

RELATÓRIO FINAL - Ecossistemas de Software Livre - Setembro 2019

Carla Silva Rocha Aguiar (Coordenadora do Projeto)

15 de Setembro de 2019

Objetivo Geral

O objetivo geral da parceria foi constituir a rede de laboratórios de pesquisa e desenvolvimento de tecnologias inovadoras para políticas públicas do Ministério da Cultura. O principal objetivo é pesquisar e aplicar técnicas, metodologias de desenvolvimento de software, além de aferição qualidade produto de software, em ambiente experimental do Laboratório Avançado de Pesquisa, Produção e Inovação em Software (LAPPIS). Tais pesquisas e práticas serão usadas para subsidiar o Ministério da Cultura de ferramentas de gestão e desenvolvimento de software colaborativo, aberto e contínuo, em diferentes arranjos produtivos, aprimorando os mecanismos de governança digital; além de fornecer subsídios tecnológicos que apoiem a execução da Lei 8.313/91, conhecida como Rouanet e das demais políticas de fomento e incentivo à cultura.

O presente relatório apresenta o acompanhamento do trabalho realizado no projeto “Ecossistemas de Software Livre”, Termo de Cooperação para Descentralização de Crédito, Processo Ofício No 0646/2017/FUB-UnB, Vigência Outubro 2017 à Outubro 2019. O relatório apresentado é relatório final do projeto e vai apresentar o que foi realizado no âmbito do projeto, os principais resultados, e entregas, a partir das expectativas do plano de trabalho vigente.

Objetivos Específicos

No plano de trabalho foram levantados alguns objetivos específicos referentes à parceria. Ao longo do relatório de entrega final serão detalhadas como esses objetivos foram alcançados. Abaixo elencamos esses objetivos e resumimos como eles foram realizados, e por qual pacote de trabalho:

- **Realizar estudos de algoritmos de aprendizado de máquina para analisar dados da execução da Lei Rouanet:** O projeto “Salic-ML” realizou esse objetivo. Além de inúmeras reuniões com a equipe da Sefic, para entender o processo da Lei de Incentivo a Cultura e o SALIC, foram levantados os principais gargalos do processo. O gargalo mais crítico foi a prestação de contas, no qual o Ministério tem um passivo de cerca de 17 mil projetos a serem analisados, incluindo o objeto e a prestação de contas. O time então priorizou em desenvolver algoritmos de ciência de dados para gerar métricas/indicadores que explicitem a **complexidade** de um projeto cultural, no que tange a análise financeira. O objetivo é direcionar o trabalho dos técnicos da Sefic no trabalho técnico, e aos gestores a priorizar e delegar as análises de forma eficiente e orientadas a dados. Esse trabalho foi finalizado e entregue como um microsserviço a ser integrado ao SALIC.
- **Realizar estudos de métodos/práticas ágeis e de desenvolvimento lean de software, além das práticas de engenharia de software e de governança utilizadas nas comunidades de software livre, de forma a prover uma infraestrutura computacional para desenvolvimento e experimentação contínua de software:** Esse objetivo foi alcançado com as reuniões estratégias trimestrais com os principais stakeholders do ministério, além de manter as boas práticas de comunidades de software livre nos projetos desenvolvidos ao longo da

cooperação. Um grande trabalho de governança foi feito com a chatbot “Tais”, engajando outros ministérios a adotarem a solução técnica da Tais, aumentando assim a comunidade.

- **Fornecer suporte tecnológico para apropriação das informações por parte da sociedade civil de maneira a contribuir para transparência pública e participação social:** Todos os estudos, código, apresentações, material de treinamento foram disponibilizados com licenças livres, de modo que a sociedade civil possa acessar, dar feedback, contribuir, etc.
- **Fornecer suporte tecnológico para estimular a participação da sociedade civil na governança digital em torno das tecnologias livres do portfólio do ministério:** Esse objetivo foi alcançado ao seguir as boas práticas e documentações adotadas pela comunidade de software livre. Dessa maneira, diminui a barreira de contribuição para os interessados em contribuir com os projetos desenvolvidos ao longo da parceria.
- **Mineração em repositórios de software para extração e análise de dados:** Esse objetivo foi alcançado nos notebooks de análise de dados. Isso foi feito tanto no contexto da Lei de Incentivo Cultural, por meio da mineração do banco de dados do Salic, quanto da análise dos próprios repositórios e métricas dos projetos desenvolvidos ao longo do parceria.
- **Processamento de linguagem natural dos dados extraídos dos diferentes sistemas de software culturais:** Esse objetivo foi alcançado por meio da assistente virtual TAIS. Nela, aplicamos algoritmos de **Natural Language Understanding (NLU)** para a classificação da intenção dos usuários que interagem com a chatbot por meio de linguagem natural.
- **Transferência de conhecimento da academia para o Estado:** Além dos diversos workshops realizados ao longo do projeto com a equipe de TI do Ministério da Cidadania, foram realizados pela frente de governanças do projeto, diversas interações com comunidades de software livre e outros ministérios.
- **Formação de alunos de graduação em pós-graduação em projetos com problemas reais do contexto cultural:** Esse objetivo foi alcançado com grande parte da equipe do projetos sendo alunos de graduação de Engenharia de Software, design e letras. Ao total, 43 alunos de graduação foram bolsistas do projeto.
- **Contribuir para o fomento da cultura de software livre na Administração Pública Federal:** Esse objetivo foi alcançado;
- **Contribuir para o desenvolvimento da cultura de tomada de decisões orientadas a dados e evidência:** Esse objetivo foi alcançado com o projeto Salic-ML no qual estimamos métricas e indicadores de complexidade de projetos de projetos culturais. Na TAIS, criamos um dashboard de BI que fornece, em tempo real, o comportamento dos usuários que interagem com a TAIS. Esse dashboard disponibiliza dados preciosos para compreender o perfil do usuário que acessa a o portal da Lei de Incentivo, e o comportamento desse usuário;
- **Contribuir para o estabelecimento da cultura de desenvolvimento e experimentação contínua:** Alcançamos esse objetivo pela forma como escolhemos as ferramentas, técnicas, e escopos a serem desenvolvidos, sempre realizados após um período de experimentação. Esses aprendizado e amadurecimento de experimentação contínua foi registrado nas wikis e notebooks disponibilizados em licenças livres nos repositórios dos projetos, no qual estudos comparativos precedem escolhas técnicas.

Metodologia, metas e etapas do projeto

O planejamento de execução das metas foi inicialmente dividido em 5 etapas e 5 pacotes de trabalho. Descrevemos os avanços a seguir e ressaltamos alguns itens que foram executados de forma diferente do cronograma. Vale ressaltar que todas as alterações ao longo da execução do projeto que divergem do planejado foi previamente acordada com o Ministério, e registrado nos relatórios de acompanhamento entregues trimestralmente.

Inovação tecnológica em Software

Os pacotes de trabalho da presente parceria acarretou em X projetos, distribuídos em X repositórios. Ao todo, foram X arquivos distintos, X commits com contribuições de X desenvolvedores distintos, X deles vinculados como bolsistas do projeto, correspondendo a X% dos commits realizados no projeto. Além disso, o projeto ainda contempla estudos acadêmicos, estudos de identidade visual, dinâmicas de capacitação e outros insumos fora do produto de software propriamente dito.

Obviamente é impraticável anexar toda esta produção à este documento, portanto este documento apresenta um resumo com estatísticas e os resultados mais importantes obtidos e direções sobre como obter o código para fazer uma análise detalhada. Todo software desenvolvido no projeto está disponível publicamente sob licenças livres na plataforma Github (principal plataforma de disponibilização de código no mundo). Os repositórios e seus respectivos endereços estão listados abaixo e podem ser acessados publicamente por qualquer pessoa:

- *Ecosistemas de Software Livre* (<https://github.com/lappis-unb/EcosistemasSWLivre>) - todos os documentos administrativos do projeto estão disponibilizados nesse repositório. Registro de reuniões, decisões, relatórios de acompanhamento, plano de trabalho.
- *TAIS* (<https://github.com/lappis-unb/tais>) - aplicação principal da TAIS, com o bot, interface web, dashboard de BI.
- *Salic-ML* (<https://github.com/lappis-unb/salic-ml>) - aplicação principal do microserviço SALIC-ML. Nesse se encontra tanto a API do microserviço, quanto toda a documentação técnica na Wiki, e estudos realizados nos notebooks.
- *Promova Cultura* (<https://github.com/lappis-unb/PromovaCultura>) - estudo de visualização de dados dos dados do banco de dados do Salic. Contém toda a documentação técnica, resultados da design sprint realizadas para definição do escopo e as visualizações implementadas.
- *Botflow* (<https://github.com/lappis-unb/BotFlow> e <https://github.com/lappis-unb/BotFlowAPI>) - não previsto no plano de trabalho, mas foi julgado pelo time necessário para facilitar a manutenção e evolução da base de conhecimento da TAIS.
- *Salic-API* (<https://github.com/lappis-unb/salic-api>) - refatoramos, criamos testes unitários, pipeline de integração contínua, documentação técnica. Ou seja, adequamos a API do Salic, que é o projeto do ministério de maior interesse externo, para os padrões, boas práticas e documentação de comunidades de software livre.

