



Universidade Federal  
do Rio de Janeiro  

---

Escola Politécnica

# **CENTRAL DE NOTIFICAÇÕES PARA A TECNOLOGIA FENCE DESENVOLVIDA PARA OS EXPERIMENTOS DO CERN**

Héron Henrique Martins Silva

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

Orientador: Carmen Lúcia Lodi Maidantchik

Rio de Janeiro

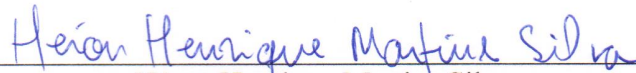
Agosto de 2017

# CENTRAL DE NOTIFICAÇÕES PARA A TECNOLOGIA FENCE DESENVOLVIDA PARA OS EXPERIMENTOS DO CERN

Héron Henrique Martins Silva

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

Autor:



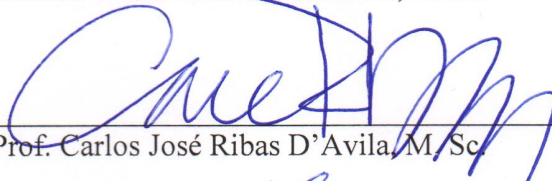
Héron Henrique Martins Silva

Orientador:



Carmen Lúcia Lodi Maidantchik, D. Sc.

Examinador:



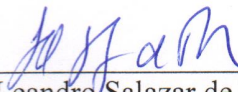
Prof. Carlos José Ribas D'Avila, M. Sc.

Examinador:



Prof. Flávio Luis de Mello, D. Sc.

Examinador:



Prof. Leandro Salazar de Paula, D. Sc.

Rio de Janeiro – RJ, Brasil

Agosto de 2017

## Declaração de Autoria e de Direitos

Eu, Héron Henrique Martins Silva CPF 130.475.347-64, autor da monografia *Central de Notificações para a tecnologia FENCE desenvolvida para os experimentos do CERN*, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na disciplina de Projeto de Graduação da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.
2. Excetuam-se do item 1. eventuais transcrições de texto, figuras, tabelas, conceitos e idéias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.
3. O autor permite que a UFRJ, por um prazo indeterminado, efetue em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.
4. O autor pode, excepcionalmente, encaminhar à Comissão de Projeto de Graduação, a não divulgação do material, por um prazo máximo de 01 (um) ano, improrrogável, a contar da data de defesa, desde que o pedido seja justificado, e solicitado antecipadamente, por escrito, à Congregação da Escola Politécnica.
5. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange a cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 e art.299, bem como na Lei 9.610.
6. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.
7. Por ser verdade, firmo a presente declaração.



---

Héron Henrique Martins Silva

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica – Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

## **AGRADECIMENTO**

Agradeço primeiramente à minha família por todo apoio, principalmente nos momentos de dificuldade, bem como por todo o esforço e confiança em mim depositados, desde o primário ao ensino superior.

Aos amigos do CERN e do LPS, Ian Miranda, Gabriela Lúcidí, Igor Abreu e Lucas Vieira, por toda a tranquilidade passada nos momentos difíceis. Em especial ao amigo Bruno Lange, pelas discussões construtivas na elaboração deste trabalho.

À minha orientadora e amiga Carmen Maidantchik por toda orientação.

À todos os professores da Universidade Federal do Rio de Janeiro que foram de grande importância na minha formação.

## RESUMO

O presente trabalho descreve o projeto e o desenvolvimento de uma Central de Notificações, para a criação e manutenção, de forma configurável pelos próprios usuários, de toda a comunicação enviada pelos sistemas *Web* que utilizam o *framework* FENCE, desenvolvido pela UFRJ para o experimento ATLAS do CERN. Estes sistemas envolvem desde a gerência de contratos e equipamentos, à gerência de plantões de controle dos detectores. A motivação deste trabalho veio da necessidade de comunicar o estado do processo de trabalho e informar as ações realizadas pelos usuários dos sistemas, dispersos geograficamente ao redor do mundo. Em particular, este projeto foi desenvolvido em colaboração com o ALICE, um experimento científico que conta com 1861 pesquisadores, em sua maioria físicos e engenheiros, de 189 institutos em 45 países diferentes. As experiências envolvem o desenvolvimento, manutenção e a coleta de resultados de colisões de partículas dentro de um grande detector. Portanto é de se esperar que sejam necessários diversos sistemas de suportes para todas as atividades da colaboração. A Central de Notificações visa a expandir o FENCE, reunindo em uma única funcionalidade todas as ações necessárias para apoiar a comunicação entre os membros colaboradores dos experimentos. A arquitetura proposta utiliza o padrão de desenvolvimento de observadores de modelos de dados, bem como conceitos difundidos em desenvolvimento de *software* orientado a objetos como herança, polimorfismo e composição, permitindo que as atividades realizadas nos sistemas, bem como o estado dos dados, sejam notificadas aos pesquisadores dos experimentos a partir de configurações descritas em arquivos JSON. O projeto encontra-se em produção e disponível para ser utilizado pelos sistemas *Web*.

Palavras-Chave: herança, polimorfismo, composição, notificação, web, configuração, CERN, ALICE.

## **ABSTRACT**

The present work describes the design and development of a Notification Center, for the creation and maintenance, configurable by the users themselves, of all the communication sent by the Web systems that use the FENCE framework, developed by UFRJ for the ATLAS experiment at CERN. These systems involve activities such as the management of contracts, equipments and the management shifts to control e retrieve data from the detectors. The project motivation came from the need to communicate the state of the work process and inform the actions carried out by the users of the systems, dispersed geographically around the world. In particular, this project was developed in collaboration with ALICE, a scientific experiment with 1861 researchers, mostly physicists and engineers, from 189 institutes in 45 different countries. Experiments involve the development, maintenance, and collection of particle collision results within a large detector. Therefore, it is expected that several support systems will be required for all experiment's activities. The Notification Center aims to expand FENCE, bringing together in a single functionality all the necessary actions to support the communication between the collaborating members of the experiments. The proposed architecture uses the development pattern of data model observers as well as concepts disseminated in object-oriented software development such as inheritance, polymorphism and composition, allowing the activities performed in the systems as well as the state of the data to be notified to the users from the settings described in JSON files. The project is in production and available to be used by all the Web systems.

Key-words: inheritance, polymorphism, composition, notification, web, configuration, CERN, ALICE.

## SIGLAS

ACDB – ALICE Collaboration Database  
ALICE – A Large Ion Collider Experiment  
ATLAS – A Toroidal LHC ApparatuS  
CERN – Centre Européenne pour la Recherche Nucléaire  
CMS – Compact Muon Solenoid  
CSV – Comma Separated Values  
DNS – Domain Name Servers  
FENCE – Front-End Engine for Glance  
HTML – Hypertext Markup Language  
HTTP – Hypertext Transfer Protocol  
IMAP – Internet Message Access Protocol  
ITS – Inner Tracking System  
JSON – JavaScript Object Notation  
LHC – Large Hadron Collider  
LHCb – Large Hadron Collider beauty  
LPS – Laboratório de Processamento de Sinais  
MDA – Mail Delivery Agent  
MSA – Mail Submission Agent  
MTA – Mail Transfer Agent  
MUA – Mail User Agent  
MVC – Model View Control  
MX – Mail Exchanger  
M&O – Maintenance and Operation  
OOP – Object-Oriented Programming  
PDF – Portable Document Format  
PDU – Protocol Data Unit  
PHP – Hypertext Preprocessor  
POP – Post Office Protocol  
PS – Proton Synchrotron  
PSB – Proton Synchrotron Booster  
SAMS – Shift Accounting Management System  
SMC – Short Message Center



SMS – Short Message Service

SMTP – Simple Mail Transfer Protocol

SPS – Super Proton Synchrotron

TCP/IP – Transmission Control Protocol and Internet Protocol

UFRJ – Universidade Federal do Rio de Janeiro

URL – Uniform Resource Locator

WWW – World Wide Web

XML – Extensible Markup Language

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
	1.1 - Objetivo .....	2
	1.2 - Organização do Documento .....	2
<b>2</b>	<b>O CERN e os Experimentos de Física de Altas Energias</b>	<b>4</b>
	2.1 - O Grande Colisor de Hádrons .....	5
	2.2 - Os Detectores de Partículas .....	6
	2.3 - Os Experimentos de Física de Altas Energias .....	7
	2.4 - Inovações e Descobertas .....	9
<b>3</b>	<b>As Tecnologias Glance e FENCE</b>	<b>11</b>
	3.1 - Glance .....	11
	3.2 - FENCE .....	14
<b>4</b>	<b>Os Sistemas de Apoio às Atividades do ALICE</b>	<b>18</b>
	4.1 - ALICE Membership .....	18
	4.2 - ALICE SAMS .....	22
	4.3 - A Comunicação entre os Sistemas e seus Usuários .....	25
<b>5</b>	<b>Protocolos de Comunicação</b>	<b>27</b>
	5.1 - SMTP .....	28
	5.2 - SMS .....	30

5.3 - Notificações <i>Web Push</i> .....	30
<b>6 Especificação da Central de Notificações</b>	<b>33</b>
6.1 - Os Motivos do Envio de Notificações .....	33
6.2 - Os Tipos de Mensagens .....	35
6.3 - Os Níveis dos Comunicados .....	36
6.4 - Os Métodos de Envio .....	36
6.5 - Os Parâmetros Configuráveis das Notificações .....	37
6.6 - As Configurações .....	40
6.7 - Validações da Comunicação .....	41
6.8 - Histórico de Configurações e das Mensagens Enviadas .....	42
<b>7 A Central de Notificações para o FENCE</b>	<b>43</b>
7.1 - Herança, Polimorfismo e Composição .....	43
7.2 - O Padrão Observador .....	44
7.3 - Implementação .....	45
<b>8 Conclusões</b>	<b>55</b>
<b>Bibliografia</b>	<b>57</b>

# Lista de Figuras

2.1 – O Complexo de aceleradores do CERN .....	5
2.2 – Reconstrução do resultado das colisões realizadas pelo experimento ALICE .....	6
3.1 – Busca de contratos de membros do sistema ALICE Membership .....	12
3.2 – Arquivo XML de configuração das interfaces de recuperação .....	13
3.3 – Tratamento de exceções no FENCE .....	16
3.4 – Sistemas desenvolvidos pelo grupo de computação do LPS para os experimentos do CERN .....	17
4.1 – Perfil de um membro colaborador do experimento ALICE .....	20
4.2 – Interface de geração da lista de autores .....	21
4.3 – Interface de configuração de períodos de plantões .....	23
4.4 – Estatísticas dos institutos do experimento ALICE no período de aquisição de dados de 2016 .....	25
5.1 – Camadas da arquitetura TCP/IP .....	28
5.2 – Processo de transferência de mensagens eletrônicas .....	29
5.3 – O serviço <i>Web Push</i> .....	32
6.1 – Funções dos membros da colaboração e suas responsabilidades .....	39
7.1 – Estrutura de desenvolvimento do padrão observador .....	45
7.2 – A classe abstrata <i>Observer</i> , referenciada pelos observadores .....	47
7.3 – Configurações JSON para notificações enviadas quando usuários fazem reservas de classes no sistema SAMS .....	49
7.4 – Interface de modificação das configurações de notificações .....	52
7.5 – Modelagem de colunas e tabelas para as notificações enviadas e para as configurações realizadas pelos membros da colaboração .....	54

7.6 – Funcionamento da Central de Notificações para o FENCE .....	54
---	----

# Capítulo 1

## Introdução

O CERN (Centro Europeu para Pesquisas Nucleares), situado na fronteira Franco-Suíça, é um centro de pesquisa internacional de grande porte que conta com esforços de muitos pesquisadores de universidades e institutos de pesquisa, dispersos geograficamente ao redor do mundo.

As pesquisas nessa instituição, em sua maioria, envolvem experiências através da colisão de partículas no centro, e também fora, de grandes detectores, que foram projetados, desenvolvidos e são mantidos pelos pesquisadores dos experimentos do CERN. Estas máquinas contêm centenas de milhares de componentes, e localizam-se sobre o grande colisor de partículas, o LHC (Large Hadron Collider), um anel de 27 quilômetros de comprimento que acelera partículas a velocidades próximas à da luz.

Sistemas *Web* de gerenciamento e suporte das atividades são utilizados pela organização, permitindo a organização da colaboração de qualquer lugar do mundo, através da Internet. Essas aplicações cobrem desde o gerenciamento de contratos dos experimentos, à localização de cabos e controle de radiação dos equipamentos. Informações encontram-se presentes em bancos de dados, que precisam ser atualizados pelos membros da colaboração.

A comunicação das ações realizadas através das aplicações pelos seus usuários é de importante para melhorar a eficiência do gerenciamento dos dados armazenados, bem como para garantir sua confiabilidade.

## 1.1 – Objetivo

O objetivo deste trabalho é projetar, e desenvolver, uma central de notificações para os sistemas de controle das atividades do CERN, que seja capaz de reunir em uma única funcionalidade todas as ações necessárias para apoiar a comunicação entre os membros colaboradores dos experimentos, de forma configurável pelos próprios usuários, apresentando regularmente à colaboração dos experimentos os históricos de modificações, funções e atividades dos membros.

Portanto, serão estudadas as formas de comunicação entre sistemas *Web* e seus usuários, bem como as tecnologias utilizadas no desenvolvimento, e as necessidades, dos sistemas *Web* do experimento ALICE. O sistema Membership, que apoia a gerência de contratos, funções e atividades dos membros da colaboração, organiza as responsabilidades das instituições de pesquisa associadas ao experimento, e gera automaticamente a lista de autores das publicações científicas. O sistema SAMS (*Shift Accounting Management System*), que gerencia os períodos de plantões realizados nas salas de controle do detector utilizado pelas pesquisas, e também organiza as reservas dos postos de plantões que devem ser preenchidos 24 horas por dias, 7 dias por semana, bem como a contagem dos créditos obtidos pelos plantonistas.

## 1.2 – Organização do Documento

O capítulo 2 apresenta o contexto da colaboração internacional no qual o projeto foi pensado e desenvolvido, dando destaque a complexidade e dispersão geográfica dos pesquisadores do experimento.

No capítulo 3 serão apresentadas as tecnologias de apoio ao desenvolvimento de sistemas *Web* desenvolvidas através da colaboração entre a UFRJ e o CERN.

No capítulo 4 serão apresentados os sistemas desenvolvidos para o experimento ALICE, que serviram de base para o levantamento de requisitos.

O capítulo 5 conta com o estudo de protocolos de comunicação, que serão utilizados para a transmissão de notificações aos usuários.

A especificação do projeto é apresentada no capítulo 6.

A implementação do trabalho é apresentada no capítulo 7.

Por fim, o capítulo 8 apresenta as conclusões do trabalho e outros desenvolvimentos futuros.



## Capítulo 2

# O CERN e os Experimentos de Física de Altas Energias

O CERN (*Centre Européenne pour la Recherche Nucléaire*) é um centro de pesquisa internacional, no qual, físicos e engenheiros, trabalham conjuntamente em experimentos de física de altas energias. Esta grande organização foi fundada em 1954, a partir de esforços conjuntos de doze países europeus, e suas instalações localizam-se na fronteira Franco-Suíça, próxima a cidade de Genebra [1]. Atualmente, a colaboração conta com aproximadamente 12 mil cientistas, de mais de 600 universidades e institutos de pesquisa, de 70 países dos diferentes continentes.

Segundos após o *Big Bang* o universo era composto por todos os tipos de partículas, movendo-se a velocidades próximas à da luz. Com o intuito de recriar tais condições, pacotes de prótons são injetados em aceleradores de partículas, onde ganham velocidade sendo em seguida expostos a colisão. Este choque gera diferentes tipos de partículas presentes nos momentos iniciais do universo, estas são estudadas pelos diferentes experimentos do centro de pesquisa.

O complexo de aceleradores do CERN é constituído por uma sucessão de máquinas em cadeia [2], ilustrado na Figura 2.1, que aumentam gradativamente a energia das partículas. O primeiro elemento da cadeia, conhecido como Linac 2, acelera os pacotes a 50 MeV e injeta-os no PSB (*Proton Synchrotron Booster*) onde são impulsionados a 1.4 GeV, em seguida, passam pelo PS (*Proton Synchrotron*) e SPS (*Super Proton Synchrotron*), atingindo ao fim 450 GeV (99,9998% da velocidade da luz). Após este estágio final, os pacotes estão preparados para serem inseridos no último, e mais potente elemento da cadeia.

