

Otávio Murilo da Paz  
Raphael Pereira Policena Rosa

# **SGCO Electa - Sistema de Gerenciamento de Curso Online**

Urutaí

27 de fevereiro de 2020

Otávio Murilo da Paz  
Raphael Pereira Policena Rosa

## **SGCO Electa - Sistema de Gerenciamento de Curso Online**

Trabalho de conclusão do curso Bacharelado em Sistemas de Informação - Campus Urutaí apresentado como parte dos requisitos para conclusão do curso.

Instituto Federal Goiano - Campus Urutaí  
Bacharelado em Sistemas de Informação  
Núcleo de Informática

Orientador: Paulo Henrique Garcia Mansur

Urutaí  
27 de fevereiro de 2020

Sistema desenvolvido pelo ICMC/USP  
Dados Internacionais de Catalogação na Publicação (CIP)  
**Sistema Integrado de Bibliotecas - Instituto Federal Goiano**

Paz, Otávio Murilo da

PP348s       SGCO Electa - Sistema de Gerenciamento de Curso  
Online / Otávio Murilo da Paz, Raphael Pereira  
Policena Rosa. -- Urutaí, 2020.  
45 f.  
Orientador: Paulo Henrique Garcia Mansur.

Monografia (Bacharelado em Sistemas de  
Informação) -- Instituto Federal Goiano, Campus  
Urutaí, 2020.

1. Sistemas. 2. Informação. 3. Oficinas. 4.  
Socioeducativo. I. Mansur, Paulo Henrique Garcia ,  
orient. II. Título.



**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR PRODUÇÕES TÉCNICO-CIENTÍFICAS NO REPOSITÓRIO INSTITUCIONAL DO IF GOIANO**

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia Goiano, a disponibilizar gratuitamente o documento no Repositório Institucional do IF Goiano (RIIF Goiano), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IF Goiano.

**Identificação da Produção Técnico-Científica**

- |  |   |
|--|---|
| <input type="checkbox"/> Tese  | <input type="checkbox"/> Artigo Científico              |
| <input type="checkbox"/> Dissertação                                 | <input type="checkbox"/> Capítulo de Livro              |
| <input type="checkbox"/> Monografia – Especialização                 | <input type="checkbox"/> Livro                          |
| <input checked="" type="checkbox"/> TCC - Graduação                  | <input type="checkbox"/> Trabalho Apresentado em Evento |
| <input type="checkbox"/> Produto Técnico e Educacional - Tipo: _____ |   |

Nome Completo do Autor: **Otávio Murilo da Paz e Raphael Pereira Policena Rosa**

Matrícula: **2016101202010284, 201611202010240**

Título do Trabalho:

**Restrições de Acesso ao Documento**

Documento confidencial:  Não  Sim, justifique: \_\_\_\_\_

Informe a data que poderá ser disponibilizado no RIIF Goiano: **28/02/20**

O documento está sujeito a registro de patente?  Sim  Não

O documento pode vir a ser publicado como livro?  Sim  Não

**DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA**

O/A referido/a autor/a declara que:

- o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia Goiano os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia Goiano.

Urutai, 26 / 02 / 2020  
Local Data

Otávio Murilo da Paz Raphael P. Policena Rosa  
Assinatura do Autor e/ou Detentor dos Direitos Autorais

Ciente e de acordo:

[Assinatura]  
Assinatura do(a) orientador(a)

**OTÁVIO MURILO DA PAZ  
RAPHAEL PEREIRA POLICENA ROSA**

**SGCO ELECTA – SISTEMA DE GERENCIAMENTO DE  
CURSO ONLINE**

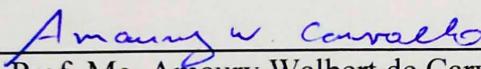
Monografia, defendida por Otávio Murilo da Paz e Raphael Pereira Policena Rosa, apresentado ao Instituto Federal de Educação, Ciência e Tecnologia Goiano, como parte das exigências para a obtenção do título de Bacharel em Sistemas de Informação, aprovados pela banca examinadora.