Existem metodologias na área de Engenharia de Software para estimar o esforço de desenvolvimento de um produto. É importante notar que estes modelos são apenas aproximados, mas ajudam a estabelecer uma ordem de grandeza do esforço de desenvolvimento de um software em comparação com projetos semelhantes encontrados na indústria.

Analizamos o projeto da ótica do modelo COCOMO (Constructive Cost Model) desenvolvido por Berry W. Boehm a partir da análise empírica da produtividade de equipes de desenvolvimento em vários projetos de software reais. Os dados foram extraídos utilizando as ferramentas sloccount (para código Python) e cloc (para as linguagens). A tabela abaixo resume os principais resultados:

XX

Podemos estimar o custo de desenvolvimento a partir do número de horas necessários para desenvolver o projeto, o custo médio da hora trabalhada de cada desenvolvedor e o parâmetro de sobrecarga (do inglês overhead), que estima o esforço adicional devido à reescrita de código e realização de outras atividades como reuniões, planejamento, documentação, estudo, etc que não estão diretamente relacionadas à produção de código. A ferramenta sloccount considera como valores padrão um parâmetro de overhead de 2,4 e um salário anualizado de U\$56.286,00. *Utilizando – se estes valores, obtemos uma estimativa de UX* ou cerca de X milhões de reais. Obviamente a realidade brasileira é diferente. Se tomarmos o salário médio de um desenvolvedor como sendo R\$6.400,00, *esta custocairi para RX* (sem custos trabalhistas), o que ainda assim demonstra uma grande eficiência econômica na execução do projeto.

Nota-se ainda que o projeto abrange atividades de pesquisa, formação de alunos treinamento e desenvolvimento de identidade visual que estão fora do escopo da programação que em um acordo comercial aumentariam ainda mais o custo do projeto.

Finalmente, além do desenvolvimento de uma solução para o Ministério da Cidadania, o projeto parte de uma lógica de parceria com a universidade e pressupõe a criação de insumos de pesquisa e produção acadêmica. A maior parte dos membros da equipe é formada por alunos do curso de Engenharia de Software, sendo que a participação no projeto contribuiu diretamente para a formação dos mesmos. Alguns destes alunos adotaram temas relacionados ao projeto como tema em seus trabalhos de conclusão de curso ainda em andamento. Foram publicados dois artigos científicos, um deles, com o título “FLOSS FAQ chatbot project reuse - how to allow nonexperts to develop a chatbot” que contém a experiência no desenvolvimento da Tais, este trabalho foi apresentado no OpenSym, um dos maiores simposios de pesquisa acadêmica em software livre. O segundo, com o título “A Survey of DevOps Concepts and Challenges” foi aceito na revista **Computing Surveys** da ACM, de alto impacto acadêmico (A1 no critério de Qualis da Capes) e mostra o resultado acadêmico dos estudos em DevOps. Foi publicado também um capítulo do livro “Software e Cultura no Brasil - Produção, gestão e políticas públicas”, no qual discutimos os modelos de contratação nas equipes de TI do governo federal, no capítulo “Colaboração aberta e sua relação com a contratação de software na administração pública”.

Pacote de Trabalho: Estratégia/modelo de transformação de softwares legados em comunidades de software aberto

Evoluir e manter um software legado é uma experiência desgastante para desenvolvedores e desestimulantes no contexto de fomento a comunidades. Por outro lado, a reescrita desses softwares é impraticável e, em se tratando de software implantado, a necessidade de adicionar novas funcionalidades e dar manutenção persiste.

Metas Específicas

Quanto as metas específicas dessa frente de trabalho definidas plano de trabalho são:

1. **Estudos e documentação do processo de containerização, testes automatizados, refatoração de sistemas legados em uma estrutura de DevOps para viabilizar trabalhos futuros**

Concluído. No início do projeto, fizemos um checklist com as principais boas práticas, documentações, automações de comunidades de software livre. Esse checklist foi aplicado em uma lista de projetos evoluídos e mantidos pelo Ministério da Cultura. Para cada projeto, colocamos a solução em containers, documentamos o básico (README), instrumentamos com serviços de análise estática de qualidade de código, e integração contínua. A importância desse trabalho é que, além de fazer com que os projetos mantidos pelo ministério façam adesão às práticas modernas de engenharia de software, permite que outras equipes de desenvolvimento possam executar, testar e evoluir os projetos mais facilmente. Adicionalmente, tais ferramentas aceleram a curva de aprendizado de novos membros no time, além de incentivar boas práticas de desenvolvimento por meio de monitoramento das métricas.

Essa estratégia, chamada *legacy in the box*, é sugerida na literatura do estado da prática como primeira ação a ser feita em direção ao DevOps em sistemas legados. Containerizar e instrumentar um software legado permite que esse seja configurado em um pipeline de deploy contínuo. A falta de testes cria a vulnerabilidade de se colocar em ambiente de produção/homologação features defeituosos. Porém esse risco está presente naturalmente em softwares legados sem testes. Logo, a estratégia *legacy in the box* agiliza a entrega de novas features, possibilitando entrega contínua, criação de comunidade, e incentivo as boas práticas de engenharia de software.

Documentação comprobatória

2. **Pesquisa em metodologias de refatoração de sistemas legados**

Concluído. O principal problema tratado foi a pesquisa de estratégias de fazer inovação em plataformas compostas por software legado. Utilizamos o SALIC, principal software mantido pelo antigo Ministério da Cultura, que além de ser o maior software ainda é o software que executa a Lei de Incentivo à Cultura. Nesse contexto, refatorar e/ou reescrever o Salic é uma tarefa inviável com custos proibitivos. Uma particularidade do Salic é a quantidade de bancos de dados (cerca de 10 bancos), e o fato de que várias regras de negócios estão do próprio banco. A documentação técnica no início do projeto era mínima, quase inexistente.

Dado o contexto, além da estratégia *legacy in the box*, descrita na seção acima, pesquisamos e aplicamos outras duas técnicas de refatoração de sistemas legados. Identificamos o *SalicAPI*, projeto que disponibiliza os dados sobre a execução de projetos da Lei de incentivo, como sendo o software com o maior potencial de se ter comunidade, uma vez que os dados acessados via o *SalicAPI* são de interesse tanto da sociedade civil quanto de jornalistas. Para o o *SalicAPI* aplicamos a técnica tradicional de refatoração orientados a métricas. Para isso, atualizamos a versão do *Python*, automatizamos a execução de testes automatizados, dockerizamos, fizemos toda a documentação técnica e dos *endpoints*. Essa técnica de refatoração (reescrita de código, melhoria das práticas, execução de testes unitários) modernizou o *SalicAPI* e permitiu um pipeline de entrega/deploy contínuo seguro.

A terceira técnica de refatoração de sistemas legados foi no contexto de inserir novas *features*. Mais especificamente, novas *features* com inovação em funcionalidades que fazem o processamento de dados com algoritmos de *machine learning*. Como o código do SALIC é PHP, e maioria dos frameworks e bibliotecas de aprendizagem de máquina são desenvolvidos na linguagem *python*. Por isso, escolhemos a técnica de adotar uma arquitetura microsserviços, no qual novas funcionalidade são adicionadas à plataforma como novos microsserviços que compartilham o banco de dados com o software legado. Essa técnica foi colocada em prática com o serviço “*SaliML*”. Nele, construímos uma API no qual adicionamos várias métricas de complexidade de análise de projetos culturais.

Documentação comprobatória

- [Relatório Etapa 1](#)
- [Relatório Etapa 2](#)
- [Relatório Etapa 3](#)
- [Relatório Etapa 4](#)
- [Relatório Etapa 7](#)

3. Utilizar como estudo de casos alguns sistemas legados do Ministério da Cultura, tais como o projeto SIMEC (Sistema Integrado de Monitoramento Execução e Controle) e o projeto Salic (Sistema de Apoio às Leis de Incentivo à Cultura), Sistel

Concluído. Dos três projetos citados, trabalhamos com o *SALIC* e o *Sistel*. No *Sistel* aplicamos a técnica de *legacy in a box*. Já no *SALIC*, aplicamos as três técnicas pesquisadas: *legacy in a box*, *refatoração de código orientado a métricas* e *arquitetura microsserviços*. Toda a documentação técnica, assim como os resultados técnicos obtidos, e decisões colaborativas com o Ministério estão disponibilizadas na wiki do respectivo repositório.