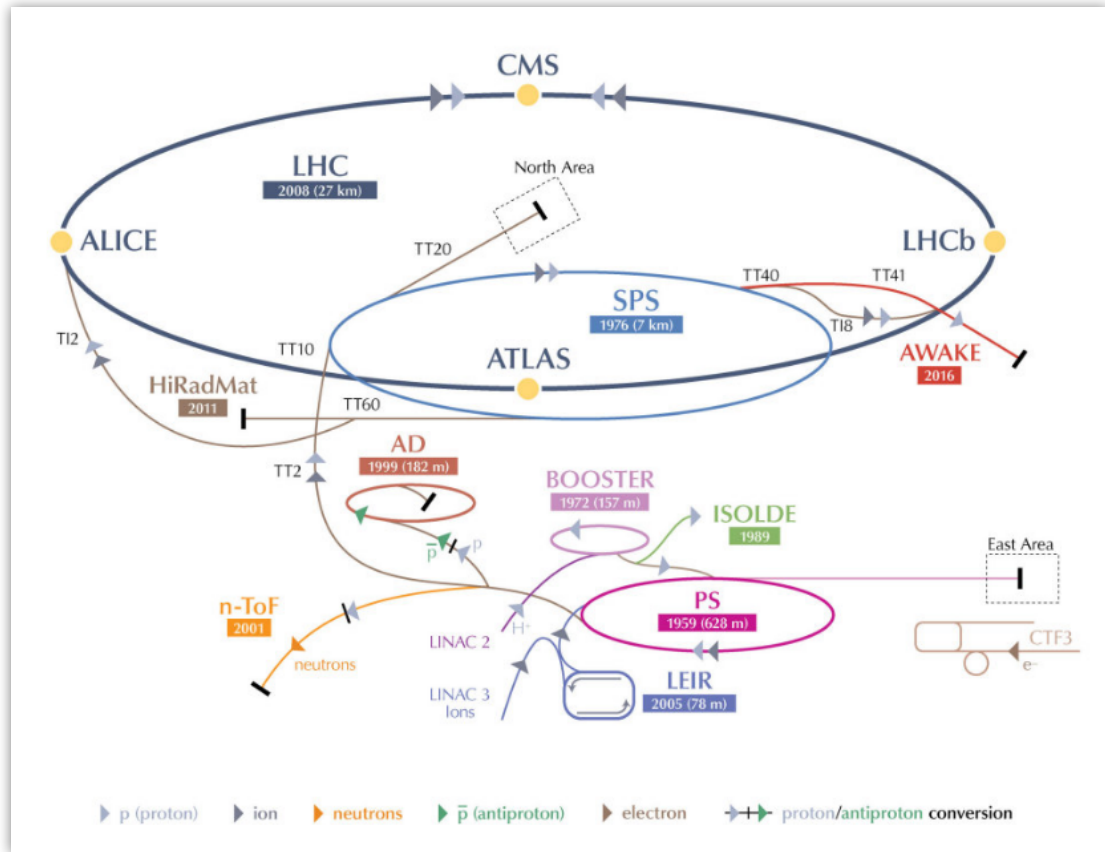


Figura 2.1. O Complexo de aceleradores do CERN. Extraído de [3].

## 2.1 – O Grande Colisor de Hádrons

O Grande Colisor de Hádrons, mais conhecido como LHC (*Large Hadron Collider*), é o maior e mais poderoso acelerador de partículas do mundo, sendo a última adição ao complexo de aceleradores do CERN. Este equipamento consiste de um anel de 27 quilômetros de comprimento localizado a 100 metros de profundidade abaixo das instalações na superfície. Sua construção foi aprovada em 1995, custando aproximadamente 7,5 bilhões de euros, e atualmente é considerado o instrumento de pesquisa mais caro já construído.

Dentro do LHC, os feixes atingem energias equivalentes a 6,5 TeV, 99.9999991% da velocidade da luz. Dois tubos separados mantêm as partículas confinadas trafegando em sentidos opostos. Para evitar choques com moléculas de gás, o interior dos canos é mantido à ultra vácuo, tão vazio quanto o espaço. As partículas são então guiadas em

volta do anel por um forte campo magnético, mantido por eletroímãs supercondutores ligados a um sistema de distribuição de Hélio líquido, refrigerados a  $-271,3$  graus Celsius, temperatura menor que a do espaço [4].

## 2.2 – Os Detectores de Partículas

As colisões dos feixes de partículas ocorrem em quatro pontos sobre o LHC, onde localizam-se os detectores de partículas ATLAS, CMS, ALICE e LHCb, ilustrados na Figura 2.1 como os pontos amarelos. Estas máquinas foram projetadas pelos engenheiros e físicos dos experimentos do CERN para amplificar significativamente a probabilidade de choques entre os pacotes de partículas no centro dos grandes detectores. Estas máquinas são equipadas com diversos sistemas eletrônicos, que possibilitam a reconstrução precisa dos eventos das colisões para armazenamento, como ilustrado na Figura 2.2. A seguir serão apresentados com mais detalhes alguns dos sistemas eletrônicos presentes no interior dos detectores.

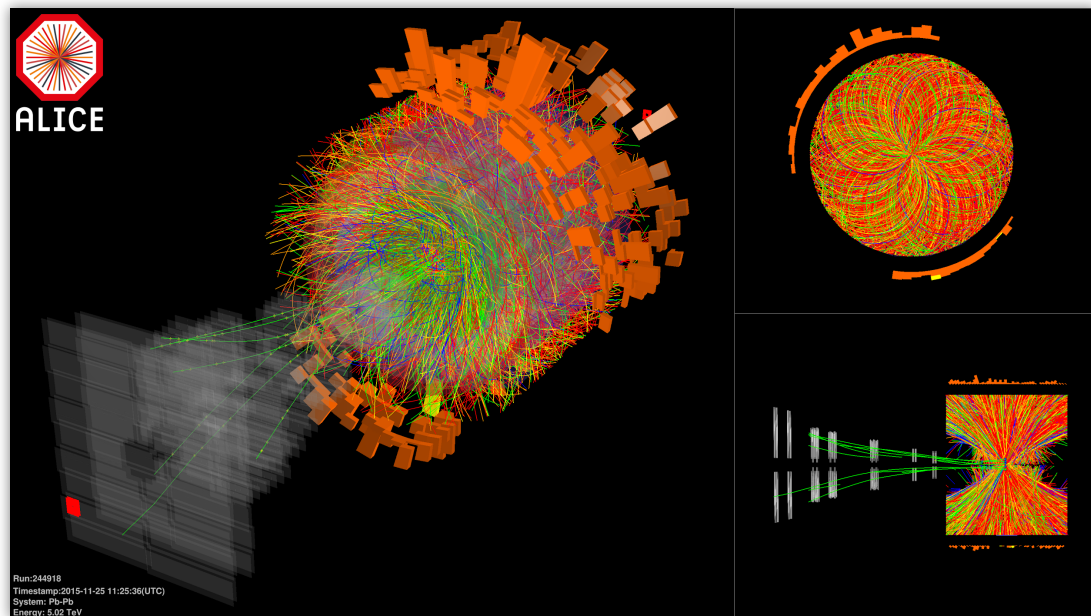


Figura 2.2. Reconstrução do resultado das colisões realizadas pelo experimento ALICE.

Extraído de [5].

Os detectores de pixel são os dispositivos mais próximos dos pontos de colisão, sendo atravessado por 10 milhões de partículas por centímetro quadrado por segundo.

São compostos por 65 milhões de píxels de silício, capazes de rastrear com extrema precisão a trajetória de cada unidade dentro do detector, determinando quais píxels foram tocados. Este dispositivo é capaz de criar uma foto tridimensional do evento ocorrido logo após a colisão, o que é de vital importância na detecção de partículas de vida curta [6].

Os calorímetros são projetados para reter, completa ou parcialmente, a energia da maioria das partículas resultantes das colisões. Consistem tipicamente de camadas de material de alta densidade passivo ou absorvente, e são usados para medir a quantidade de energia que uma partícula perde ao passar pelo detector. Os calorímetros são capazes de detectar todas as partículas conhecidas, com exceção de múons e neutrinos [7].

Os espectrômetros de múons são compostos por mil a quatro mil câmaras, na parte mais externa dos detectores, onde espera-se que somente os múons sejam capazes de registrar algum sinal. Duzentas vezes mais pesados que elétrons e pósitrons, são capazes de penetrar vários metros de metal sem qualquer interação, ou seja, atravessam os diversos dispositivos sem serem detectados. Estas câmaras são responsáveis por saber quando, e como os múons foram gerados, indicando a presença de uma partícula que não foi detectada [8, 9].

A cada segundo, aproximadamente 1 bilhão de eventos acontecem no centro dos detectores, gerando uma enorme quantidade de dados. A maioria não revela fenômenos físicos significativos, portanto, os detectores fazendo uso de sistemas eletrônicos responsáveis por selecionar os eventos potencialmente interessantes. Estes sistemas são projetados em duas ou mais camadas de dispositivos eletrônicos em conjunto com sistemas de *software*, reduzindo a quantidade de eventos selecionado para armazenamento a algumas centenas por segundo [10, 11]. Estes dados são armazenados e disponíveis para serem analisados pelos físicos do CERN ao redor do mundo.

## **2.3 – Os Experimentos de Física de Altas Energias**

Os experimentos do LHC são os responsáveis por guiar o desenvolvimento e a operação dos detectores, e subsequentemente a análise dos dados. Cada um deles com métodos e focos de pesquisa diferentes. Os maiores são o ATLAS, o ALICE, o CMS e o LHCb, que serão explorados com mais detalhe nesta seção.

O ATLAS (*A Toroidal LHC ApparatuS*) é o maior dos experimentos, o detector operado pesa 7 mil toneladas com 46 metros de comprimento e 25 de diâmetro. Seguem uma pesquisa de propósito geral que investiga uma gama de fenômenos físicos, desde teorias do Modelo Padrão à partículas que podem compor matéria escura, e também outras dimensões. Esta colaboração conta com mais de 5 mil cientistas de 38 países diferentes que trabalham no desenvolvimento, controle a análise dos dados gerados pelo detector [12, 13].

O CMS (*Compact Muon Solenoid*), assim como o ATLAS, é um experimento de propósito geral, isto é, não favorece a detecção de um tipo específico de partícula. Mesmo contando com menores medidas que o ATLAS, com 21 metros de comprimento e 15 de diâmetro, o detector pesa em torno de 14 mil toneladas, extremamente compacto [14]. Embora ambos os experimentos tenham objetivos científicos similares, eles fazem uso de soluções técnicas e de design diferentes, sendo do ponto de vista do método científico, importante para a confirmação cruzada das descobertas.

O ALICE (*A Large Ion Collider Experiment*) é um detector de íons pesados, que busca explicações para a forte interação entre matéria. Para certas partes do ano, o LHC realiza colisões entre íons pesados, como os de chumbo, chegando a temperaturas cem mil vezes mais quentes que o núcleo do sol, e liberando as ligações entre quarks e glúons, gerando um estado de matéria conhecido como plasma quark-glúon. Para o experimento, a existência de tal estado é a chave para entender o confinamento da matéria, exercido pela força forte [15].

O LHCb (*Large Hadron Collider beauty experiment*) é uma pesquisa de propósito específico. Segundo os físicos deste experimento, O *Big Bang* deu origem a quantidades iguais de matéria e antimatéria, e, portanto, esta colaboração busca explicar o que permitiu que a matéria forme tudo o que conhecemos hoje, não se aniquilando com antimatéria. Os cientistas do LHCb acreditam que a chave se encontra em um tipo de partícula conhecida como “*beauty quark*” ou “*b quark*” [16].

## 2.4 – Inovações e Descobertas

A operação de experimentos de tais portes traz inúmeros desafios, justificando os esforços colaborativos de mais de 12 mil cientistas de diferentes nacionalidades. Desde 1954 o CERN tem se dedicado na busca por mais conhecimento para a humanidade. Esta missão implica em, não somente descobertas físicas, mas também diversas tecnologias de apoio para o controle e gerência da colaboração.

Em 1976, o pesquisador italiano Carlo Rubia propôs adaptações no SPS para ser possível a colisão entre prótons e antiprótons no mesmo anel, e juntamente com a tecnologia de resfriamento estocástico [44] do pesquisador holandês Simon van der Meer, foi possível construir o acumulador de antiprótons. Estas novidades tornaram possíveis as detecções dos bósons W e Z, e em 1984 [43], ambos os pesquisadores foram premiados com o Nobel de Física por suas contribuições fundamentais que levaram à descoberta dos bósons mediadores da força nuclear fraca.

Em 1989, o pesquisador britânico Tim Berners-Lee concebeu, projetou e implementou a *Web* [19], deparado com problemas de compartilhamento de informações entre os diversos pesquisadores dispersos geograficamente ao redor do mundo. O WWW, ou *World Wide Web*, foi projetado como um espaço de informação onde os documentos e recursos são identificados por Localizadores de Recursos Uniformes (URLs), ligados e conectados a outros recursos através de hiperlinks. Permitindo-se que estes dados pudessem ser acessados por qualquer usuário, independente da localização, conectado à Internet. A *Web* foi fundamental para atender a comunicação, e a publicação de resultados entre os pesquisadores da colaboração, impulsionando a construção do conhecimento e o desenvolvimento de novas tecnologias. Desde sua invenção, até os dias de hoje, é utilizada por diversos sistemas desenvolvidos para o CERN.

Em 1992, o físico francês Georges Charpak foi premiado com o Nobel de Física devido a por sua invenção para os detectores de partículas, conhecida como *Multiwire Proportional Chamber*. Estes dispositivos são uma grande evolução aos detectores de colisões de partículas com leitura eletrônica, pois permitem haver centenas de milhares de canais de leitura, e não um (ou poucos) como era até então, viabilizando detecções mais complexas. Hoje em dia, esta tecnologia é utilizada também na área médica para a

geração de imagens raio-x em 2D e 3D, e a exposição a radiação é reduzida em um décimo em comparação com as tecnologias convencionais.

Em julho de 2012, a partir dos grandes avanços tecnológicos trazidos pelos experimentos, ambos ATLAS e CMS, anunciaram que conseguiram observar uma nova partícula. O Bóson de Higgs [17], previsto inicialmente pelo modelo padrão [18] em 1964, confirma o mecanismo através do qual partículas elementares adquirem massa. O físico inglês Peter Higgs e o físico belga François Englert, foram premiados em 2013 com o Nobel de Física. Ambos os físicos trabalharam individualmente, atingindo resultados complementares.

## Capítulo 3

# As Tecnologias Glance e FENCE

A colaboração entre a UFRJ e o CERN foi responsável pelo desenvolvimento de 22 sistemas *Web* de apoio às atividades dos experimentos, 17 para o ATLAS (onde iniciou-se o envolvimento em 2006), 2 para o ALICE, 2 para o LHCb e 1 para o CMS. Estes sistemas apoiam diversas atividades das colaborações, como por exemplo a medição da dosagem de radioatividade dos equipamentos do detector, a gerência de contratos e funções de membros, a gerência de plantões nas salas de controle dos detectores, processo de elaboração de artigos científicos, geração da lista de autores, etc. Nesta seção serão apresentadas duas tecnologias que foram utilizadas no desenvolvimento destes sistemas.

### 3.1 – Glance

O Glance é uma ferramenta *Web*, desenvolvida através da colaboração entre a UFRJ e o ATLAS, de recuperação e operação de grandes massas de dados, que através de um mecanismo genérico, atua como uma camada intermediária, isolando o usuário das particularidades de cada banco de dados, independente da tecnologia utilizada [20].

O sistema reconhece as tabelas e colunas da modelagem dos dados e permite que seus usuários selecionem o conjunto de dados de interesse, gerando automaticamente uma interface de recuperação através da *Web*, sem que o usuário precise realizar qualquer tipo de programação. As informações pertinentes sobre a modelagem dos dados e a geração da interface de busca, são armazenadas em alto nível, em arquivos XML, permitindo que tais interfaces sejam personalizadas, e facilmente adaptadas frente a modificações ou adições de tabelas e colunas no modelo dos dados armazenados.

A Figura 3.1 ilustra a aplicação do Glance na busca de contratos de membros do sistema ALICE Membership. Cada coluna das tabelas utilizadas na geração da interface



é automaticamente transformada em parâmetros de busca, que podem ser combinados para filtrar o conteúdo dos bancos de dados de acordo com a necessidade do usuário. Estas configurações são realizadas através de arquivos XML. As buscas são feitas de forma paramétrica através do atributo, do operador escolhido e do valor inserido. Por fim, os resultados transmitidos podem ser visualizados em diversos formatos, como na própria interface *Web*, em arquivos XML, JSON ou CSV.

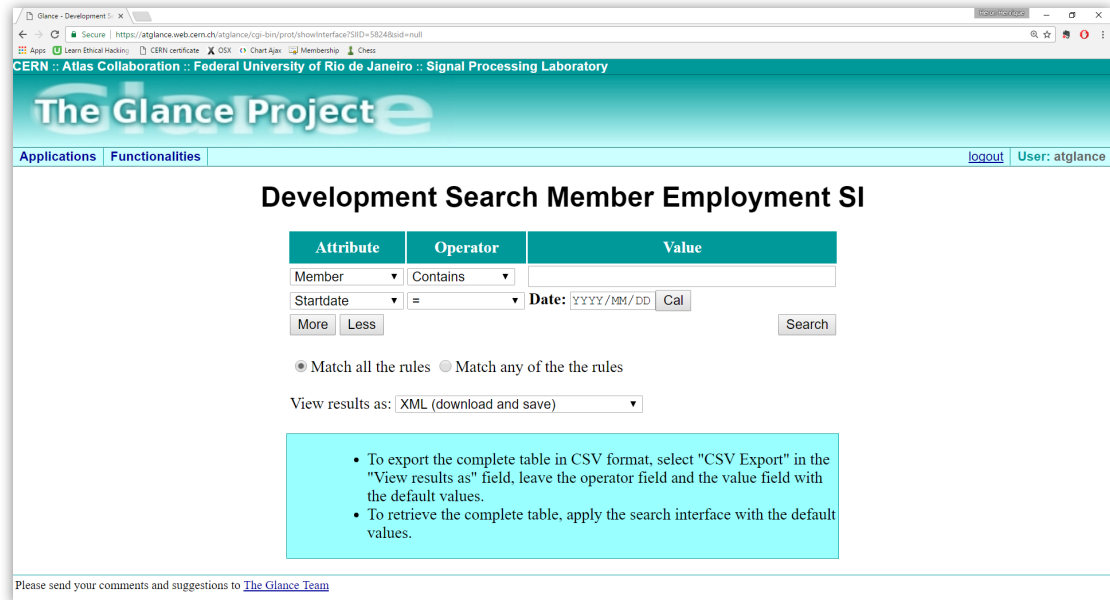


Figura 3.1. Busca de contratos de membros do sistema ALICE Membership.

