



Universidade Federal
do Rio de Janeiro

Escola Politécnica

AUTOMAÇÃO DO PROCESSO DE PRODUÇÃO DE PUBLICAÇÕES EM COLABORAÇÕES CIENTÍFICAS GEOGRAFICAMENTE DISTRIBUÍDAS

Gabriela Lemos Lúcidí Pinhão

Projeto de Graduação apresentado ao Curso de Engenharia de Controle e Automação da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Carmen Lucia Lodi
Maidantchik
Flávio Luis de Mello

Rio de Janeiro
Setembro de 2019

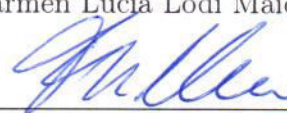
AUTOMAÇÃO DO PROCESSO DE PRODUÇÃO DE PUBLICAÇÕES EM
COLABORAÇÕES CIENTÍFICAS GEOGRAFICAMENTE DISTRIBUÍDAS

Gabriela Lemos Lúcidí Pinhão


PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO DA ESCOLA
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE ENGENHEIRO DE CONTROLE E AUTOMAÇÃO.

Examinado por:


Prof. Carmen Lucia Lodi Maidantchik, D.Sc.


Prof. Flávio Luis de Mello, D.Sc.


Prof. Afonso Celso Del Nero Gomes, D.Sc.


Prof. Guilherme Horta Travassos, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2019

Lemos Lúci Pinhão, Gabriela

Automação do processo de produção de publicações em colaborações científicas geograficamente distribuídas/Gabriela Lemos Lúci Pinhão. – Rio de Janeiro: UFRJ/ Escola Politécnica, 2019.

XII, 50 p.: il.; 29,7cm.

Orientadores: Carmen Lucia Lodi Maidantchik

Flávio Luis de Mello

Projeto de Graduação – UFRJ/ Escola Politécnica/ Curso de Engenharia de Controle e Automação, 2019.

Referências Bibliográficas: p. 46 – 47.

1. Sistemas Web. 2. *Workflow*. 3. Grafos. 4. CERN.
I. Lucia Lodi Maidantchik, Carmen *et al.* II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia de Controle e Automação. III. Título.

*Ao meu anjo da guarda, minha
amada mãe.*

Agradecimentos

Agradeço, primeiramente, à minha mãe Cristina por ter feito tudo ao seu alcance para me proporcionar educação de qualidade. Agradeço a ela, também, por todo o resto e por ter sido a melhor do mundo.

Obrigada aos meus avós João, Leilda e Helena por terem sido pai e mães. Minha formatura não seria possível sem o amor incondicional, apoio, conselhos, torcida e o lar que me proporcionam. Não poderia deixar de agradecer aos meus tios Sandra e João Henrique e primos Nicole e João Pedro, por serem a definição do que uma família deve ser.

Aos Beltrão, Pimenta, Bellotti, Romeiro, Fonseca e Napoli, sou grata por me deixarem ser parte de suas famílias e por todo o apoio. Aos amigos do Rio até Genebra, obrigada por cuidarem da minha paz de espírito e por compreenderem as inúmeras vezes que não pude encontrá-los, por estar ocupada estudando.

A todos os meus amigos e colegas da T-16, T-17, T-18, T-19 e T-20 obrigada pelo acolhimento, união e companheirismo; teria sido muito mais difícil chegar ao final sem vocês. Agradeço a todos que estiveram comigo durante a minha passagem pela empresa júnior Fluxo Consultoria. Essa foi, sem dúvida, a época em que mais aprendi em toda a minha vida. Não poderia, jamais, deixar de mencionar os irmãos que fiz no CERN, que me ensinaram grande parte do que hoje sei de programação e que me proporcionaram momentos inesquecíveis.

Ao Victor, obrigada por todo o amor e por acreditar nos meus sonhos e planos, mesmo que, às vezes, eles possam ir de encontro aos seus. Sou grata também à sua família pelo carinho com que me recebem aos finais de semana.

Agradeço imensamente a todos os mestres que me motivaram a raciocinar e me superar. Em especial, obrigada à minha orientadora Carmen, por estar sempre presente, disposta a me dar suporte acadêmico, profissional e emocional e por acreditar no meu potencial. Obrigada ao professor Flávio, por toda ajuda durante este projeto. Ao professor Afonso Celso, obrigada pela atenção, conselhos e também por exercer a arte de lecionar com maestria.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/ UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Controle e Automação.

AUTOMAÇÃO DO PROCESSO DE PRODUÇÃO DE PUBLICAÇÕES EM COLABORAÇÕES CIENTÍFICAS GEOGRAFICAMENTE DISTRIBUÍDAS

Gabriela Lemos Lúcidu Pinhão

Setembro/2019

Orientadores: Carmen Lucia Lodi Maidantchik
Flávio Luis de Mello

Curso: Engenharia de Controle e Automação

O tema deste projeto de conclusão de curso está relacionado à aplicação de conceitos de grafos para o controle e automação de processos através de software. O trabalho objetivou projetar e implementar uma solução automatizada para apoiar a evolução, revisão e aprovação das publicações científicas do experimento ATLAS do CERN, contemplando todas as etapas e regras a serem satisfeitas pelos autores e revisores envolvidos.

Em linhas gerais, a solução proposta consistiu em uma máquina de estados, na qual cada estado representa uma etapa da evolução de uma publicação. Para modelar tal solução, traçou-se um paralelo com a estrutura matemática de grafos. A classe criada para representar tal modelo foi integrada ao FENCE, um *framework* orientado a objetos que foi usado na implementação dos cinco sistemas denominados Analysis: Analysis/Phase 0, Papers, PUB notes, CONF notes e PLOTs.

A ênfase deste trabalho está no sistema Analysis/Phase 0, que apoia a concepção de publicações científicas e gera a estrutura de repositórios necessária ao armazenamento e versionamento dos arquivos. Os sistemas Papers, PUB notes, CONF notes e PLOTs dão suporte ao processo de revisão e aprovação das publicações. Juntos, os sistemas Analysis se tornaram uma ferramenta essencial, contemplando e apoiando o processo de uma publicação desde seu estágio inicial. São intensamente utilizados pelos membros do experimento ATLAS, já tendo apoiado mais de 800 artigos científicos, 900 artigos de conferência e 300 notas públicas.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Engineer.

Gabriela Lemos Lúcida Pinhão

September/2019

Advisors: Carmen Lucia Lodi Maidantchik

Flávio Luis de Mello

Course: Automation and Control Engineering

The subject of this undergraduate project is the application of graph concepts to the process control and automation through a software system. The work aimed to design and implement an automated solution to support the evolution, review and approval of the scientific publications of ATLAS, a CERN experiment, covering all steps and rules that have to be followed by all involved authors and reviewers.

The solution consists of a state machine, in which each state represents a stage in a publication process. The proposal was modeled by drawing a parallel with the mathematical structure of graphs. The class created to represent the model was integrated into FENCE, an object-oriented framework that was used to implement the five systems named Analysis: Analysis/ Phase 0, Papers, PUB Notes, CONF Notes, and PLOTs.

The emphasis of this project is on the Analysis/Phase 0 system, which supports the scientific papers elaboration and generates a repository structure that is required to store and manage the versioning of all files. The Papers, PUB notes, CONF notes and PLOTs systems support the review and approval publication process. Together, the Analysis systems became an essential tool that supports the publishing process from its earliest stage. They are intensively used by the ATLAS members, having carried out more than 800 scientific papers, 900 conference articles and 300 public notes.

Sumário

Lista de Figuras	x
Lista de Tabelas	xii
1 Introdução	1
1.1 Tema	1
1.2 Problema	1
1.3 Objetivos	1
1.3.1 Objetivo geral	1
1.3.2 Objetivos específicos	1
1.4 Metodologia	2
1.5 Solução e produto	2
1.6 Conteúdos dos capítulos	3
2 CERN, ATLAS e suas publicações científicas	4
2.1 O CERN e o Grande Colisor de Hádrõs (LHC)	4
2.2 O experimento ATLAS do CERN	5
2.3 Publicações científicas do experimento ATLAS	7
2.4 Desafios do processo de publicação	12
3 Grafos	14
3.1 O grafo e sua estrutura	14
3.2 Modelagem de <i>workflow</i> por grafos	16
3.3 Publicações científicas e seu <i>workflow</i>	17
4 FENCE	19
4.1 O <i>framework</i>	19
4.2 Sistemas FENCE	21
4.3 Principais classes	22
5 O conjunto de sistemas Analysis	27
5.1 Contribuições ao <i>framework</i> FENCE	27

5.2	Sistemas Analysis	29
5.2.1	Os sistemas Papers, CONF notes, PUB notes e PLOTs	30
5.2.2	O sistema Analysis/Phase 0	31
6	Conclusões	43
6.1	Trabalhos futuros	44
6.2	Contribuições à aluna	44
	Referências Bibliográficas	46
A	Arquivo de configuração de uma etapa do <i>workflow</i> da fase 0	48

Lista de Figuras

2.1	Estrutura do detector ATLAS. Extraído de [1].	6
3.1	Exemplo de grafo direcionado ou dígrafo.	15
3.2	Notação criada para a representação dos elementos de um fluxo de trabalho.	16
3.3	Modelagem da fase de submissão de um <i>Paper</i>	18
4.1	Sistemas FENCE desenvolvidos até Junho de 2019.	21
4.2	Funcionalidade de busca avançada de denominações dos membros do experimento ATLAS. Elementos ligados na horizontal representam o operador AND e na vertical o operador OR.	22
4.3	Funcionalidade de inserção de um novo instituto associado ao experimento ATLAS exemplificando a negligência de um campo obrigatório em um formulário.	23
4.4	Alerta de acesso negado a uma interface de um sistema FENCE pela falta de credenciais.	24
4.5	<i>E-mail</i> enviado ao reiniciar-se o processo de qualificação de um autor no sistema Membership do ATLAS.	24
4.6	Diagrama de classes da estrutura criada para a comunicação com a API do GitLab [2]	25
4.7	Principais classes do <i>framework</i> FENCE dentre as 436 existentes. Em azul, as classes de maior relevância para a implementação dos sistemas Analysis.	26
5.1	Diagrama de classes da estrutura criada para modelar fluxos de atividade.	28
5.2	Principais classes do <i>framework</i> FENCE dentre as 436 existentes. Em verde, as classes adicionadas à biblioteca em consequência deste trabalho.	29
5.3	Relação de dependência entre as entidades dos sistemas Analysis.	30

5.4	Funcionalidade de inserção de publicações do sistema Analysis/Phase 0. À esquerda, um resumo das etapas necessárias para completar a submissão dos dados. À direita, os campos presentes na primeira etapa.	32
5.5	Funcionalidade de busca de publicações do sistema Analysis/Phase 0. Apresenta campos para definição de critérios de busca, que podem ser adicionados ao espaço denominado <i>Logic Workspace</i> formando expressões lógicas.	33
5.6	Interface de detalhes de uma publicação e sua fase 0. À direita, um resumo com as informações mais importantes de uma publicação. À esquerda, as etapas correspondentes ao fluxo de atividades da fase 0.	34
5.7	Modelo entidade relacionamento simplificado do sistema Analysis/Phase 0.	36
5.8	Modelo de um arquivo de configuração da classe <code>Workflow</code> .	38
5.9	Estrutura criada no GitLab [3] para o armazenamento de repositórios Git de publicações do experimento ATLAS.	39
5.10	Gráficos de usuários por dia do sistema Analysis/Phase 0 entre Janeiro e Março de 2019.	41
5.11	Gráficos de usuários por dia do sistema Analysis/Phase 0 entre Março e Junho de 2019.	42

Lista de Tabelas

2.1	Tabela com o fluxo entre as etapas da fase 1 de um <i>Paper</i>	9
2.2	Tabela com o fluxo entre as etapas da fase 2 de um <i>Paper</i>	11
2.3	Tabela com o fluxo entre as etapas da fase de submissão de um <i>Paper</i>	12

Capítulo 1

Introdução

1.1 Tema

O tema deste projeto está relacionado à aplicação de conceitos de grafos para o controle e automação de processos através de software.

1.2 Problema

O experimento ATLAS gera cerca de 100 artigos científicos por ano, além de notas públicas e artigos de conferência. Cada um desses documentos é submetido a um rigoroso e extenso processo de revisão e aprovação, que envolve esforços de inúmeros atores geograficamente dispersos e que trabalham em diferentes fusos horários. Caso haja a realização imprecisa de atividades, retrabalho ou desvios nas etapas de tal processo, a qualidade do material científico do experimento pode ser comprometida.

1.3 Objetivos

1.3.1 Objetivo geral

Objetivou-se neste trabalho projetar, implementar e validar uma solução automatizada que contemple todas as etapas e regras a serem satisfeitas pelos envolvidos nas atividades de conceber, evoluir, revisar e aprovar uma publicação científica.

1.3.2 Objetivos específicos

- Facilitar a comunicação interna entre os autores e revisores;
- Diminuir o volume de trabalho dos autores e revisores;

- Promover a integração de ferramentas dispersas utilizadas no processo de publicação;
- Centralizar informações importantes sobre as publicações.

1.4 Metodologia

Primeiramente, buscou-se na literatura uma base teórica para representar o fluxo entre as etapas da concepção, evolução, revisão e aprovação de publicações científicas. Tal fluxo pode ser visto como um conjunto de estados e ações disparadas na transição entre eles, o que pode ser perfeitamente modelado pela estrutura de um grafo, seus vértices e arestas. A partir desta representação, um sistema computacional seria projetado e implementado para apoiar cada fase e garantir que todas as regras sejam respeitadas, promovendo uma comunicação efetiva entre os autores e revisores.

A proposta foi utilizar o *framework* FENCE, que segue o paradigma da orientação a objetos para promover a reutilização, integralmente desenvolvido pela colaboração entre o experimento ATLAS e a UFRJ. Desta forma, a modelagem resultante foi implementada em uma classe e incorporada à biblioteca de classes do *framework*, podendo ser utilizada por outros sistemas. Ao mesmo tempo, o sistema computacional desenvolvido também fez uso de várias classes já existentes na biblioteca, incorporando funcionalidades importantes de forma facilitada.

Vale ressaltar que o desenvolvimento da solução buscou seguir as boas práticas da Engenharia de Software no entendimento do problema, análise e documentação de requisitos, codificação, planejamento e execução de testes, o que facilitou o alcance dos objetivos do projeto.

1.5 Solução e produto

Este projeto deu origem a um conjunto de cinco sistemas Web denominados Analysis:

- **Analysis/Phase 0**, para apoio à concepção de publicações científicas e integração com o gerenciador de repositórios Git[4], GitLab [3], criando um conjunto inicial de arquivos e diretórios necessários à escrita das publicações e com controle de versão;
- **Papers**, para suporte à revisão e aprovação de artigos científicos;
- **CONF notes**, para suporte à revisão e aprovação de artigos de conferência;

- **PUB notes**, para suporte à revisão e aprovação de notas públicas;
- **PLOTs**, para suporte à revisão e aprovação de gráficos que serão conteúdo de publicações.

Embora apresentem especificidades distintas, as funcionalidades básicas dos sistemas são bastante similares. Em todos eles, os usuários envolvidos numa publicação seguem etapas a serem concluídas por diferentes atores e os sistemas garantem que cada responsável seja notificado quando for o momento de realizar uma ação designada a ele, impedindo que o fluxo de atividades seja interrompido ou que siga uma direção indesejada.

Neste trabalho, o sistema Analysis/Phase 0 foi escolhido para ser mais bem detalhado por ser o mais completo, englobando a maioria das funcionalidades contidas nos outros sistemas Analysis e ainda outras. Sua principal funcionalidade é a criação automática de ambientes colaborativos para escrita de documentos, onde o autor parte de arquivos pré formatados de acordo com as regras definidas pelos jornais científicos, evitando incompatibilidades relacionadas ao formato durante a submissão das publicações aos jornais.

1.6 Conteúdos dos capítulos

Este trabalho será organizado em 6 partes. No primeiro e atual **capítulo 1**, foi feita uma apresentação do problema em forma de motivação, dos objetivos do projeto, da metodologia usada e, além disso, o produto foi apresentado em linhas gerais.

No **capítulo 2**, o maior laboratório de física de partículas do mundo, o CERN, será introduzido junto ao experimento ATLAS e o complexo processo a que são submetidas as publicações antes de serem apresentadas à comunidade científica será descrito.

O **capítulo 3** será reservado para a revisão teórica da estrutura de grafos e sua relação com o tema deste trabalho.

O *framework* FENCE para desenvolvimento Web é apresentado no **capítulo 4**, junto às suas principais classes e paradigmas.

A proposta de solução é descrita no **capítulo 5**.

No **capítulo 6** encontram-se as conclusões finais do trabalho, alguns temas a serem desenvolvidos futuramente, além das contribuições profissionais e pessoais do projeto à aluna.

Ao final, apresentam-se as **referências** do trabalho e um **apêndice**, que contém um exemplo de código mencionado no capítulo 5.

Capítulo 2

CERN, ATLAS e suas publicações científicas

Neste capítulo serão apresentados a organização internacional de pesquisa CERN, seu maior experimento, o ATLAS e, também, o processo de evolução, revisão e aprovação de publicações científicas do experimento, cujos colaboradores estão geograficamente dispersos.