Documentação comprobatória

Pacote de Trabalho: Estudo sobre catálogos de Software Culturais

De acordo com o plano de trabalho, "O foco dessa etapa é executar o ciclo de projeto de software completo, desde a iniciação. Assim, o projeto já será iniciado como software livre e com as práticas de DevOps, ferramentas e tecnologias modernas. Será focado o levantamento das tecnologias e ferramentas usadas pela comunidade de software livre para automatizar o processo de desenvolvimento e implantação do software, pois há pouca pesquisa focada nesse tema. O principal objetivo nessa etapa é exercitar em todo ciclo de projeto a experimentação e inovação contínua, de forma a subsidiar a pesquisa realizada na Etapa 5.

Metas Específicas

1. Aplicação de práticas de experimentação e inovação contínua no desenvolvimento do projeto de Catálogo de Software Culturais

Cancelado. Durante a execução do projeto, percebemos que os objetivos desse pacote de trabalho estavam sendo realizados no desenvolvimento da TAIS e do SalicML. Adicionalmente, catálogo de softwares são apresentados atualmente como gitpages dentro das próprias organizações, otimizando a manutenção e evolução. Dessa forma, em conjunto acordo com o Ministério, essa frente foi extinta. Tal decisão foi documentada no relatório de entrega 2 (2 de maio de 2018), página 04, que foi devidamente aprovada pela equipe da gestão do projeto tanto na Unb quanto no Ministério.

2. Realizar estudos e documentação do processo de desenvolvimento e das boas práticas e automações realizadas

Concluído - O projeto foi dividido em ciclos de 3 meses, no qual a cada ciclo era realizado um planejamento estratégico com a equipe estratégica do Ministério da Cidadania (tanto a equipe de coordenação quanto de negócio). Nessa reunião de planejamento estratégico, amadurecido ao longo do projeto, era definido as metas estratégicas para o próximo ciclo, além das épicas priorizadas. Essas reuniões foram documentadas na wiki do repositório ou em apresentações com o alinhamento. Então, a equipe se reunia para criar as issues, histórias de usuário e tarefas a serem desenvolvidas. Tais tarefas estão todas documentadas em forma de issue, no repositório de cada repositório. Utilizamos *sprints* de duas semanas para planejar e revisar o desenvolvimento. A cada mês, a equipe técnica/gestão da TI do ministério era convidado para visitar o laboratório e acompanhar a evolução das frentes.

Em relação as automações realizadas, elas foram todas mapeadas nas *issues* e toda documentação técnica necessária para evoluir e manter as soluções.

Como o desenvolvimento do catálogo de software foi cancelado, tanto as práticas quanto métodos e automações, além da documentação foi usada nos outros projetos desenvolvidos, como o projeto da Tais e SalicML.

Documentação comprobatória

- [Relatório Etapa 1](#)
- [Relatório Etapa 2](#)
- [Relatório Etapa 3](#)
- [Relatório Etapa 4](#)
- [Relatório Etapa 5](#)
- [Relatório Etapa 6](#)
- [Relatório Etapa 7](#)

3. Relatório com os modelos de desenvolvimento e comunidade para serem aplicados aos projetos de software do Minc

Concluído - O principal modelo defendido pela comunidade de software livre no desenvolvimento de projeto é o Bazar. No projeto, aplicam os

Documentação comprobatória (mestrado Poppi) +

4. Transferência de conhecimento e capacitar a equipe de servidores e técnicos do MinC em práticas de gestão e desenvolvimento de software aberto, colaborativo e contínuo

Concluído - Durante toda a execução do projeto, tivemos a preocupação em manter a documentação técnica atualizada para deixar a comunidade dos projetos desenvolvidas receptivas. Além disso, realizamos ao longo do projeto diversos workshops de Devops, chatbots, ML, boas práticas, com a equipe técnica do ministério treinamentos. Ao amadurecer tecnicamente conhecimento sobre chatbots e a arquitetura da Tais, realizamos diversos webinars que estão disponibilizados no youtube, que abordam diversos aspectos técnicos importantes para a manutenção e evolução da Tais. Participamos

também de 2 edições do pydata gravadas no qual apresentamos a visão geral da Tais e do SalicML. Tais materiais estão organizados e disponibilizados de forma que pessoas interessadas em manter e evoluir os projetos desenvolvidos no projeto terão todo o apoio técnico necessário.

Documentação comprobatória

- [Relatório Etapa 2](#)
- [Relatório Etapa 5](#)
- [Relatório Etapa 6](#)
- [Relatório Etapa 7](#)

Pacote de Trabalho: Estudos sobre práticas de gestão colaborativa em comunidades de software aberto

De acordo com o plano de trabalho, “O principal resultado dessa pesquisa será sistematizar e produzir conhecimento sobre as práticas das comunidades de software livre que o Estado participa por adesão e, a partir dos aprendizados com seus arranjos, orientar e capacitar os servidores e técnicos do MinC nas práticas de planejamento, gestão de softwares abertos, aprimorando os mecanismos de governança digital dos softwares presentes no portfólio do MinC”.

Metas Específicas

1. Estudos de caso sobre comunidades de software livre onde o Estado participa por adesão, com prioridade para os softwares utilizados para a gestão cultural

Concluído

Documentação comprobatória -

2. Estudos sobre boas práticas para planejamento conjunto de milestones e releases entre as organizações que fazem parte das comunidades

Concluído -

Documentação comprobatória -

3. Estudos sobre boas práticas de comunicação e mobilização no contexto das comunidades onde o Estado participa

Concluído -

Documentação comprobatória -

4. Participação em eventos e encontros das comunidades de software livre que contribuem para o portfólio mantido pelo MinC

Concluído -

Documentação comprobatória -

5. Estudos sobre arranjos econômicos utilizados pelas comunidades com fins de sustentabilidade de seus comuns de software

Concluído -

Documentação comprobatória -

6. Estudos sobre metodologias e suportes tecnológicos para a gestão colaborativa em comunidades de software livre nas quais o Estado participa por adesão

Concluído -

Documentação comprobatória -

Pacote de Trabalho: Estudo de técnicas de Aprendizado de Máquina para apoiar a fiscalização da Lei Rouanet

De acordo com o plano de trabalho, “O principal objetivo é o estudo de técnicas de Aprendizado de Máquina que possam apoiar o sistema de recomendação e fiscalização da lei Rouanet. Nessa etapa será realizada uma pesquisa exploratória em técnicas de aprendizado de máquina e processamento de linguagem natural. Tais técnicas e algoritmos serão aplicados para melhorar a experiência de usuário (UX) por meio da proposta de chatbots como interface entre os proponentes na lei Rouanet e o Ministério da Cultura”.

Metas específicas

1. **Realizar estudos e propor técnicas de processamento de linguagem natural, aprendizado supervisionado e o desenvolvimento de chatbots para interagir com proponentes da Lei Rouanet**

Concluído: Principal pesquisa do projeto TAIS. Durante o projeto foram testados diversos frameworks de chatbots, dentre eles o *hubot natural*, *botpress*, e *rasa*. O objetivo era pesquisar os algoritmos, dados de treinamento e performance de algoritmos de processamento de linguagem natural em português. Pela novidade da tecnologia, e das ferramentas, existia poucos ou nenhum experimento executado em português.

Após diversos experimentos, testes (a primeira versão da Tais utilizamos o *hubot natural* com o classificador X), escolhemos o Rasa, pois além de crescente comunidade, possuía a vantagem de trazer a possibilidade de customizar o pipeline de processamento e algoritmos. Além disso, já estava implementado algoritmos modernos de processamento de linguagem natural, tais como *Embeddings*, *LSTM*, entre outros.

Testamos diversas técnicas de classificadores de intenção do usuário: Supervised Vector Machine (SVM), Embeddings. Após diversos experimentos observamos que o embeddings classificava melhor com poucos dados de treinamento, que é o caso do chatbot em português que tem poucos dados para treinamento. Além de testes de algoritmos de *Natural Language Processing* (NLU), um chatbot tem que prever a ação mais adequada a partir da intenção do usuário e o contexto da conversa. Com isso, tem-se uma segunda etapa de processamento de chatbots que é a gestão de conversa. Nessa etapa, testamos duas formas de fazer a gestão de conversa: a partir de um fluxo de conversa (árvore de decisão), ou utilizando redes neurais preditivas (LSTMs). O primeiro caso se mostrou muito limitado, pois impõe a estrutura burocrática do órgão para o usuário, e obriga o chatbot a controlar a conversa para garantir a boa experiência do usuário. Mas testes em uso mostraram que esse tipo de diálogo é não natural e, mesmo o chatbot tendo o contexto e sabendo responder a determinadas perguntas, deixa a experiência para o usuário de muitos erros na interação. Já a abordagem utilizando redes neurais demanda mais conhecimento técnico do time de projeto, mais dados de treinamento, mas traz o benefício de permitir o usuário conduzir a conversa com o chatbot e tivemos mais feedbacks positivos dos nossos testes em uso.

Todos os experimentos conduzidos foram documentados e os resultados disponibilizados na wiki do repositório do projeto Tais.