**COMISSÃO EXAMINADORA**



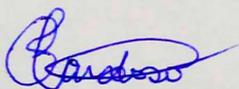
---

Prof. Dr. Paulo Henrique Garcia Mansur  
Orientador



---

Prof. Me. Amaury Walbert de Carvalho  
Avaliador



---

Profa. Dra. Cristiane de Fátima dos Santos Cardoso  
Avaliadora

Urutaí (GO), 13 de dezembro de 2019.

*Este trabalho é dedicado a Deus, familiares e a todos os professores, que de alguma forma contribuíram para a nossa formação.*

# Agradecimentos

Agradecemos os esforços de nossos pais que se preocupam com nosso futuro e sempre buscaram oferecer educação pessoal e profissional de qualidade.

Agradecemos todos os integrantes do Núcleo de Informática do Instituto Federal Goiano - Campus Urutaí pela paciência, motivação, dedicação e esforço.

*“Não vos conformeis com este mundo,  
mas transformai-vos pela renovação do vosso espírito,  
para que possais discernir qual é a vontade de Deus,  
o que é bom, o que lhe agrada e o que é perfeito.  
(Bíblia Sagrada, Romanos 12, 2)*

## Resumo

Os Sistemas de Informação interferem diretamente nos processos internos das empresas, pois facilitam o trabalho e gestão do negócio a qual atendem, além de possibilitar acesso rápido a informação economizando horas de trabalho. Deste modo, tais *softwares* podem influenciar diretamente no lucro das empresas, sendo utilizados por vários usuários simultaneamente. Com isso, este trabalho tem como objetivo desenvolver um *software* para a gestão de Oficinas Socioeducativas, as quais possuem o objetivo de oportunizar o exercício da cidadania entre os participantes, por meio da promoção integral, convívio familiar e social. Assim, com base em atividades culturais, comunicação e envolvimento com a comunidade obtém-se o crescimento individual e seu desenvolvimento como um bom cidadão. Nas oficinas socioeducativas todo o controle de presença, relatórios de atividades e controle de informação dos alunos eram feitos manualmente. Toda a parte manual levava muito tempo para realizar as tarefas, além de, gasto de papel, perda de informações e retrabalho. Desta forma, foi requisitado o desenvolvimento do Sistema de Gerenciamento de Curso Online (SGCO) para a gestão das oficinas socioeducativas, para um melhor gerenciamento de alunos e dos relatórios das atividades. O SGCO foi desenvolvido utilizando a linguagem de programação Java, com o JPA (*Java Persistence API*) utilizado para persistir os dados no banco de dados relacional MySQL no *Backend* e o HTML5, CSS, Bootstrap, JavaScript e JSTL no *Frontend*. A metodologia utilizada para o desenvolvimento foi o SCRUM. O sistema, juntamente com o banco de dados MySQL foi hospedado na Jelastic, no provedor Virtual Cloud, uma plataforma em nuvem DevOps. As prefeituras de Caieiras e de Araçoiaba da Serra do estado de São Paulo utilizam o sistema para gerenciar suas Oficinas Socioeducativas.