A Figura 3.2 apresenta as configurações em linguagem XML das interfaces de buscas apresentadas. Estes arquivos são gerados automaticamente pelo Glance, a partir das tabelas e colunas presentes no banco de dados, que podem ser vistas nas chaves `<table></table>` e `<column></column>`, assim como as configurações sobre os diferentes operadores de buscas, gerados a partir do tipo de dado presente na coluna, como visto nas chaves `<colType></colType>` e `<colResolution></colResolution>`.

```

<table>
  <tableExpr>t_employment e</tableExpr>
  <schemaName>acdb</schemaName>
  <column>
    <colExpr>e.startdate</colExpr>
    <colId>startdate</colId>
    <colDescr>Startdate</colDescr>
    <colType>datetime</colType>
    <colResolution>date</colResolution>
  </column>
  <column>
    <colExpr>modifyenddate(e.enddate)</colExpr>
    <colId>enddate</colId>
    <colDescr>Enddate</colDescr>
    <colType>datetime</colType>
    <colResolution>date</colResolution>
  </column>
  <column>
    <colExpr>f.name</colExpr>
    <colId>category</colId>
    <colDescr>Category</colDescr>
    <colType>string</colType>
  </column>
  <column>
    <colExpr>e.mno</colExpr>
    <colId>mno-a-budget</colId>
    <colDescr>MNO-A Budget</colDescr>
    <colType>string</colType>
  </column>
</table>
<table>
  <tableExpr>t_employ_category f</tableExpr>
  <schemaName>acdb</schemaName>
</table>
<table>
  <tableExpr>t_member g</tableExpr>
  <schemaName>acdb</schemaName>
  <column>
    <colExpr>g.ltx_name</colExpr>
    <colId>ltxname</colId>
    <colDescr>LaTeX Name</colDescr>
    <colType>string</colType>
  </column>

```

Figura 3.2. Arquivo XML de configuração das interfaces de recuperação.

Esta tecnologia pode ser incorporada a outras aplicações através de chamadas HTTP às interfaces de busca geradas pelo Glance. Os filtros das colunas de interesse podem ser adicionados às URLs, bem como o tipo de resposta esperado (XML, CSV ou JSON).

Esta ferramenta impulsionou o desenvolvimento dos sistemas de controle das atividades dos experimentos, como a gerência de membros, de cabos e dos equipamentos utilizados nos detectores, tornando-se de grande importância na recuperação dos dados dos diferentes bancos de dados do CERN. Dentre estes sistemas foram percebidas diversas funções comuns, que motivaram a geração das múltiplas funcionalidades e interfaces, de forma mais eficiente e configurável.

### 3.2 – FENCE

O FENCE (*Front-End Engine for Glance*) é uma ferramenta desenvolvida em PHP (*Hypertext Preprocessor*), fruto da colaboração entre a UFRJ e o ATLAS, que permite a rápida geração e adaptação de interfaces *Web* [21], e é integrado ao Glance para recuperação de dados. Apoiado em paradigmas de orientação a objetos (*Object-Oriented Programming*, OOP) e herança, o *framework* permite que as funcionalidades sejam geradas a partir de arquivos de configuração, contendo as regras dos sistemas.

A descrição destas regras é feita através de pares nome/valor em arquivos JSON (*JavaScript Object Notation*) [22]. Trata-se de uma ferramenta de texto que é completamente independente da linguagem de programação utilizada, que permite armazenar informações de forma organizada e de fácil acesso. A classe *JReader* interpreta estes arquivos, e permite que as subclasses dos sistemas tenham acesso as diferentes chaves e valores do objeto descrito.

As classes dos sistemas estendem as classes abstratas do *framework*, que faz uso de polimorfismo por herança, permitindo que o comportamento descrito nas classes concretas, seja ditado pelas classes que referenciam. Esta dinâmica permite que a adaptação das regras dos sistemas, seja realizada através da modificação dos parâmetros definidos nos arquivos de configuração.

O FENCE encoraja o desenvolvimento MVC (*Model-View-Controller*) através de diversas classes bases que suportam os modelos de dados, as classes de visualização e os controladores das funcionalidades. A classe *DBOf* permite que os modelos de dados dos sistemas sejam construídos a partir das configurações definidas em seus arquivos JSON. Validando automaticamente todas as regras definidas antes de qualquer comunicação concreta com os bancos de dados, garantindo a confiabilidade dos valores armazenados. A classe *View* lida com atributos e métodos compartilhados por todas as interfaces de visualização dos dados. Por fim, os controladores referenciam a classe *Content*, que contém uma série de mecanismos comuns entre os controles das funcionalidades dos sistemas.

O Twig [23], uma linguagem *open-source* de *templates* HTML para PHP, é atualmente utilizado para a estruturação do conteúdo a ser renderizado. A classe base de visualização do FENCE contém os métodos necessários para integração dos dados

manipulados pelos controladores aos *templates*. Separando desta forma as responsabilidades das classes de visualização e de controle dos sistemas.

Dentre as últimas adições ao *framework*, encontram-se os construtores de modelos de dados, implementado a partir de teorias conhecidas na área de desenvolvimento de software como *Builder Design Pattern* [24]. Este padrão visa separar a construção de um objeto complexo, de sua representação (ou modelo), para que o mesmo procedimento possa ser utilizado para construir quaisquer representações.

As fábricas de modelos dos sistemas, classes filhas de *Factory*, recebem os pedidos dos dados a serem construídos, e conhecem as classes construtoras que devem ser acessadas. Estes pedidos são então passados aos devidos construtores, que contém o mapeamento de quais tabelas, e colunas do banco de dados devem ser buscadas, compondo os valores das propriedades das representações.

O FENCE também conta com uma classe que gerencia o envio de emails. A classe *Mailer* permite que um objeto que a instancie, injete a partir de seus métodos o remetente, o(s) destinatário(s), o assunto e o corpo da mensagem de email a ser enviada. Este último podendo ser composto por texto simples ou em formato HTML, permitindo a decoração das mensagens. O disparo das mensagens é realizado pela chamada ao método *fire*, que lida com o envio concreto da mensagem.

Todas as ações dos usuários através das interfaces dos sistemas são registradas automaticamente através da classe *Logger*. O tratamento de exceções é feito automaticamente por blocos *try/catch*, integrados a classe *Mailer*, informando os usuários frente a qualquer *bug* que o suporte técnico foi acionado, e disparando imediatamente um email aos responsáveis com o código do erro presente no arquivo de *log* do usuário. Permitindo a rápida identificação e reação por partes do grupo de desenvolvedores.

A Figura 3.3 apresenta um exemplo desta dinâmica, a esquerda o email recebido pelos desenvolvedores informando qual usuário, a URL acessada, a mensagem de erro, a data e hora do incidente. Ao centro o arquivo de *log* das ações do usuário com o código do erro acontecido. À direita a funcionalidade de personificação de usuários, que permite que os desenvolvedores se passem por outros usuários para testar os sistemas.

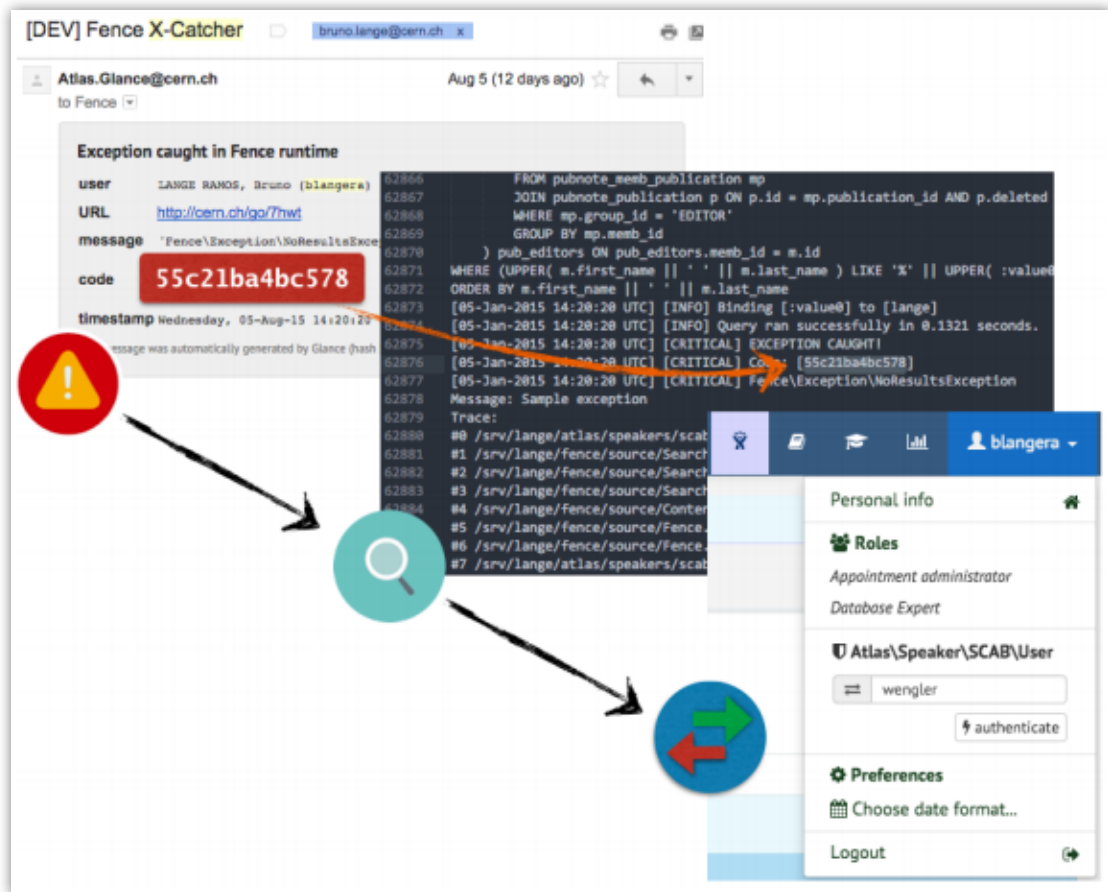




Figura 3.3. Tratamento de exceções no FENCE. Extraído de [21].

A Figura 3.4 ilustra os sistemas *Web* desenvolvidos para o controle das atividades dos experimentos do CERN. Todos os sistemas fazem uso da tecnologia Glance, e sete encontram-se em produção utilizando o FENCE, enquanto que outros 4 estão em fase de migração com conclusões previstas para este ano. Dentre os sistemas apresentados, o sistema ALICE Membership e ALICE SAMS, serviram de base para a proposta e implementação deste projeto. O Membership encontra-se em produção em sua segunda versão utilizando ambas as tecnologias apresentadas nesta seção. O SAMS, já em produção, está em fase de migração de Glance para FENCE com conclusão prevista para o fim deste ano. A expectativa é de que todos os sistemas sejam migrados em curto e médio prazos.

# ATLAS TCn Applications




**The Glance Project**




[more info](#)

**ATLAS Cables**




[more info](#) ★

**Rack Wizard**




[more info](#) ★

**Analysis - PUB notes**




[more info](#) ★

**ATLAS Membership**




[more info](#)

**ATLAS Survey**




[more info](#)

**ATLAS SCAB**




[more info](#) ★

**ATLAS Thesis**




[more info](#)

**ATLAS Appointment**




[more info](#) ★

**Glance TDAQ**




[more info](#)

**ATLAS TDAQ Speakers**




[more info](#)

**Speakers Committee**




[more info](#)

**DSS alarms viewer**




[more info](#)

**Apex TDAQ**




[more info](#)

**ATLAS Upgrade Speakers**




[more info](#)

**Collaboration Board Meetings**




[more info](#) ★

**ATLAS Traceability**




[more info](#)

**Muon Spare**




[more info](#)

**Analysis - Papers**




[more info](#) ★

**MTF Database**




[more info](#)

**Muon Equipment**



[more info](#)


**Analysis - CONF notes**



[more info](#) ★


**Related Projects**

**LHCb Traceability**




[more info](#)

**LHCb Membership**




[more info](#)

**Alice Membership**



[more info](#) ★


**Alice SAMS**



[more info](#)


**FENCE Applications Under Development**

**ATLAS Membership**




[more info](#) ★

**LHCb Cables**




[more info](#) ★

**ALICE SAMS**



[more info](#) ★

**CMS Rack Wizard**



[more info](#) ★

[Contact Us](#)

Figura 3.4. Sistemas desenvolvidos pelo grupo de computação do LPS para os experimentos do CERN. Extraído de [25].

# Capítulo 4

## Os Sistemas de Apoio às Atividades do ALICE

Este projeto foi desenvolvido em colaboração com o experimento ALICE, que conta com mais de 1861 membros, de 189 institutos e 45 países diferentes. Dois sistemas foram desenvolvidos para apoiar as atividades da colaboração, como o gerenciamento de contratos, institutos, plantões de controle do detector e a geração da lista de autores das publicações. Estes sistemas utilizam a plataforma *Web*, para que possam ser acessados por todos os membros a partir de qualquer navegador.

### 4.1 – ALICE Membership

Desde o início do experimento em 2008 e dos anos subsequentes de aquisição de dados, as informações sobre contratos, institutos, funções e atividades dos membros da colaboração eram mantidas manualmente em um banco de dados, conhecido como ACDB (*ALICE Collaboration Database*). De forma a melhorar a eficiência e economizar custo e tempo com esses tipos de atividades, deu-se início em 2013 o desenvolvimento do sistema ALICE Membership [26].

A colaboração depende de várias agências de fundos que devem ser reconhecidas tanto pela agência como pelo experimento a partir de um documento que deve ser assinado por ambos. Portanto o sistema permite que estes documentos sejam registrados no sistema assim como as agências às quais estão ligados. Estas disponibilizam os fundos para as pesquisas e operação do detector.

Os membros são registrados na colaboração com datas de início e fim, ligadas a sua profissão e seu instituto de pesquisa. Estas informações podem mudar com o decorrer

tempo, como por exemplo um estudante de PhD que termina suas pesquisas e passa a ser associado como físico ou engenheiro sênior no experimento, com novas datas de início e conclusão deste período, ou mesmo sem uma conclusão prevista. Estes atributos podem ser inseridos e modificados através das interfaces do sistema.

Cada colaborador tem uma função específica, como chefes de grupo, líderes de projeto ou coordenação de recursos por exemplo, e participam de diferentes atividades nos detectores, como a aquisição ou análise de dados. Estas funções e atividades dos pesquisadores são registradas pela coordenação de recursos da colaboração, com períodos de início e fim, e assim como os contratos, a aplicação permite que estes aspectos possam ser inseridos ou adaptados através de suas funcionalidades.

Segundo a constituição do experimento, os estudantes de PhD associados devem cumprir seis meses de atividades obrigatórias, conhecidas como *Service Tasks*, para que possam obter seus direitos como autores das publicações do ALICE. Estas atividades são definidas pelos líderes dos diferentes projetos, e podem ser registradas no sistema para que os estudantes possam fazer suas reservas. Concluída a tarefa, ambos estudante e supervisor, devem responder a formulários de pesquisas gerados pelos sistemas, para que possam ser avaliados pela coordenação, e enfim, o estudante passa a ser considerado autor das pesquisas.

As teses dos membros também são registradas no sistema, como informação para que outros usuários possam buscar mais detalhes sobre um determinado campo de pesquisa.

A Figura 4.1 exibe a interface de perfil dos membros do sistema Membership. Onde são aplicados os diferentes requisitos apresentados acima. A aba inicial, mostra um resumo dos detalhes do membro, e as outras abas estão disponíveis para os usuários responsáveis por seus gerenciamentos. A geração da imagem foi feita com o nível de super usuário do sistema, que tem o poder de ver e modificar todos os aspectos envolvendo os dados apresentados.



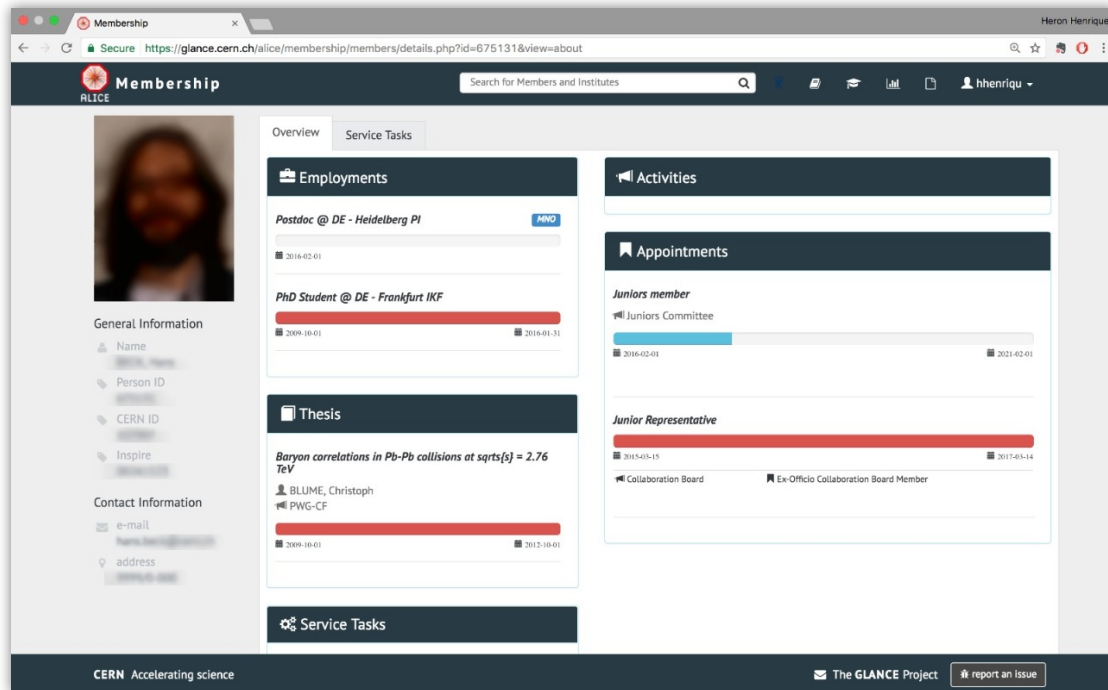


Figura 4.1. Perfil de um membro colaborador do experimento ALICE. Extraído do sistema.

As instituições de pesquisa e universidades aos quais os colaboradores são associados também são registradas no sistema. Estas são reconhecidas juntamente com os autores participantes das pesquisas, e seus líderes são responsáveis pelo gerenciamento dos contratos, funções e tarefas de seus respectivos membros.