Após a etapa de experimento, implementamos a chatbot TAIS. Até o final do projeto, todo o conteúdo demandado pela SEFIC foi inserido no bot, e também foi feita uma curadoria orientada a dados. Como parte da arquitetura da Tais é um BI com os dados, métricas e indicadores sobre o comportamento dos usuários, utilizamos esses dados para melhorar de forma contínua novos conteúdos.

Os dados gerais sobre o projeto da Tais são detalhados - Total de intenções:

- Total de utters:

Documentação comprobatória

- [Relatório Etapa 1](#)
- [Relatório Etapa 2](#)

- [Relatório Etapa 3](#)
 - [Relatório Etapa 4](#)
 - [Relatório Etapa 6](#)
 - [Relatório Etapa 7](#)
2. **Realizar estudos e propor técnicas de aprendizado supervisionado e detecção de anomalias para automatizar as trilhas de auditoria na fase de aprovação e prestação de contas**

Concluído -

Documentação comprobatória

- [Relatório Etapa 3](#)
- [Relatório Etapa 4](#)

3. **Realizar estudos e propor técnicas de reconhecimento de padrão e Inteligência de Negócio para análise dos projetos submetidos via Lei Rouanet**

Concluído -

Documentação comprobatória

- [Relatório Etapa 2](#)
- [Relatório Etapa 4](#)

4. **Realizar estudos e compartilhar resultados sobre inovação DevOps e melhores práticas no contexto de integração e deploy contínuos**

Concluído: Em algumas frentes, buscou-se cultivar a cultura DevOps para otimizar a entrega de valor no produto de software. Como fator comum na maioria dos casos, temos o uso de gerência de configuração utilizando: - Docker - Docker-Compose - Makefiles - *scripts* em outras linguagens (principalmente Shell Script e Python)

Em relação aos ambientes de deploy, houve sucesso na experimentação das seguintes abordagens: - Uso de máquina virtual disponibilizada pelo Ministério - Uso de máquina virtual em plataformas de cloud (Google Cloud e Digital Ocean) - Uso de máquina virtual na infraestrutura do LAPPIS - Orquestração de contêineres com Rancher 1.6 (Rancher Cattle) na infraestrutura do LAPPIS - Orquestração de contêineres com Rancher 2.0 (Kubernetes) na infraestrutura do LAPPIS

Apesar dos diversos ambientes mencionados, todos possuíam um orquestrador Docker instalado (Docker Compose, Rancher Cattle ou Kubernetes), padronizando e abstraindo toda a configuração necessária para o desenvolvedor. Além das ferramentas, o processo de *deploy* em diferentes ambientes agregou nas documentações de instalação dos serviços, uma vez que estas eram realimentados com eventuais peculiaridades observadas durante a instalação e disponibilização.

Além da disponibilização, houve preocupação com o aspecto da entrega contínua, utilizando integração e deploy contínuos (CI e CD, respectivamente). Como plataforma de CI, foi utilizado o GitLabCI pelos seguintes motivos: - Plataforma feita em código aberto - Possibilidade de se instanciar um Runner em infraestrutura externa para auxiliar no *pipeline* - Integrável ao GitHub e GitLab - Comunidade ativa compartilhando experiências e dando suporte - Experiência prévia de alguns membros do laboratório

Também houve a experimentação do Jenkins integrado ao Rancher 2.0. Todas as configurações de cada frente estão disponíveis em arquivos `.gitlab-ci.yml` localizados na raiz de cada repositório que fez uso da tecnologia. Os avanços feitos nesta tecnologia incluem: - Otimização de pipeline - Reaproveitamento de imagens Docker entre os *jobs* do *pipeline* para reduzir a duração do processo - Uso de imagem Docker separada para as dependências das aplicações para permitir sua construção apenas quando necessário, também reduzindo duração do processo - Uso de *job image* customizada para permitir o *deploy* em instâncias do Rancher 1.6 sem o uso de requisições explícitas para a API - Deploy multiplataforma - Definição de *stage* contendo deploy para diferentes ambientes em um mesmo *pipeline* - Disponibilização de imagens Docker em múltiplos *registries* - DockerHub - GitLabCI Registry - Rancher Registry

Alguns resultados e lições aprendidas foram apresentados e discutidos pelos alunos de graduação nos Workshops organizados pelo laboratório e em eventos abertos à comunidade.

Documentação comprobatória - [Configuração GitLabCI SALIC-ML](#) - [Configuração GitLabCI Tais](#) (com reaproveitamento de *build* entre *stages*) - [Apresentação sobre otimização de *pipeline* no Workshop realizado na ENAP em 27/09/2019](#) - [Apresentação sobre deploy da Tais no Workshop realizado na ENAP em 27/09/2019](#) - [Apresentação sobre construção de *pipeline* de entrega no SALIC-ML \(17º PyData Brasília\)](#) - [Configuração Rancher Pipeline BotFlow](#) - [Configuração Rancher Pipeline BotFlowAPI](#) - [DockerHub do laboratório com todas as imagens publicadas nos *pipelines*](#)

Pacote de Trabalho: Visualização de dados e criação de Dashboards

De acordo com o plano de trabalho, “O tema deste estudo é buscar formas visuais de apresentar os dados obtidos e processados nas etapas anteriores. Os gráficos produzidos servem de embasamentotanto para análise por parte da equipe do projeto quanto pelos gestores do próprio ministério.”

Metas específicas

1. Painéis com estatísticas sobre projetos cadastrados no Salic

Concluído - Inicialmente, o objetivo desta meta era gerar visualizações com base nos dados disponíveis no portal VerSalic. A equipe fez uma análise extensa tanto desse portal quanto de todo o processo da Lei de Incentivo. Em seguida, aplicou-se uma versão modificada da técnica de Design Sprint para geração de ideias de visualização. A dedicação dos bolsistas e apreço pelo tema, foi tamanha que resolveram batizar o grupo de trabalho de “Promova Cultura”.

Foram geradas 18 ideias, possibilidades de visualização divididas em dois grupos chamados de “Panorama da cultura” e “Visualização de projetos”. Destas 18, foram escolhidas 5 considerando tanto o potencial de benefício quanto viabilidade técnica. Estas 5 ideias foram prototipadas sob a forma de painéis interativos on-line. Para isso, foi adotada a tecnologia VueJS para o front-end, tecnologia essa usada pela equipe do SALIC. Além disso, a maioria dos painéis acessava dados reais via a versão refatorada da Salic-API.

Dentre os painéis prototipados, ressalta-se o de “Mapa de Calor”, que demonstra a distribuição de projetos de incentivo, incentivadores e valores incentivados pelo território nacional. O painel confirmou suspeitas conhecidas, como a concentração de incentivadores no eixo Rio-São Paulo, mas também surpreendeu ao revelar projetos no Acre sendo fortemente incentivados pela região Nordeste. O painel de “Deslocamentos” é bastante relevante pois mostra claramente o deslocamento de projetos pelo país e também trás algumas interpretações inusitadas. Por exemplo, pouquíssimos projetos circula entre o RS e SC, mesmo com a proximidade geográfica entre os estados. Cabe indicar que as visualizações prototipadas são todas responsivas, funcionando tanto em versão desktop quanto mobile.

Os painéis foram combinados em uma biblioteca de visualizações e o resultado amplamente documentado no Relatório de Acompanhamento 04. Sendo este apresentado e aprovado para os gestores do projeto. Aliás, foi durante a reunião de apresentação deste relatório, em 21/09/2018, que a equipe gestora de uma das áreas finalísticas do então Ministério da Cultura, solicitou à coordenação do projeto que o desenvolvimento dos painéis não fossem avançado e que a equipe fosse alocada para a refatoração completa do Portal Comparar, um portal de dados mais antigo do Ministério.