2.1 O CERN e o Grande Colisor de Hádrons (LHC)

No final da Segunda Guerra Mundial, a ciência europeia estava fragilizada, porém alguns cientistas inovadores almejavam a criação de um laboratório europeu de física de partículas. Raoul Dautry, Pierre Auger e Lew Kowarski na França, Edoardo Amaldi na Itália e Niels Bohr na Dinamarca estavam entre esses pioneiros. Tal laboratório não apenas uniria os cientistas europeus, mas também permitiria que compartilhassem os altos custos das instalações de projetos da física nuclear. Desta forma, surgiu o CERN (em francês: *Organisation Européenne pour la Recherche Nucléaire*) em 1954 com a missão de [5]:

- fornecer uma gama única de instalações para a aceleração de partículas que permita a pesquisa na vanguarda do conhecimento humano;
- realizar pesquisa da melhor qualidade na área da física fundamental;
- unir pessoas de todo o mundo para estimular as fronteiras da ciência e tecnologia, para o benefício de todos.

Atualmente, o CERN conta com a aliança de 23 países membros e 55 países com acordos de cooperação ou contatos científicos. Os membros da equipe do CERN participam do projeto, construção e operação da infraestrutura de pesquisa. Eles

também contribuem para a preparação e operação dos experimentos, bem como para a análise dos dados coletados por uma vasta comunidade de usuários, abrangendo mais de 12.200 cientistas de 110 nacionalidades e de institutos de mais de 70 países.

Nos primeiros momentos após o *Big Bang*, o universo era extremamente quente e denso. Com o seu esfriamento, as condições se tornaram apropriadas para dar origem aos blocos constituintes da matéria - os quarks e elétrons, dos quais todos nós somos feitos. Com o objetivo de recriar e gerar conhecimento sobre tais condições, os cientistas do CERN aceleram prótons na velocidade da luz, que depois são levados a uma colisão. O choque gera inúmeros tipos de partículas existentes nos primeiros momentos de existência do universo e estas são estudadas pelos diferentes experimentos do centro de pesquisa.

O Grande Colisor de Hádrons (LHC) é o maior acelerador de partículas do mundo. Foi inaugurado em 10 de setembro de 2008 e continua sendo a mais recente adição ao complexo de aceleradores do CERN. Dentro do acelerador, dois feixes de partículas de alta energia viajam perto da velocidade da luz antes de serem colididos. Os feixes viajam em direções opostas em tubos separados - dois tubos mantidos em ultra-alto vácuo. Eles são guiados em torno de um anel acelerador de 27 quilômetros por um forte campo magnético mantido por eletroímãs supercondutores e estruturas de aceleração para aumentar a energia das partículas ao longo do caminho, chegando a percorrer tal anel mais de 11 mil vezes a cada segundo. As colisões dos feixes dentro do LHC acontecem em quatro locais ao redor do anel acelerador, correspondendo às posições de quatro detectores de partículas - ATLAS, CMS, ALICE e LHCb. Como este projeto foi desenvolvido dentro do escopo do experimento ATLAS este será mais bem descrito a seguir.

2.2 O experimento ATLAS do CERN

O ATLAS (*A Toroidal LHC ApparatuS*) é o maior dos quatro detectores acoplados ao LHC. É, também, um dos dois detectores de uso não específico do LHC, sendo o outro o CMS. Ele investiga uma ampla gama da física, desde a busca do bóson de Higgs até partículas que poderiam compor a matéria escura [6]. Embora tenha os mesmos objetivos científicos, o ATLAS e o CMS utilizam diferentes projetos e soluções técnicas em seus detectores.

No LHC, feixes de partículas colidem no centro do detector ATLAS formando fragmentos de colisão na forma de novas partículas, que saem do ponto de colisão em todas as direções. Seis subsistemas de detecção diferentes, dispostos em camadas ao redor do ponto de colisão, registram os caminhos, o momento e a energia das partículas, permitindo que elas sejam individualmente identificadas. A estrutura do ATLAS é representada na figura 2.1, onde é possível observar suas diversas camadas,

cada qual com sua função específica na reconstrução dos eventos de colisão.

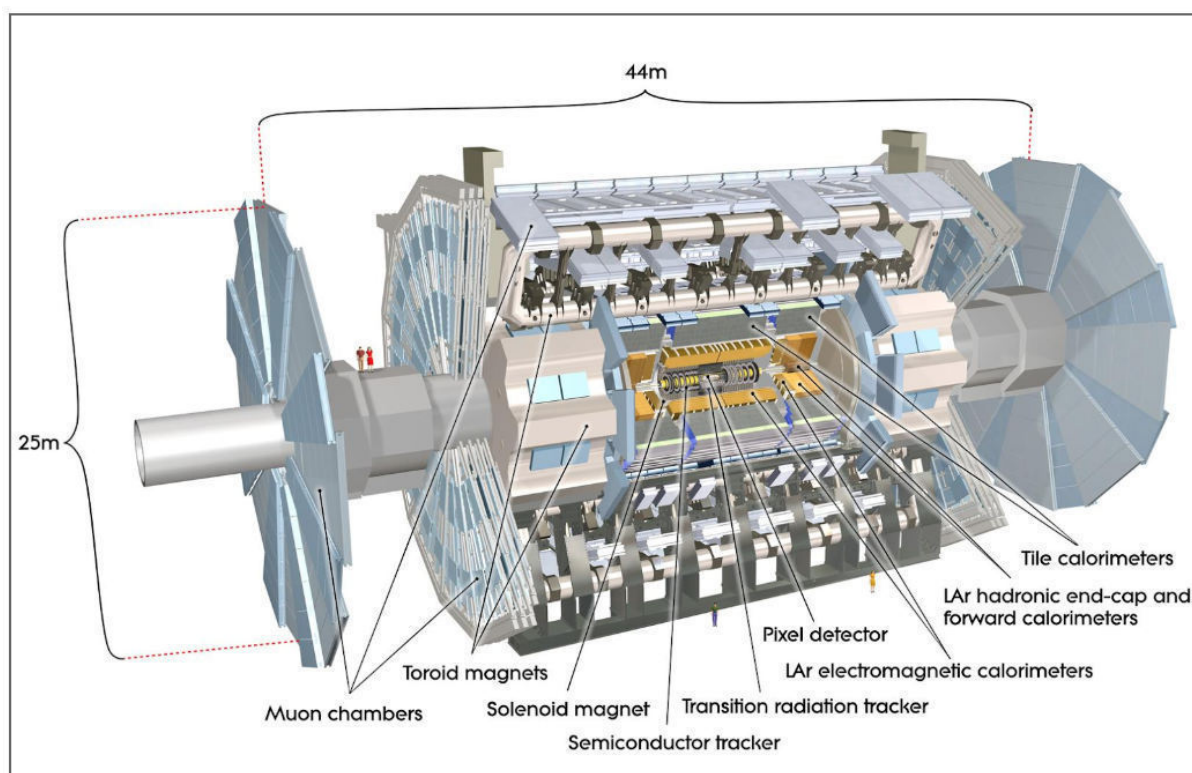


Figura 2.1: Estrutura do detector ATLAS. Extraído de [1].

Um sistema magnético força partículas eletricamente carregadas a curvarem-se para um lado ou o oposto, dependendo do sinal da carga. Ao interagirem com a camada mais interna do detector, a carga elétrica e o momento das partículas podem ser inferidos. Ao redor desta camada estão os calorímetros, instrumentos utilizados para capturar e medir a energia das partículas.

As interações nos detectores ATLAS criam um enorme fluxo de dados. Para filtrá-los, o ATLAS usa um sistema avançado denominado gatilho (*trigger*, em inglês) para informar ao detector quais eventos gravar e quais ignorar. Sistemas complexos de aquisição de dados e computação são usados para analisar os eventos de colisão registrados.

O ATLAS compreende cerca de 3000 cientistas de 183 instituições ao redor do mundo. É um dos maiores esforços colaborativos do ramo da ciência. Cerca de 1200 alunos de doutorado estão envolvidos no desenvolvimento do detector, aquisição de dados e posteriores análises. A colaboração também depende dos esforços de muitos engenheiros, técnicos e equipe administrativa.

O experimento possui uma estrutura organizacional que permite que as suas equipes se autogerenciem e que os seus membros estejam diretamente envolvidos nos processos de tomada de decisão. Os cientistas se organizam em diversos grupos relacionados às áreas de pesquisa do experimento. Qualquer descoberta científica

da colaboração é compartilhada por todos os membros e está sujeita a processos rigorosos de revisão e verificação de fatos antes que os resultados se tornem públicos. Este processo será mais bem detalhado a seguir.

2.3 Publicações científicas do experimento ATLAS

Para realizar o seu honrado programa de pesquisas, reconhecido mundialmente, os físicos do experimento ATLAS precisam analisar o conjunto de dados gerados pelo detector perante vários modelos físicos, pois o resultado das colisões de partículas podem gerar descobertas que contribuam a um ou mais deles. Para melhor gestão dessa análise, os pesquisadores envolvidos são divididos em grupos e subgrupos sob a liderança de coordenadores: *Top quark* (TOPQ), *Standard Model* (STDM), *B quark* (BPHY), *Higgs* (HIGG), *Electron/Gamma* (EGAM), *Jet and Etmis* (JETM) e outros. Os estudos que não têm a física como objeto principal de estudo também são contemplados por grupos e subgrupos, por exemplo, *Software* (SOFT) e *Data Preparation* (DAPR).

Quando a análise de dados ou o estudo de cada grupo gera resultados significativos, os físicos são solicitados a escrever uma publicação. No caso de uma pesquisa fundamental, como é o caso da maioria que é realizada no CERN, voltada para a descoberta dos constituintes elementares da física de partículas, tornar seus resultados públicos é uma obrigação para que o conhecimento seja compartilhado com toda a comunidade científica.

O ATLAS considera três tipos de publicações: artigos que serão submetidos a um periódico científico (podendo ser baseados em dados ou relacionados a projetos), artigos de conferência e notas públicas. Os artigos científicos são chamados de *Papers*, os artigos de conferência de *CONF notes* e as notas públicas de *PUB notes*.

Todas as análises do ATLAS são primeiramente discutidas e apresentadas nos grupos de trabalho relevantes e, quando o grupo conclui que delas podem ser retirados resultados interessantes, inicia-se o processo de escrita, revisão e aprovação. Muitos colaboradores com papéis diferentes são envolvidas nestes processos, alguns deles serão apresentados a seguir.

Os ***Group Conveners*** são os coordenadores dos grupos ou subgrupos aos quais uma publicação pertence. Os grupos e subgrupos devem providenciar ajuda, conselhos e recursos às análises no seu estágio inicial, ajudar na estruturação do conteúdo da publicação e participar das revisões.

A ***Colaboração*** representa todos os autores do experimento ATLAS, que são convidados a ler e comentar as versões de uma publicação, desde que compareçam às reuniões. O ***Analysis Team*** é o grupo de autores que de fato escreve a publicação, tendo o dever de assegurar que os resultados apresentados estejam cientificamente

corretos. Já os *Contact Editors* são autores que pertencem ao *Analysis Team*, porém têm mais responsabilidades na escrita e correção da publicação do que os outros.

O *Editorial Board* é um conselho formado para cada publicação e os seus integrantes precisam garantir o seu progresso durante várias etapas da revisão e aprovação. No caso de uma *PUB note* são chamados de *Readers*. Estes são representados por dois pesquisadores, selecionados pelo *Physics Coordinator*, para exercer o papel do *Editorial Board* e ainda aprovar as versões iniciais e final da nota pública. Os *Language editors* são membros da Colaboração responsáveis por corrigir erros de gramática em inglês e, também, garantir que as regras de estilo de texto e imagem sejam seguidas.

O *Publication Committee* é um comitê formado por dez membros do experimento ATLAS, que são responsáveis por garantir que o conteúdo de um artigo seja de alto padrão científico e identificar os jornais adequados à publicação. O *Spokesperson* é o porta-voz do experimento ATLAS e, também, responsável por ler e dar a última aprovação antes que o artigo seja submetido a um periódico científico. Em uma *CONF note* ou *PUB note*, o *Publication Committee* e o *Spokesperson* não são envolvidos no processo de aprovação. Ao invés disso, o *Physics Coordinator* é responsável nomear dois pesquisadores responsáveis pelas aprovações das versões iniciais e finais da publicação, os *Readers*.

Para iniciar o processo de escrita, algumas definições precisam ter sido estabelecidas com relação à publicação como o seu título, a quais grupo e subgrupos pertence, e quais os membros do *Analysis Team*. A partir daí, os membros do *Analysis Team* já podem começar a escrever a primeira versão da publicação, ainda na forma de uma ou mais notas internas do experimento ATLAS. Simultaneamente à escrita, inicializando-se a fase 0, reuniões com os grupos e subgrupos acontecem para outras formalizações como: objetivos, métodos, conjunto de dados utilizados na análise e modelos testados. Depois, o *Editorial Board* é formado, seguido da nomeação dos *Contact Editors* e, por fim, os *Group* e *Subgroup Conveners* aprovam se as definições e as notas internas estão adequadas e se justificam a criação de uma publicação. Caso positivo, é necessário definir-se se a publicação se tornará um *Paper* ou *CONF note*. Caso contrário, permanece como nota interna. *PUB notes* não são submetidas à fase 0 e seguem diretamente à fase 1. Este, também, é o caso de alguns tipos de *Papers* e *CONF notes*.

Após a fase 0, um *Paper* é submetido a mais três fases: fase 1, fase 2 e submissão. Já uma *CONF note* ou *PUB note* seguem apenas uma única fase. A sequência das etapas de cada fase de um *Paper* será descrita em detalhes a seguir e resumida pelas tabelas 2.1, 2.2, 2.3.

O processo de revisão e aprovação de um *Paper* tem início na fase 1, que começa

com uma apresentação da chamada primeira versão, ainda na forma de nota interna, aos membros do grupo e subgrupos. Depois da apresentação, os membros do *Analysis Team* devem fazer as correções relativas aos comentários realizados. Os *Group Conveners* devem, depois da correção, aprovar a primeira versão quando estiverem satisfeitos e, a seguir, o *Editorial Board* deve aprová-la também.

Dada as aprovações, inicia-se o período da Primeira Circulação ao ATLAS, quando o *Publication Committee* encaminha um *e-mail* aos autores da Colaboração encorajando-os a fazer comentários na primeira versão. O período da circulação dura de dez a quatorze dias dependendo da extensão do texto e da complexidade do assunto. Ao fim da Primeira Circulação, o *Analysis Team* tem aproximadamente três dias para responder aos comentários da Colaboração. Depois, o *Publication Committee* organiza uma reunião denominada *Paper Approval Meeting* (PAM) em que o *Editorial Board* apresenta à Colaboração a resposta do *Analysis Team* aos comentários realizados durante Primeira Circulação. Após o aval do *Publication Committee*, a publicação pode seguir para a fase 2.

Tabela 2.1: Tabela com o fluxo entre as etapas da **fase 1** de um *Paper*.

Posição	Fase	Descrição da etapa	Posição da próxima etapa
1	Fase 1	Apresentação da primeira versão ao grupo e subgrupos	2
2	Fase 1	Correção dos comentários decorrentes da apresentação	3
3	Fase 1	Aprovação da primeira versão corrigido pelos <i>Group Conveners</i>	4 - Se houver aprovação; 2 - Caso contrário
4	Fase 1	Aprovação da primeira versão corrigido pelo <i>Editorial Board</i>	5 - Se houver aprovação; 2 - Caso contrário
5	Fase 1	Primeira Circulação ao ATLAS	6
6	Fase 1	Elaboração das respostas aos comentários da Colaboração	7
7	Fase 1	Reunião com a Colaboração após a Primeira Circulação	8
8	Fase 1	Aval do <i>Publication Committee</i>	9

A fase 2 inicia-se com o *Analysis Team* preparando a segunda versão, corrigindo-a de acordo com o resultado da PAM. Quando esta estiver pronta, o *Editorial Board*

deve aprová-la se estiver satisfeito com as modificações. Os *Language Editors* devem, então, corrigir erros de gramática e forma diretamente no documento. Depois, inicia-se o período da Segunda Circulação ao ATLAS, quando o *Publication Committee* encaminha um *e-mail* para todos os autores da Colaboração encorajando-os a fazer comentários na segunda versão. O período da circulação dura, tipicamente, sete dias e o *Analysis Team* tem aproximadamente três dias para responder aos comentários da Colaboração.

O *Publication Committee* organiza outra PAM, se necessário, em que o *Editorial Board* apresenta à Colaboração as respostas do *Analysis Team* aos comentários. Em seguida, o *Editorial Board* se certifica que todos os comentários da PAM foram devidamente corrigidos e, caso sim, aprova a segunda versão corrigida. Depois, o *Publication Committee* e o *Spokesperson*, simultaneamente, leem a versão cuidadosamente e apresentam novos comentários e o *Analysis Team* tem de três a cinco dias para respondê-los e corrigi-los. Quando satisfeitos com as correções, o *Publication Committee* e o *Spokesperson* realizam a aprovação final e a publicação pode seguir para a fase de submissão.

Tabela 2.2: Tabela com o fluxo entre as etapas da **fase 2** de um *Paper*.