As publicações realizadas pelas pesquisas no ALICE, contam com aproximadamente novecentos autores, tendo cada um colaborado em algum aspecto da pesquisa. Os autores são selecionados a partir dos membros da colaboração que respeitam regras específicas descritas na constituição do experimento, como por exemplo, todo o físico com mais de um ano ativo nas pesquisas, é considerado autor, ou todo estudante de PhD com mais de 6 meses de atividades. Atualmente, estas regras de autoria são aplicadas sobre todos os contratos dos membros na base de dados, automaticamente, e esta lista pode ser gerada a partir do sistema para qualquer data escolhida, e em diferentes formatos de visualização (PDF, XML, CSV, entre outros).

A Figura 4.2 ilustra a interface de geração da lista de autores. Onde ao top são exibidas as regras da constituição do experimento, e o campo de data disponível pode ser utilizado para a geração da lista para a data especificada. A geração da lista disponibiliza

diferentes formatos de saída, como XML (utilizado pelo quadro editorial), CSV, PDF ou HTML na interface, como exibido na imagem.

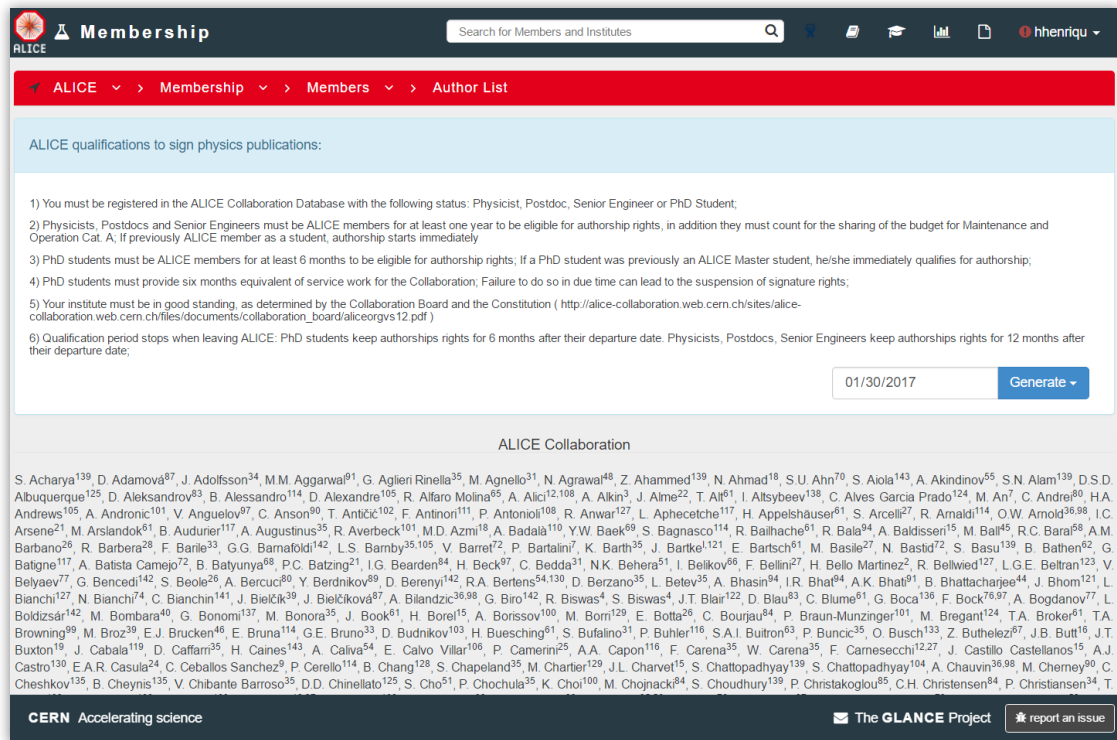


Figura 4.2. Interface de geração da lista de autores. Extraído do sistema.

A partir dos registros do sistema Membership, também são definidos os níveis de acesso dos membros a outros sistemas do CERN, como por exemplo o sistema Indico. Esta é uma aplicação do centro de pesquisa utilizada para o gerenciamento de conferências, *workshops* e reuniões.

O controle de acesso aos dados exibidos e gerenciados pelo sistema é realizado através de seus próprios registros. Por exemplo, os membros presentes no experimento podem buscar e visualizar informações de outros membros, como seus contratos, funções e atividades. Os líderes dos institutos e universidades são responsáveis pela atualização dos contratos dos pesquisadores de sua instituição, os líderes de projetos, pelo gerenciamento das atividades de pesquisa e controle da colaboração. Por fim a coordenação de recursos e a secretaria tem o poder de modificar todos os dados manipulados pela aplicação.

## 4.2 – ALICE SAMS

A operação do detector ALICE, requer a presença de um representante de cada uma de suas subpartes, 24 horas por dia, 7 dias por semana, nos momentos em que as partículas estão sendo aceleradas pelo LHC. Essas atividades são conhecidas como *shifts*, ou plantões.

A constituição do experimento define que cada instituição colaboradora tem o dever de cumprir uma determinada cota de créditos, calculada a partir da quantidade de membros ativos e participantes das pesquisas. Esta regra visa garantir a cobertura total da operação, bem como a sua boa qualidade. Estes membros são registrados como M&O (*Maintenance and Operation*), e cada instituto investe um determinado capital no experimento ALICE, para ter direito a utilizar as instalações do experimento [27].

O SAMS (*Shift Accounting Management System*), é a ferramenta através da qual permite-se que a coordenação configure os períodos de operação do detector, conhecidos como *Runs*. Estas configurações envolvem definir datas de início e fim, quais especialistas serão necessários, e em quais períodos do dia, manhã, tarde ou noite, e quais as quantidades de créditos que serão obtidas pelas cadeiras de plantões. Estas informações estão disponíveis aos membros dos institutos da colaboração, para que estes possam se programar a fazer as reservas de seus plantões. Geralmente, a operação dura desde os primeiros dias de janeiro, ao fim de dezembro, com pausas de uma a duas semanas.

A Figura 4.3 mostra a interface de configuração dos períodos de plantões. Onde devem ser definidos as datas de início e fim, durante que períodos do dia, e quais os diferentes tipos de especialistas necessários durante o respectivo período de tempo serão necessários, bem como a quantidade de pontos que serão creditados por cada plantão. Os nomes como ITS (*Inner Tracking System*) ou *Muon Arm*, são referentes a diferentes partes dos detectores, das quais necessita-se de membros especializados na operação.

Figura 4.3. Interface de configuração de períodos de plantões. Extraído do sistema.

Os períodos de plantões dos membros especialistas são creditados à sua instituição, de forma que sejam cumpridas suas cotas anuais para com a colaboração. Antes de estarem aptos a assumirem seus postos nas salas de controle do detector, os usuários devem passar por um treinamento, que consta primeiramente em assistir aulas ministradas pelos coordenadores. Estas classes são registradas para que os membros possam fazer suas reservas, todas as classes têm um limite máximo de vagas a serem assumidas. Atendendo às aulas e devidamente aprovados pela coordenação, os alunos obtêm permissão para realizar seus três primeiros plantões de treinamento, ou *Training Shifts*. Estes são plantões especiais, realizados obrigatoriamente em três dias consecutivos, onde um supervisor é responsável por auxiliar e avaliar o trabalho. Após concluir os dias de treinamento, os membros são avaliados pelos supervisores, que definem quais terão as permissões de realizar plantões reais de controle da operação.

Os plantões devem obrigatoriamente ser reservados durante seis dias consecutivos, 2 dias de manhã (de 7:00 às 15:00), 2 de tarde (de 15:00 às 23:00) e 2 de noite (de 23:00 às 7:00). Os membros da colaboração podem reservar suas cadeiras através do sistema, onde obtêm informações de quantos créditos serão obtidos para seus institutos. A coordenação técnica de controle é comunicada sobre cada reserva feita, para

que possam monitorar os dias ocupados e selecionar membros de sua escolha para os dias não reservados, mantendo a total cobertura da operação.

Os coordenadores têm o poder de manipular as reservas dos plantonistas, podendo remover ou sobrescrever membros para momentos os quais podem requerer mais perícia. Tendo um dia de seu plantão desabilitado, ou sendo removido de sua reserva, os usuários são devidamente notificados, para não comparecerem nos dias em que outros especialistas são necessários.

Todos os plantões são supervisionados por um representante com mais experiência, conhecido como *Shift Leader*, este é responsável por controlar e avaliar os outros plantonistas. Este posto especial também pode ser reservado pelo sistema, tendo este membro a responsabilidade de informar ao sistema SAMS, quais usuários realizaram bons plantões, para que estes possam definitivamente obter créditos para suas instituições.

A Figura 4.4 apresenta a tabela de estatística de créditos obtidos pelas instituições em um período de controle específico, no caso "*Data Taking 2016*". A tabela ilustra os dados referentes ao progresso de cada uma das instituições participantes, como as reservas já realizados e os pontos obtidos. Cada instituição que cumpre menos que 75% de sua cota total, recebe punições por parte do experimento ALICE, motivando seus representantes à realizarem seus plantões de controle e aquisição de dados.

O sistema entrou em produção em 2014, e desde então, diversas melhorias foram implementadas ao sistema. Por exemplo, não havendo limites para as reservas por parte de uma instituição, algumas realizavam mais plantões que suas requeridas cotas, impossibilitando outras de atingirem as suas. Tornou-se necessário a implementação da ferramenta de *overbooking*, possibilitando que plantões de institutos que já haviam atingido seus créditos necessário, pudessem ser sobrescritos por membros de institutos que não haviam atingido os seus. Sendo necessário comunicar todos e quaisquer usuários que houvesse perdido suas reservas, e informar a coordenação sobre as alterações.

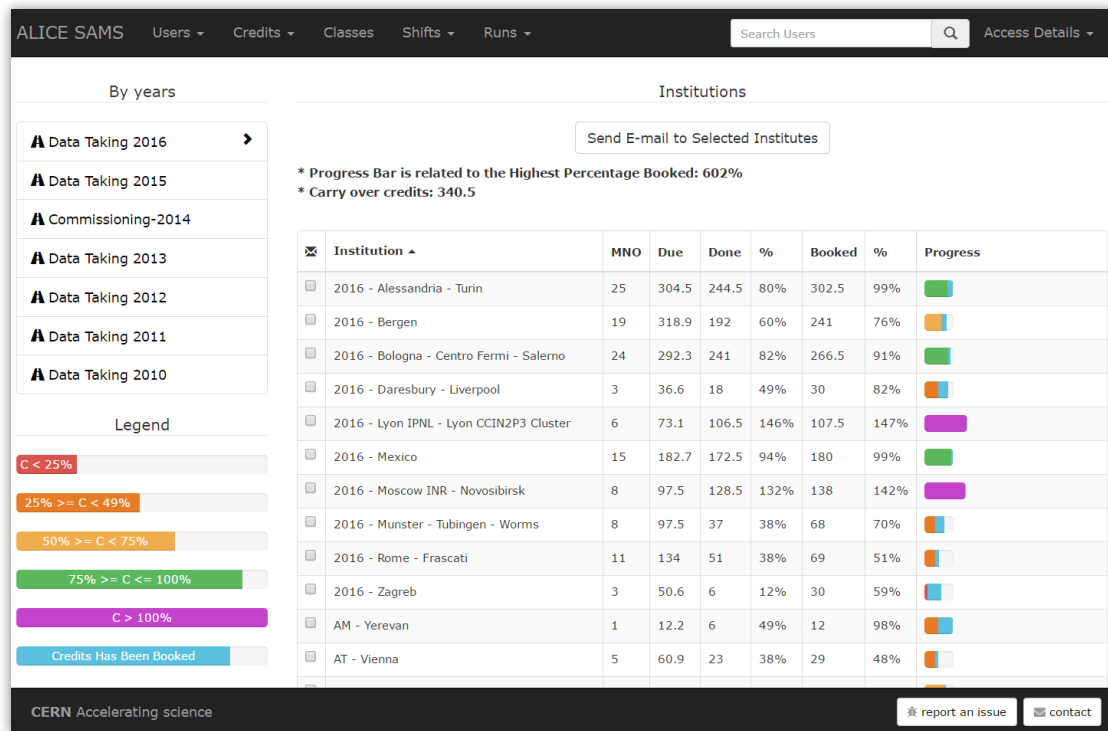


Figura 4.4. Estatísticas dos institutos do experimento ALICE no período de aquisição de dados de 2016. Extraído do sistema.

### 4.3 – A Comunicação Entre os Sistemas e seus Usuários

Ambos os sistemas comunicam seus usuários sobre diversas ações realizadas a partir de suas funcionalidades, e de seus registros nos respectivos bancos de dados.

No caso do sistema Membership, por exemplo, ao ser alterado o período de contrato, ou da função, de um membro participante da colaboração, deve-se informar a secretaria e a coordenação de recursos do experimento. Ao serem finalizadas as tarefas obrigatórias realizadas pelos estudantes de PhD, os supervisores das atividades, e o aluno, são notificados e devem preencher o formulário de pesquisa referente a atividade realizada, para que as respostas sejam avaliadas pela coordenação técnica das atividades. Listas com registros do banco de dados são enviadas periodicamente à coordenação, como quais são os membros participantes dos institutos, e das atividades dos experimentos, quais cargos estão próximos de suas conclusões, quais estudantes de PhD ainda não cumpriram suas tarefas obrigatórias, entre outros.

No caso do sistema SAMS, nem todos os plantões são reservados, deixando lacunas no controle da operação e ocasionando problemas à coordenação em relação à aquisição de dados. É necessário notificar os coordenadores dias antes dos plantões sobre cada plantão não reservados, para que estes possam tomar as devidas providências e encontrar um usuário de sua escolha para assumir a vaga. Instituições que durante o período de operação estejam muito abaixo de suas cotas gerais, também são comunicadas, para que possam de imediato regularizar suas situações com o experimento, evitando punições subsequentes.

Os usuários dos sistemas necessitam configurar as notificações transmitidas pelos sistemas, como por exemplo, quando, para quem, e quais mensagens serão enviadas. Deve-se manter registros dos comunicados enviados, e dos períodos nos quais um determinado requisito esteve ativo, de forma que possam ser rastreados pelos próprios desenvolvedores dos sistemas. Por exemplo, desde quando usuários são comunicados sobre as adaptações dos períodos de contratos ou das funções dos membros da colaboração. Portanto, serão estudados os protocolos de comunicação mais utilizados e formas que possibilitem a implementação de uma ferramenta que seja capaz de cobrir tais atividades.

# Capítulo 5

## Protocolos de Comunicação

Em telecomunicações, a troca de informações entre dois dispositivos eletrônicos ocorre através da transmissão de pacotes de dados através de uma ou mais redes. Para que essa comunicação seja possível, o mesmo conjunto de regras ou procedimentos, sobre como os dados serão enviados e recebidos, devem ser aplicados em ambos os lados. Estes padrões são conhecidos como protocolos, e podem ser implementados através de hardware, software, ou mesmo uma combinação de ambos, e são estabelecidos e controlados por organizações internacionais [28].

O modelo atual que regula a maior parte da transmissão através da Internet, quer com fio ou sem fio, é conhecido como TCP/IP (*Transmission Control Protocol and Internet Protocol*). Este modelo funciona através de quatro camadas que trocam informações entre si, responsáveis por encapsular os pacotes, endereçá-los e controlar a suas transmissões. A Figura 5.1 ilustra essa arquitetura, a infraestrutura da rede está associada às três primeiras camadas, de baixo para cima. A primeira camada ou camada de acesso à rede (*Network Access Layer*), define como os *bits* serão transmitidos fisicamente, e é representada pela conexão cabeada *Ethernet* ou *Wi-Fi*. A camada da Internet (*Internet Layer*) é regulada pelo protocolo IP, responsável por empacotar os dados com seus endereços de origem e destino, e guiá-los pelos diferentes caminhos da rede ao destino. A terceira, ou camada de transporte (*Transport Layer*), é responsável por encapsular os dados, estabelecer a comunicação entre os dispositivos, regular o fluxo e a verificação, reconhecendo os pacotes recebidos e solicitando a retransmissão dos que foram perdidos ou ilegíveis. Por fim, a quarta camada ou camada da aplicação (*Application Layer*), diz respeito aos protocolos de comunicação entre os aplicativos e a camada de transporte, regulada por uma série de procedimentos diferentes, dependendo da aplicação [29, 30].



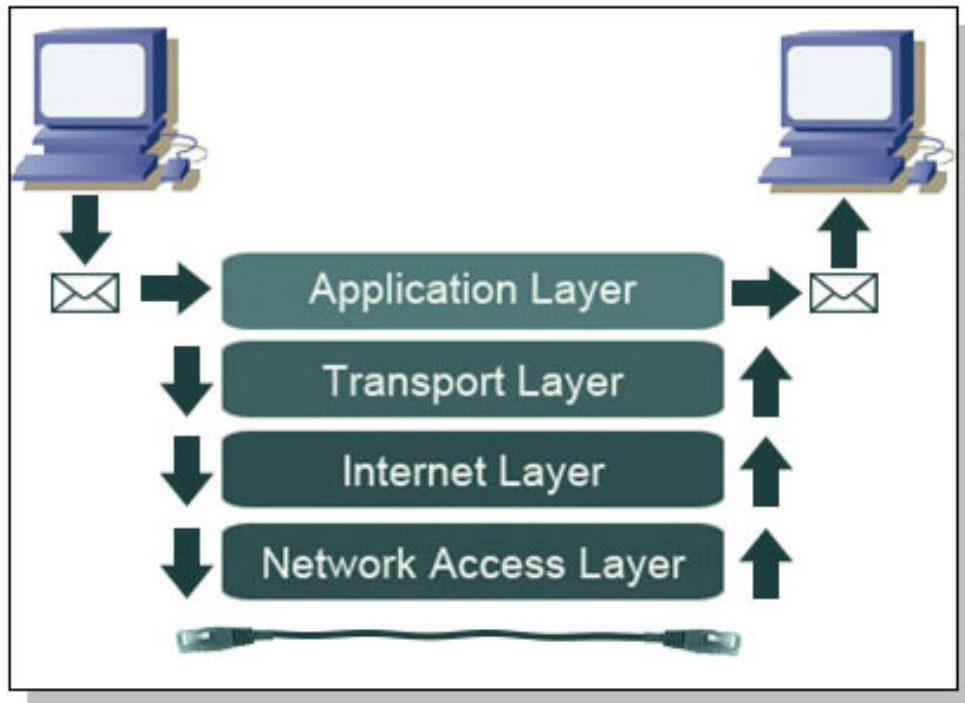


Figura 5.1. Camadas da arquitetura TCP/IP. Extraído de [31].