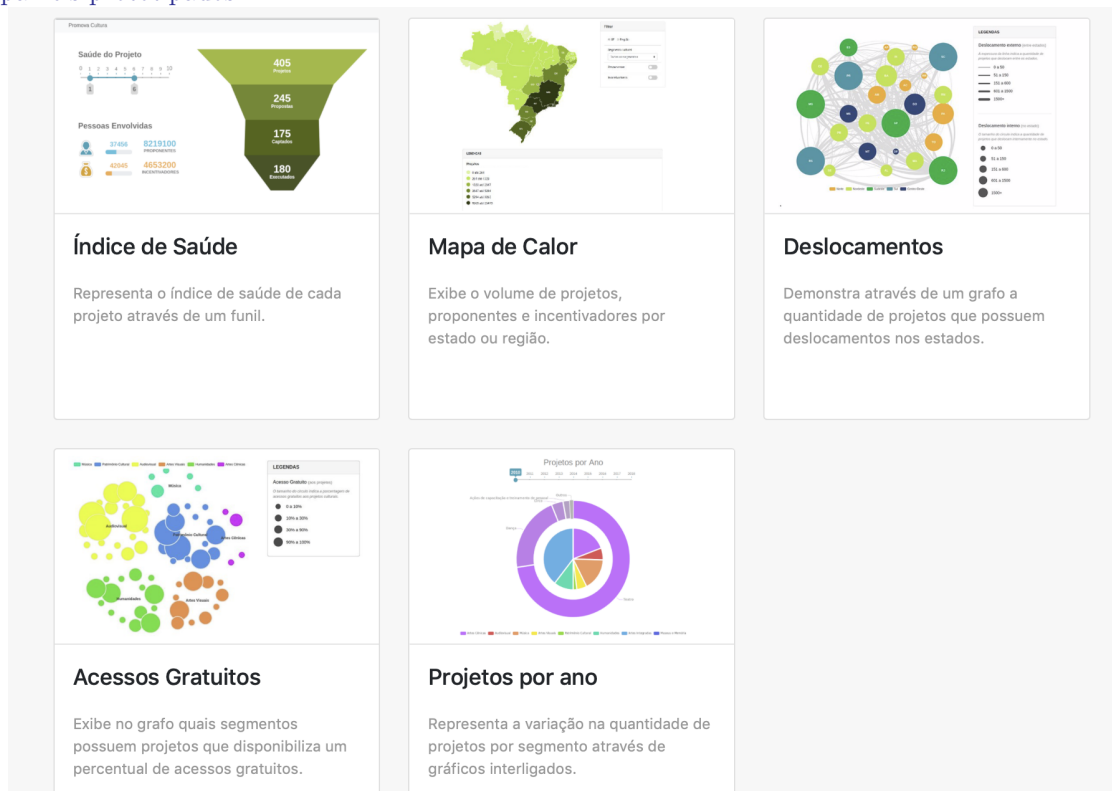
O pedido foi acatado pela coordenação do projeto, na medida que não ferisse o escopo e as metas dos projetos. Procedeu-se uma análise aprofundada de compatibilidade tecnológica e disponibilidade dos dados. O resultado dessa análise indicou que a refatoração completa seria inviável, já que dos 105 relatórios disponíveis no Portal Comparar, apenas 4 tinham dados compatíveis com o que era disponibilizado na SALIC-API. Além disso, uma refatoração completa necessitaria de uma mudança de base tecnológica que fugiria tanto das competências disponíveis na equipe, quanto da essência deste projeto.

Isso foi comunicado aos gestores do projeto e, no final de dezembro 2018, acordou-se uma tentativa de converter o relatório “Captação de recurso por UF - desde 1992” em uma visualização baseada no “Mapa de Calor”. A equipe então realizou essa adaptação e conversão, adicionando aí a exportação dos dados em formato tabular. O resultado foi uma visualização com o mesmo nome do relatório original, disponível também na biblioteca de visualizações, junto com os outros painéis.

Todos os resultados desta segunda empreitada, foram documentados no Relatório de acompanhamento 05 e aprovados pela equipe gestora de transição para o Ministério da Cidadania. Nesta mesma oportunidade, foi acordado que o custo de avançar com a adaptação do Portal Comparar seria muito grande para o projeto. Sendo assim, por decisão da CGTI, essa iniciativa foi dada como finalizada e concluída, ainda no Relatório de acompanhamento 05. A equipe “Promova Cultura”, em seguida, foi realocada para outras frentes, a saber: SALIC-ML, dashboard de BI da Tais e BotFlow.

Documentação comprobatória

- Repositório do “Promova Cultura”
- Técnica de Design Sprint - Ideias de “Panorama da cultura”
- Técnica de Design Sprint - Ideias de “Visualização de projetos”
- Repositório da API dos protótipos do “Promova Cultura”
- Biblioteca de painéis prototipados



- Visualizações:
- Relatório de Acompanhamento 4
- Portal Comparar
- Planilha de comparação dos dados necessários no portal Comparar contra os dados disponíveis na SALIC-API
- Relatório “Captação de recurso por UF - desde 1992”
- Relatório de Acompanhamento 5

2. Estudos sobre a apresentação visual de resultados de algoritmos de aprendizado de máquina e análises estatísticas

Concluído - Esse estudo foi realizado no contexto do projeto da Tais. Foi acoplado a arquitetura a stack elastic/Kibana para a mineração dos dados de conversa entre a Tais e o usuário. A partir dos dados em uso, foram projetados dashboards com métricas de uso, comportamento, e de negócio. Ao total, foram propostos X graficos de negócio, X de compartimento do usuário, e X de uso/técnico.

Essas métricas foram validadas e evoluídas. Ao final do projeto, utilizamos esses dashboards para melhorar o desempenho da Tais, acrescentar novos conhecimentos à sua base de treinamento. Também conseguimos prever tendências e conteúdos mais pesquisados pelo usuário.

Documentação comprobatória

- [Relatório Etapa 5](#)
- [Relatório Etapa 6](#)
- [Relatório Etapa 7](#)

3. Dashboard para a visualização e análise das relações entre proponentes e financiadores por meio de grafos

Concluído - A parte da equipe do “Promova Cultura” que foi realocada para o SALIC-ML, conforme relatado na meta 1 deste pacote de trabalho, se juntou aos esforços de estudo das relações entre proponentes e incentivadores por meio de grafos. O assunto é bastante complexo pois os mesmos atores podem assumir papéis distintos em projetos diferentes. Por exemplo, uma incentivadora de um projeto, pode ser proponente de um outro projeto e, ao mesmo tempo, ser fornecedora para este mesmo projeto. Além disso, existe circularidade, onde é necessário desmembrar pessoas jurídicas em seus sócios componentes para aí sim, analisar a relação entre eles.

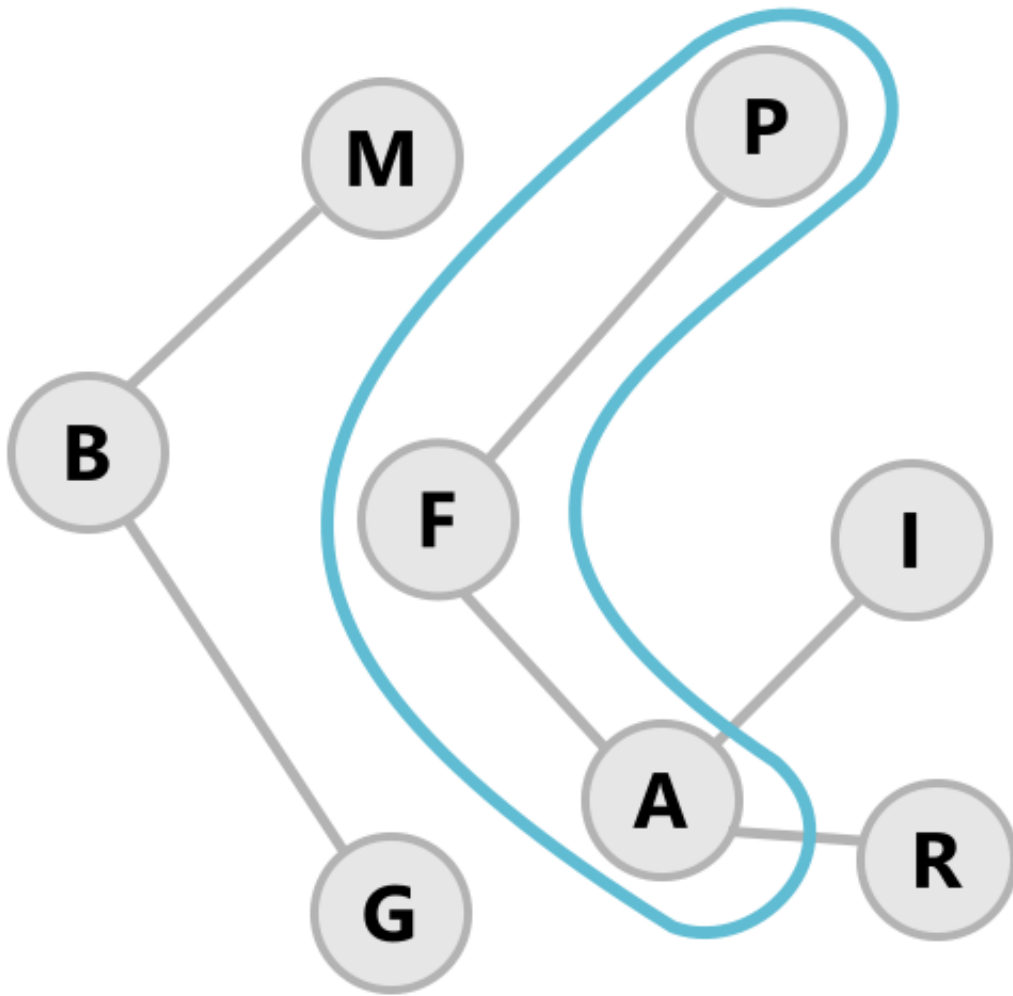
Com base nas análises preliminares dos dados e nas explorações computacionais, foram geradas propostas de visualização, documentadas as issues 312 e 326 do repositório. Em, 02/04/2019 a equipe recebeu a visita de um especialista da CGEE que fez uma jornada de treinamento e co-trabalho, apontando vários pontos que precisavam ser revistos e aprofundados. Dada essa perspectiva e a necessidade de aprofundamento do relatório de complexidade de análise financeira dos projetos, o tópico principal do SALIC-ML, a equipe foi realocada para garantir a qualidade do relatório de complexidade.

