mais variados propósitos. Entre tantas opções, sem dúvidas um dos serviços mais populares é o de redes sociais online, que reunem milhões de pessoas com objetivos semelhantes ao redor do mundo.

Tais redes não são apenas para diversão e conexões pessoais, mas também para trabalho e conexões profissionais. Por exemplo, o GitHub [1] é uma das maiores plataformas online de desenvolvimento colaborativo de software do mundo, reunindo uma comunidade de mais de 28 milhões de desenvolvedores reunidos em mais de 85 milhões de projetos em 337 linguagens de programação (Junho de 2018)[2]. Dessa forma, devido ao grande volume de dados, técnicas de visualização são muito importantes para trazer à tona informações e análises relevantes sobre a colaboração de software no GitHub, que inicialmente não poderiam ser percebidas olhando apenas para os dados puros.

Neste trabalho, será explorada a evolução da criação de repositórios de *software* no GitHub, classificados de acordo com sua linguagem de programação. A partir das visualizações desenvolvidas, espera-se obter informações e análises valiosas sobre a evolução das linguagens de programação ao longo do tempo e também da própria plataforma, relacionando pontos de crescimento (ou descrescimento) expressivo com fatores (internos e externos) que possam ter influenciado tal fato.

2. Idealização e Desenvolvimento

2.1. Base de Dados

Para desenvolver este trabalho, foi utilizada a base de dados do **GHTorrent**[3], um serviço de coleta de dados públicos do GitHub que tem por objetivo facilitar o acesso aos dados oferecidos pela API oficial por desenvolvedores e pesquisadores. Em linhas gerais, esse serviço monitora os eventos lançados pela API oficial, coleta informações adicionais necessárias e os disponibiliza diretamente (i.e., no mesmo formato recebido pela API) ou de forma estruturada (i.e., em um banco de dados relacional). Com isso, tem-se acesso a informações de usuários, **repositórios** e suas respectivas atividades dentro do GitHub, como *commits* e *pull requests*.

Os dados foram modelados em um banco de dados relacional (MySQL) e são oferecidos no formato .csv para download. A cada mês, um novo *dump* é disponibilizado

[1] GitHub: `http://github.com`

[2] `https://github.com/about`

[3] GHTorrent: `http://ghtorrent.org/`

pelo GHTorrent. Neste trabalho em questão foi utilizada o *dump* de 01/05/2017. Para visualizar os dados de criação dos repositórios, vamos trabalhar com a tabela ***projects*** do banco, que contém os seguintes campos:

- **id:** campo numérico único que identifica o repositório;
- **url:** endereço da API onde os dados do repositório foram coletados;
- **owner_id:** campo numérico que referencia o usuário dono do repositório;
- **name:** nome do repositório;
- **description:** descrição do repositório;
- **language:** principal linguagem de programação do repositório;
- **created_at:** *timestamp* da criação do repositório ;
- **forked_from:** indica se o repositório veio de outro;
- **deleted:** indica se o repositório foi deletado ou transformado em privado;
- **updated_at:** *timestamp* da última atualização do repositório.

2.2. Escolha da visualização

A partir do conjunto de dados e da especificação deste trabalho, optou-se por escolher uma visualização que respondesse algumas perguntas importantes, a saber:

1. Como o GitHub cresceu ao longo do tempo?
2. O que fez o GitHub se popularizar?
3. Quais linguagens tiveram maior crescimento? E quais estão em decadência?

Todas essas perguntas remetem à ideia de uma evolução ao longo do tempo, e portanto o que se deve fazer é analisar uma **série temporal**. Várias visualizações permitem fazer tais análises, mas como queremos identificar tendências, optou-se pela construção de um **gráfico de linhas e pontos**. Este tipo de gráfico permite a comparação do crescimento individual de cada linguagem no GitHub através dos anos, sendo fácil comparar as linguageus umas com as outras. Isso é importante quando queremos verificar a popularidade ao longo do tempo de linguagens que competem entre si e também de linguagens que substituíram outras no mercado. Além disso, é possível verificar o crescimento dos repositórios no GitHub como um todo, como forma de atestar a sua popularização.