Posição	Fase	Descrição da etapa	Posição da próxima etapa
9	Fase 2	Preparação da segunda versão	10
10	Fase 2	Aprovação da segunda versão pelo <i>Editorial Board</i>	11 - Se houver aprovação; 9 - Caso contrário
11	Fase 2	Correção de erros de gramática e forma na segunda versão	12
12	Fase 2	Segunda Circulação ao ATLAS	13
13	Fase 2	Elaboração das respostas aos comentários da Colaboração	14
14	Fase 2	Reunião com a Colaboração após Segunda Circulação	15
15	Fase 2	Correção dos comentários da reunião	16
16	Fase 2	Aprovação da versão corrigida da segunda versão pelo <i>Editorial Board</i>	17 - Se houver aprovação; 15 - Caso contrário
17	Fase 2	Revisão da segunda versão corrigido pelo <i>Publication Committee</i> e <i>Spokesperson</i>	18
18	Fase 2	Correção dos comentários da revisão	19
19	Fase 2	Aprovação da versão final da segunda versão pelo <i>Publication Committee</i> e <i>Spokesperson</i>	20 - Se houver aprovação; 18 - Caso contrário

No início da fase de Submissão, o *Paper* é submetido ao periódico científico escolhido e, normalmente, comentários são recebidos algumas semanas depois. O *Analysis Team* deve, então, responder aos comentários e fazer as correções necessárias, gerando a versão revisada. Depois, esta é submetida ao periódico novamente e o processo finaliza quando for aprovada pelo periódico.

Tabela 2.3: Tabela com o fluxo entre as etapas da **fase de submissão** de um *Paper*.

Posição	Fase	Descrição da etapa	Posição da próxima etapa
20	Submissão	Submissão do <i>Paper</i> ao periódico científico	21
21	Submissão	Correção dos comentários feitos pelo periódico	22
22	Submissão	Aprovação da publicação pelo periódico	21 - Se houver aprovação; FIM - Caso contrário

No caso de uma *CONF* ou *PUB note*, o sequenciamento de etapas é semelhante ao de um *Paper*, porém, contendo apenas a fase 1. Ao fim desta fase, a primeira e única versão deve ser aprovada pelos responsáveis e depois é considerada finalizada, podendo ser apresentada numa conferência ou tornar-se uma nota pública. É válido ressaltar que os gráficos que aparecem nas publicações também precisam ser aprovados pela colaboração e pelos *Group Conveners*, passando por um processo de circulação separado daquele voltado às publicações.

2.4 Desafios do processo de publicação

A elaboração de uma publicação científica do ATLAS segue um fluxo extenso e complexo de etapas que não devem ser negligenciadas. Porém, sem a correta gestão do processo, isso se torna um risco. Um exemplo seria se algum responsável por dar seguimento a uma etapa não fosse contactado, o que atrasaria o processo até que algum outro envolvido percebesse sua demora. Outro exemplo seria se a data de uma reunião não fosse comunicada a algum revisor e seus comentários não fossem considerados, podendo provocar retrabalho para os editores quando considerarem tais comentários tardiamente, ou pior, nunca os levar em consideração, gerando impactos na qualidade da publicação.

Outro possível problema é a utilização de ferramentas diferentes para realizar as tarefas do processo, como os resultados de uma reunião de revisão sendo registrado em uma plataforma e o de outra reunião em outra, sem integração entre elas. Isso causa mais trabalho aos envolvidos e aumenta o risco de erro no armazenamento de dados, uma vez que são inseridos manualmente em diversos repositórios.

Portanto, a continuidade e eficácia das etapas do processo de produção de publicações científicas são grandes desafios. A solução desenvolvida neste projeto busca resolvê-los, garantindo a efetiva comunicação entre todos os envolvidos, promovendo a integração entre as ferramentas usadas, estimulando a uniformização de procedi-

mentos e centralizando informações.

Capítulo 3

Grafos

Este capítulo visa introduzir o conceito de grafos, explicando sua estrutura, além de relacionar tal teoria com o tema deste trabalho.

3.1 O grafo e sua estrutura

Muitas situações do cotidiano podem ser descritas por meio de um diagrama de pontos com linhas fazendo ligações entre eles. Um exemplo estático seriam os pontos representando pessoas e as linhas ligando uma pessoa até todos os seus amigos. Um exemplo dinâmico seria a confecção de um bolo onde os pontos seriam as etapas (separar os ingredientes, preparar a massa, levar o bolo ao forno, remover o bolo do forno) e as linhas seriam as relações entre uma etapa e a sua próxima. O conceito matemático empregado em situações como estas é chamada de grafo.

Um grafo é uma representação abstrata de um conjunto de objetos e das relações que existem entre eles. Consiste numa tripla contendo um conjunto de **vértices**, um conjunto de **arestas** e uma **função de incidência**, que associa cada aresta a um par não ordenado de vértices. Caso o par de vértices seja ordenado, o grafo é chamado de grafo direcionado ou dígrafo. Se dois vértices são ligados pela mesma aresta, são chamados de vértices adjacentes. Da mesma forma, se duas arestas se ligam ao mesmo vértice, são chamadas de arestas adjacentes. Uma aresta que liga um vértice a ele mesmo é chamada de *loop* e, caso contrário, é chamada de *link*.

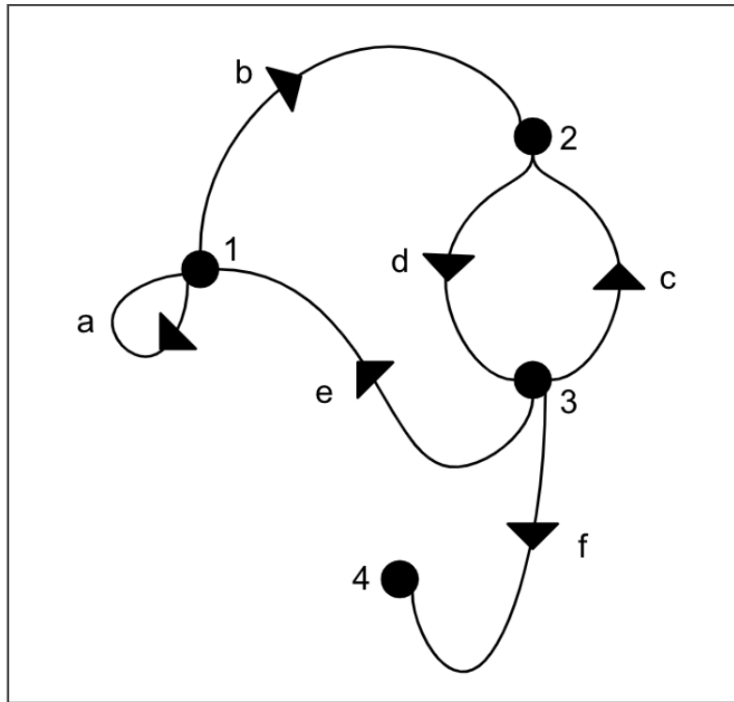


Figura 3.1: Exemplo de grafo direcionado ou dígrafo.

Na figura 3.1 temos o exemplo de um grafo direcionado onde podemos aplicar todos os conceitos vistos até então:

- Grafo: $G = (V(G), E(G), \psi_G)$;
- Vértices: $V(G) = \{1, 2, 3, 4\}$;
- Arestas: $E(G) = \{a, b, c, d, e, f\}$;
- Função de incidência: ψ_G , definida por $\psi_G(a) = 11$; $\psi_G(b) = 12$; $\psi_G(c) = 32$; $\psi_G(d) = 23$; $\psi_G(e) = 31$; $\psi_G(f) = 34$;
- Vértices adjacentes: 1 e 1; 1 e 2; 2 e 3; 3 e 1; 3 e 4;
- Arestas adjacentes: a, b, e; b, c, d; c, d, e, f;
- *Loops*: a;
- *Links*: b, c, d, e, f;

Um grafo pode conter um ou mais subgrafos, sendo chamado de super grafo. Um passeio ou caminhada num grafo consiste numa sequência intercalada de vértices e arestas começando e terminando em vértices. Um passeio é denominado fechado se começa e termina no mesmo vértice. Se um passeio fechado tem os vértices internos diferentes do vértice inicial ou final, é denominado ciclo.

3.2 Modelagem de *workflow* por grafos

Muitas definições de fluxos de trabalho, atividades ou, do inglês, *workflow*, podem ser encontradas na literatura. De acordo com o modelo de referência da *Workflow Management Coalition* - WfMC [7], um *workflow* automatiza procedimentos em que documentos, tarefas ou informações são passados de um participante para outro de acordo com um conjunto definido de regras para alcançar-se um objetivo. Esta definição considera a palavra já dentro de um contexto automatizado, na forma de um sistema de TI. Porém, um fluxo de atividades manual também é possível, o que pede uma definição mais abrangente da palavra: progressão das etapas (tarefas, eventos, interações) que compõem um processo de trabalho envolvendo duas ou mais pessoas e que cria ou agrega valor às atividades de uma organização.

Alguns dos componentes principais de um *workflow* são: atividades, atores e rotas. Uma **atividade** em um fluxo de trabalho corresponde a ações que devem ser executadas em um certa etapa de um processo. Um **ator** é aquele responsável por executar uma atividade. As **rotas** definem como deve ser o encadeamento entre as etapas de cada atividade, ou seja, qual a ordem entre elas e quais **condições** e **eventos** justificam o fim de uma etapa e o início de outra.

O roteamento de um fluxo de trabalho pode ser perfeitamente representado por um dígrafo, em que suas arestas encapsulam a relação direcional entre os vértices. As atividades e seus respectivos atores podem estar associadas aos vértices deste grafo e o movimento entre os vértices pode ser determinado por meio de eventos e/ou condições.

Sendo assim, é possível apoiar-se na estrutura de um grafo para a construção de um *workflow*, adicionando ao seu modelo alguns elementos vacantes: atividades/ações, atores, eventos e condições. A figura 3.2 mostra a notação criada para este modelo.

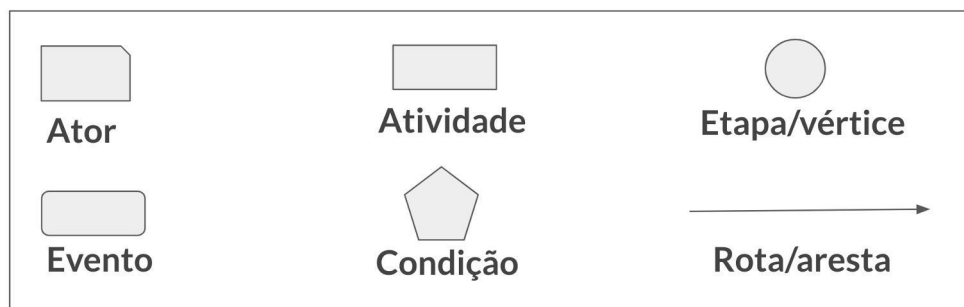


Figura 3.2: Notação criada para a representação dos elementos de um fluxo de trabalho.

3.3 Publicações científicas e seu *workflow*

A produção de publicações científicas é um fluxo de trabalho e, conforme visto na seção anterior 3.2, é possível pensá-la na forma de um super grafo e seus subgrafos adicionando-se notações para atividades, atores, eventos e condições. Tais subgrafos representam as fases do processo e seus vértices representam as etapas de cada fase. As arestas que ligam as etapas podem ser de dois tipos: permanecer na mesma etapa (*loop*) ou prosseguir para a próxima etapa (*link*). Existe também a aresta de voltar a uma etapa anterior, mas essa pode ser entendida como similar a de prosseguir para uma próxima etapa, porém, já visitada anteriormente (ciclo). As funções de incidência, assim como os pré requisitos (eventos e/ou condições) que determinam a transição entre etapas, são definidas pelas regras do experimento ATLAS. Cada vértice pode estar associado a atividades como o envio de um *e-mail*, armazenamento de dados, atualização ou criação de um grupo de *e-mail*.

O grafo de uma publicação possui inúmeros possíveis passeios. O mais trivial é aquele que segue todas as etapas, da primeira até a última, sem repeti-las. Porém, pode também conter ciclos como, por exemplo, uma publicação que, após o processo de revisão, não é aprovada. Neste caso, é preciso retornar a uma etapa anterior correspondente ao início da revisão.

Para ilustrar o que foi dito até então, tomaremos como exemplo a fase de submissão de um *Paper*, descrita na seção 2.3, que considera as seguintes etapas:

- Submissão do *Paper* ao periódico científico;
- Correção dos comentários feitos pelo periódico;
- Aprovação da publicação pelo periódico.

Cada uma destas etapas é composta por atividades associadas a atores. As condições e eventos determinam a transição entre etapas ou o início de uma atividade. A atividade relativa à etapa 1 corresponde à ação de submeter o *Paper* à plataforma oficial do periódico, realizada por um membro do *Publication Committee*. A transição da etapa 1 para a 2 acontece quando é disparado o evento de recebimento dos comentários do periódico com relação ao artigo. A etapa 2 têm duas atividades, sendo a primeira realizada pelos membros do *Analysis Team*, que devem corrigir o documento de acordo com as modificações solicitadas e, ao fim desta, um membro do *Publication Committee* deve submeter novamente o documento corrigido à plataforma do periódico. A transição da etapa 2 para a 3 acontece quando o evento de ressubmissão do artigo é finalizado. A etapa 3 não tem atividades, pois todos os envolvidos estão esperando pela aprovação por parte do periódico, porém, esta

etapa tem uma condição: Caso o *Paper* seja aprovado, então o processo está completo. Caso haja reprovação, os autores devem corrigi-lo novamente. Assim, quando o evento de recebimento da resposta do periódico acontecer, o fluxo é finalizado ou retorna para a etapa anterior para uma nova correção, executando a transição da etapa 3 para a 2. A modelagem deste exemplo usando a notação apresentada na seção 3.2 pode ser vista na figura 3.3.

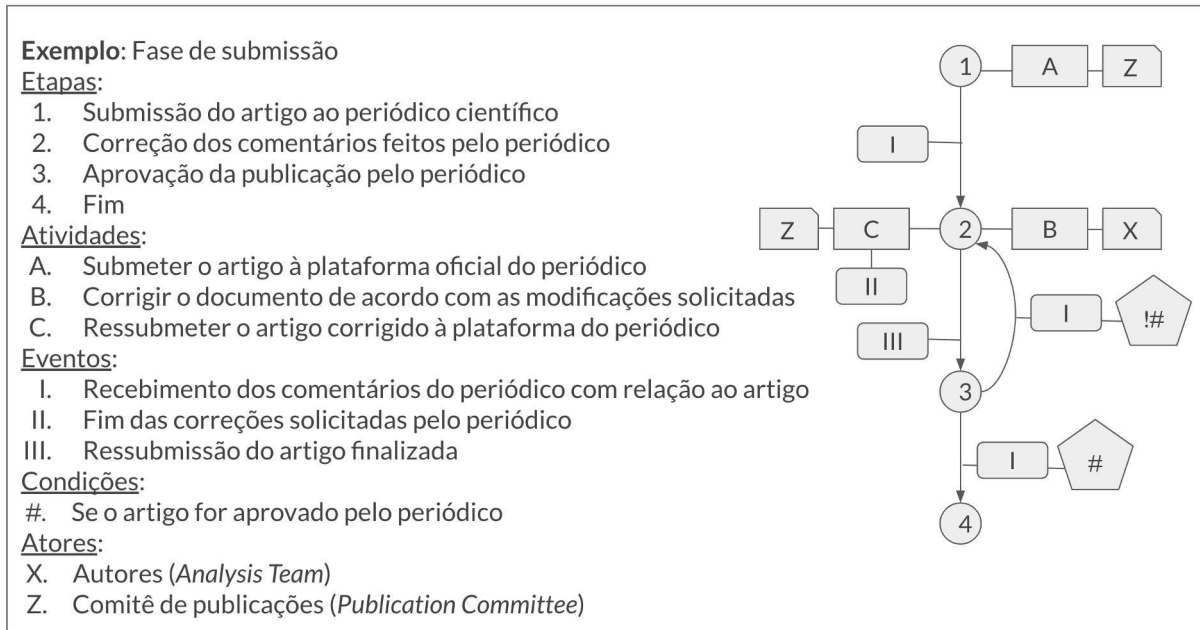


Figura 3.3: Modelagem da fase de submissão de um *Paper*.

Portanto, fica claro que o processo evolutivo de uma publicação pode ser representado pelo modelo incrementado de um grafo. Essa foi a estratégia adotada neste trabalho e os detalhes desta implementação serão mostrados no capítulo 5.

Capítulo 4

FENCE

Neste capítulo será apresentado o *framework* FENCE, utilizado no desenvolvimento da solução deste projeto. Primeiramente, será explicada sua estrutura geral, depois, serão citados alguns sistemas já implementados fazendo seu uso e, por fim, serão apresentadas suas principais classes.

4.1 O *framework*

Como já visto no capítulo 2, o CERN envolve o projeto e a construção de complexos aparatos físicos, a colaboração de pesquisadores provenientes de diferentes universidades e instituições de pesquisas internacionais e a divulgação das descobertas científicas através de publicações. Para gerenciar essa estrutura, os experimentos utilizam sistemas computacionais cujas aplicações vão desde o posicionamento dos cabos e equipamentos do detector até a escolha dos palestrantes de uma conferência. Conforme o avanço e evolução dos experimentos, é imprescindível que os softwares tenham que acompanhar tais mudanças, o que acarreta em uma constante alteração de regras e requisitos. Nesse contexto, o *framework* FENCE (*Front-End Engine for Glance*) foi projetado para apoiar o desenvolvimento e a manutenção dos sistemas Web implementados para o CERN.