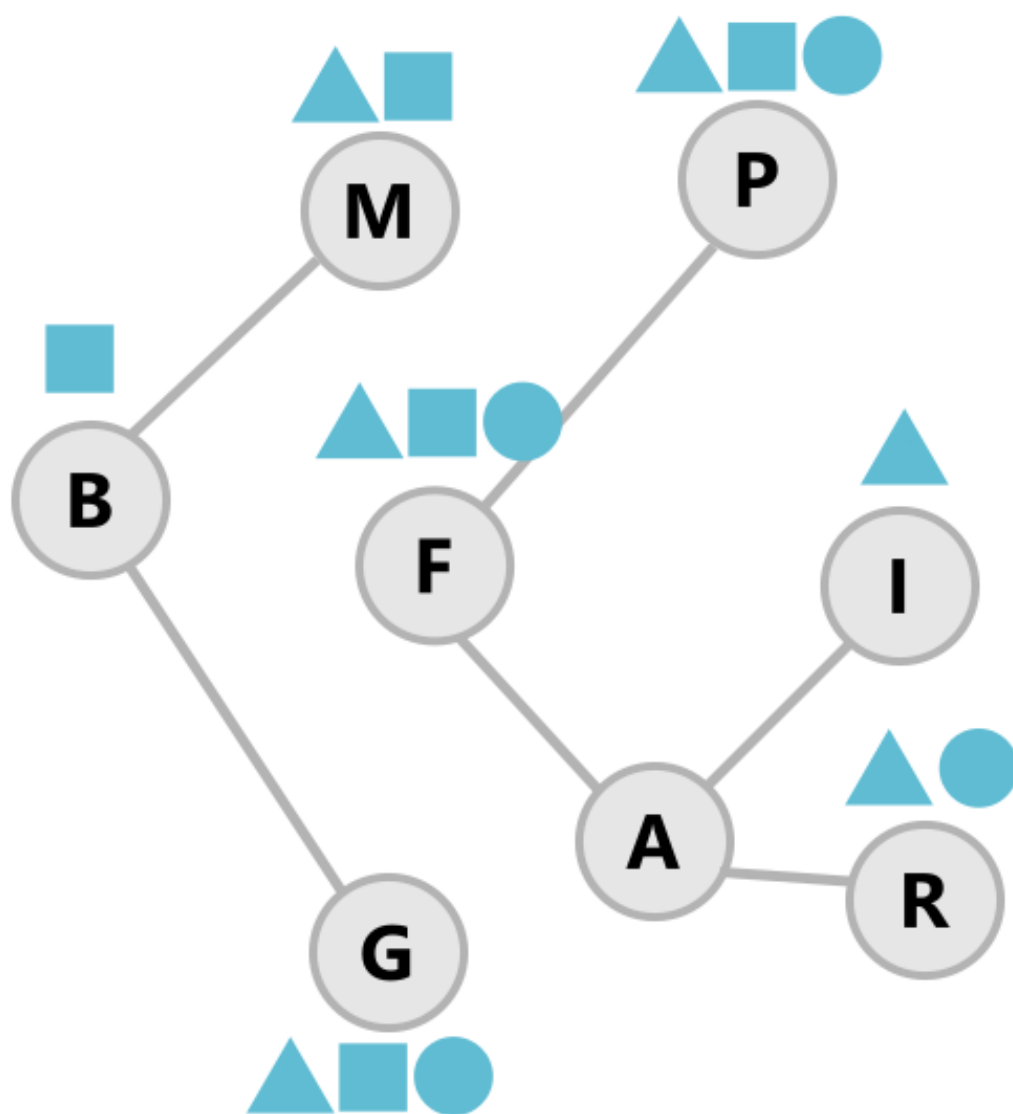
Cabe frisar que os estudos computacionais de caráter exploratório foram 100% documentados em Jupyter notebooks (em anexo), apresentando os cálculos em associação à visualizações científicas e explicitando as relações entre proponentes e financiadores por meio de grafos. O que cumpre plenamente esta meta.

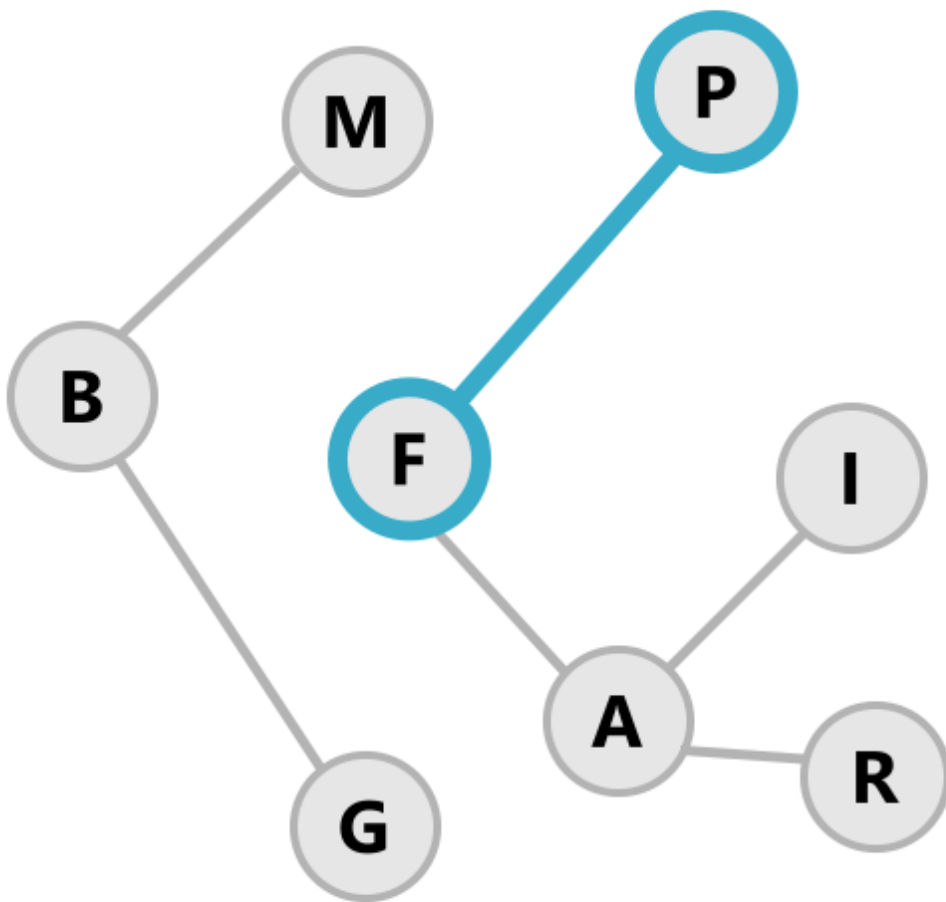
O material produzido, todos os estudos e esboços de visualização, foram apresentados como parte do Relatório de Acompanhamento 6 e aprovado pelos gestores.