Nesta seção serão exploradas as formas de notificar usuários de aplicações *Web*. As formas mais comuns são por email, através do protocolo SMTP (*Simple Mail Transfer Protocol*), por SMS (*Simple Message Service*) ou por *Web Push Notifications*.

## 5.1 – SMTP

Email, ou *Electronic Mail*, é uma forma de troca de mensagens digitais realizada através de aplicações em dispositivos eletrônicos. Este sistema baseia-se em um modelo de armazenamento e encaminhamento (conhecido como *store-and-forward*). Neste modelo, os servidores recebem as mensagens enviadas e as guardam, até que sejam requisitadas pelos destinatários, não exigindo que ambos estejam *online* simultaneamente. É suficiente que somente uma das partes conecte-se brevemente ao servidor de email para que suas mensagens sejam enviadas e recebidas.

Estas mensagens são compostas minimamente por duas seções principais, o cabeçalho e o corpo. O cabeçalho contém as informações do remetente, destinatário(s), assunto e data. Estes atributos são utilizados pelos protocolos de email no processo de transporte das mensagens entre máquinas. O corpo contém a mensagem em si, como texto

não estruturado, às vezes contendo um bloco de assinatura no final, e é separado do cabeçalho por uma linha em branco [32].

A Figura 5.2 ilustra a transmissão dos emails, que é realizada a partir de um cliente (*Mail User Agent*, MUA) para um servidor (*Mail Submission Agent*, MSA) através do protocolo SMTP (*Simple Mail Transfer Protocol*), localizado na quarta camada da arquitetura TCP/IP. As mensagens são então entregues deste servidor para o seu agente de transferência (*Mail Transfer Agent*, MTA), geralmente ambos são instâncias do mesmo *software* utilizados a partir de diferentes configurações na mesma máquina. O agente de transferência usa o sistema de domínios DNS, para localizar o registro do trocador de mensagens (*Mail Exchanger*, MX), é a parte do email localizada depois do “@”. Sendo reconhecida pelo servidor, a mensagem é enviada ao agente de entregas (*Mail Delivery Agent*, MDA), sendo arquivada e pronta para entrega mediante o pedido do destinatário [33, 34]. Todas as transmissões entre cliente-servidor e agentes ocorre através do padrão SMTP.

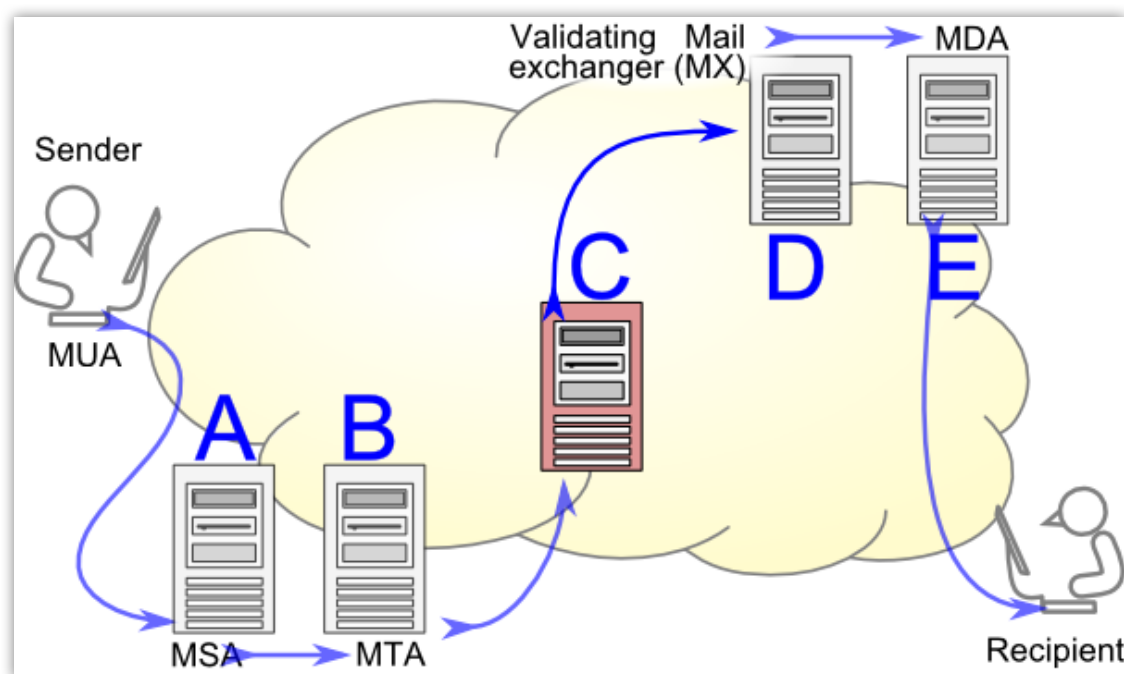


Figura 5.2. Processo de transferência de mensagens eletrônicas. Extraído de [33].

A retirada das mensagens dos servidores de correios eletrônicos é realizada por dois outros protocolos distintos. O protocolo POP (*Post Office Protocol*) suporta requisitos de carregamento dos emails do servidor, e em seguida, sua exclusão remota (*download-and-delete*). Este protocolo foi projetado para usuários que acessam suas

mensagens eletrônicas sempre do mesmo dispositivo, onde são arquivadas, não necessitando conexão intermitente à rede para ler e organizar as mensagens após o carregamento [34]. O protocolo IMAP (*Internet Message Access Protocol*) é uma outra alternativa, que cobre uma das limitações do procedimento anterior, permitindo que correios eletrônicos possam ser acessados por múltiplas aplicações, mantendo as mensagens até que um pedido explícito de exclusão seja realizado. Possibilitando que usuários acessem e manipulem suas caixas de email nos servidores, a partir de múltiplos dispositivos [35].

No FENCE a classe *Mailer* responsabiliza-se pela transmissão de mensagens, através de emails, entre as aplicações que o utilizam e seus usuários. Esta classe tem como dependências todos os parâmetros das mensagens eletrônicas, como o emissor, os receptores, o assunto e o conteúdo da mensagem, e realiza a transmissão através do protocolo SMTP.

## 5.2 – SMS

É um protocolo utilizado para enviar mensagens curtas através de redes, permitindo textos de no máximo 160 caracteres de comprimento, dentre eles letras, números e símbolos. Este procedimento define os bits de código binário que compõem cada letra e como esses dados estão organizados, para que os dispositivos de envio e recebimento possam se comunicar.

O formato dos dados transmitidos inclui o comprimento da mensagem, um carimbo de data e hora, o número de telefone de destino e o conteúdo da mensagem. Esses detalhes são construídos pela unidade de descrição do protocolo PDU (*Protocol Data Unit*), e assumem a forma de uma sequência de octetos hexadecimais. Onde os primeiros octetos contêm informações sobre para onde enviar a mensagem, o centro de mensagens curtas, ou SMC (*Short Message Center*), e o próprio número do remetente. Estas mensagens são armazenadas na central de serviços, que então encaminha a mensagem para outros centrais, até atingir o destino desejado. Se um destinatário sem fio estiver desligado, fora da faixa ou se houver alguma interrupção da rede, a mensagem armazenada será entregue quando o destinatário ficar disponível novamente [36].

A transmissão faz uso de dois canais separados, um para transferência de mensagens de controle e outro para a transmissão dos pacotes. O canal de controle é geralmente usado para rastrear a célula em que o telefone de destino está atualmente, permitindo que este altere suas células conforme se mova. Desta forma, as mensagens são transmitidas para os aparelhos corretos nos locais corretos.

Em algumas operadoras, mensagens podem ser enviadas para o telefone de um usuário através de um gateway *Email-to-SMS*. O CERN proporciona este tipo de serviço, que pode ser utilizado enviando-se a mensagem para um servidor interno, com o número do telefone do destinatário, para esta ser transmitida para o dispositivo móvel do membro da colaboração, por SMS.

### **5.3 – Notificações *Web Push***

O protocolo *Web Push* é uma forma de comunicação que utiliza a versão HTTP (*HyperText Transfer Protocol*) 2, onde o pedido de uma determinada transação é iniciado pelo servidor. Este serviço funciona através de um pedido de permissão enviado do servidor aos usuários, que podem, ou não, habilitar este canal de comunicação [37, 38].

Esta tecnologia consolida todos os eventos em tempo real, em uma única sessão, o que garante um uso mais eficiente dos recursos de rede, distribuindo esses eventos aos aplicativos à medida que eles chegam. Sempre que disponível um novo conteúdo, estas informações são empurradas para o cliente através do canal habilitado. Essas mensagens são exibidas na área de trabalho ou no dispositivo, mesmo quando o usuário não estiver utilizando a aplicação *Web*. Permitindo uma forma de comunicação mais rápida com os usuários dos sistemas.

Em contrapartida, esta tecnologia é implementada de formas diferentes em cada navegador, e por ser recente, encontra-se presente somente nas versões mais modernas dos navegadores Chrome, Safari e Mozilla Firefox. As mensagens devem ser curtas, o título pode conter de 30 a 50 caracteres, e o conteúdo pode variar entre 30 a 120 caracteres, dependendo do navegador utilizado pelo usuário [39].

A Figura 5.3 ilustra essa arquitetura, onde as aplicações devem ser registradas aos serviços *Push* providenciados pelos diferentes navegadores, os quais emitem e guardam

as informações dos usuários cadastrados, e encaminha todas as mensagens transmitidas pelos sistemas *Web* aos clientes.

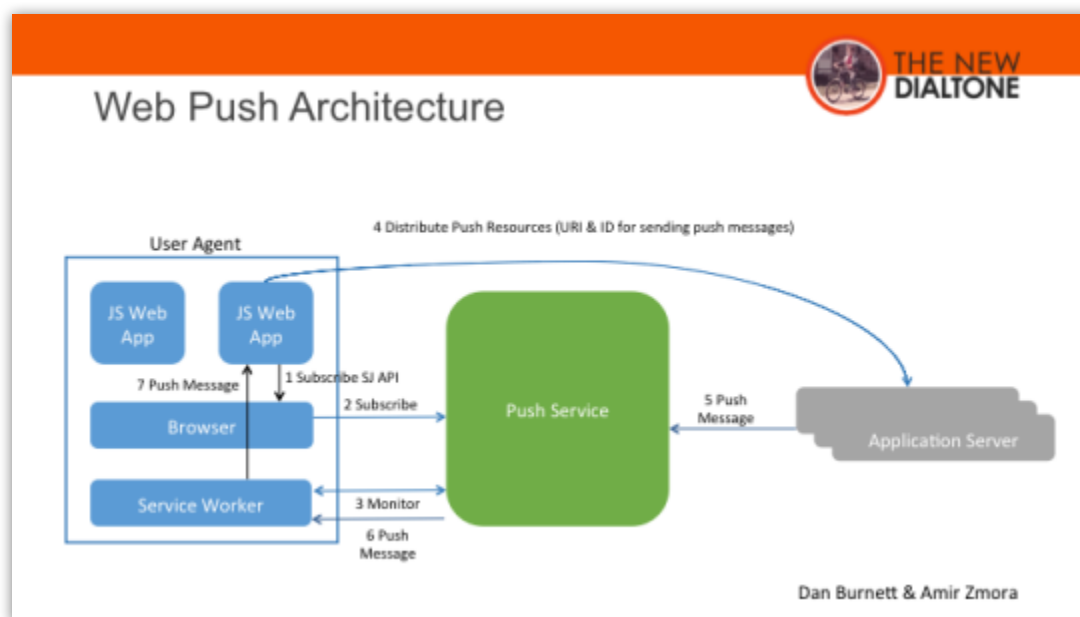


Figura 5.3. O serviço *Web Push*. Extraída de [39].

Ambas as tecnologias *Web Push* e SMS são importantes para atingir os usuários de aplicações de forma mais rápida. Porém, devido as limitações na quantidade de conteúdo que pode ser transmitida, o email é a plataforma de comunicação mais consistente, permitindo a transmissão de uma maior quantidade de informação e a customização das mensagens, a múltiplos destinatários na mesma conversa, e garantia do armazenamento das mensagens recebidas pelos usuários, a menos que removidas pelo próprio.

No contexto deste projeto, estas tecnologias foram estudadas visando-se entender as formas como a comunicação entre sistemas *Web* e seus usuários pode ser realizada, de forma a serem integradas ao trabalho. A seguir serão especificados os requisitos da Central de Notificações, a partir dos estudos dos sistemas e das tecnologias apresentadas.

# Capítulo 6

## Especificação da Central de Notificações

Neste capítulo, apresenta-se a especificação da Central de Notificações. O projeto envolve a extensão do *framework* FENCE, através da incorporação de novas classes que visam possibilitar que os próprios usuários dos sistemas possam configurar as comunicações encaminhadas aos integrantes da colaboração. Buscou-se analisar os diferentes requisitos das comunicações transmitidas pelos sistemas existentes, bem como identificar outras necessidades ainda não existentes, que serão apresentadas a seguir.

### 6.1 – Os Motivos do Envio de Notificações

Os dados manipulados pelos sistemas estão passíveis a alterações por usuários autorizados, cujos direitos de acesso e escrita devem respeitar as especificações da aplicação. No contexto da colaboração, determinados grupos de usuários devem ser prontamente informados mediante a modificação dos registros. Sendo assim, a primeira causa de notificações corresponde aos eventos relacionados aos valores dos atributos dos dados. Os dados estão sujeitos a três tipos distintos de eventos, que envolvem a comunicação do software com os respectivos bancos de dados: alterações, inserções e exclusões.

Como exemplos, no sistema ALICE Membership, extensões ou inserções de contratos de membros da colaboração devem ser avisadas ao respectivo representante do grupo. Alterações das funções e atividades dos usuários devem comunicadas à coordenação de recursos. A conclusão das tarefas realizadas pelos doutorandos deve ser notificada à coordenação técnica, que também envolve o envio de formulários de pesquisas ao responsável pela atividade e ao seu supervisor, e assim que respondidos, os representantes e o porta-voz da colaboração são comunicados, para que possam analisar o desempenho em relação à tarefa realizada.

No sistema SAMS, todos os membros inscritos em treinamentos de supervisão da operação do detector devem ser informados caso a localização ou data de um treinamento seja alterada. Ao concluir o treinamento, o membro colaborador passa a estar apto a realizar plantões na sala de controle e esta mudança de privilégios deve ser comunicada tanto ao próprio membro, quanto à coordenação de operação do detector. Adaptações nas reservas dos plantões, como a desativação de um horário ou alterações dos plantonistas, devem ser comunicadas imediatamente. Ao ser alterada a instituição que será creditada pelos plantões de um membro, disparam-se comunicados à instituição que perdeu seus créditos, à que ganhou e à coordenação, informando as alterações na contagem dos pontos.

A segunda causa de notificações se refere a análises temporais aplicadas sobre os dados. Comandos são previamente especificados de forma a agendar um intervalo periódico para a realização de determinadas tarefas, que buscam registros na base de dados, analisam e apresentam os resultados à coordenação da colaboração.

Por exemplo, diariamente verificam-se quais dentre todos os cargos da colaboração terminarão dentro dos próximos seis meses. Tal informação é encaminhada à coordenação de recursos do experimento para indicar novos membros, estender a data de término da nomeação ou simplesmente não realizar nenhuma ação, pois pode ser o caso de não ter nenhum substituto para o cargo. Mensalmente, listas com os contratos dos membros de cada instituição é enviada aos respectivos líderes, para serem atualizados quando necessários. Semanalmente, arquivos Excel são enviados à coordenação de recursos contendo todas as modificações realizadas nos dados. Doutorandos que iniciaram suas atividades na colaboração há mais de 18 meses, e ainda não tenham cumprido suas tarefas obrigatórios, são notificados, juntamente com a coordenação técnica para regularizar suas situações, pois o não cumprimento implica não serem registrados na lista de autores das publicações do experimento.

Estas análises de dados também são aplicadas sobre os períodos de controle do detector. Dois dias antes dos plantões, a supervisão da operação é alertada caso alguma cadeira não tenha sido reservada, para que possam nomear um membro de sua escolha, evitando problemas na aquisição de dados. As instituições colaboradoras são comunicadas em 30%, 50% e 75% dos períodos anuais de plantões, caso estejam abaixo de um valor estipulado de suas cotas, para que possam regularizar suas situações e evitar

punições posteriores. Devido a atrasos no registro dos contratos dos membros, dez dias antes de um plantão, é comunicado à coordenação quais dos plantonistas reservados não tenham um contrato ativo, pois a falta deste registro implica em créditos que serão obtidos pelo membro e não serão passados a nenhuma instituição.

A Central de Notificações deve ser capaz de identificar tipos de dados que envolvem datas de início e término, como por exemplo contratos ou papéis dos membros, para que possa disparar mensagens informativas quando tais períodos estiverem próximos de suas conclusões, bem como dados que tem sua ocorrência em uma data definida, como por exemplo plantões ou classes, de forma que estes também possam ser comunicados dias antes de suas ocorrências.

## **6.2 – Os Tipos de Mensagens**

As mensagens enviadas podem ser de dois tipos distintos: informativas ou alertas requisitando uma ação por parte do usuário. Em geral, mensagens informativas têm o papel de comunicar sobre alterações realizadas por terceiros, sobre estados dos registros, bem como deveres dos membros na colaboração, e não contém botões ou links de redirecionamento para as funcionalidades dos sistemas. Por exemplo, ao serem inseridos novos doutorandos, estes recebem uma mensagem contendo seus deveres de acordo com a constituição do experimento.