A colaboração entre o ATLAS e a UFRJ, existente desde 1988, vem atuando nas áreas de Física, Engenharia Eletrônica e Engenharia de Software, envolvendo professores, pesquisadores, alunos de pós-graduação e graduandos. A área de computação atuou em diferentes aspectos do experimento, tais como banco de dados, simulação, análise e aquisição de dados, apoio à comunicação, acesso e publicação de informações. A equipe brasileira teve a oportunidade de trabalhar com a Web desde suas versões iniciais e construiu o primeiro site do experimento ATLAS. Outras aplicações utilizando a plataforma Web foram desenvolvidas e amplamente utilizadas por toda a colaboração. A experiência adquirida deu origem ao Glance em 2003, ferramenta que facilita o acesso a registros armazenados em bancos de dados através

de uma interface de busca automaticamente gerada, sem a necessidade de entendimento da modelagem do banco por parte do usuário. Tal interface possibilita um refinamento sucessivo dos dados através de operações lógicas e exibe o resultado em forma de tabela. Inicialmente, o Glance foi projetado para permitir acesso à informação dos milhares de cabos do detector ATLAS, mas diversos sistemas foram implementados para apoiar a gestão de equipamentos, dosagem da radioatividade, membros e institutos colaboradores, dentre outros. Os sistemas Glance também foram desenvolvidos para os experimentos ALICE e LHCb do CERN.

Em 2014, a equipe da UFRJ concebeu e implementou o FENCE, um *framework* orientado a objetos para apoiar o desenvolvimento de sistemas Web. Inicialmente, o FENCE se baseava no Glance para acessar os bancos de dados e gerar interfaces de busca. Com a evolução do *framework*, diferentes classes foram implementadas, tais como as que extraem dados se conectando diretamente aos repositórios e outras que apoiam a implementação de interfaces de busca. Desde então, o FENCE funciona de forma independente ao Glance pois aperfeiçoou e ampliou seus conceitos. O *framework* reúne diversas classes para inserir, atualizar e excluir registros, orientar as ações dos usuários, controlar o acesso às funcionalidades do sistema, gerar menus que apoiam a navegação, codificar regras e dar suporte a procedimentos definidos pelo experimento, incentivando a reutilização de código.

Internamente, O FENCE é vastamente apoiado no modelo de programação orientada a objetos (OOP), que define classes, métodos e atributos. As **classes** representam as entidades que compõem uma solução de software, tendo seu comportamento definido por seus **métodos**, que, por sua vez, podem alterar suas propriedades, chamadas de **atributos**. Quando uma classe é instanciada, cria-se um **objeto**. Um bom exemplo seria uma classe chamada Pessoa. Alguns de seus métodos seriam andar, falar e pensar e os seus atributos poderiam ser nome, idade e endereço.

O fator de escolha do paradigma OOP pelo *framework* deu-se pela possibilidade de herança, em que uma classe pode herdar os métodos e atributos de outra sem ser reescrita, sendo, portanto, uma ferramenta de redução de duplicação de código. Como muitos dos sistemas computacionais adotados pelos experimentos do CERN tem funcionalidades similares, o desenvolvimento de um novo sistema seria beneficiado por um *framework* com uma vasta biblioteca de classes que pudessem ser usadas por herança. Esse é o conceito do FENCE.

A fim de minimizar o impacto de alterações e permitir uma rápida resposta da equipe de desenvolvedores na manutenção de sistemas, o *framework* vincula arquivos de configuração às classes de sua biblioteca. Desta forma, uma classe pode acessar o arquivo e construir seus métodos e atributos de acordo com o seu conteúdo, que consiste nas regras particulares da interface de um sistema. Esse fato resume algumas mudanças de requisito na simples tarefa de alterar o texto de um arquivo de

configuração.

4.2 Sistemas FENCE

A demonstrada capacidade de reduzir esforços de manutenção e promover a rápida evolução de interfaces Web motivou a implementação de 20 sistemas com o *framework* FENCE. A figura 4.1 resume todos os sistemas desenvolvidos até então.

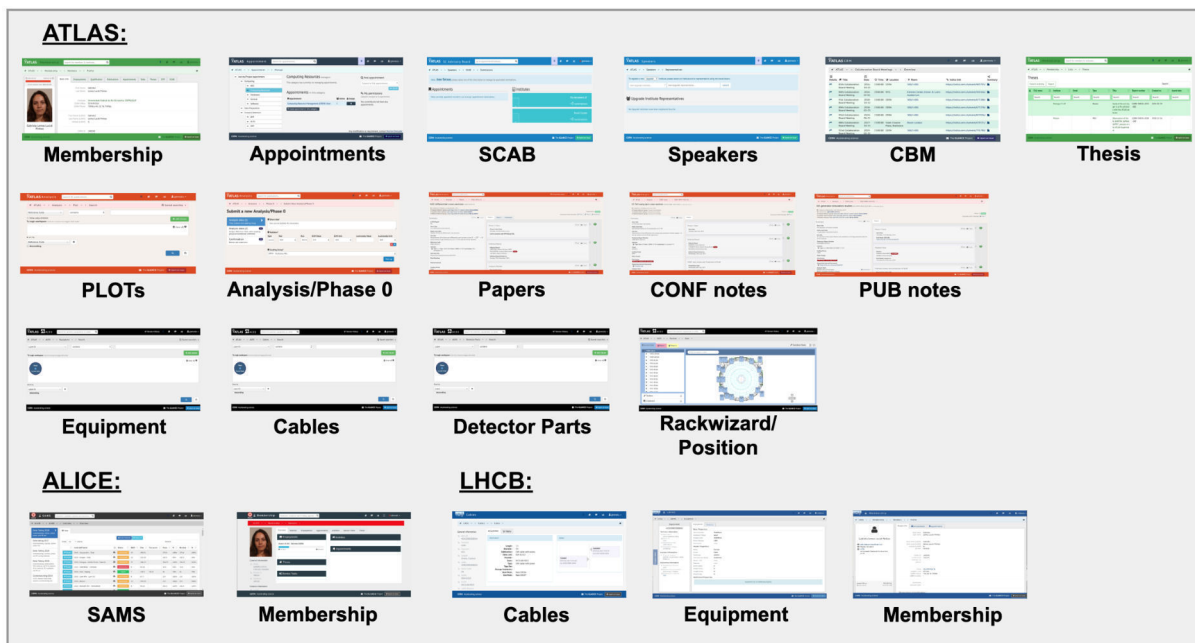


Figura 4.1: Sistemas FENCE desenvolvidos até Junho de 2019.

Os sistemas FENCE que pertencem ao experimento ATLAS podem ser agrupados em três categorias: pessoas, publicações e equipamentos. Os relacionados a pessoas tem funcionalidades para gerir informações pessoais dos membros do experimento, seus contratos, denominações, institutos, nomeações, conferências, teses e listas de autores. Os sistemas relativos a publicações automatizam o processo de produção de artigos científicos, artigos de conferência, notas públicas e gráficos. Aqueles referentes a equipamentos lidam com informações sobre partes do detector, seus cabos e outros tipos de equipamento do experimento.

O ALICE também apresenta sistemas para a gestão de informações pessoais de seus membros e para o controle de plantões da sala de controle do detector. Já os sistemas do experimento LHCb apresentam funcionalidades para gerência de membros, equipamentos e cabos.

4.3 Principais classes

O *framework* FENCE tem uma biblioteca de classes úteis ao desenvolvimento dos sistemas Web de gerenciamento dos experimentos do CERN. As mais relevantes para este trabalho serão introduzidas a seguir.

A classe **SuperSearch** possibilita a criação de interfaces de busca avançada por meio de expressões lógicas. Ela possibilita o usuário a elaborar consultas a repositórios com os operadores AND e OR, como pode ser observado na figura 4.2. Neste exemplo, a busca retornará uma lista das denominações do experimento ATLAS cujos membros sejam ativos, tenham data de início de contrato maior que 1/1/2019, data de fim de contrato menor que 30/6/2019 e afiliação com a Universidade Federal do Rio de Janeiro.

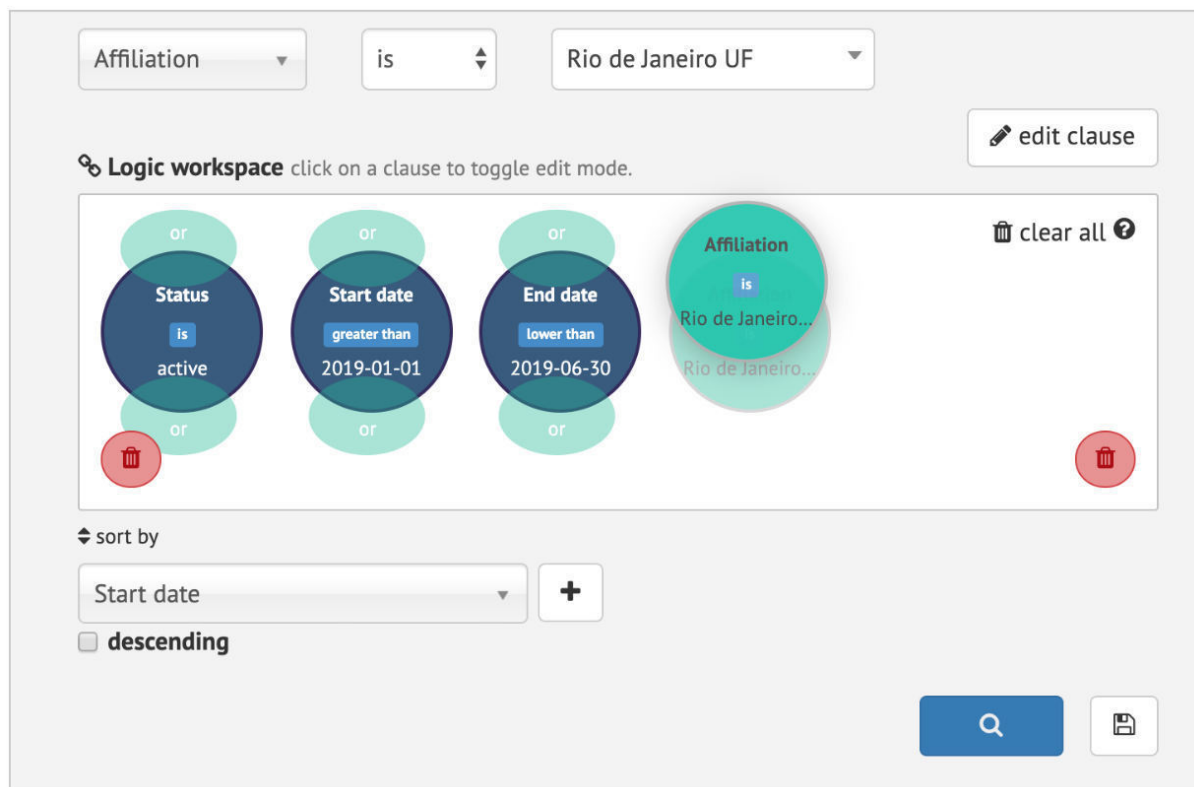


Figura 4.2: Funcionalidade de busca avançada de denominações dos membros do experimento ATLAS. Elementos ligados na horizontal representam o operador AND e na vertical o operador OR.

A classe **Form** é responsável por criar interfaces com formulários em HTML. Um formulário HTML é formado por um conjunto de campos que são enviados ao servidor para que seus dados sejam tratados e armazenados. A classe é responsável também por validar se as regras de cada campo estão sendo obedecidas e apenas envia o formulário ao servidor caso positivo. Tais campos e suas regras são definidos no arquivo de configuração relacionado à classe. Um exemplo pode ser observado na

figura 4.3, que apresenta a interface de inserção de um novo instituto do experimento ATLAS. O formulário é dividido em etapas, explicitadas no canto esquerdo. Quando a regra de algum campo é negligenciada, a classe impede o prosseguimento para a próxima etapa e explicita o erro ao usuário. Na imagem, o campo *short name*, que é um campo obrigatório, foi ignorado.

The image shows a web form interface for adding a new institute. On the left, there is a vertical sidebar with six steps: 'Basic info' (selected, with a right arrow), 'Location', 'Cluster', 'Affiliation', 'Contact', and 'Confirmation'. Each step has a small icon and a description. The main area contains four input fields: 'short name*' (empty, with a red error message 'short name: Field is required.' above it), 'full name*' (containing 'test'), 'domain' (empty), and 'webpage' (empty). A blue 'Next' button with a dropdown arrow is at the bottom right.

Figura 4.3: Funcionalidade de inserção de um novo instituto associado ao experimento ATLAS exemplificando a negligência de um campo obrigatório em um formulário.

A classe `User` é responsável pelo controle de acesso das interfaces. Tal classe consegue acessar o identificador de um usuário logado em um determinado sistema e também os grupos de *e-mail* aos quais pertence, no CERN, denominados *egroups* [8]. Tais informações são fornecidas pelo SSO (*Single Sign-On*), autenticação centralizada feita através de tecnologia Shibboleth [9], em que o usuário precisa fornecer suas credenciais ao acessar qualquer conteúdo dos servidores do CERN. Em posse disso, os sistemas que fazem uso do FENCE podem criar seus próprios métodos de permissão através da classe `User`, por exemplo, permitindo que apenas os usuários pertencentes a um determinado *egroup* possam ver o conteúdo completo de uma interface. Caso um usuário não tenha acesso a uma interface, observará uma notificação como a da figura 4.4. A mesma classe é utilizada para definir permissões de visualização e edição de campos em formulários.

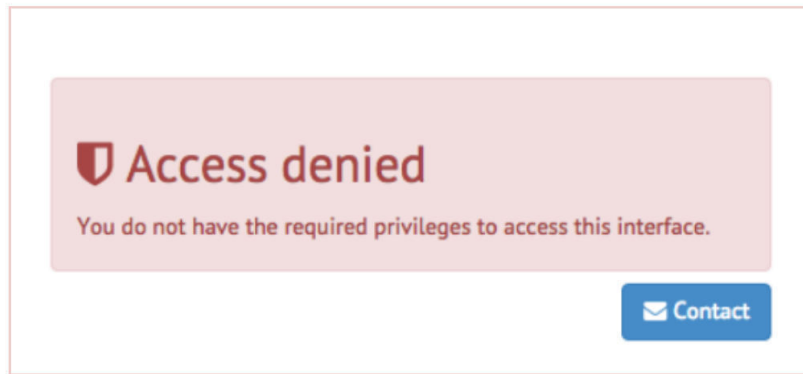


Figura 4.4: Alerta de acesso negado a uma interface de um sistema FENCE pela falta de credenciais.

A `DBManager` é usada para inserir, modificar, recuperar e excluir dados de um banco de dados. A classe apresenta métodos para estabelecer a conexão a um banco por meio de autenticação e, depois de validar as credenciais do sistema ou usuário, permite a definição e execução de comandos em SQL e as ações de *commit* e *rollback* para gravar as mudanças feitas pelos comandos ou desfazê-las.

A classe `Mailer` é usada para o envio de *e-mails* automáticos. Ela apresenta métodos para a definição de todos os parâmetros necessários ao envio de um *e-mail*, como: remetente, destinatários (principal, cópia e cópia oculta), assunto e conteúdo. Todos os *e-mails* encaminhados fazendo uso desta classe seguem um padrão de formatação e terminam com uma mensagem de como contatar o grupo de desenvolvedores. O exemplo de *e-mail* enviado por um sistema está na figura 4.5.

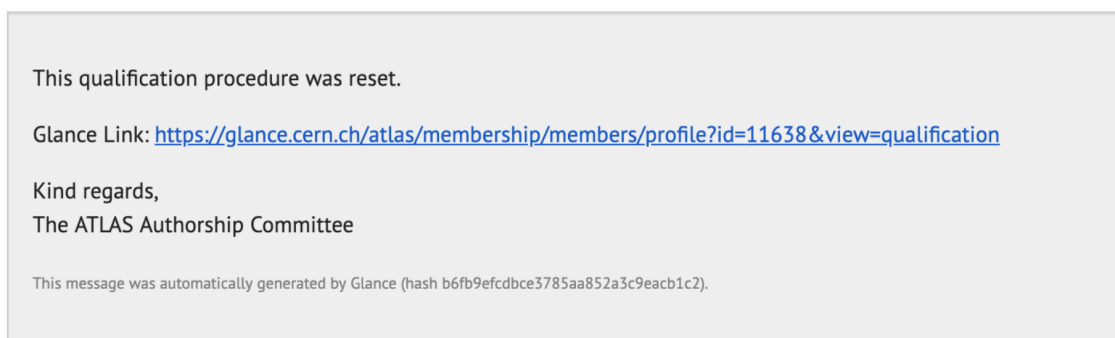


Figura 4.5: *E-mail* enviado ao reiniciar-se o processo de qualificação de um autor no sistema Membership do ATLAS.

Uma API (*Application Programming Interface*) apresenta um conjunto de instruções que permitem o uso dos serviços de uma aplicação por softwares externos. A classe `EgroupSOAPHandler` é responsável por permitir a conexão com a API do sistema de *egroups* [8] do CERN. Por meio dela é possível performar todas os serviços disponíveis na API, como por exemplo: criar um novo *egroup*, excluir um existente,

encontrar um *egroup* pelo nome tendo em resposta todos os seus metadados, adicionar membros e remover membros. Caso ocorram erros ao se comunicar com a API, a classe apresenta métodos para tratá-los.

A classe `Gitlab`, por sua vez, é responsável por permitir a conexão com a API do GitLab [3], um gerenciador de repositório de software baseado em Git [4], possibilitando diversas ações como: a criação e exclusão de repositórios, *branches*, *commits* e outros elementos detalhados na documentação oficial [2]. Caso ocorram erros ao se comunicar com a API, esta classe também apresenta métodos de tratamento de erro. Os principais métodos da classe `Gitlab` e também das classes às quais se relaciona podem ser observados no diagrama de classes da figura 4.6. O diagrama apresenta relações de associação unidirecional (representadas pela seta) entre a classe `Gitlab` e as outras, o que significa que os objetos das classes `Project`, `Group`, `Member`, `Branch`, `Label`, `File` e `Commit` são usados pela classe `Gitlab`.