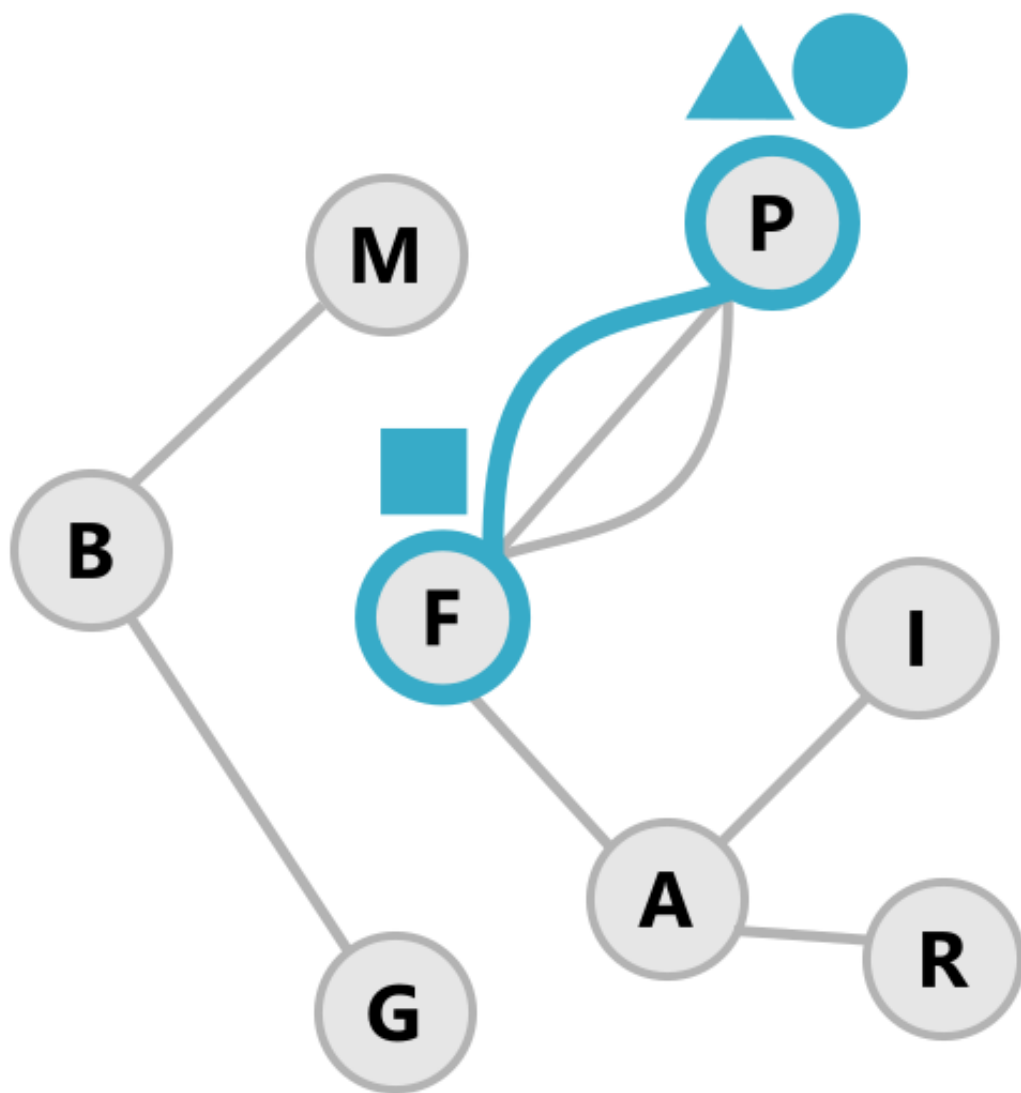
Documentação comprobatória

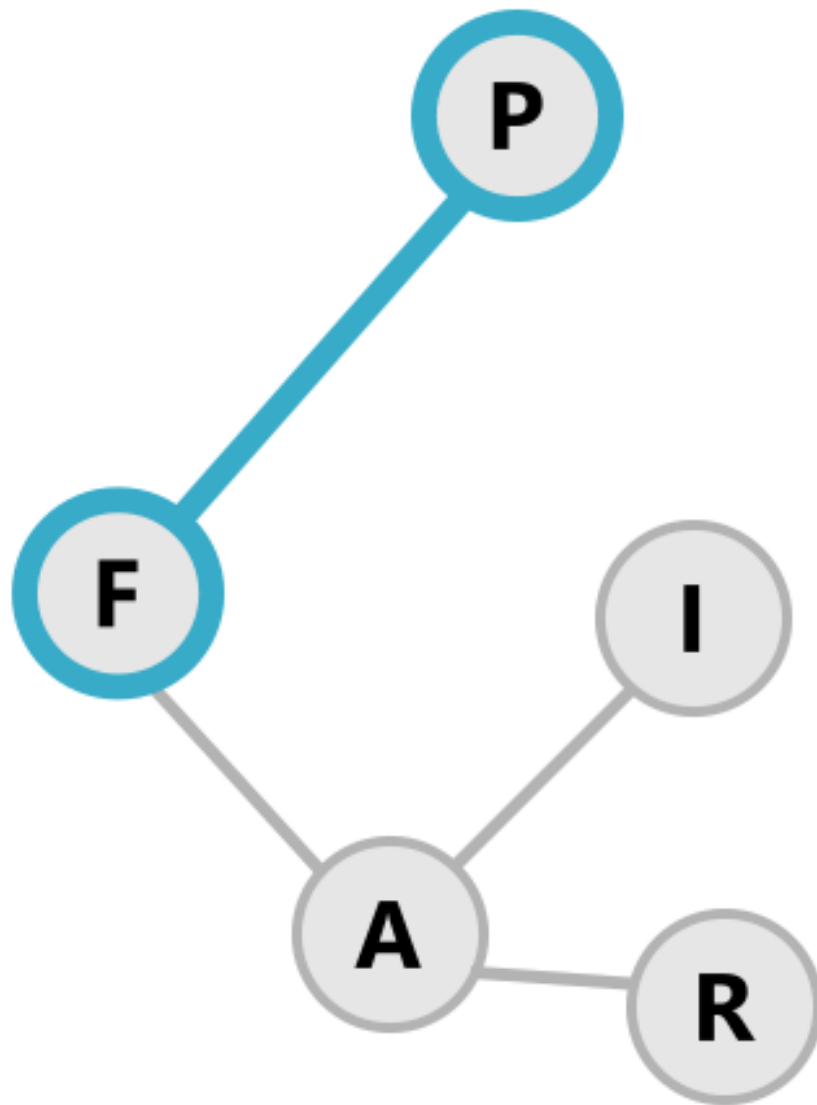
- [Repositório do “SALIC-ML”](#)
- Exemplos de estudos de grafos:

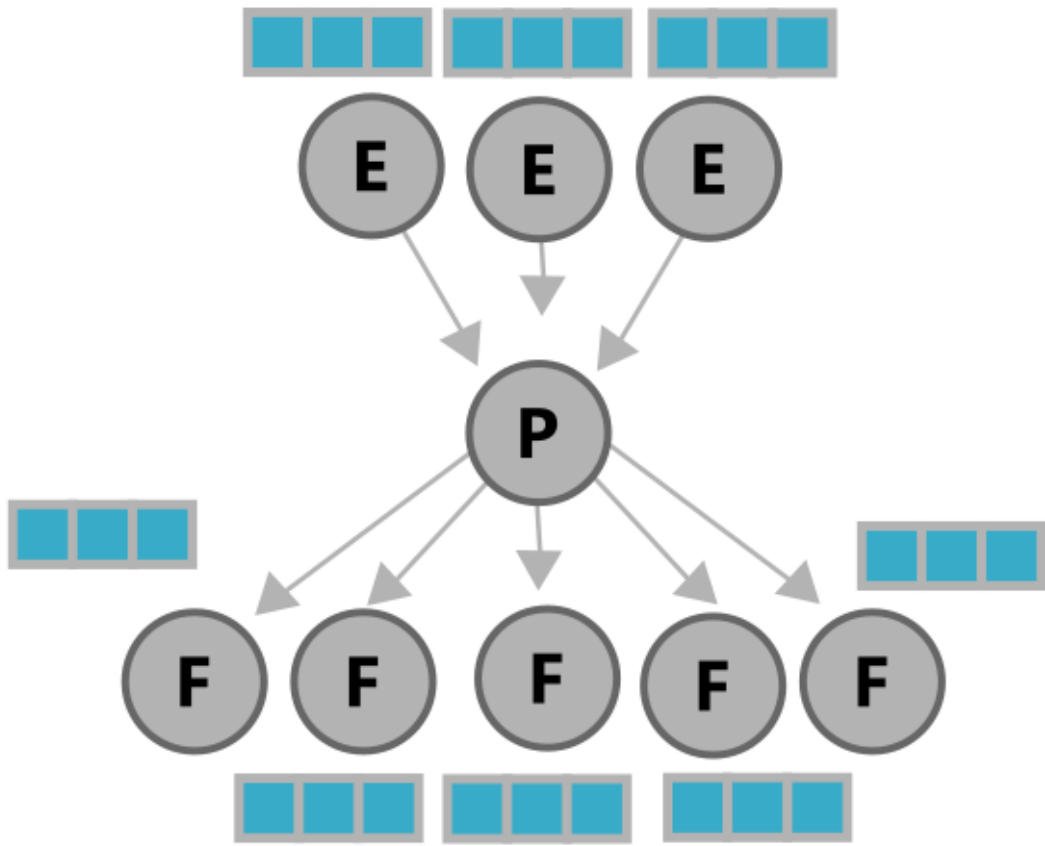


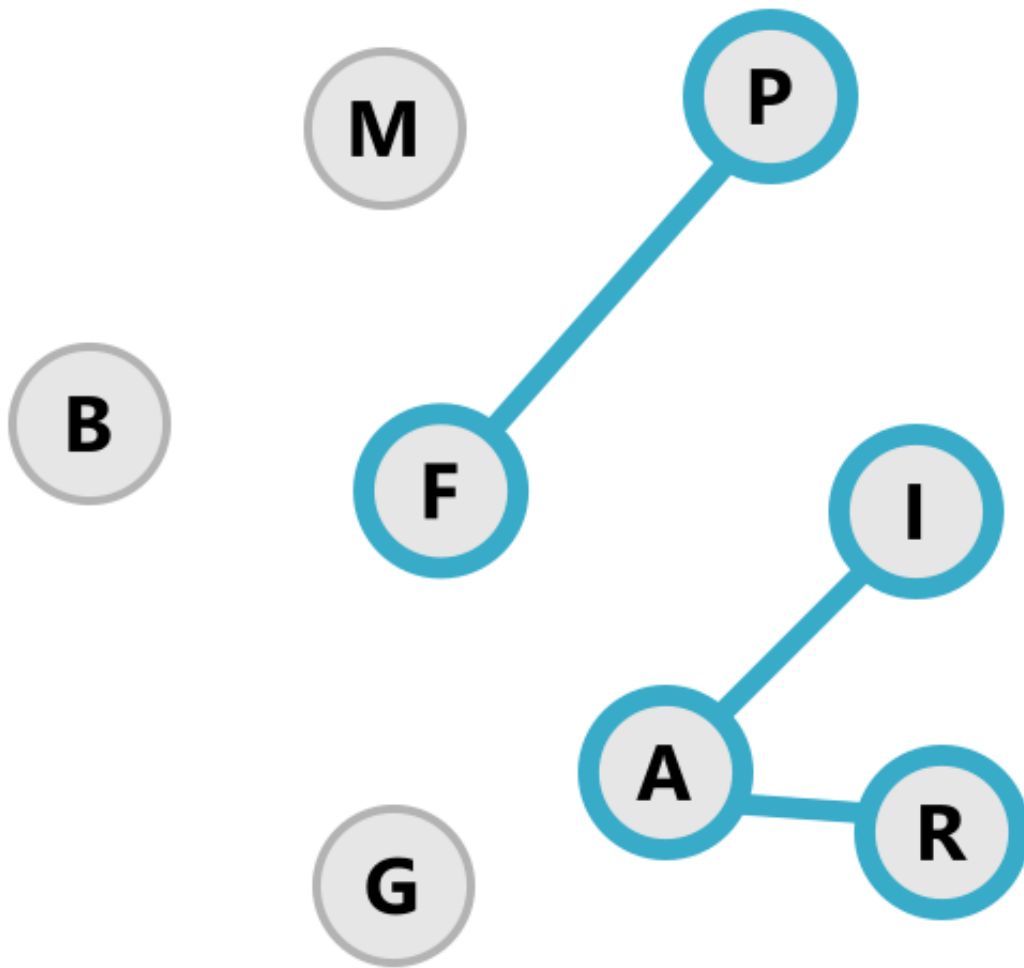


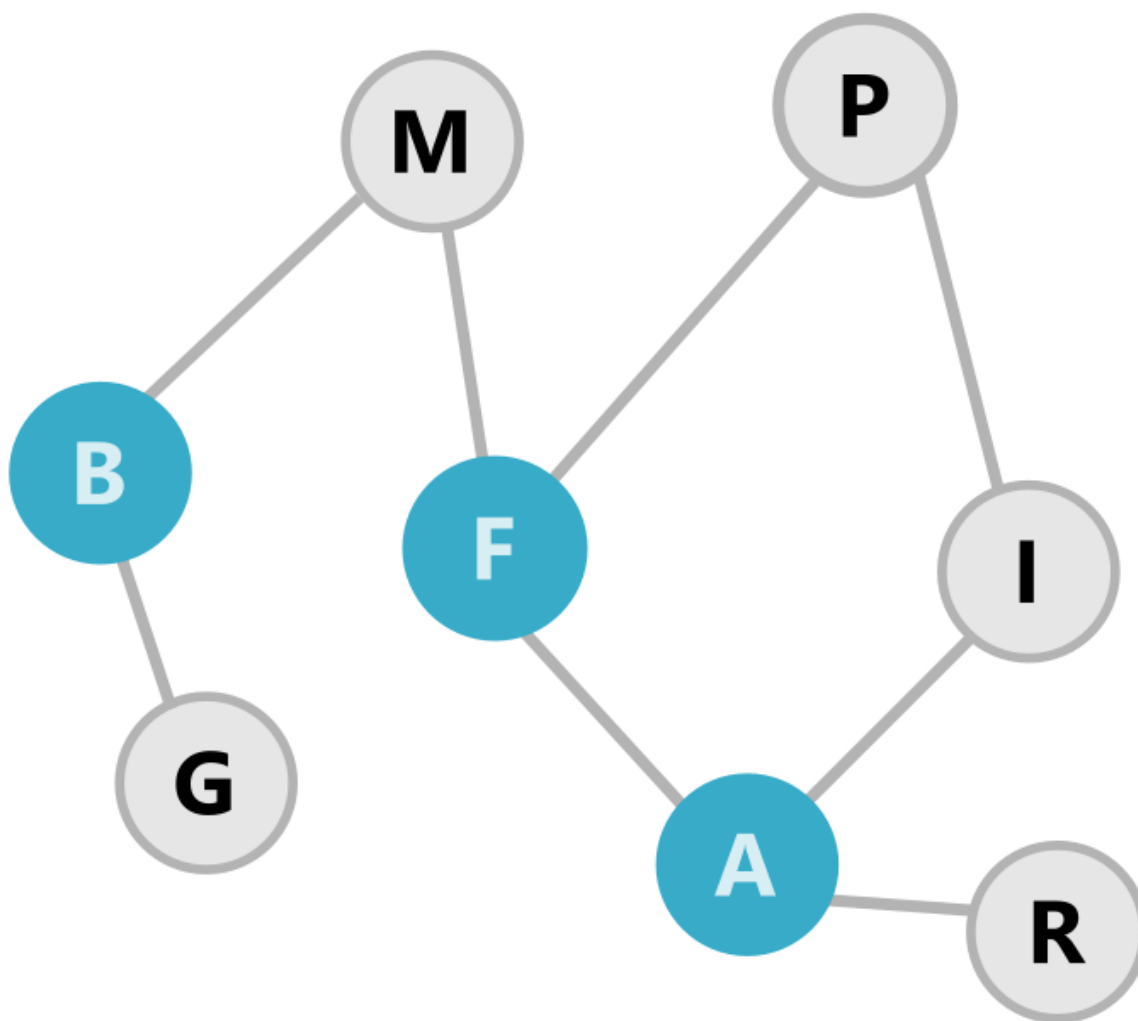












- Issue 312 - Issue 326 - Relatório de Acompanhamento 6

Pacote de Trabalho: Estudos dos processos técnicos e gerenciais MinC para aferição e aceitação de produtos de software

De acordo com o plano de trabalho, "O objetivo geral desta frente de pesquisa é auxiliar os times de desenvolvimento e gestores de TI do MinC a aprimorarem sua capacidade em tomar decisões acerca da qualidade das versões dos produtos de software entregues por seus fornecedores.

Esse pacote de trabalho teve seu cronograma alterado e escopo limitado. Tais mudanças foram registradas no relatório X de acompanhamento, e devidamente aprovados pelo ministério e universidade.

Metas específicas

1. Experimentação contínua aplicada à engenharia de produto de software

Concluído - Durante todo o projeto foi abordado a experimentação contínua, no qual passavamos continuamente por três etapas: (1) levantamento de hipóteses para solucionar um problema, (2) teste

de parte da solução para validar a hipótese, (3) validação da hipótese, (4) implementação da solução no produto final. Além disso, trabalhamos com times full-stack e orientados a produto, a fim de ter uma qualidade alta do produto entregue. Com isso, para cada pacote de trabalho, tinha-se um time composto por alunos de Engenharia de Software, Designers, Alunos de letras, e profissionais plenos e seniores para gerir os riscos de projeto.

Documentação comprobatória

- [Relatório Etapa 3](#)
- [Relatório Etapa 4](#)

2. a mineração em repositórios de software e a análise científica de dados do software

Concluído - Análise que o Renato está fazendo

Documentação comprobatória -

3. prospectar uma sistemática, baseada em evidência científica, que auxilie a homologação de produtos de software, em obediência ao normativo estabelecido

Concluído parcialmente -

Documentação comprobatória -

Acompanhamento Financeiro

Tivemos completa transparência quanto ao acompanhamento financeiro do projeto. A prestação de contas foi feita a cada relatório de acompanhamento, no qual apresentamos não só o montante gasto, quanto também o valor gasto por pacote de trabalho. O tamanho do time por frente de trabalho foi definido de acordo com as prioridades estabelecidas pelo ministério, nas reuniões estratégicas. Porém, vale ressaltar, que por se tratar de times multidisciplinares, membros poderiam ser alocados em diferentes frentes por diversas **sprints** de acordo com necessidade de projeto e deadlines de entrega.

O financeiro completo do projeto pode ser visto na figura abaixo.

Assinatura

Responsável pela Execução:

Nome: Carla Silva Rocha Aguiar (Coordenadora do Projeto)

Assinatura:

Data: /2019

Anexos

Artigos Acadêmicos

Artigos de Comunidade

Participação em eventos