Diferentemente dos comunicados informativos, os alertas requerem uma ação por parte do usuário notificado, na maioria dos casos, com urgência, e são enviadas constantemente caso estas ações não sejam tomadas. Por exemplo, caso os plantões não sejam preenchidos após o primeiro comunicado enviado à coordenação, um outro comunicado é enviado no dia seguinte. Desta forma, evitando problemas no controle e na aquisição de dados do experimento.



### **6.3 – Os Níveis de Comunicados**

As notificações podem ser diferenciadas por níveis de prioridade, que podem ser definidos como: baixa, média, alta e urgente. Estes podem ser adicionados aos assuntos das mensagens transmitidas, permitindo que os receptores possam dar mais ou menos importância aos comunicados, e permitindo a organização e exibição das mensagens enviadas.

Caso um destinatário não responda as ações requisitadas por um alerta, esta pode ser reenviada com um nível de prioridade maior. Esta nova transmissão pode incluir uma mensagem diferente, com mais informações e mais urgência, e também, ser transmitida a outros usuários também envolvidos na atividade, melhorando a eficiência na tomada de ações necessárias nos sistemas. Os novos destinatários das mensagens podem então ser categorizados pelo nível de importância da notificação.

Por exemplo, os formulários de pesquisas enviados após as conclusões de atividades nos experimentos, caso não respondidos, podem ser reenviados após uma semana, um nível de prioridade acima, até que atinjam o nível de urgência. Neste momento, a própria coordenação da colaboração pode ser informada, para que possa também entrar em contato com o membro e o supervisor da atividade, requisitando o preenchimento do formulário. Instituições que estiveram abaixo de suas cotas de plantões nos primeiros 30% do período anual, serão notificadas novamente em 50% do período, através dos níveis de prioridade, os comunicados podem ter um impacto maior, permitindo que mais instituições mantenham suas situações regulares.

### **6.4 – Os Métodos de Envio**

Esta comunicação entre as aplicações e os usuários deve ser realizada de 3 formas diferentes: por email, SMS ou *Web Push*. Para a transmissão de mensagens por email, é necessário ter, no mínimo, os endereços do remetente e dos destinatários, bem como a mensagem que será transmitida. O assunto, é opcional, e permite organizar e localizar as mensagens presentes nas pastas ou caixas de entrada do usuário. Através do corpo das mensagens permite-se anexar documentos, imagens e outros tipos de arquivos. Conexões

a outros serviços *Web* também podem ser feitas através de hiperlinks, que podem ser inseridos ao texto da mensagem.

Opcionalmente, outras pessoas também podem ser inseridas em cópia das mensagens. Copiar alguém em um e-mail é uma ótima maneira de mantê-los informados do que está acontecendo, sem realmente estar envolvido. Esta cópia também pode ser feita de forma privada, usuários podem ser adicionados aos destinatários, sem que os outros saibam que ele faz parte das pessoas que receberam a mensagem.

Os comunicados também podem ser realizados através de SMS, *Short Message Service*. No CERN este serviço pode ser utilizado enviando-se uma mensagem de e-mail breve, com no máximo 160 caracteres, à um domínio do centro de pesquisa que retransmite esta mensagem ao número de celular do membro colaborador.

Assim como mensagens curtas por SMS, comunicados também podem ser enviados via *Web Push*, uma tecnologia recente que permite que o servidor envie ao cliente mensagens que serão exibidas em sua área de trabalho, através do navegador no qual o usuário utilizou e permitiu a comunicação da aplicação. Estas mensagens podem ser clicadas e redirecionam os usuários às páginas das aplicações das quais foram disparadas.

## **6.5 – Os Parâmetros Configuráveis das Notificações**

O remetente das notificações é único e é composto pelos responsáveis do sistema, que são o grupo de desenvolvedores e os membros do experimento responsáveis pelas especificações das regras e funções da aplicação. Para tal utiliza-se um grupo de emails criado através da ferramenta de *egroups* do CERN, contendo todos os responsáveis pelo desenvolvimento e manutenção dos diferentes sistemas *Web*. O usuário agente das ações realizadas é adicionado ao corpo das mensagens transmitidas pelos sistemas, para que os usuários alvos possam saber quem foi o responsável pelas alterações.

Outro parâmetro de qualquer comunicação enviada pelos sistemas, são os destinatários das mensagens. Estes podem ser: um único usuário, grupos de usuários, *egroups* gerenciados pela colaboração, ou membros com uma função ou atividade

específica, por exemplo, líderes de grupo, chefes de projeto, a coordenação técnica, a coordenação de recursos ou a coordenação de controle do detector.

Os papéis mais importantes dos membros dos experimentos, envolvem um substituto, ou vice, que têm os mesmos privilégios em relação a qualquer atividade que gerencie. Tais cargos devem ser notificados sempre em conjunto. Por exemplo, sempre que alertada a coordenação técnica ou de operação das pesquisas, estão incluídos tanto os coordenadores principais, como seus vices. Esta relação entre cargos deve fazer parte de todos os casos existentes, como líderes de instituto que tenham vice-líderes, ou cargos de coordenação e liderança de projetos que também contenham múltiplos membros apontados para o mesmo cargo.

Em alguns casos, cargos devem incluir outros destinatários que não sejam membros apontados como substitutos ou vices. Por exemplo, sempre que notificada a coordenação de recursos do experimento, a assistência e secretaria do experimento também devem ser comunicadas. Todos estes cargos estão envolvidos com a gerência das instituições e dos contratos da colaboração. Ou seja, estes cargos devem ser agrupados de acordo com as atividades manipuladas pelas aplicações.

A Figura 6.1 ilustra a relação entre diferentes funções dos membros da colaboração e suas responsabilidades. A coordenação de recursos, juntamente com a secretaria do experimento, é responsável pelos membros e institutos da colaboração, portanto devem ser notificados em conjunto a partir de qualquer modificação relacionadas a estes dados. A coordenação de controle de operação, juntamente com a coordenação dos detectores, é responsável pelo controle das classes e plantões. Líderes e vice-líderes de institutos são responsáveis pelos contratos e atividades dos seus membros. A coordenação técnica e os líderes de projeto, mantém o controle das tarefas e projetos dos experimentos.

Os assuntos são distintos entre os diferentes tipos mensagens enviadas pelas aplicações. Por exemplo, todas as mensagens enviadas pelo sistema Membership são inicializadas com "ALICE Membership - ", seguidas pelo assunto da atividade específica. O mesmo se aplica para as mensagens enviadas pelo sistema de gerenciamento de plantões, bem como pelos outros sistemas de controle das atividades dos outros experimentos.



Figura 6.1. Funções dos membros da colaboração e suas responsabilidades.

Os comunicados são disparados a partir de diferentes fases dos sistemas, por exemplo, antes de estarem aptos a realizar plantões, membros devem realizar classes ministradas pela coordenação e treinamento nas salas de controle do detector. Estas diferentes fases podem ser definidas nas aplicações, e adicionadas ao assunto das mensagens, para que os receptores possam organizá-las e dar mais, ou menos urgência às ações a serem tomadas devido aos eventos do sistema.

O conteúdo das mensagens é diferente em cada notificação implementada. Este atributo é composto pelos textos que serão comunicados, e integrado com variáveis ou valores constantes presentes nos modelos de dados. Por exemplo, as notificações das alterações realizadas através das interfaces das aplicações, contém os valores antigos e novos dos atributos modificados no corpo das mensagens transmitidas.

Links de redirecionamento também são adicionados as mensagens, permitindo que os usuários envolvidos possam acessar, a partir do comunicado, a interface do sistema que gerou o comunicado. Por exemplo, as mensagens enviadas a partir das extensões de contratos dos membros, contém um botão que ao ser clicado, abre a interface de

modificação de contratos do membro alvo das alterações, em uma nova aba do navegador. Para que possam ser visualizados os novos valores alterados, evitando que os destinatários interessados acessem os sistemas para buscar a informação.

Regras são aplicadas antes do disparo de certas notificações, e se tratam de lógicas aplicadas sobre os registros manipulados. Como exemplo, dois dias antes de cada plantão, a coordenação da operação do detector deve ser comunicada somente caso, algum dos plantonistas mais experientes, conhecidos como *General Shifters* da colaboração, não houverem participado do curso de segurança das cavernas do LHC. Estes cursos são opcionais e devem ser realizados em diversos períodos do ano, devido as novas tecnologias utilizadas nos detectores, e a maior parte dos plantonistas não os faz. Estas regras são aplicadas antes do envio das mensagens, pois este comunicado é desnecessário para a maior parte dos membros.

No sistema Membership, os doutorandos participantes de instituições associadas a colaboração, não devem cumprir as tarefas de serviços obrigatórios do experimento. Portanto, antes do envio do comunicado contendo os deveres do novo colaborador, deve-se checar seu instituto de pesquisa, para que não sejam emitidas mensagens incorretas sobre as atividades obrigatórias do membro.

## **6.6 – As Configurações**

Os usuários mais importantes dos sistemas podem ser capazes de configurar os parâmetros das notificações, evitando o envolvimento do grupo técnico de desenvolvimento dos sistemas para estas modificações. Estes parâmetros configuráveis envolvem os textos das mensagens transmitidas, os destinatários, as variáveis adicionadas, a forma de transmissão, o nível do comunicado, o estado atual da atividade, ou mesmo a ativação ou desativação do envio das mensagens.

Os atributos configuráveis podem ser exibidos em interfaces dos sistemas, para que todos os usuários tenham acesso a saber durante quanto tempo quais eram as mensagens enviadas, e quais os destinatários por exemplo, bem como os usuários selecionados pelo controle de acesso que possam alterar estas informações.

Os usuários comuns receptores dos comunicados, podem definir quando querem ou não receber as notificações, podendo ser adicionados a grupos específicos de exceções quando não desejarem receber uma determinada notificação. Relativo as mensagens recebidas temporalmente, os mesmos usuários também podem definir quão regularmente querem ser comunicados sobre uma data atividade ou funcionalidade do sistema a qual acompanha.

## **6.7 – Validações da Comunicação**

Todos os parâmetros devem ser validados antes da transmissão das mensagens, garantindo o recebimento dos comunicados ou alertas por parte dos colaboradores. Por exemplo, os destinatários devem ser validados, fazendo referência a membros existentes, com contas de email e número de telefone celular registrados nos bancos de dados. As mensagens devem ser validadas para que não contenham erros de digitação que causem falhas em tempo de execução dos códigos, e variáveis válidos dos sistemas devem ser injetadas as mesmas, evitando o recebimento de mensagens com falta de informação.

As mensagens enviadas podem não atingir os destinatários em casos como, contas de email inexistentes, ou configurações de *egroups* que não podem receber mensagens enviadas de forma externa, ou seja, por contas de email que não façam parte do grupo. Portanto, antes da transmissão das notificações, estes tipos de validações devem ser realizadas, possibilitando que todos os membros sejam devidamente notificados. Caso algum erro que impossibilite a transmissão seja encontrado, este pode ser comunicado ao grupo técnico de desenvolvimento para que possam corrigir as inconsistências.

As aplicações podem também fazer sugestões de usuários a serem notificados, para os casos dos destinatários não encontrados, assegurando a tomada de ações em todas atividades gerenciadas. Em alguns casos, a omissão ou esquecimento de usuários que deveriam ser comunicados, gera reclamações por parte dos mesmos, devido ao não conhecimento de um determinado acontecimento, e também podem ser sugeridas pelas aplicações.

## **6.8 – Histórico de Configurações e das Mensagens Enviadas**

As mensagens enviadas devem ser registradas em bancos de dados, para que possam ser rastreadas pelos desenvolvedores, bem como serem exibidas através das interfaces, permitindo desta forma que os usuários também possam obter as informações através dos sistemas. Em determinados casos, ao serem modificados os membros presentes em um cargo, o novo membro não tem acesso as mensagens que foram enviadas ao membro anterior. A partir do registro destas mensagens, funcionalidades de retransmissão podem ser adicionadas às aplicações, percebendo as mudanças de atividades ou cargos dos membros, e realizando a retransmissão das mensagens à nova pessoa.

As adaptações realizadas nos parâmetros de configuração também podem ser registradas, possibilitando que os desenvolvedores e usuários dos sistemas saibam quais eram as configurações ativas durante um determinado período de tempo. Por exemplo, adaptações dos textos das mensagens enviadas, destinatários, status de disparo, juntamente com as datas das modificações e o membro responsável. Possibilitando a exibição de cronogramas com as configurações utilizadas pelas aplicações, e evitando que hajam dúvidas ou reclamações por parte dos usuários.

# Capítulo 7

## A Central de Notificações para o FENCE

Esta seção apresentará a arquitetura da Central de Notificações, através da extensão do *framework* FENCE. Inicialmente, serão introduzidos alguns conceitos que serviram de embasamento para a implementação deste projeto, estes são: Herança, Polimorfismo e Composição em OOP (*Object-Oriented Programming*), e o padrão de desenvolvimento do observador, ou *Observer Design Pattern*.

### 7.1 – Herança, Polimorfismo e Composição

Na área da Engenharia de Software existe uma série de mecanismos que favorecem a reutilização de funcionalidades, através do compartilhamento de código, evitando duplicação e facilitando a compreensão semântica da arquitetura. Estes padrões estão presentes em diversas linguagens de programação orientada a objetos, como em PHP (*Hypertext Preprocessor*), utilizado no desenvolvimento deste projeto.

Dentre eles, a Herança é utilizada através da extensão de classes, onde métodos presentes na Classe Pai, Classe Base ou Superclasse, são compartilhados pelas classes herdeiras, chamadas de Classes Filha ou Subclasses. Nesta arquitetura, procura-se generalizar as similaridades entre objetos em superclasses, difundindo funcionalidades comuns para seus herdeiros. Este acoplamento propicia que mudanças realizadas nas classes bases, sejam propagadas às suas referências.

O Polimorfismo, é um mecanismo de herança, que permite que as classes herdeiras representem o comportamento das classes que referenciam. Neste caso, as superclasses são abstratas, ou seja, não podem ser instanciadas diretamente, e descrevem um ou mais métodos abstratos que devem, obrigatoriamente, serem descritos por suas referências. Esta estrutura de desenvolvimento é importante quando, as subclasses compartilham uma mesma funcionalidade que, entretanto, é implementada de formas



diferentes. Um exemplo deste conceito pode ser representado por uma classe abstrata *Animal*, que descreve um método abstrato *Som*. Subclasses de animais como *Gato* e *Cachorro* estendem a superclasse *Animal*, e compartilham diversos atributos comuns, como a mesma família (mamíferos), porém implementam o método *Som* de formas diferentes.

Por sua vez, a Composição é uma arquitetura que delega a implementação de funcionalidades mais complexas à um objeto composto por objetos mais simples, permitindo que a associação entre eles seja alterada em tempo de execução. Seguindo o Princípio da Responsabilidade Única, cada objeto encapsula funcionalidades, que são acessadas somente através de suas interfaces, pelas classes controladoras, que decidem o que, e quando executá-las, fundamentando o princípio de programar para uma interface e não para uma implementação [40].

## 7.2 – O Padrão Observador

Este padrão consiste em anexar uma lista de observadores a outros objetos, conhecidos como sujeitos, definindo uma relação de um para muitos. Esta arquitetura envolve a composição dos sujeitos por seus observadores, tornando-se útil na implementação de sistemas distribuídos de manipulação de eventos, como arquiteturas de notificações, pois permite que um objeto notifique suas classes observadoras sobre alterações em seus estados [41].

No contexto de sistemas desenvolvidos em MVC, registra-se as classes observadores às representações de dados, que detêm a informação sobre seus estados, e a partir de modificações, avisam seus observadores sobre os novos valores do subconjunto dos atributos que observam. Outra forma de implementação se dá quando os modelos de dados avisam seus observadores que foram modificados, e estes podem por sua vez buscar os valores de interesse.

A Figura 7.1 ilustra a arquitetura apresentada. O método *attach* é o responsável por registrar observadores aos sujeitos, que executam o método *update* de cada um de seus observadores quando notadas alterações em seus estados. O método *update* implementa as ações que devem ser executadas a partir das mudanças de estados, e

atualizam por exemplo, *views* distintas. As interfaces de visualização podem obter o estado do sujeito, através do método *getState*. Estados podem ser vistos como os conjuntos dos atributos das classes de representação de dados, como por exemplo o telefone e o nome de um usuário.

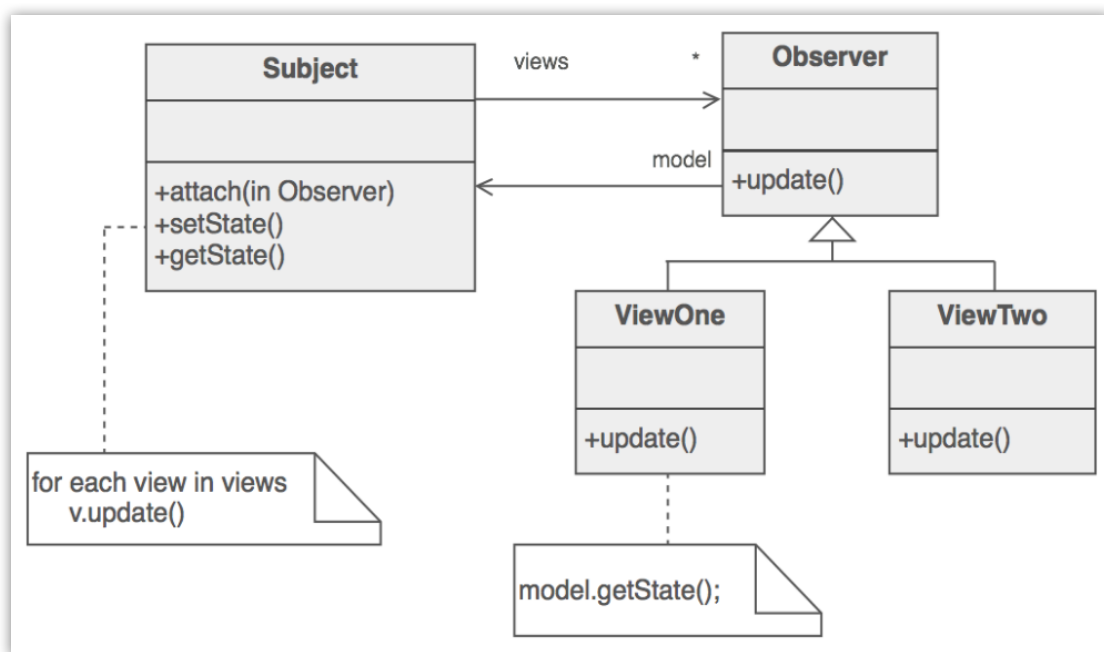


Figura 7.1. Estrutura de desenvolvimento do padrão observador. Extraído de [42].