Dessa forma, a visualização construída apresenta uma lista de linguagens de programação a serem selecionadas, e à medida em que cada uma é escolhida, uma linha com pontos é plotada, mostrando a evolução da criação de repositórios naquela linguagem entre os anos de 2008 e 2016. O interessante é que podemos comparar o crescimento de duas ou mais linguagens, bastando selecioná-las para que as linhas sejam carregadas no gráfico. Em cada ano, pode-se visualizar o exato número de repositórios criados. A escala do eixo Y se ajusta de acordo com as linguagens selecionadas, facilitando a comparação por parte do usuário. A Figura 1 mostra um exemplo de visualização oferecida.

2.3. Interatividade

Para cumprir o requisito de oferecer interatividade ao usuário, construiu-se a visualização levando em consideração algumas técnicas de interação analítica, todos elas vistas em sala de aula:

- **Comparação:** ao se plotar duas ou mais curvas de evolução das linguagens, é possível efetuar uma comparação de seu crescimento numérico ao longo dos anos.

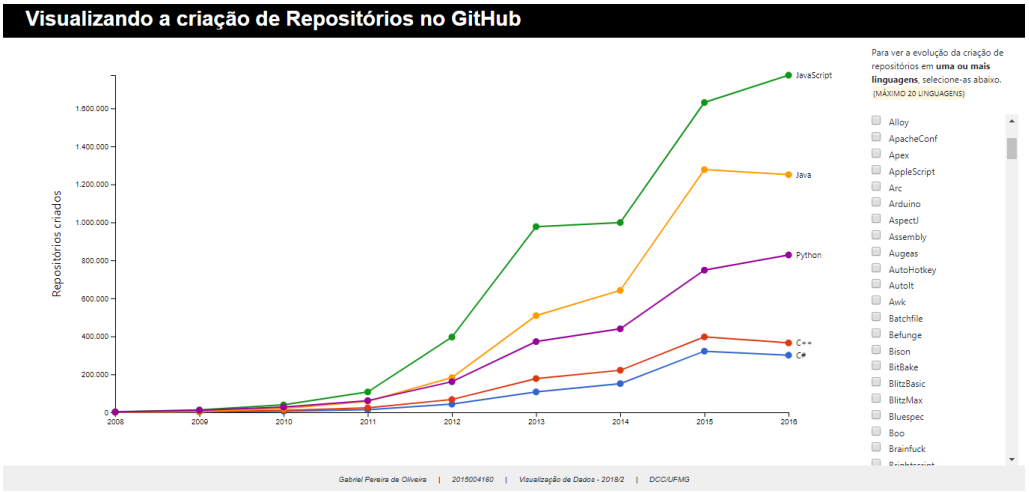

Figura 1: Exemplo de visualização mostrando o crescimento da criação de repositórios de cinco linguagens de programação de 2008 a 2016.

- **Adição de variáveis:** ao se clicar nas *checkboxes* no canto direito da página, a curva de crescimento da referida linguagem é automaticamente adicionada.
- **Filtragem:** é possível mostrar apenas as linguagens desejadas/selecionadas.
- **Mudança de escala:** a escala do eixo Y é atualizada automaticamente de acordo com as linguagens selecionadas.
- **Detalhe sob demanda:** para cada linguagem, a cada ano é possível visualizar o número exato de repositórios criados naquele ano através de uma *tooltip box* (caixa informativa).

2.4. Ferramentas utilizadas

Tendo definida a visualização a ser feita, foi definido que a ferramenta principal para a visualização dos dados seria a biblioteca **D3.js**, conforme sugerida pela especificação. Tal biblioteca possui uma maior variedade de ferramentas e conteúdos a serem utilizados, apesar da maior dificuldade em se aprender seus conceitos.