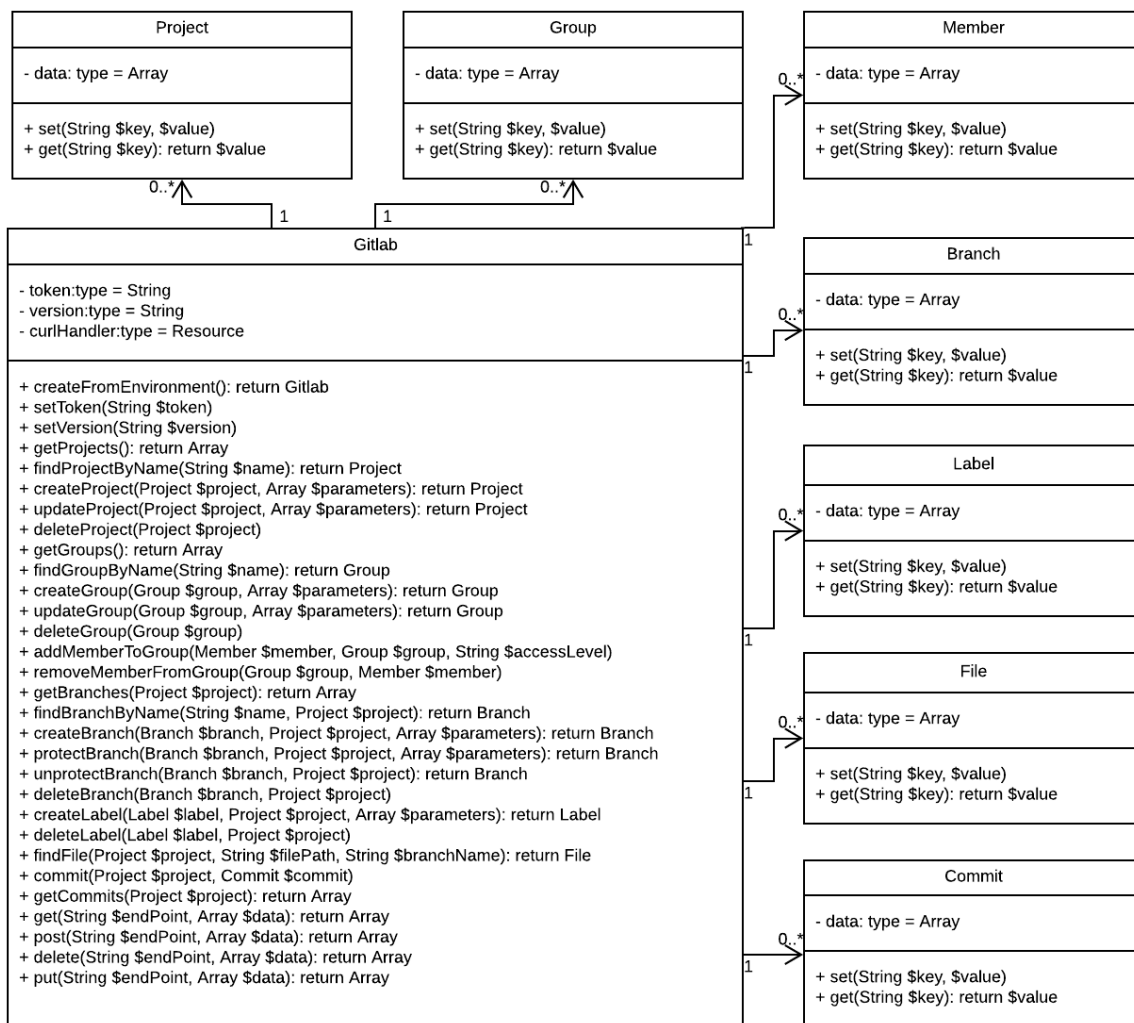


Figura 4.6: Diagrama de classes da estrutura criada para a comunicação com a API do GitLab [2]

Essas e muitas outras classes compõem o *framework* FENCE, mas as aqui mencionadas possuem grande relevância para este trabalho. Um resumo delas pode ser encontrado na figura 4.7.

Fence Main Classes				
AjaxDispatcher	DBManager	Inspector	Plot	Tree
Assembler	DBOf	JReader	Property	User
BaseInterface	Downloader	LogicTree	QuickSearch	View
Base Profile	EgroupSOAPHandler	Mailer	SearchInterface	
BaseSearch	EmailTemplateParser	Menu	SessionManager	
BaseSettings	EnviromentResolver	MPTree	SettingsInterface	
BaseUser	Factory Fence	Navigation	Settings	
Builder	Fieldset	Notification	Step	
ClauseFactory	FileDispatcher	OBJSon	StockSearch	
Clause	Form	Observer	SuperSearchInterface	
Configuration	Gitlab	Parameter	SuperSearch	
Content	Header	Payload	Table	
DBJReader		Picture	TokenManager	
DBJWriter				

Figura 4.7: Principais classes do *framework* FENCE dentre as 436 existentes. Em azul, as classes de maior relevância para a implementação dos sistemas Analysis.

Capítulo 5

O conjunto de sistemas Analysis

Até aqui foi relatada a problemática e os fundamentos teóricos utilizados neste trabalho. Neste capítulo será apresentada uma proposta de solução em formato de sistema Web, fazendo uso do *framework* FENCE.

5.1 Contribuições ao *framework* FENCE

Como antes apresentado no capítulo 4, o FENCE apresenta uma biblioteca de classes que auxiliam o desenvolvimento de sistemas Web. Porém, ainda não existiam classes em seu acervo adequadas para modelar a dinâmica das atividades que envolvem a evolução, revisão e aprovação de uma publicação do experimento ATLAS. Neste contexto, a classe `Workflow` foi projetada e implementada para modelar e automatizar qualquer fluxo de atividades e, desta forma, pode ser utilizada não só por este projeto mas também por qualquer sistema a ser desenvolvido utilizando o *framework* FENCE.

A classe `Workflow` foi desenvolvida com base na estrutura de um grafo e, para representá-lo, foi implementada a classe `Graph`, que apresenta apenas métodos abstratos a serem definidos por suas classe herdeiras. Um exemplo de classe que herda a `Graph` é a classe `MapperGraph`, responsável por armazenar vértices e arestas dentro de uma estrutura da linguagem PHP chamada `SplObjectStorage` [10], que, para essa implementação, pôde lidar melhor com objetos do que *arrays* associativos. Caso outra aplicação necessite de um *array* como estrutura de armazenamento, a classe `MatrixGraph` também foi disponibilizada na biblioteca do *framework* FENCE.

Também foram desenvolvidas classes para representar vértices e arestas chamadas `Node` e `Edge`, respectivamente. A classe `Node` estabelece métodos para definir e recuperar metadados relativos a um vértice. De forma análoga, classe `Edge` também apresenta tais métodos para definir e recuperar metadados relativos a uma aresta. Um exemplo de metadado que pode estar relacionado a um vértice é uma instância da classe `Action`, que, por sua vez, possui métodos para definir, recuperar e disparar

funções podendo escolher suas entradas e acessar sua saída. O diagrama de classes de todas as classes mencionadas nesta seção pode ser observado na figura 5.1. A relação entre as classes **Graph** e **MapperGraph** é de herança (representada pela seta de ponta triangular). Já o relacionamento entre as outras classes é de associação unidirecional, o que significa que a classe de origem faz uso dos objetos da classe alvo.

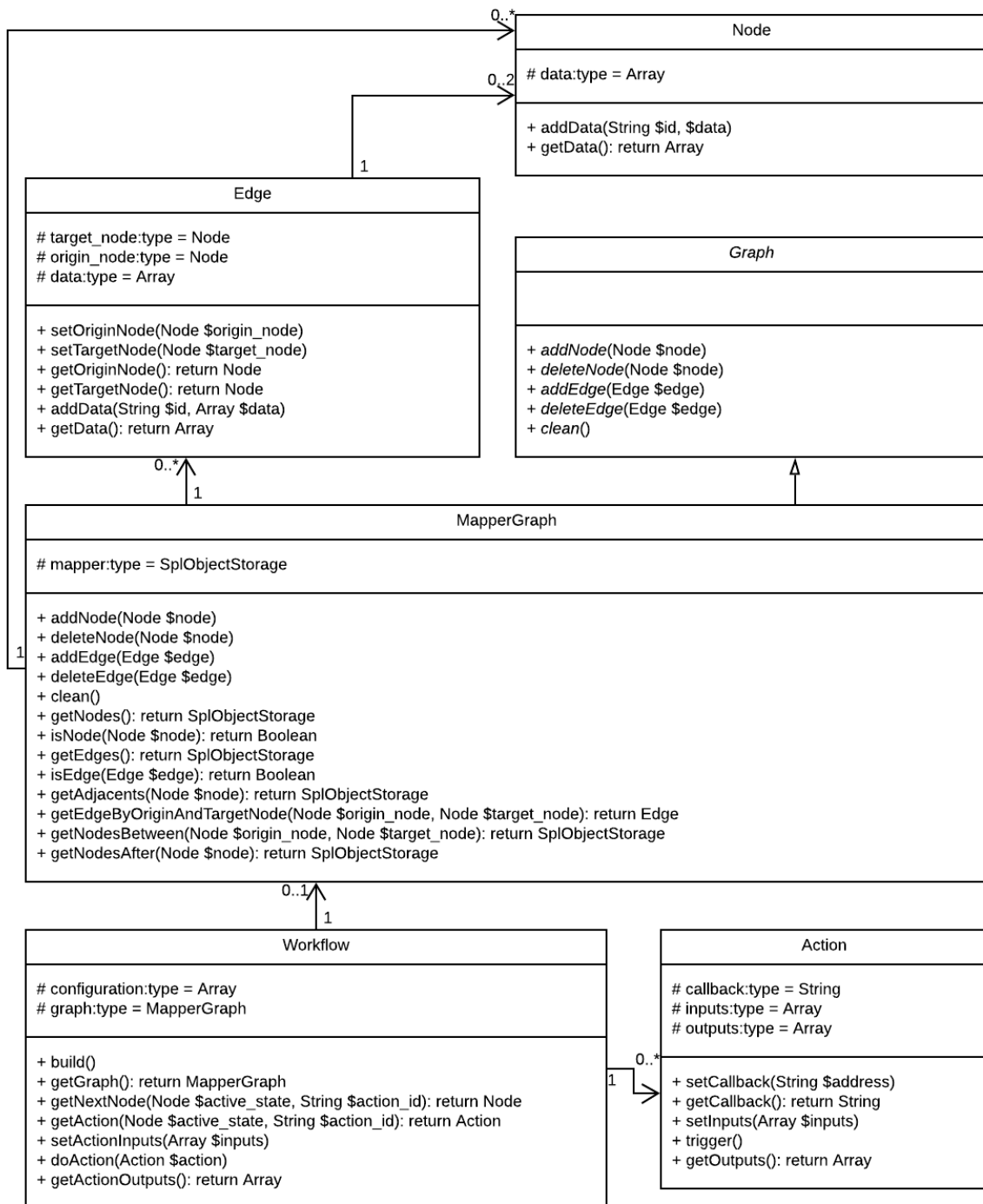


Figura 5.1: Diagrama de classes da estrutura criada para modelar fluxos de atividade.

A classe `Workflow` faz uso das classes `MapperGraph`, `Node`, `Edge` e `Action` para representar, baseando-se na estrutura de um grafo, um fluxo de atividades. A teoria desta modelagem foi vista na seção 3.2. Na prática, a classe cria objetos de vértices, arestas e ações a partir de propriedades definidas num arquivo de configuração, que será visto em mais detalhes na subseção 5.2.2.

Depois de criados, os objetos de vértices e arestas são armazenados de acordo com suas relações e as ações ficam junto ao vértice que pertencem. É, ainda, possível atrelar eventos e condições a objetos de ação e aresta, também a partir do arquivo de configuração. Uma ação vinculada a um evento só ocorre no vértice que está relacionada e quando o evento acontecer. Uma aresta vinculada a eventos e condições estabelece uma transição entre vértices apenas no vértice de origem, quando o evento acontecer e quando a condição for satisfeita.

A classe `Workflow`, além de ser usada como solução deste trabalho, é também aproveitada pelo sistema Membership do ATLAS, modelando o processo de qualificação de autores. Desta forma, fica clara a contribuição deste trabalho ao *framework*, adicionando classes em sua biblioteca que podem ser usadas por diversos sistemas. As novas classes FENCE adicionadas podem ser observadas na figura 5.2, em verde.

Fence Main Classes				
Action	DBOf	JReader	Picture	Tree
AjaxDispatcher	Downloader	LogicTree	Plot	User
Assembler	Edge	Mailer	Property	View
BaseInterface	EgroupSOAPHandler	MapperGraph	QuickSearch	Workflow
Base Profile	EmailTemplateParser	MatrixGraph	SearchInterface	
BaseSearch	EnvironmentResolver	Menu	SessionManager	
BaseSettings	Factory Fence	MPTree	SettingsInterface	
BaseUser	Fieldset	Navigation	Settings	
Builder	FileDispatcher	Node	Step	
ClauseFactory	Form	Notification	StockSearch	
Clause	Gitlab	OBJSon	SuperSearchInterface	
Configuration	Graph	Observer	SuperSearch	
Content	Header	Parameter	Table	
DBJReader	Inspector	Payload	TokenManager	
DBJWriter				
DBManager				

Figura 5.2: Principais classes do *framework* FENCE dentre as 436 existentes. Em verde, as classes adicionadas à biblioteca em consequência deste trabalho.

5.2 Sistemas Analysis

No intuito da automatização do processo de concepção, evolução, revisão e aprovação de publicações, cinco sistemas foram desenvolvidos: Papers, CONF notes,

PUB notes, PLOTs e Analysis/Phase 0. Os 4 primeiros serão aqui brevemente descritos e o último terá sua implementação mostrada em detalhes.

O processo de produção de uma publicação se inicia com a fase 0 e, para o seu suporte, o sistema Analysis/Phase 0 foi implementado. A publicação, quando sua primeira versão estiver pronta, pode evoluir para um *Paper*, *CONF note* ou *PUB note*, sendo que para cada um destes documentos foi criado um sistema dedicado e de mesmo nome. Para a revisão e aprovação de um gráfico, criou-se o sistema PLOTs, independente dos outros. Tal relação entre a entidade principal de cada sistema é evidenciada pela figura 5.3.

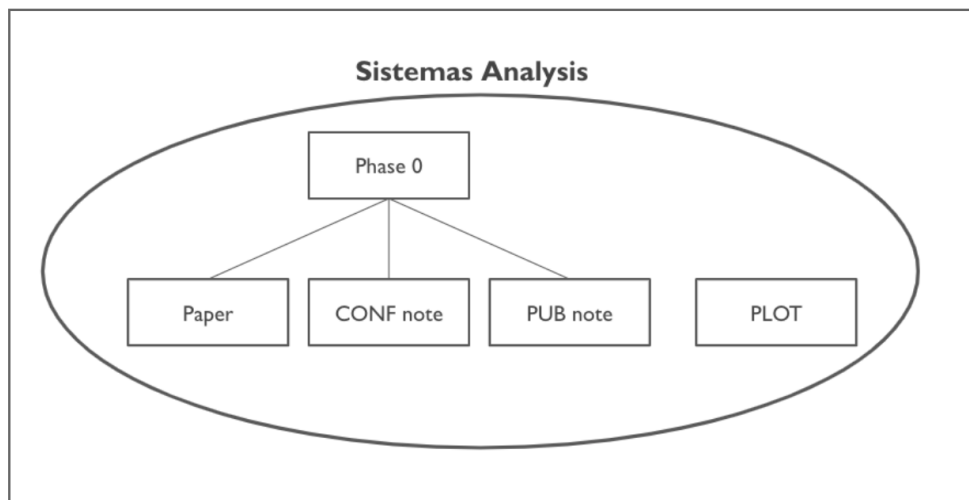


Figura 5.3: Relação de dependência entre as entidades dos sistemas Analysis.

Antes do detalhamento dos sistemas Analysis, o que será feito a seguir, é válido mencionar uma diferença de implementação entre eles. Os sistemas Papers, CONF notes e PUB notes já existiam antes do lançamento do *framework* FENCE em 2014 e, portanto, foram migrados, passando a fazer uso da nova tecnologia. Os sistemas PLOTs e Analysis/Phase 0 foram sistemas FENCE desde sua concepção e, por esse motivo e também por apresentar muitas funcionalidades em comum aos outros sistemas Analysis, o sistema Analysis/Phase 0 será, neste trabalho, mais detalhado do que os demais.

5.2.1 Os sistemas Papers, CONF notes, PUB notes e PLOTs

O sistema Papers contempla artigos científicos que serão submetidos, depois de aprovados, a um periódico científico. Apresenta funcionalidades para inserção, recuperação, edição e deleção das propriedades de um *Paper* num banco de dados além de gerenciar o fluxo de atividades das suas três fases: 1, 2 e submissão.