Esta estrutura permite o encapsulamento de código, evitando duplicação e promovendo uma separação limpa entre camadas diferentes das aplicações, como a camada de lógica de dados e a camada de interface de usuário. Desta forma, são implementadas classes mais simples, facilitando a compreensão e manutenção do código por parte do grupo de desenvolvimento.

### 7.3 – Implementação

Os dados manipulados pelas funcionalidades dos sistemas podem ser estimulados de 3 formas diferentes, através de modificações, inserções ou remoções. A cada diferente estímulo, uma ou mais ações podem ser tomadas, desta forma, precisa-se mapear os diferentes eventos às classes que serão disparadas quando tais estados forem atingidos.

A classe *DBOf* do FENCE que é referenciada por todos os modelos de dados dos sistemas, foi expandida através do método *on*, que possibilita o mapeamento dos

observadores aos eventos. Este método recebe como dependências o evento ao qual o observador será injetado, e a própria classe observadora que tomará as ações quando disparada. Dentre os possíveis eventos, foram definidos: *update*, *insert* e *delete*. Quando chamado, o método *on* adiciona o observador recebido à lista de observadores do evento. Este novo método é propagado, por herança, à todos os modelos de dados dos sistemas que fazem referência a esta classe.

A injeção de observadores aos modelos de dados deve seguir o mesmo padrão de implementação, o que é realizado através da classe abstrata *Observer*, integrada também ao código do FENCE. Essa classe por sua vez, implementa 3 métodos que controlam o disparo dos observadores. O método público *trigger*, que é o disparador da classe, e tem como responsabilidade executar as regras referentes a execução da classe, presentes no método *shouldRun*, que retorna um booleano caso o disparo deva ser realizado. Por fim, sendo atingidas todas as condições, o método abstrato *run* é executado, e deve obrigatoriamente ser sobrescrito pelas classes filhas de *Observer*, tendo como responsabilidade disparar as notificações.

O disparo da classe observadora é realizado pelo método *trigger*, que recebe como argumentos a versão atual do modelo, e opcionalmente sua versão antes das modificações realizadas. Estes são passados aos métodos *shouldRun* e *run*, para que sejam utilizados na execução das regras caso necessário. Por exemplo, ao notificar-se instituições que ainda estejam abaixo de suas cotas de créditos, ou membros que tenham perdidos suas permissões de realizar plantões, deve-se comparar ambos os estados, passado e atual, dos modelos de dados para aplicação das regras.

A figura 7.2 apresenta a classe abstrata *Observer*. O método *shouldRun* por padrão retorna um valor verdadeira, ou seja, caso não sobrescritos pelas classes filhas, nenhuma regra de disparo é aplicada, e o observador será disparado. O método *trigger* tem como responsabilidade executar o método *run*, caso um valor verdadeiro seja retornado pela aplicação das regras em *shouldRun*, implementado (não obrigatoriamente), pelas classes filha. O método *run*, deve obrigatoriamente ser sobrescrito pelas classes filhas de *Observer*, pois é definido como um método abstrato, e tem como responsabilidade executar as funcionalidades necessárias ao ser disparada uma classe observadora.

```

abstract class Observer
{
    /**
     * Defines whether the observer should run
     * @param DBOf $model
     * @param DBOf|null $oldModel
     * @return boolean
     */
    protected function shouldRun(DBOf $model, DBOf $oldModel = null)
    {
        return true;
    }

    /**
     * Triggers observer
     * @param DBOf $model
     * @param DBOf|null $oldModel
     */
    public function trigger(DBOf $model, DBOf $oldModel = null)
    {
        if ($this->shouldRun($model, $oldModel)) {
            $this->run($model, $oldModel);
        }
    }

    /**
     * Runs observer and should be implemented by subclasses
     * @param DBOf $model
     * @param DBOf|null $oldModel
     */
    public abstract function run(DBOf $model, DBOf $oldModel = null);
}

```

Figura 7.2. A classe abstrata *Observer*, referenciada pelos observadores.

Os observadores injetados aos modelos de dados dos sistemas, são disparados a partir do método `trigger` adicionado à classe *DBOf*. Este método recebe como argumentos uma *string* contendo o evento que será disparado e ambas as versões, antiga e modificada do modelo de dados, e tem como responsabilidade iterar e disparar, através de suas interfaces, todos os observadores que foram previamente injetados pelo método *on*.

Esta arquitetura segue o padrão de desenvolvimento do observador, apresentado anteriormente, e permite que múltiplos observadores sejam injetados aos diferentes eventos aos quais os modelos de dados podem ser estimulados. Permitindo também que estes sejam disparados em tempo de execução pelos controladores dos sistemas.

Para que estes observadores possam cumprir o papel de notificar os usuários dos sistemas, outras classes que tem como responsabilidade a construção das notificações foram adicionadas ao código do FENCE. Estas serão mencionadas a seguir, e são

injetadas aos observadores, para que possam ser utilizadas no momento de seus disparos, realizando a transmissão das mensagens entre os sistemas e seus usuários.

A classe abstrata *Notifier* foi inserida para lidar com as notificações transmitidas pelos sistemas. Esta classe recebe como argumentos os parâmetros mínimos de comunicação, que são o emissor, o(s) receptor(es) e a mensagem, e tem como método abstrato *notify*, que permite, através de polimorfismo, diferentes formas de transmissão da comunicação, como por exemplo, por email, SMS ou *Web Push*. A transmissão será implementada através das classes filhas de *Notifier*, que deverão obrigatoriamente descrever como será realizado o disparo das mensagens através do método *notify*.

Para que seja possível a construção de notificações a partir de arquivos de configuração em JSON, as classes notificadoras contêm o método estático *summonFromJSON*. Este método recebe como dependência o arquivo de configuração que será utilizado para a construção da classe de notificação, e lida com a interpretação dos atributos descritos no arquivo. Isto é possível com a classe *JReader*, já presente no FENCE, que permite que os atributos presentes nos arquivos JSON, sejam interpretados e encapsulados em listas associativas em PHP.

A figura 7.3 apresenta um exemplo de configurações em JSON, já utilizado no sistema SAMS para notificações enviadas ao serem realizadas reservas de classes. O atributo *notify* do objeto pode receber um ou mais formatos de notificação, onde cada um deles deve conter o tipo de notificação que será enviada, como no exemplo, por email, uma *label* que é utilizada tanto assunto quanto no conteúdo da mensagem, para o reconhecimento do sistema de interesse, bem como a fase do trabalho e o nível do alerta, que pode variar entre informativo, aviso e urgente. As configurações também devem definir os receptores das mensagens, o assunto, a mensagem que será utilizada bem como constantes ou variáveis que poderão ser utilizadas tanto no assunto como no texto da mensagem.

```

"notify": {
  "type" : "email",
  "label": "ALICE SAMS",
  "phase": "Classes",
  "level": "INFO",
  "receptors": [
    {
      "type": "user",
      "kind": "to"
    },
    {
    },
    {
      "type" : "appointment",
      "mapper" : "RUN_COORDINATION",
      "factory": "\\Alice\\Sams\\Appointment\\Factory\\Appointment",
      "order" : {
    },
      "kind": "cc"
    }
  ],
  "subject": {
    "prefix" : "{{ self.label }} [{{ self.level }}]",
    "splitter": "-",
    "value" : "{{ model.class.detector.name }} Class Successfully Booked"
  },
  "data": {
    "runCoordination": {
    },
    "detectorCoordination": {
    }
  },
  "const": {
    "link": {
      "calendar": "/alice/sams/classes/calendar",
      "profile" : "/alice/sams/member/profile"
    }
  },
  "message": {
    "type" : "twig",
    "dir" : "/alice/sams/classes/assignment/notification/view",
    "template": "regularBook.twig"
  }
}

```

Figura 7.3. Configurações JSON para notificações enviadas quando usuários fazem reservas de classes no sistema SAMS.

Os receptores das mensagens podem variar entre diferentes tipos, dentre eles o próprio usuário agente das modificações, funções ou atividades dos membros, das quais

todos os membros que fazem parte serão alertados, e também *e-groups* do CERN. Como apresentado na Figura 7.3, tanto o membro que fez a reserva, bem como a coordenação do experimento será notificada sobre a reserva realizada. No caso de notificações por email, os receptores também devem conter o atributo “*kind*”, que define se a mensagem será enviada diretamente ao receptor caso este seja do tipo “*to*”, ou se o mesmo será adicionado em cópia da mensagem, caso “*cc*” ou “*bcc*”. Caso o atributo *kind* não tenha sido definido, por padrão utiliza-se *to*.

Constantes ou dados também podem ser definidos nas configurações. Os dados são construídos a partir de chamadas a métodos estáticos, ou “*callbacks*”, que tem por responsabilidade retornar os valores que serão utilizados no assunto ou conteúdo da mensagem transmitida.

No caso de notificações enviadas por email, o assunto, ou “*subject*” das mensagens deve ser definido, este por sua vez é um objeto que pode conter os atributos “*prefix*”, “*splitter*” e “*value*”, os quais serão concatenados e adicionados ao assunto do email. O prefixo e o valor do assunto podem fazer uso de valores como constantes e dados construídos no atributo “*data*”, bem como valores do próprio modelo de dados que foi modificado. Estes valores são adicionados através de duplas chaves, que contém as chaves das listas que devem ser acessadas separados por um ponto. Constantes e dados construídos são armazenadas na lista “*self*” e variáveis dos modelos são armazenadas e podem ser acessadas através da chave “*model*”. Como no exemplo apresentado na Figura 7.3, o assunto das mensagens é prefixado com a *label* e o nível da notificação, e o valor contém o nome do detector para a qual a classe foi reservada.

As mensagens podem ser construídas através de diferentes tipos, *strings*, listas de *strings* contendo as múltiplas linhas da mensagem, ou mesmo através de *templates* Twig. No exemplo apresentado na Figura 7.3, a mensagem notificada será construída através de um *template* descrito no atributo “*template*” e localizado no diretório definido em “*dir*”. Este arquivo deve conter o código HTML utilizado para a composição da mensagem, o qual pode receber constantes, variáveis ou valores dos modelos que são injetados da mesma forma que no assunto da mensagem, através de duplas chaves.

O emissor também pode ser definido no arquivo, mas caso omitido, utiliza-se por padrão o grupo de email utilizado pelo sistema, já definido em outros arquivos de configuração carregado pelo FENCE.

Os atributos apresentados e presentes nos arquivos de configuração são interpretados pelas classes *summonFromJSON*, e passados às respectivas fábricas de dados. Estas fábricas têm como responsabilidade interpretar os diferentes tipos de dados e construí-los, de forma que estes serão utilizados como dependências das classes notificadoras. Ao fim da construção dos dados, o método *summonFromJSON* cria a instância da classe de notificação, com suas dependências, que será utilizada pelos observadores para a transmissão das mensagens.

Os arquivos de configuração podem ser modificados pelos usuários dos sistemas através de outro projeto em desenvolvimento para o *framework* FENCE, conhecido como *WebEditor*. Esta ferramenta permite que os arquivos sejam exibidos em interfaces dos sistemas e que seus atributos estejam disponíveis para edição pelos próprios usuários dos sistemas, sendo estes capazes de adaptar atributos como receptores, assunto ou o conteúdo da mensagem transmitida, evitando a intervenção dos desenvolvedores quando se trata destes tipos de modificações.

Os arquivos são armazenados nos respectivos bancos de dados dos sistemas, assim como os usuários e as datas das modificações das configurações, mantendo-se desta forma o histórico dos estados das notificações em períodos específicos de tempo, bem como os responsáveis pelas adaptações.

A figura 7.4 apresenta a interface de modificação das configurações apresentadas na figura 7.3. Nesta interface, permite-se que sejam alteradas as configurações, atributos como a *label*, a fase, e o nível da notificação podem ser adaptados a partir dos *inputs* disponibilizados aos usuários. A fase do trabalho contém a lista com as diferentes fases do sistema, por exemplo no sistema SAMS existem as notificações enviadas a partir de classes, plantões, permissões de usuários, entre outros. Os níveis dos comunicados também são pré-definidos pela própria central de notificações e são selecionados a partir das listas de possíveis níveis. Outros atributos como receptores podem ser adicionados à transmissão da mensagem ao clicar no botão “*Add Receptor*”, assim como pode-se também adaptar ou remover os receptores atuais. As variáveis e constantes, bem como os textos utilizados no assunto e conteúdo da mensagem também podem ser adaptados pelos usuários aptos a modificar o arquivo.



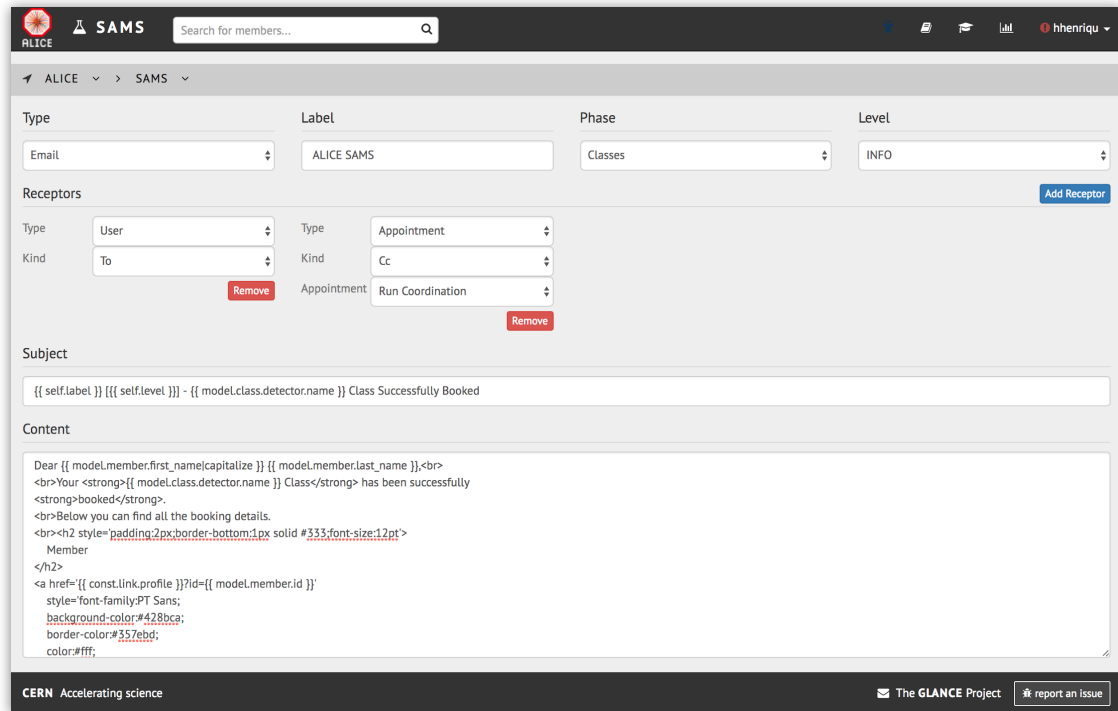


Figura 7.4. Interface de modificação das configurações de notificações.

É necessário que os atributos de configuração sejam validados após as modificações, já que são alterados pelos próprios usuários dos sistemas e estes podem introduzir erros. Estas validações são importantes para evitar a transmissão de mensagens aos receptores incorretos ou com conteúdo que utilize variáveis inexistentes, por exemplo.

As validações dos atributos de configuração são realizadas pelas próprias fábricas, antes da construção dos dados. Por exemplo, ao ser definida uma mensagem como do tipo “twig”, devem ser passados um diretório onde está localizado o *template* da mensagem, e o nome do *template* que será utilizado, caso algum dos atributos seja omitido, uma exceção é gerada contendo a mensagem referente ao erro, que será adicionada aos arquivos de *log*. O próprio *template* contendo o código HTML da mensagem que será transmitida também é validado, de forma que estes não contenham erros nas mensagens enviadas, garantindo a confiabilidade da comunicação da central de notificações.

Outros atributos como receptores, ou o assunto das mensagens, também são avaliados, onde garante-se que tanto os atributos de configuração, como o resultado construído estejam corretos. Permitindo desta forma que a comunicação transmitida

esteja de acordo com o que é desejado a partir das configurações descritas nos arquivos JSON.

Todas as mensagens transmitidas pela central de notificações são armazenadas na base de dados de cada sistema, em uma tabela específica criada para este propósito. Esta tabela contém a informação completa sobre qual tipo de mensagem transmitida (se por email ou SMS), qual a fase do sistema (como classes ou plantões, no caso do sistema SAMS), qual o nível do comunicado, qual o usuário receptor da mensagem, e qual o conteúdo da mensagem transmitida. Permitindo que estas informações sejam rastreadas pelos desenvolvedores, ou mesmo exibidas aos usuários receptores assim que autenticados nos respectivos sistemas.

Todas as modificações realizadas nos arquivos de configuração também são armazenadas nos bancos de dados. O próprio *WebEditor*, que permite a modificação dos arquivos, armazena as informações sobre qual usuário, em qual data e hora, modificou os arquivos de configurações, bem como qual o novo estado dos arquivos. Este histórico permite que sejam gerados diversos cronogramas sobre as notificações transmitidas pela central, por exemplo, durante qual período de tempo, quem eram os receptores de mensagens enviadas a partir da modificação de contratos, ou funções de membros dos experimentos, e qual o conteúdo da mensagem enviada a esses usuários.