Além disso, foi utilizado o **Bootstrap**, um conjunto de estilos CSS para harmonizar o conteúdo da página Web desenvolvida. Assim, além da visualização, tem-se uma página completa e funcional, oferecendo ao usuário a opção o gerar o gráfico com as linguagens selecionadas. Assim, pode-se ter uma visão mais detalhada, permitindo análises mais profundas sobre os dados. O código-fonte[4] e a página Web[5] com as visualizações estão disponíveis online.

2.5. Boas práticas

As boas práticas e conceitos vistos em sala de aula guiaram o desenvolvimento da visualização escolhida. Alguns pontos importantes observados foram:

- Escolha de cores que fossem categóricas, para distinguir bem cada linguagem. O site *ColorBrewer* foi utilizado como norte;

[4]Repositório: `https://github.com/opgabriel/github-repositories`
[5]Visualização: `https://opgabriel.github.io/github-repositories/`

- A escolha da proporção 2:1 (largura x altura) para o gráfico, que otimiza a leitura das informações para o usuário;
- Espessuras diferentes para linhas e pontos (no caso da linha cumulativa);
- A interação com o usuário é dinâmica, através da seleção de linguagens e mostrando *tooltip boxes* para informar ao usuário dados exatos, auxiliando na compreensão dos dados;
- A utilização de gráficos de linhas e pontos para comparar verificar tendências em séries temporais.

3. Análises

Feita a página Web com as visualizações, pode-se fazer análises interessantes e confirmar tendências já imaginadas no contexto do GitHub, como por exemplo:

Aumento no número de repositórios criados em todas as linguagens. De uma forma geral, todas as linguagens tiveram um aumento significativo no número de repositórios criados. Isso se deve à popularização do GitHub enquanto plataforma de desenvolvimento colaborativo e também da difusão do processo colaborativo de software e do Git. Vale lembrar que o GitHub foi criado em 2007 e em 2016 já era o ambiente mais popular de compartilhamento online de software no mundo.

HTML passou a crescer mais do que PHP. A linguagem HTML sempre esteve presente no desenvolvimento de páginas Web. No entanto, durante boa parte da década de 2000, a linguagem PHP era a preferida dos desenvolvedores para criar páginas. No entanto, a linguagem HTML saltou de 54 mil repositórios criados em 2014 para mais de 736 mil em 2015, enquanto PHP teve somente 560 mil repositórios criados no mesmo ano. Isso pode ser explicado pelo fato de, em outubro de 2014, o HTML5 foi apresentado, oferecendo uma série de novas possibilidades e funcionalidades, que foram logo adotadas pelos desenvolvedores.

Os gráficos refletem linguagens populares e em declínio. A partir de listas contendo as linguagens mais populares[6] e as linguagens que estão em "declínio"[7], observou-se que as estatísticas de criação de repositórios nessas linguagens refletem tais classificações. As linguagens populares são as que mais crescem em número e as que estão "morrendo"possuem cada vez menos repositórios criados.

4. Conclusão

O GitHub é sem dúvidas uma excelente plataforma de colaboração de software, oferecendo um grande volume de dados para serem analisados. As curvas de crescimento da criação de repositórios acompanham a popularização das linguagens e da plataforma.

Em relação ao processo de construção das visualizações, foram encontradas algumas dificuldades, principalmente com a biblioteca D3.js, principalmente ao fazer a mudança de escala nos gráficos. Felizmente, tais dificuldades foram superadas e tudo o que foi proposto conseguiu ser executado. Pretende-se utilizar essa visualização no projeto final da disciplina, adicionando novas *features*, como a adição de um campo de pesquisa para linguagens, um botão para limpar a seleção das linguagens e a adição de responsividade na página.

[6]Most Popular and Influential Programming Languages of 2018: `https://bit.ly/2PvQtCJ`

[7]Is there any programming language that is dying? If yes, why? `https://bit.ly/2Jsgj55`