O sistema CONF notes contempla os artigos que devem ser apresentados em uma conferência, o PUB notes, notas públicas que devem ser apresentadas à comunidade

científica sem que tenham que ser submetidas a um periódico ou mesmo apresentadas numa conferência e, o PLOTs, os gráficos a serem usados para apresentar resultados em todos os outros tipos de publicação mencionados até então. Todos apresentam funcionalidades para inserção, recuperação, edição e deleção das propriedades de suas entidades num banco de dados e também gerenciam o fluxo de atividades da fase única de cada sistema.

Os sistemas apresentados são bastante semelhantes, diferindo apenas no número de fases e das etapas envolvidas em cada uma. Assemelham-se, também, as ações relacionadas a cada etapa, podendo estas serem: o salvamento de dados no banco de dados, envio automático de *e-mails* ou até mesmo a criação ou atualização de *egroups* [8].

As versões FENCE dos sistemas PUB notes, CONF notes e Papers estão sendo utilizadas pelos usuários desde de 2016. Já o sistema PLOTs teve seu lançamento em 2018.

5.2.2 O sistema Analysis/Phase 0

A necessidade do sistema Analysis/Phase 0 surgiu em 2017, quando o setor de TI do ATLAS deprecou o sistema de controle de versão SVN (*Apache Subversion*) [11] e incentivou que seus membros e autores passassem a utilizar o Git [4] por sua característica descentralizada e mais adaptada à realidade de seus colaboradores. Com essa mudança, o experimento passou a utilizar o gerenciador de repositórios GitLab [3] por apresentar funcionalidades de integração contínua, possibilidade de armazenamento de repositórios em servidores privados e por disponibilizar uma API dotada de inúmeros serviços. Como os sistemas Analysis Papers, PUB notes e CONF notes abrigavam as principais informações das publicações em seus bancos de dados, foi requisitado um novo sistema Analysis que pudesse se comunicar com a API do GitLab e criar grupos e repositórios Git automaticamente configurados e com os dados particulares de cada publicação previamente preenchidos em certos arquivos. Para formalizar a criação dos repositórios no início da escrita de publicações, surgiu o conceito da fase 0.

A nova ideia, porém, passou pouco tempo abrigando apenas a integração com o GitLab. Pensou-se que esta também poderia incluir um fluxo de atividades não formalizadas que aconteciam numa fase preliminar da escrita de publicações, quando ainda não se sabe se uma descoberta científica ou conhecimento deverá se materializar num artigo científico, artigo de conferência ou nota pública.

Principais funcionalidades

O sistema apresenta três interfaces principais. A primeira contém a funcionalidade de inserção de uma nova publicação, usada quando os cientistas do experimento ATLAS decidem publicar os resultados de seus trabalhos e precisam definir os principais dados do artigo ou nota pública para iniciar sua escrita. A interface apresenta um formulário cujos campos definem os dados de uma nova publicação como título, código de referência, grupos, subgrupos, palavras-chaves e outros. A segunda interface apresenta a funcionalidade de busca, onde é possível pesquisar por publicações por meio da aplicação de filtros e elaborar relatórios com a tabela de resultados. A terceira, a principal, apresenta e torna possível a edição das informações de uma publicação em detalhes, facilita o acompanhamento e evolução da fase 0 além de viabilizar a criação automática de repositórios Git [4].

A funcionalidade de inserção pode ser vista na figura 5.4. Por meio desta, um usuário faz a submissão do formulário por meio de etapas. Os campos de cada etapa podem ter seu preenchimento obrigatório, sendo representados por um asterisco (*). Informações de como preencher cada campo e suas regras são exibidas pelo clique no ícone “i” ao lado do nome do campo. Ao fim de todas as etapas, existe a etapa de confirmação, onde o usuário pode conferir se todos os campos foram preenchidos corretamente. Caso positivo, as informações do formulário podem ser armazenadas no banco de dados, que passa a reunir as informações iniciais que definem uma publicação como seu título e código de referência.

The screenshot shows the ATLAS Analysis submission interface. The top navigation bar includes the ATLAS logo and a search bar. The breadcrumb trail indicates the user is in 'ATLAS > Analysis > Phase 0 > Submit New Analysis/Phase 0'. The main heading is 'Submit a new Analysis/Phase 0'. On the left, a sidebar lists the submission steps: 'Analysis data (1)' (Title, Collision and Leading Group), 'Analysis data (2)' (Analysis Reference Code, other working groups and additional comments), and 'Confirmation' (Review and submission). The main form area contains the following fields:

- Short title***: A text input field with a note 'Text cannot exceed 40 characters.'
- Collision***: A table of input fields:

Type	Year	Run	ECM Value	ECM Unit	Luminosity Value	Luminosity Unit
p-p	N/A	N/A	0.9	N/A		N/A
- Leading Group***: A dropdown menu with the selected value 'BPHY - B physics WG'.

A 'Next' button is located at the bottom right of the form area. The footer contains the CERN logo, 'Accelerating science', 'The GLANCE Project', and a 'report an issue' button.

Figura 5.4: Funcionalidade de inserção de publicações do sistema Analysis/Phase 0. À esquerda, um resumo das etapas necessárias para completar a submissão dos dados. À direita, os campos presentes na primeira etapa.

A funcionalidade de busca avançada do sistema Analysis/Phase 0 pode ser vista na figura 5.5. Os critérios de busca são formados por três campos sendo o primeiro para a definição de um atributo de publicação, o segundo para seleção de um comparador e o terceiro para a inserção do valor. Por exemplo, para adição do critério “publicações que pertencem ao grupo Higgs”, basta a seleção do atributo “Grupo”, do comparado “Igual a” e da definição do valor “HIGG”. Um ou mais critérios de busca podem ser selecionados e arranjados, formando expressões lógicas por meio dos operadores AND e OR. É possível, ainda, definir a ordenação dos resultados a partir de um atributo em ordem crescente ou decrescente.

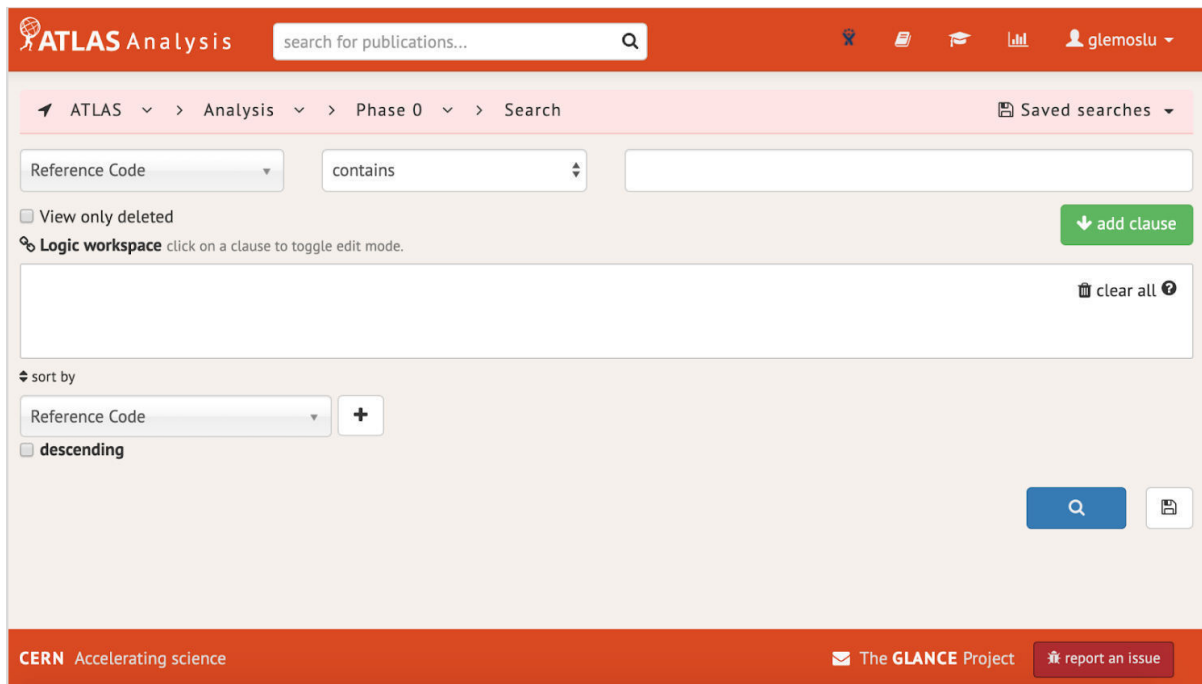


Figura 5.5: Funcionalidade de busca de publicações do sistema Analysis/Phase 0. Apresenta campos para definição de critérios de busca, que podem ser adicionados ao espaço denominado *Logic Workspace* formando expressões lógicas.

Por fim, a mais complexa interface, a interface de detalhes da publicação pode ser vista na figura 5.6. Nesta, deve ser possível visualizar e editar metadados além de controlar o fluxo de atividades da fase 0, dando ao usuário uma visão geral de todas as suas etapas, evidenciando as passadas, a atual e as posteriores, possibilitando-o tomar as providências necessárias para seguir da primeira até a última etapa da fase.

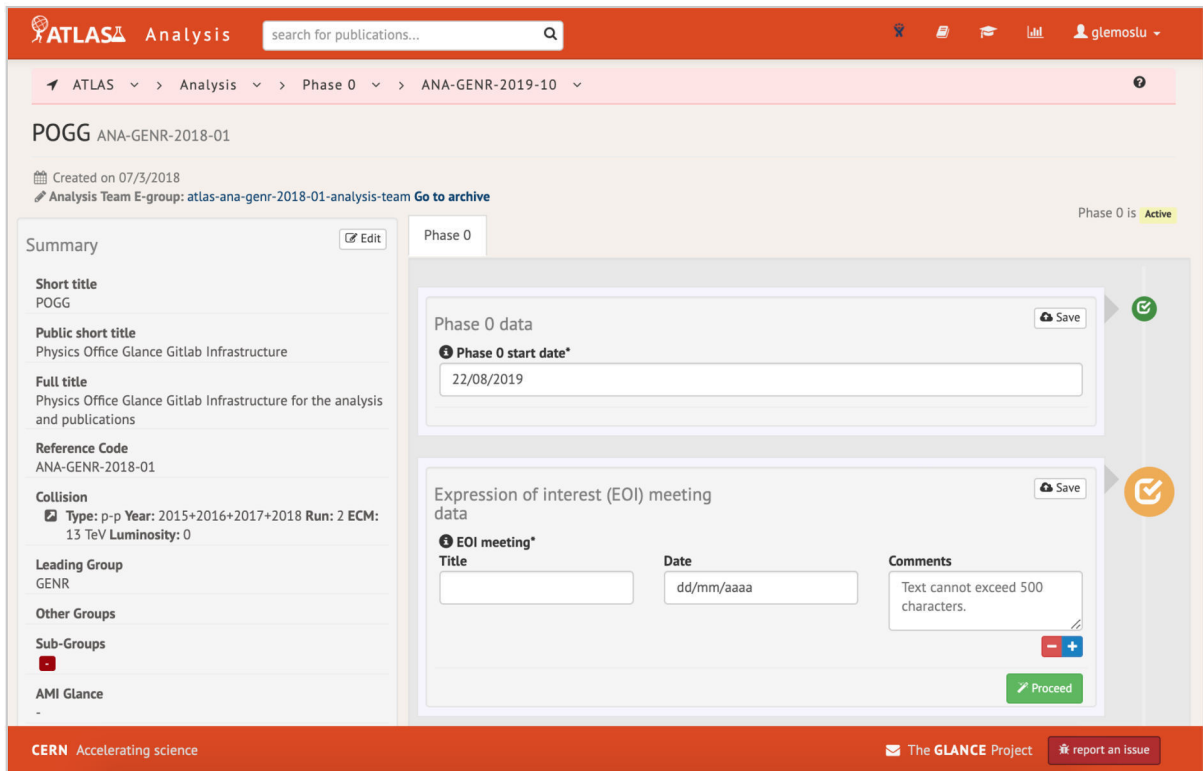


Figura 5.6: Interface de detalhes de uma publicação e sua fase 0. À direita, um resumo com as informações mais importantes de uma publicação. À esquerda, as etapas correspondentes ao fluxo de atividades da fase 0.

A transição entre as etapas da fase 0 dispara ações e a mais comum é o armazenamento de dados no banco de dados. Como pode ser observado na figura 5.6, uma etapa contém campos a serem preenchidos. O usuário, caso tenha permissão, tem a opção de salvar os dados no repositório e permanecer na mesma etapa ao pressionar o botão “Save” ou de salvar os dados e ir para a etapa seguinte ao pressionar o botão “Proceed”.

Outra ação recorrente é o envio automático de **e-mails** na mudança de etapas. Desta forma, os responsáveis por preencher os dados da etapa são notificados do que deve ser feito.

Outro exemplo é a de criação ou atualização de grupos de *e-mail*, os chamados *egroups* [8]. Um exemplo desta função pode ser visto na etapa de definição do Editorial Board, em que membros do experimento ATLAS são selecionados como parte deste conselho. Um *egroup* formado pelos conselheiros precisa ser criado para que um *e-mail* enviado a um único endereço seja recebido por todos. Quando o conselho muda os seus membros, o *egroup* precisa ser atualizado.

Modelagem de Dados

Depois de estudadas as funcionalidades do sistema Analysis/Phase 0, foi possível identificar as entidades do sistema com seus atributos e relações. Será feito, a seguir, uma simplificação desta modelagem.

A principal entidade do sistema é uma **Publicação**, que guarda em seus atributos características como título, código de referência e data de criação.

Uma publicação sempre está relacionada a um **Grupo** e, na maioria das vezes, **Subgrupos**, cujos atributos são nome e descrição.

Membros do experimento ATLAS estão relacionados a uma publicação por meio de **Papéis**, como, por exemplo, o grupo de pesquisadores que constitui o *Analysis Team* ou o *Editorial Board*. Um membro tem como alguns de seus atributos o nome, sobrenome e principal *e-mail* para contato. Os atributos de um papel são seu nome, tipo, data de início e fim.

Uma publicação contém fases, neste sistema, apenas a **Fase 0**, outra entidade cujos atributos são data de início da fase e o status de qual etapa se encontra.

Durante as etapas da fase 0, muitas **Reuniões** acontecem e seus atributos são título, data e comentários.

Alguns **Conteúdos** externos estão associados à Fase 0, como, por exemplo, artigos de suporte à publicação e atas de reunião que estão armazenados no servidor de documentos do CERN. Essa entidade tem como atributo o nome do conteúdo, seu tipo e seu endereço na Web.

À fase 0 também são atribuídos **Prazos** para que seus atores finalizem suas atividades. Um prazo tem como atributos o seu tipo e sua data.

A modelagem de dados simplificada descrita anteriormente é ilustrada na figura 5.7 por meio de um modelo entidade relacionamento. Todas as entidades foram representadas por meio de tabelas no banco de dados do sistema Analysis/Phase 0 e seus registros podem ser preenchidos ou modificados por meio da funcionalidade de inserção de uma nova publicação ou por meio das operações com banco de dados presentes na funcionalidade de controle do fluxo de atividades da fase 0.

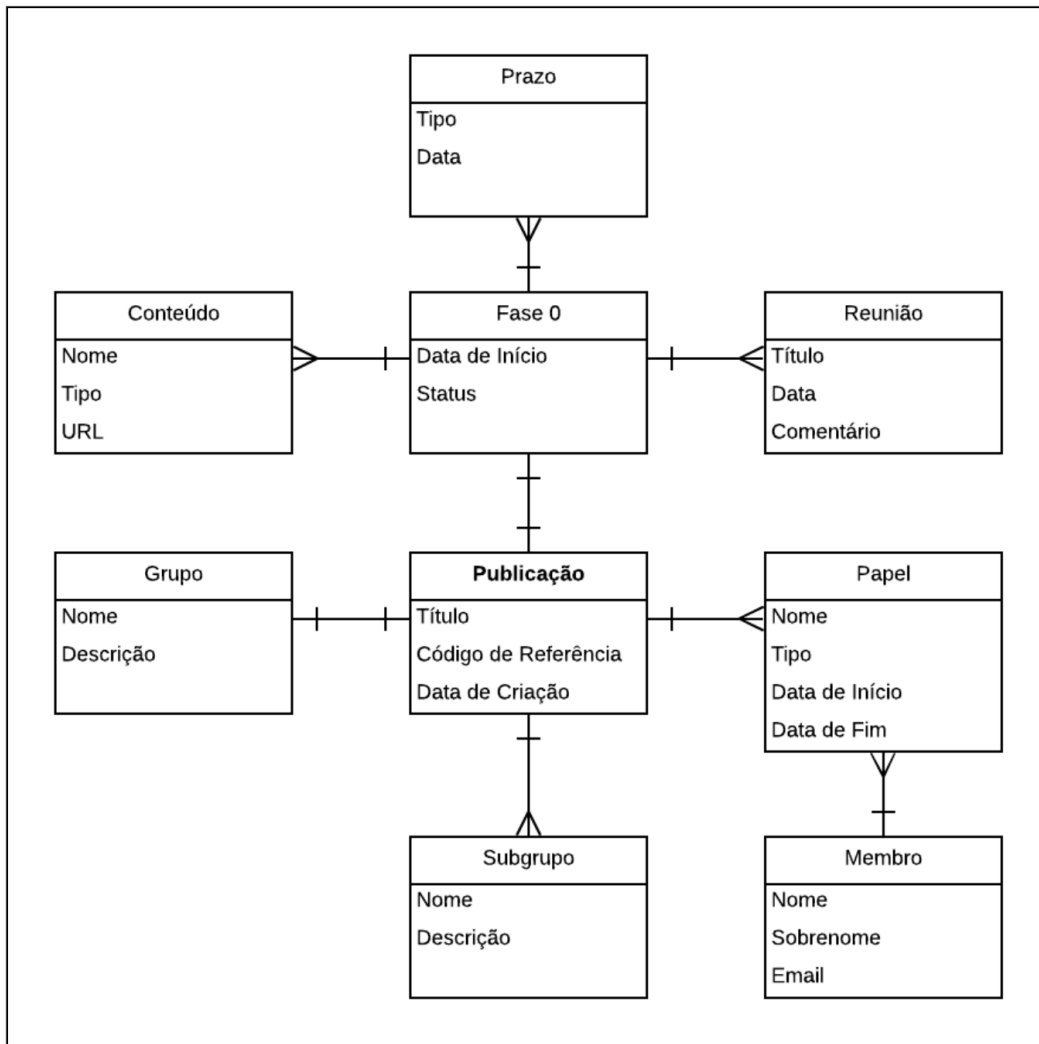


Figura 5.7: Modelo entidade relacionamento simplificado do sistema Analysis/Phase 0.