A Figura 7.5 ilustra a modelagem de dados das notificações enviadas aos membros da colaboração e das configurações realizadas, onde as colunas da tabela armazenam os atributos mencionados anteriormente, suficientes para a retransmissão das mensagens e para a exibição dos estados das configurações durante os diferentes períodos de tempo.

A Figura 7.6 ilustra o funcionamento da Central de Notificações. À esquerda pode-se visualizar a estrutura de observadores, que são gerados a partir dos arquivos de configuração em JSON, após serem interpretados pelas fábricas adicionadas ao FENCE. À direita pode-se visualizar os observadores de modelos de dados, que são injetados aos modelos por controladores para os diferentes eventos observáveis, e são disparados automaticamente após a ocorrência dos eventos. Os observadores são compostos pelas classes notificadoras, e realizam a transmissão das mensagens após a aplicação das regras definidas nos arquivos de configuração, e o envio de cada mensagem é também registrados nas bases de dados.

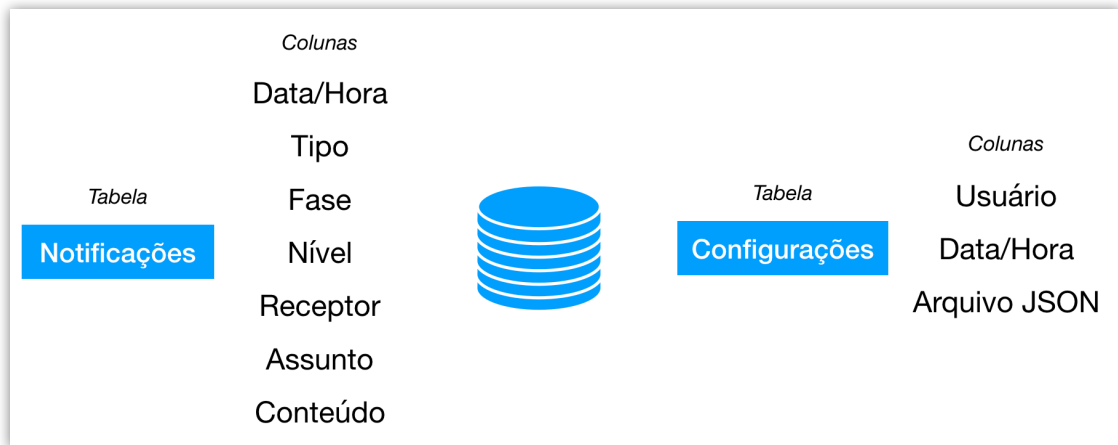


Figura 7.5. Modelagem de colunas e tabelas para as notificações enviadas e para as configurações realizadas pelos membros da colaboração.

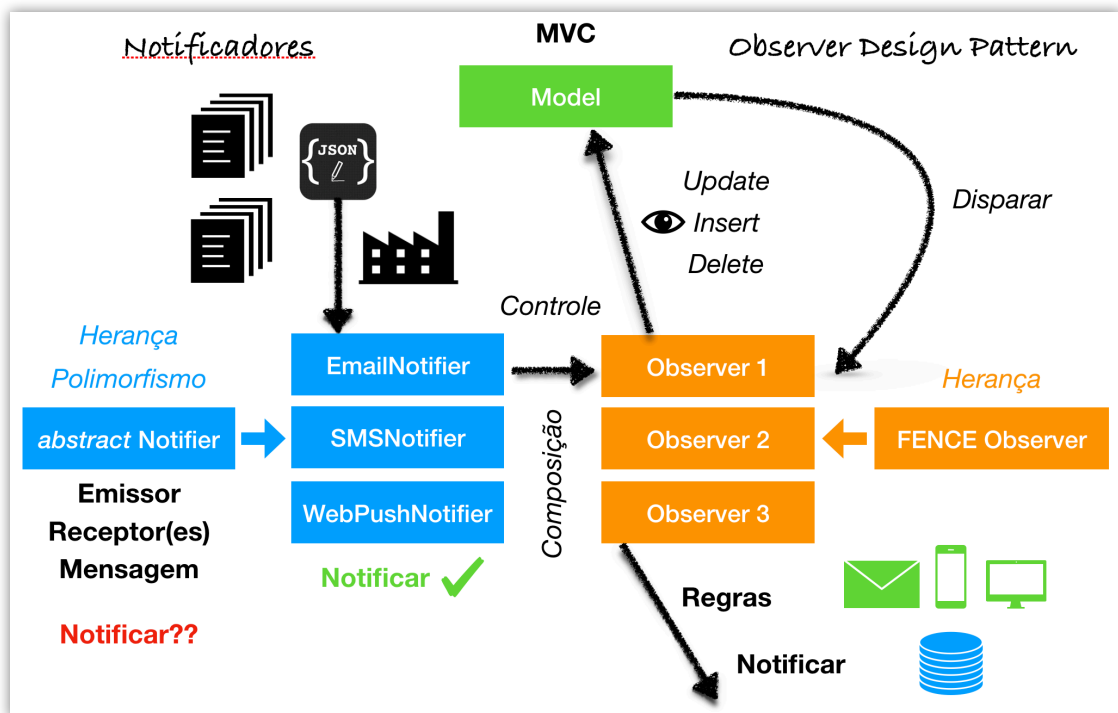


Figura 7.6. Funcionamento da Central de Notificações para o FENCE.

# Capítulo 8

## Conclusões

Em uma colaboração internacional de grande porte como o CERN, utilizam-se diversas ferramentas para a organização da colaboração e o controle das atividades nos detectores. Devido à dispersão geográfica dos pesquisadores, tecnologias *Web* são utilizadas para que as informações possam ser acessadas de qualquer lugar do mundo. As tecnologias Glance e FENCE, desenvolvidas pela UFRJ, impulsionaram a implementação de diversos sistemas para os experimentos ATLAS, ALICE e LHCb.

No experimento ALICE, percebeu-se a necessidade de comunicar os membros sobre as atividades realizadas pelos usuários através dos sistemas, e também sobre o estado dos dados já armazenados, melhorando desta forma a eficiência no controle e a garantia de confiabilidade das informações existentes. Para que esta comunicação pudesse ser realizada de forma configurável, pelos próprios membros dos experimentos, foi projetada e desenvolvida a Central de Notificações para o *framework* FENCE, utilizado como base para os sistemas *Web* dos experimentos.

A Central de Notificações permite que a comunicação realizada através dos sistemas possa ser gerada a partir de arquivos de configuração em JSON. Podendo-se definir múltiplos formatos de envio das mensagens, se por email, SMS ou *Web Push*, ou mesmo diferentes níveis de comunicados, se mais urgentes, ou simplesmente informativos. As configurações também permitem a integração com constantes e variáveis dos próprios sistemas, que podem ser utilizadas no assunto e conteúdo das mensagens transmitidas. Os receptores das mensagens podem ser descritos diretamente nos arquivos a partir de diversos tipos de dados, dentre os quais funções ou atividades dos membros da colaboração.

Todas as mensagens encaminhadas são armazenadas em históricos, contendo o usuário receptor, a hora de transmissão, e o conteúdo da mensagem, que podem ser exibidos nas interfaces dos próprios sistemas. Desta forma os próprios usuários são

capazes de rastrear as mensagens que já receberam, assim como o grupo técnico de desenvolvimento das aplicações. As modificações realizadas nas configurações são registradas para que possam ser gerados cronogramas sobre quem eram usuários notificados sobre uma determinada atividade, e qual mensagem era transmitida.

A Central de Notificações permite que a modificação das mensagens transmitidas pelos sistemas seja realizada pelos próprios usuários. A geração de novos comunicados também é feita de forma simples pelo grupo de desenvolvimento, bastando-se descrever um novo arquivo com as configurações das mensagens que devem ser transmitidas.

O trabalho já se encontra presente no *framework* FENCE e pode ser utilizado pelos sistemas que utilizam esta tecnologia. As notificações transmitidas pelo sistema SAMS são realizadas a partir de arquivos de configuração através da Central de Notificações, e tem previsão de entrar em produção ao fim de 2017.

Como trabalhos futuros, serão implementadas notificações enviadas a partir de regras aplicadas sobre os dados armazenados nos bancos, informando sobre seus estados, como por exemplo, institutos que se encontram abaixo de suas cotas de créditos de plantões, e também, gerando listas periodicamente com as atividades e funções dos membros do experimento, permitindo que os próprios usuários dos sistemas definam quão regularmente querem receber estas informações.

Outro aspecto importante ainda não implementado, é gerar uma prévia do conteúdo das mensagens configuradas pelos usuários. Como os textos podem ser descritos em arquivos HTML, é importante que seja apresentado ao usuário que realiza as configurações um “*preview*” da mensagem que será transmitida a partir do texto descrito pelo próprio usuário, evitando erros ou um conteúdo indesejado.

O projeto pode ser integrado, para o envio de comunicados, com outras classes e funcionalidades do *framework* FENCE, como por exemplo, o Report Generator, que permite configurar a lista resultante de buscas, para a geração gráficos, que podem ser exportados em formatos como PDF (*Portable Document Format*), e enviados à outros membros da colaboração.

# Bibliografia

- [1] CERN, <http://home.cern/about>, (Acesso em 27 de Março 2017).
- [2] O Complexo de Aceleradores do CERN, “*CERN’s Accelerator Complex*”  
<http://home.cern/about/accelerators>, (Acesso em 27 de Março 2017).
- [3] O Complexo de Aceleradores e os Experimentos, “*Accelerator’s Complex and Experiments*”  
<http://home.cern/about/accelerators>, (Acesso em 27 de Março 2017).
- [4] O Grande Colisor de Hádrons, “*The Large Hadron Collider*”  
<http://home.cern/topics/large-hadron-collider>, (Acesso em 27 de Março 2017).
- [5] RONCHETTI, Federico, “*Exit Protons Enter Ions*”, 2015  
<https://ep-news.web.cern.ch/content/exit-protons-enter-ions>, (Acesso em 27 de Março 2017).
- [6] TAYLOR, Lucas, “*Silicon Pixels*”, 2011  
<http://cms.web.cern.ch/news/silicon-pixels>, (Acesso em 29 de Março 2017).
- [7] Calorímetro do ATLAS, “*ATLAS Calorimeter*”  
<http://atlas.cern/discover/detector/calorimeter>, (Acesso em 29 de Março 2017).
- [8] LAUREN LEE, Jennifer, “*Muon Chambers*”, 2008  
[https://www-cdf.fnal.gov/virtualtour/muon\\_chambers.html](https://www-cdf.fnal.gov/virtualtour/muon_chambers.html), (Acesso em 30 de Março 2017).
- [9] TAYLOR, Lucas, “*Muon Detectors*”, 2011  
<http://cms.web.cern.ch/news/muon-detectors>, (Acesso em 30 de Março 2017).
- [10] PETERSEN, B A, “*ATLAS Trigger System*”, 36<sup>th</sup> International Conference on High Physics Energy, Melbourne, Australia, 2012 <https://cds.cern.ch/record/1459646/?ln=en>, (Acesso em 01 Abril 2017).
- [11] TAYLOR, Lucas, “*Triggering and Data Acquisition*”, 2011  
<http://cms.web.cern.ch/news/triggering-and-data-acquisition>, (Acesso em 01 de Abril 2017).
- [12] ATLAS, <https://atlas.cern/>, (Acesso em 01 de Abril 2017).
- [13] ATLAS Collaboration, <http://atlas.cern/discover/collaboration>, (Acesso em 01 de Abril 2017).
- [14] CMS Experiment, <https://home.cern/about/experiments/cms>, (Acesso em 04 de Abril 2017).
- [15] ALICE Experiment, <https://home.cern/about/experiments/alice>, (Acesso em 04 de Abril 2017).
- [16] LHCb Experiment, <https://home.cern/about/experiments/lhcb>, (Acesso em 04 de Abril 2017).
- [17] The Higgs Boson, <https://home.cern/topics/higgs-boson>, (Acesso em 05 de Abril 2017).
- [18] The Standard Model, <https://home.cern/about/physics/standard-model>, (Acesso em 05 de Abril 2017).
- [19] The Birth of Web, <https://home.cern/topics/birth-web>, (Acesso em 05 de Abril 2017).
- [20] GALVAO, K K, MAIDANTCHIK, C, GRAEL, F F and POMMES, K, “*Glance project: a database retrieval mechanism for the ATLAS detector*”. J. Phys.: Conf. Ser, 2008
- [21] LANGE RAMOS, B, MAIDANTCHIK, C, “*FENCE: Uma Abordagem Orientada a objetos na concepção de sistemas Web altamente configuráveis para os experimentos do CERN*”. UFRJ, <http://monografias.poli.ufrj.br/monografias/monopoli10014830.pdf>, 2015

- [22] Introducing JSON, <http://json.org/>, (Acesso em 15 de Abril 2017).
- [23] Twig, The flexible, fast and secure template Engine for PHP, <https://twig.sensiolabs.org/>, (Acesso em 15 de Abril 2017).
- [24] Source Making, *The Builder Design Patter*, [https://sourcemaking.com/design\\_patterns/builder](https://sourcemaking.com/design_patterns/builder), (Acesso em 16 de Abril 2017).
- [25] ATLAS TCn Applications, <https://atglance.web.cern.ch/atglance/ATLASTCnDB/>, (Acesso em 16 de Abril 2017).
- [26] MARTINS SILVA, H H, BOETTGER, S, TELESCA, A, MAIDANTCHIK, C, ABREU DA SILVA, I, “*The ALICE Glance Membership Management System*”. 21st International Conference on Computing in High Energy and Nuclear Physics, (CHEP2015), <https://indico.cern.ch/event/304944/contributions/1672489/>, 2015
- [27] MARTINS SILVA, H H, RONCHETTI, F, TELESCA, A, MAIDANTCHIK, C, “*The ALICE Shift Accounting Management System*”. 21st International Conference on Computing in High Energy and Nuclear Physics, (CHEP2015), <https://indico.cern.ch/event/304944/contributions/1672490/>, 2015
- [28] The Editors of Encyclopedia Britannica, “*Protocol Computer Science*”, <https://www.britannica.com/technology/protocol-computer-science>, (Acesso em 29 de Abril 2017).
- [29] “*Four Layers of TCP/IP model, Comparison and Difference between TCP/IP and OSI models*”, <http://www.omnisecu.com/tcpip/tcpip-model.php>, (Acesso em 29 de Abril 2017).
- [30] VINTON G. GERF and ROBERT E. KHAN, Member IEEE, “*A Protocol for Packet Network Intercmunication*”, <https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf>, 1974
- [31] “*How Encapsulation Works Within the TCP/IP Model*”, <http://networkingtips-tricks.blogspot.ch/2010/05/how-encapsulation-works-within-tcpip.html>, (Acesso em 02 de Maio 2017).
- [32] RFC 5322, “*Internet Message Format*”, P. RESNICK ED. 2008, IETF <https://tools.ietf.org/pdf/rfc5322.pdf>, (Acesso em 03 de Maio 2017).
- [33] Simply Explained, “*SMTP (Simple Mail Transfer Protocol)*”, <http://www.effecthacking.com/2014/07/smtp-simple-mail-transfer-protocol.html>, (Acesso em 07 de Maio 2017).
- [34] RFC 5321, “*Simple Mail Transfer Protocol*”, J. KLENSIN. 2008, IETF <https://tools.ietf.org/pdf/rfc5321.pdf>, (Acesso em 07 de Maio 2017).
- [35] RFC 1939, “*Post Office Protocol, Version 3*”, J. MYERS, CARNEGIE MELLON, M. ROSE, 1996, IETF, <https://tools.ietf.org/pdf/rfc1939.pdf>, (Acesso em 07 de Maio 2017).
- [36] RFC 3501, “*Internet Message Access Protocol*”, M. CRISPIN, University of Washington 2003, IETF, <https://tools.ietf.org/pdf/rfc3501.pdf>, (Acesso em 09 de Maio 2017).
- [37] TRIGGS, R, “*What is SMS and how does it work?*”, 2013, <http://www.androidauthority.com/what-is-sms-280988/>, (Acesso em 15 de Maio 2017).
- [38] KANSAL, A, “*7 Thing to Know Before Starting With Website Push Notifications*”, 2016, <https://vwo.com/blog/before-starting-website-push-notifications/>, (Acesso em 17 de Maio 2017).
- [39] BANSAL, V, “*An introduction to website push notifications – technology, use cases and the way forward*”, 2016, <https://www.linkedin.com/pulse/introduction-website-push-notifications-technology-use-vipul-bansal>, (Acesso em 18 de Maio 2017).

- [40] ERIC FREEMAN and ELISABETH ROBSON, BERT BATES, KATHY SIERRA, "*Head First Design Patterns A Brain Friendly Guide*", O'Reilly Media, 2009
- [41] Source Making, "*Observer Design Pattern*"  
[https://sourcemaking.com/design\\_patterns/observer](https://sourcemaking.com/design_patterns/observer), (Acesso em 17 de Junho 2017).
- [42] "*Observer Design Pattern in PHP*"  
<https://php.earth/doc/oop/design-patterns/observer>, (Acesso em 17 de Junho 2017).
- [43] "*The Nobel Prize in Physics 1984, Carlo Rubia and Simon van der Meers*",  
[https://web.archive.org/web/20150510022145/http://www.nobelprize.org/nobel\\_prizes/physics/laureates/1984/rubbia-bio.html](https://web.archive.org/web/20150510022145/http://www.nobelprize.org/nobel_prizes/physics/laureates/1984/rubbia-bio.html), (Acesso em 21 de Julho 2017).
- [44] JOHN MARRINER, "*Stochastic Cooling Overview*", *11 de Agosto de 2003*  
<https://arxiv.org/abs/physics/0308044>, (Acesso em 21 de Julho 2017).