A funcionalidade de inserção de nova publicação apresenta um formulário com campos para a definição do título, código de referência e data de criação de uma publicação, seus grupos, subgrupos e os membros que farão parte do conjunto de autores. Tais dados são armazenados nas tabelas referentes a publicações, grupos, subgrupos, papéis e membros no banco de dados. A funcionalidade de controle do fluxo da fase 0 apresenta campos para definição de reuniões, prazos e conteúdos relacionados à fase 0 e estes são armazenados em suas respectivas tabelas.

Uso da classe Form

A funcionalidade de inserção de uma nova publicação faz uso da classe `Form` do *framework* FENCE. Para a sua construção, foi necessária a criação de três arquivos. O primeiro é um arquivo em PHP [12] que cria uma classe que herde a classe `Form`, definindo apenas um método chamado `post()` responsável por receber os dados do

formulário e inseri-los no banco de dados. O segundo é um arquivo de configuração em JSON [13] contendo as propriedades dos campos do formulário, como, por exemplo, o tipo (texto, data ou número) e regras (obrigatoriedade do preenchimento do campo ou número máximo de caracteres). Como se trata de um formulário longo, define-se, também, no arquivo de configuração a divisão dos campos entre etapas ou passos. O terceiro é um arquivo em PHP [12] que será acessado pelo usuário através do endereço inserido no navegador, criando apenas uma instância da classe herdeira da **Form** e estabelecendo como seu atributo o caminho até o arquivo de configuração.

Uso da classe **SuperSearch**

A funcionalidade de busca utiliza a classe **SuperSearch** do FENCE. A sua construção necessita de apenas quatro arquivos. Os dois primeiros são arquivos de configuração em JSON [13] que, em um, deve existir a definição dos tipos dos campos dos atributos de pesquisa que vão determinar a lista de comparadores (por exemplo, para um número: igual a, maior que, menor que) e, no outro, deve existir a definição dos atributos que devem estar presentes no resultado da busca. O terceiro é um arquivo na linguagem SQL contendo as instruções necessárias para realizar a pesquisa num banco de dados. O quarto é um arquivo em PHP [12] que será acessado pelo usuário através do endereço inserido no navegador, criando uma instância da classe **SuperSearchInterface** e estabelecendo como seus atributos os caminhos até os arquivos de configuração e SQL.

Uso da classe **Workflow** e suas auxiliares

A Classe **Workflow** é a principal classe utilizada pelo sistema Analysis/Phase 0, pois é a responsável pelo fluxo entre as etapas desta fase. O comportamento desta classe é comandado por um arquivo de configuração em formato JSON [13], cujo modelo pode ser observado na figura 5.8. Esse arquivo define todas as etapas de uma fase dentro da chave **nodes**, tendo, para cada etapa, um nome na chave **label**, um identificador na chave **id**, os campos correspondentes na chave **input** - que devem ser preenchidos com seus nomes, identificadores, tipos e regras - e os eventos que podem ser disparados na chave **events**. Para cada evento, é definido o identificador da próxima etapa na chave **next_state**, as condições para a transição, quando presentes, na chave **conditions**, as ações que serão disparadas na transição de uma etapa para a sua sucessora na chave **actions** e a lista de grupos com permissão de disparar o evento na chave **usergroups**. Um exemplo com os dados de uma etapa do sistema Analysis/Phase 0 é mostrado no apêndice A.

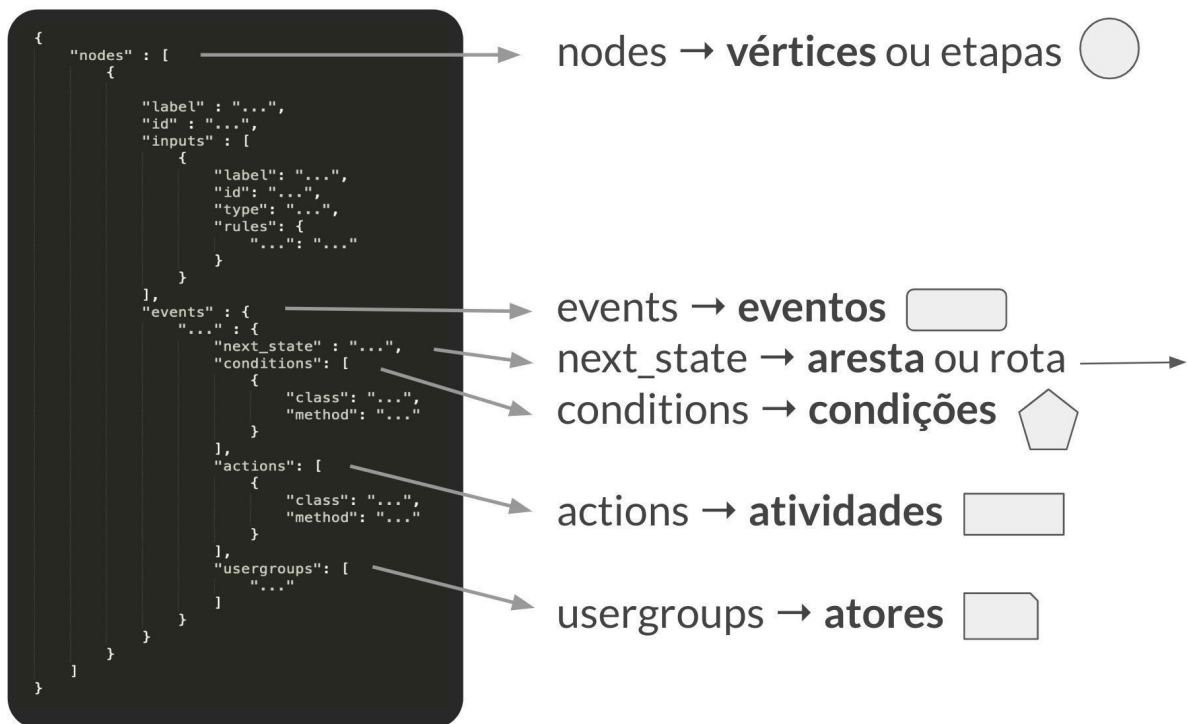


Figura 5.8: Modelo de um arquivo de configuração da classe `Workflow`.

Depois da construção do arquivo de configuração, a classe `Workflow` transformará seus elementos em objetos das classes `Node`, `Edge` e construirá um grafo a partir da classe `Graph`. Porém, para a construção dos objetos da classe `Action`, é necessária a definição dos métodos cujos endereços estão nas chaves `actions` do arquivo de configuração.

Caso a ação a ser realizada seja a inserção, atualização ou exclusão de dados no banco de dados, um método deve ser construído para, primeiramente, definir um comando SQL `INSERT`, `UPDATE` ou `DELETE` referente ao registro a ser alterado no banco. Depois, deve ser criada uma instância da classe `DBManager` e uma chamada ao método `connect()` deve ser feita para estabelecer a conexão com o banco. O comando SQL deve ser executado por meio do método `execute()` e a confirmação das alterações nos dados pelo método `commit()`.

Se o intuito de uma ação for o envio de um *e-mail*, o método deve criar uma instância da classe `Mailer` e definir o título do *e-mail* pelo método `setSubject()`, os destinatários pelo método `addRecipient()` e o conteúdo pelo método `setMessage()`. O encaminhamento do *e-mail* ocorre ao ser chamado o método `fire()`.

Para as ações de criar, modificar ou excluir *egroups* [8] o método deve criar uma instância da classe `EgroupSOAPHandler` e realizar uma chamada ao seu método `initialize()` para estabelecer a conexão com a respectiva API. Para a criação ou modificação de *egroups*, usa-se o método `synchronizeEgroup()`. Para a exclusão,

usa-se o método `deleteEgroup()`.

Integração com o GitLab

Para abrigar os grupos e repositórios a serem criados através do sistema Analysis/Phase 0, existe uma estrutura previamente definida no GitLab [3]. O grupo principal chamado “atlas-physics-office” abriga subgrupos. Cada um desses subgrupos pertence a um grupo do ATLAS, como HIGG e BPHY, mencionados na seção 2.3.

Toda vez que uma nova publicação é cadastrada no sistema Analysis/Phase 0, um novo grupo deve ser criado dentro do subgrupo correspondente ao grupo ATLAS relacionado à publicação. O nome do grupo deve ser o mesmo nome do código de referência da publicação, normalmente composto pelo seguinte padrão: ANA-(grupo)-(ano)-(número).

Dentro do grupo pertencente à publicação podem haver quatro tipos diferentes de repositórios: INT para notas internas; PAPER para *Papers*; CONF para *CONF notes*; PUB para *PUB notes*. Pode haver um ou mais repositórios do tipo INT, porém para PAPER, CONF e PUB só deve haver um de cada.

A figura 5.9 mostra um exemplo de grupos e repositórios dentro do grupo “atlas-physics-office”. Dentro do grupo de HIGG, existem vários grupos de publicações. O ANA-HIGG-2017-08 contém dois repositórios, um do tipo INT, chamado ANA-HIGG-2017-08-INT1 e outro do tipo PAPER chamado ANA-HIGG-2017-08-PAPER.

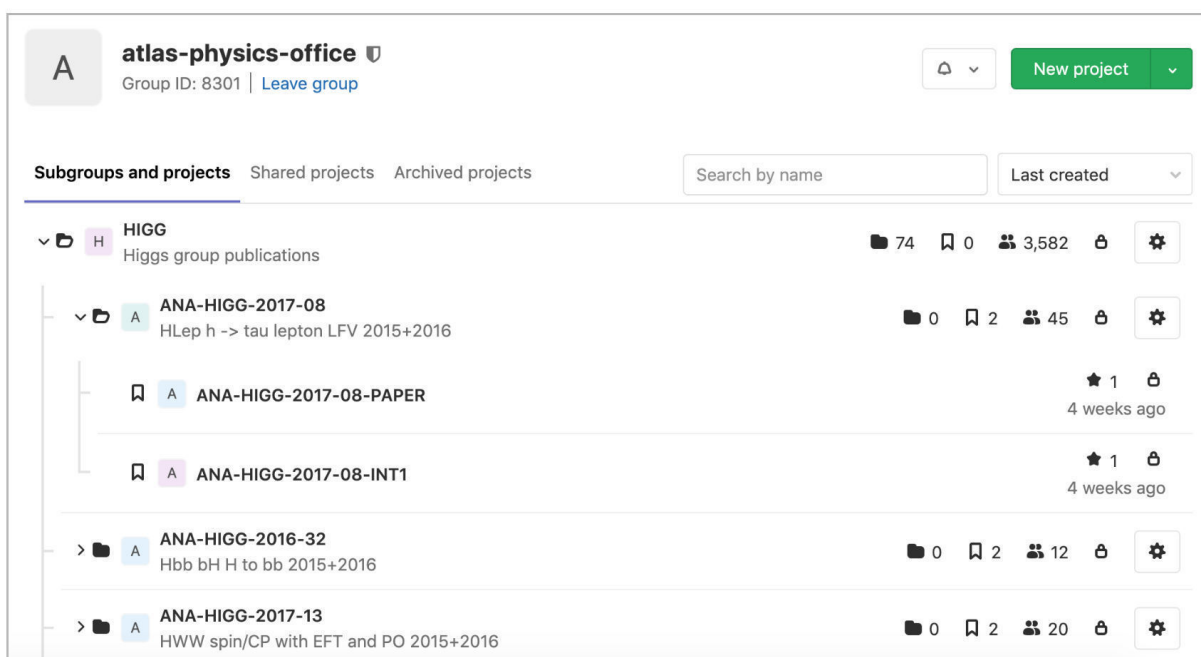


Figura 5.9: Estrutura criada no GitLab [3] para o armazenamento de repositórios Git de publicações do experimento ATLAS.

Uma vez que o grupo da publicação está criado, o sistema Analysis/Phase 0 adiciona, de acordo com o comando do usuário, repositórios dentro dele. Tais repositórios devem ser construídos com as seguintes regras:

- A *branch* “master” deve ser desprotegida;
- Uma nova *branch* chamada “PO-ready” deve ser criada e protegida;
- Uma *label* de cor vermelha chamada “PO-Publication” deve ser criada;
- Os membros do *egroup* do *Analysis Team* devem ter permissão de desenvolvedor dentro do grupo da publicação;
- Deve ser adicionado um primeiro *commit* ao repositório com as seguintes regras:
 - Os arquivos de um outro repositório chamado “atlaslatex” devem ser adicionados ao *commit*, com exceção de alguns;
 - Alguns arquivos de um outro repositório chamado “gitlab-integration” devem ser adicionados ao *commit*, porém, em alguns deles, devem ser substituídas as variáveis: `project_name` e `project_group`.

Todas essas ações podem ser realizadas por comandos à API do GitLab [2] disponibilizados pela classe `Gitlab` do FENCE. Por exemplo, para criar um grupo no GitLab deve ser chamado o método `createGroup()` e para criar um repositório Git [4] deve ser chamado o método `createProject()`.

Por meio da interface de detalhes do sistema Analysis/Phase 0, os usuários podem requisitar a criação de repositórios dos quatro tipos. A opção para a criação de um repositório tipo INT está sempre disponível, porém as opções para os tipos PAPER, CONF e PUB desaparecem quando já existe um repositório do mesmo tipo.

Resultados do sistema Analysis/Phase 0

O sistema Analysis/Phase 0 foi transferido para ambiente de produção em março de 2018 depois de elaborado e executado um plano de testes para garantir a qualidade e correção do código. Está sendo utilizado pelos autores de publicações do ATLAS, possibilitando que se concentrem no conteúdo de seus artigos científicos, abstraindo os aspectos técnicos do processo editorial que foram automatizados por este trabalho.

Com a criação do sistema Analysis/Phase 0, todo o fluxo de atividades que acontecia antes da fase 1 de uma publicação foi formalizado e todo metadado relacionado à nova fase está agora armazenado e organizado em uma interface Web. Alguns resultados quantitativos podem ser listados:

- Número de publicações com fase 0 criadas (de Março de 2018 até Junho de 2019): 598
- Número de *egroups* [8] colaborativos criados automaticamente (de Março de 2018 até Junho de 2019): 1099
- Número de repositórios GitLab [3] criados automaticamente (de Março de 2018 até Junho de 2019): 1000
- Média de usuários por dia (de Janeiro a Junho de 2019, em detalhes nas figuras 5.10 e 5.11): 23
- Média da percentagem de usuários entre os sistemas FENCE (de Janeiro a Junho de 2019, em detalhes nas figuras 5.10 e 5.11): 34,13%

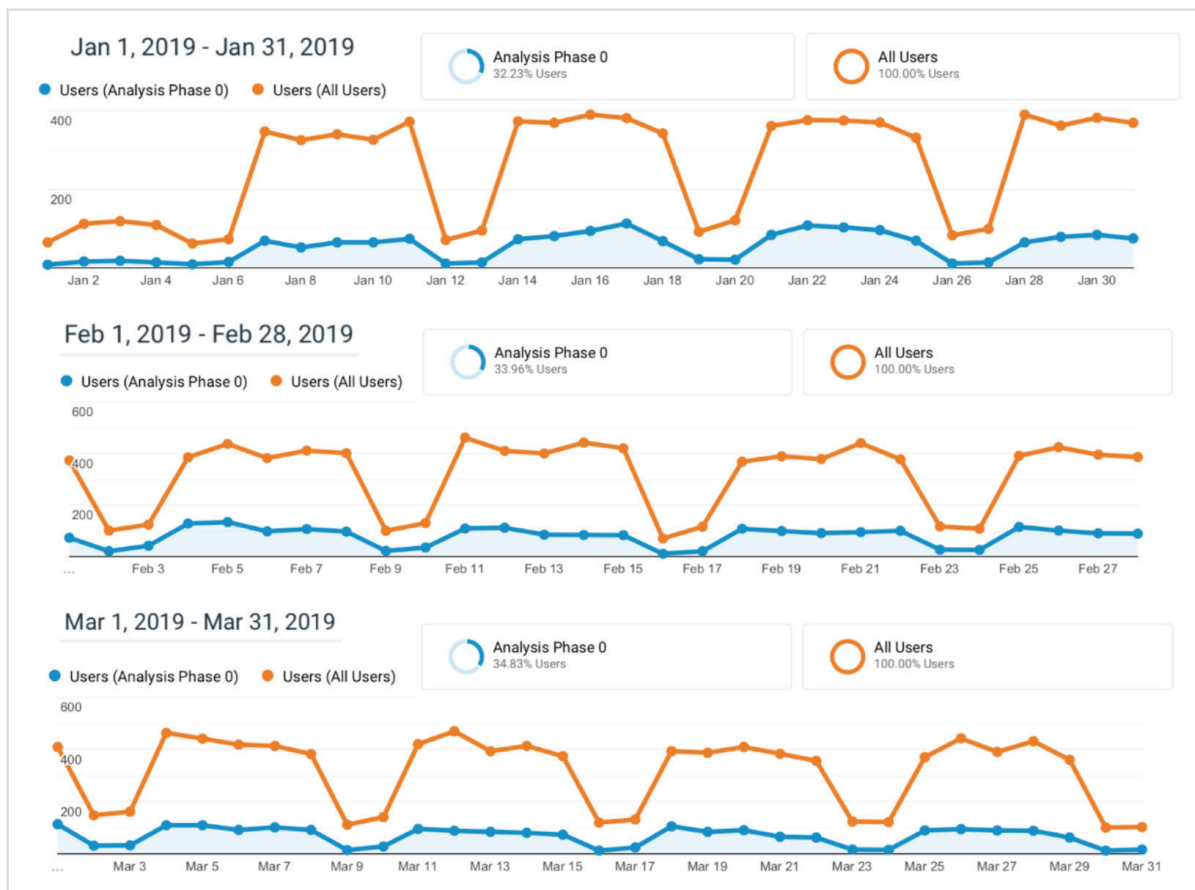


Figura 5.10: Gráficos de usuários por dia do sistema Analysis/Phase 0 entre Janeiro e Março de 2019.

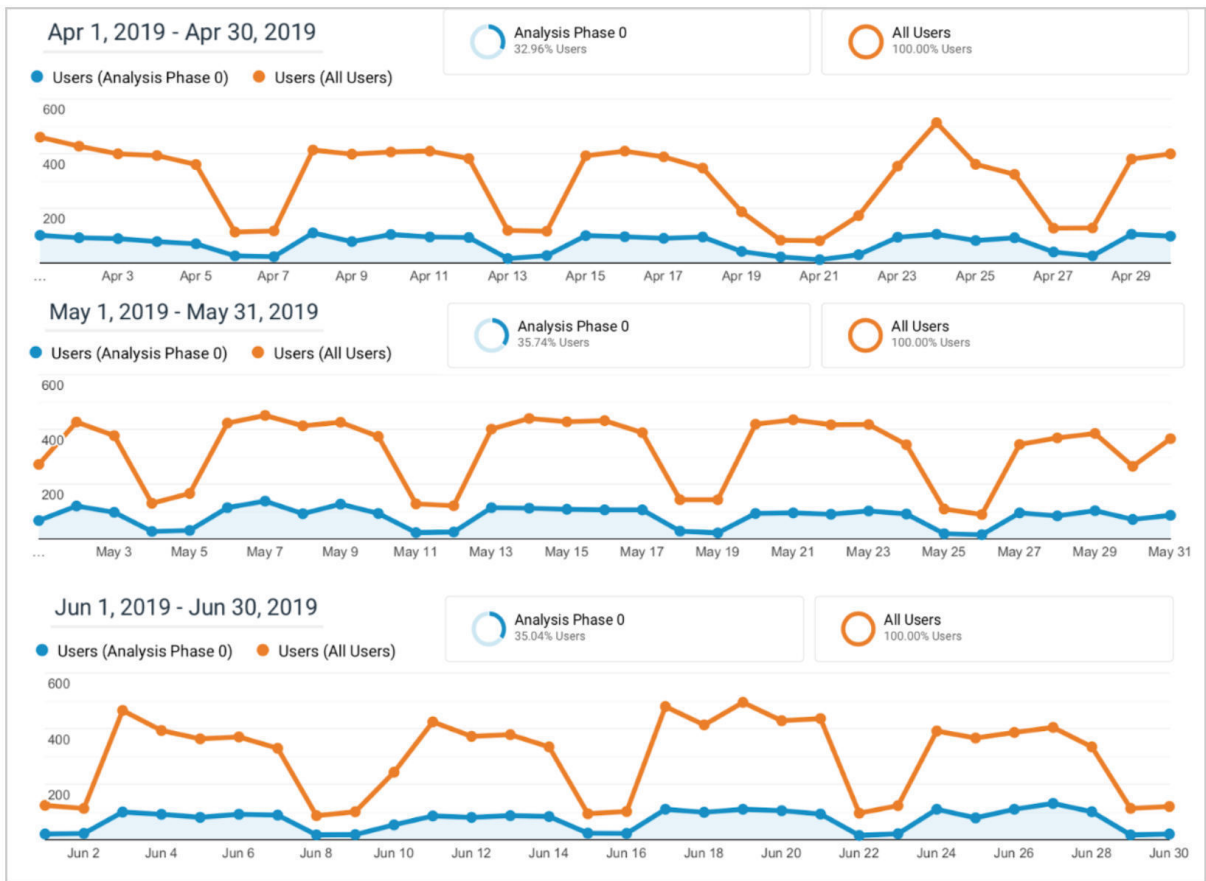


Figura 5.11: Gráficos de usuários por dia do sistema Analysis/Phase 0 entre Março e Junho de 2019.

Capítulo 6

Conclusões

Atualmente os membros do experimento ATLAS podem utilizar os sistemas Analysis, que apoiam o processo de produção de publicações desde sua concepção até estarem prontas para serem apresentadas à comunidade científica. A qualquer horário e a partir de qualquer lugar do mundo, pesquisadores podem ter acesso ao sistema e dar prosseguimento às etapas do processo de publicação.

Os *egroups* [8] criados automaticamente pelos sistemas permitem a melhor comunicação entre os envolvidos com uma publicação uma vez que *e-mails* podem ser direcionados apenas ao grupo de destinatários interessados, evitando ruídos. Além disso, os *e-mails* automáticos encaminhados pelos sistemas garantem a continuidade do fluxo de atividades do processo de publicação que antes poderia ser interrompido por falha de comunicação entre os autores e revisores.

A criação automática de *egroups* e repositórios diminui o volume de trabalho dos autores e revisores que antes a realizavam manualmente. Além disso, a padronização do processo e transparência de suas regras faz com que os envolvidos tenham maior facilidade em realizar suas etapas.

O uso da API dos *egroups* e do GitLab [3] permitiu a integração entre tais ferramentas e os sistemas Analysis, possibilitando a troca de informação entre eles de forma automática.

Todos os metadados importantes a uma publicação são agora centralizados num único banco de dados e podem ser visualizadas pelos usuários através das funcionalidades dos sistemas Analysis. Isso facilita o acesso à informação e evita o risco de erros e retrabalho pela repetição de dados em diferentes ferramentas.

Desta forma, pode-se concluir que os cinco sistemas Analysis/Phase 0, Papers, CONF notes, PUB notes e PLOTs formalizaram e automatizaram as atividades de conceber, evoluir, revisar, e aprovar uma publicação científica, atingindo o objetivo geral deste trabalho.

6.1 Trabalhos futuros

Devido à concentração de informações importantes ao experimento ATLAS contidas no banco de dados dos sistemas Analysis, é crescente a demanda de acesso a tais informações por softwares externos ou até mesmo usuários interessados em fazer relatórios. Desta forma, surge a necessidade da criação de uma interface entre os sistemas Analysis e outros usuários ATLAS, uma API. O grupo de desenvolvedores FENCE deu início a esta implementação, seguindo o projeto de uma API em REST (*Representational State Transfer*) que disponibiliza dados através de requisições HTTP (*Hypertext Transfer Protocol*), protocolo de comunicação utilizado para sistemas de informação de hipermídia.

Outro trabalho a ser feito é relacionado ao banco de dados dos sistemas Analysis. Este começou a ser estruturado e modelado antes mesmo da existência do *framework* FENCE. Dado o tempo de existência do sistema e o amadurecimento dos requisitos de uso, é normal que as relações entre entidades e atributos mudem e, desta forma, faz-se necessária a revisão da modelagem do banco de dados, garantindo as regras de normalização.

6.2 Contribuições à aluna

Esta sessão não é usual de um trabalho de conclusão de curso, mas não poderiam deixar de serem mencionadas as inúmeras contribuições que o projeto agregou à aluna. Devida a colaboração entre a UFRJ e o experimento ATLAS no projeto, implementação e manutenção dos sistemas FENCE, a aluna teve a oportunidade de compor o time de desenvolvedores nas instalações do CERN de Agosto de 2015 até Março de 2017, sendo convidada a retornar durante suas férias de verão em 2018 e 2019. Durante este tempo no exterior, grande parte do trabalho apresentado aqui foi desenvolvido e, paralelamente, enriquecedoras experiências profissionais foram adquiridas e serão apresentadas a seguir.

Dos conhecimentos aprendidos, é válido exaltar a grande experiência com linguagens de programação Web como PHP e Javascript, o mecanismo CSS e a linguagem de marcação HTML. O uso do *framework* FENCE pediu o exercício contínuo da programação orientada a objetos. O contato constante com um banco de dados Oracle relacional proporcionou o domínio da linguagem SQL e das regras de normalização de banco de dados. O aprendizado de outras ferramentas essenciais para o desenvolvimento de software foi exigido, como o Jira [14], software para acompanhamento de tarefas e projetos, o Confluence [15], software integrado ao Jira para documentação colaborativa, o sistema de controle de versões distribuído Git [4], o gerenciador de repositórios Git, GitLab [3], o editor de texto Sublime Text [16] e o

SQL developer [17], ferramenta para facilitar o desenvolvimento e gerenciamento de bancos de dados Oracle.

No que consta o processo de desenvolvimento de software, a aluna participou de constantes rotinas de levantamento de requisitos por meio de reuniões com usuários membros do experimento ATLAS. Devido às diversas demandas, o exercício de priorização e planejamento de atividades foi constante. Ao fim de cada implementação, foram seguidas rotinas de documentação, planejamento e execução de testes assim como a validação de requisitos perante o usuário.

Das habilidades adquiridas, a rotina de desenvolvimento colaborativo aguçou as habilidades de trabalho em equipe da aluna. As diversas reuniões e apresentações em inglês tornou-a proficiente na língua e um tanto mais confiante. O treinamento de novos alunos participantes do projeto aprimorou sua didática e técnicas de liderança. O convívio com pessoas excepcionais, sempre dispostas a ensinar, estimulou sua vontade de aprender e por isso é imensamente grata.

Referências Bibliográficas

- [1] “ATLAS Detector & Technology”. <https://atlas.cern/discover/detector>, . Accessed: 2019-08-15.
- [2] “GitLab API”. <https://docs.gitlab.com/ee/api/>, . Accessed: 2019-08-15.
- [3] “GitLab”. <https://about.gitlab.com/>, . Accessed: 2019-08-15.
- [4] “Git”. <https://git-scm.com/>, . Accessed: 2019-08-15.
- [5] “CERN”. <http://home.cern/about>. Accessed: 2019-08-15.
- [6] “Dark Matter”. <https://home.cern/science/physics/dark-matter>. Accessed: 2019-08-15.
- [7] “Workflow Management Coalition”. <http://www.wfmc.org/>. Accessed: 2019-08-15.
- [8] “CERN Egroups”. <https://cern.service-now.com/service-portal/function.do?name=egroups>. Accessed: 2019-08-15.
- [9] “Shibboleth”. <https://www.shibboleth.net/>. Accessed: 2019-08-15.
- [10] “PHP SplObjectStorage”. <http://php.net/manual/en/class.splobjectstorage.php>, . Accessed: 2019-08-15.
- [11] “SVN”. <https://subversion.apache.org/>. Accessed: 2019-08-15.
- [12] “PHP”. <https://www.php.net/>, . Accessed: 2019-08-15.
- [13] “JSON”. <https://www.json.org/>. Accessed: 2019-08-15.
- [14] “Jira”. <https://www.atlassian.com/br/software/jira>. Accessed: 2019-08-15.
- [15] “Confluence”. <https://www.atlassian.com/br/software/confluence>. Accessed: 2019-08-15.
- [16] “Sublime Text”. <https://www.sublimetext.com/>. Accessed: 2019-08-15.

- [17] “SQL Developer”. <https://www.oracle.com/database/technologies/appdev/sql-developer.html>. Accessed: 2019-08-15.
- [18] “ATLAS”. <https://atlas.cern/discover/about>, . Accessed: 2019-08-15.
- [19] “The ATLAS Collaboration”. <https://atlas.cern/discover/collaboration>, . Accessed: 2019-08-15.
- [20] “The Standard Model”. <https://home.cern/science/physics/standard-model>. Accessed: 2019-08-15.
- [21] LANGE RAMOS, B. ; MAIDANTCHIK, C. “FENCE: Uma Abordagem Orientada a objetos na concepção de sistemas Web altamente configuráveis para os experimentos do CERN”. <http://monografias.poli.ufrj.br/monografias/monopoli10014830.pdf>, 2015.
- [22] HENRIQUE MARTINS SILVA, H. ; MAIDANTCHIK, C. “Central de notificações para a tecnologia FENCE desenvolvida para os experimentos do CERN”. <http://monografias.poli.ufrj.br/monografias/monopoli10021571.pdf>, 2017.
- [23] HOLLINGSWORTH, D. “Workflow Management Coalition: The Workflow Reference Model”. <http://www.wfmc.org/docs/tc003v11.pdf>. Accessed: 2019-08-15.
- [24] PEREIRA, L. A. M. ; CASANOVA, M. A. “Sistemas de Gerência de Workflows: Características, Distribuição e Exceções”, 2003.
- [25] TICK, J. “Workflow Model Representation Concepts”, *7th International Symposium of Hungarian Researchers on Computational Intelligence*, 2006.
- [26] TRIPATHI, D. ; BANERJEE, S. . S. A. “Formalization of business workflow with typed attributed graph”, *International Journal of Web Information Systems*, 2018.
- [27] BONDY, J. A. ; MURTY, U. S. R. *Graph Theory with Applications*. 5 ed. New York, Elsevier, 1976.

Apêndice A

Arquivo de configuração de uma etapa do *workflow* da fase 0

O pedaço de código a seguir mostra uma parte do arquivo de configuração da classe `Workflow` da fase 0, correspondente à etapa denominada “Analysis definition after EOI meeting”. O comportamento desta etapa se dá pelo registro de dois campos no banco de dados quando o evento de acionamento do botão “Save” é disparado e desde que a condição dos campos serem válidos seja satisfeita. Quando o botão “Proceed” é acionado, um comportamento similar é observado, porém acontece a transição para uma próxima etapa e uma outra ação, a de enviar um *e-mail* aos responsáveis pelo preenchimento dos campos da próxima etapa, é disparada.

No arquivo de configuração, primeiramente, o nome e identificador da etapa são definidos como “Analysis definition after EOI meeting” e `analysis_definition` respectivamente. Depois, os campos “Main physics aim” e “Dataset Used” são definidos pelos identificadores `main_physics_aim` e `dataset_used`, seus nomes (“Main physics aim” e “Dataset Used”), tipos (caixa de texto) e regras (máximo de 500 caracteres). Logo após, são listados todos os eventos possíveis na etapa (`proceed` e `save`). Cada um deles define a próxima etapa do fluxo caso haja transição (`analysis_coordinators_selection` e `analysis_definition`, sendo a segunda um *loop*), seguida das condições em forma de métodos (os métodos são definidos por sua classe, nome e possíveis atributos) e também das ações disparadas, também na forma de métodos.

```
1 {
2     "label" : "Analysis definition after EOI meeting",
3     "id" : "analysis_definition",
4     "inputs" : [
5         {
6             "label": "Main physics aim",
```

```

7         "id": "main_physics_aim",
8         "type": "textarea",
9         "rules": {
10             "maxlength": 500
11         }
12     },
13     {
14         "label": "Dataset Used",
15         "id": "dataset_used",
16         "type": "textarea",
17         "rules": {
18             "maxlength": 500
19         }
20     }
21 ],
22 "events" : {
23     "proceed" : {
24         "next_state" : "analysis_coordinators_selection",
25         "conditions": [
26             {
27                 "class": "Atlas\\Analysis\\WorkflowConditions",
28                 "method": "validateInputs"
29             }
30         ],
31         "actions": [
32             {
33                 "class": "Atlas\\Analysis\\WorkflowActions",
34                 "method": "saveInDB",
35                 "inputs": [
36                     "main_physics_aim",
37                     "dataset_used"
38                 ]
39             },
40             {
41                 "class": "Atlas\\Analysis\\WorkflowActions",
42                 "method": "sendEmail",
43                 "template": "ANALYSIS_COORDINATOR_REQUEST"
44             }
45         ],

```

```

46     "usergroups": [
47         "GROUP_CONVENER",
48         "SUBGROUP_CONVENER",
49         "PROJECT_LEADER"
50     ]
51 },
52 "save" : {
53     "next_state" : "analysis_definition",
54     "conditions": [
55         {
56             "class": "Atlas\\Analysis\\WorkflowConditions",
57             "method": "validateInputs"
58         }
59     ],
60     "actions": [
61         {
62             "class": "Atlas\\Analysis\\WorkflowActions",
63             "method": "saveInDB",
64             "inputs": [
65                 "main_physics_aim",
66                 "dataset_used"
67             ]
68         }
69     ],
70     "usergroups": [
71         "GROUP_CONVENER",
72         "SUBGROUP_CONVENER",
73         "PROJECT_LEADER"
74     ]
75 }
76 }
77 }

```