**Palavras-chave:** Sistemas. Informação. Oficinas. Socioeducativo.

# Abstract

Information Systems directly interfere in the internal processes of companies, as they facilitate the work and management of the business they serve, in addition to allowing quick access to information saving hours of work. In this way, such softwares can directly influence the profit of companies, being used by several users simultaneously. Thus, this work aims to develop a software for the management of Socioeducative Workshops, which have the objective of providing opportunities for the exercise of citizenship among the participants, through integral promotion, family and social interaction. Thus, based on cultural activities, communication and involvement with the community, individual growth and development as a good citizen are obtained. In socio-educational workshops, all attendance control, activity reports and student information control were done manually. The entire manual part took a long time to perform the tasks, in addition to paper waste, loss of information and rework. Thus, it was requested to develop the Online Course Management System (SGCO) for the management of socio-educational workshops, for better management of students and activity reports. SGCO was developed using the Java programming language, with the JPA (Java Persistence API) used to persist the data in the MySQL relational database at Backend and HTML5, CSS, Bootstrap, JavaScript and JSTL no Frontend. The methodology used for the development was SCRUM. The system, together with the MySQL database, was hosted on Jelastic, on the provider Virtual Cloud, a cloud platform DevOps. The municipalities of Caieiras and Araçoiaba da Serra in the state of São Paulo use the system to manage their Socio-Educational Workshops.

**Keywords:** Systems. Information. Workshops. Socio-Educational.

## Lista de ilustrações

Figura 1 – Painel de configuração da topologia de infraestrutura . . . . .	10
Figura 2 – Principais componentes da arquitetura do Tomcat . . . . .	12
Figura 3 – Ciclo Scrum . . . . .	16
Figura 4 – Diagrama entidade relacionamento inicial . . . . .	20
Figura 5 – Diagrama entidade relacionamento atual . . . . .	21
Figura 6 – Diagrama UML . . . . .	23
Figura 7 – Diagrama de pacotes . . . . .	24
Figura 8 – BPMN - Administrador parte 01 . . . . .	26
Figura 9 – BPMN - Administrador parte 02 . . . . .	27
Figura 10 – BPMN - Administrador parte 03 . . . . .	28
Figura 11 – BPMN - Administrador parte 04 . . . . .	29
Figura 12 – BPMN - Administrador parte 05 . . . . .	30
Figura 13 – BPMN - Prefeitura / Secretário parte 01 . . . . .	31
Figura 14 – BPMN - Prefeitura / Secretário parte 02 . . . . .	32
Figura 15 – BPMN - Professores parte 01 . . . . .	33
Figura 16 – BPMN - Professores parte 02 . . . . .	34
Figura 17 – Operações CRUD . . . . .	35
Figura 18 – Modal para operações, adicionar, editar e remover . . . . .	36
Figura 19 – Tela inicial para usuários administradores . . . . .	36
Figura 20 – Painel administrativo das oficinas - Visão geral . . . . .	37
Figura 21 – Painel administrativo das oficinas - Alunos matriculados . . . . .	37
Figura 22 – Geração de relatórios . . . . .	38
Figura 23 – Gerenciamento dos horários de aula para cada oficina . . . . .	38
Figura 24 – Tela inicial para professores . . . . .	39
Figura 25 – Adicionar aula em uma oficina . . . . .	40
Figura 26 – Diário de frequência . . . . .	40
Figura 27 – Lançar faltas . . . . .	41
Figura 28 – Acessar aula . . . . .	41
Figura 29 – Autenticação de documentos . . . . .	42

## Lista de abreviaturas e siglas

BPMN	Business Process Model and Notation
CPU	Central Process Unit
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DER	Diagrama Entidade Relacionamento
EL	Expression Language
FTP	File Transfer Protocol
GCP	Google Cloud Platform
GPL	General Public License
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISV	Independend Software Vendor
JCP	Java Community Process
JDBC	Java Database Connectivity
JDK	Java Development Kit
JRE	Java Runtime Environment
JS	JavaScript
JSR	Java Specification Requests
JSTL	Java Standard Tag Library
JVM	Java Virtual Machine
MVC	Model-View-Controller
NDB	Network Database
ORM	Object-relational mapping
PaaS	Plataform as a Service

RAM	Random Access Memory
RDBMS	Relational Database Management Systems
SGDB	Sistema de Gerenciamento de Banco de Dados
SO	Sistemas Operacionais
SQL	Structure Query Language
TCC	Trabalho de Conclusão de Curso
TFS	Team Foundation Server
UML	Linguagem Unificada de Modelagem
URI	Uniform Resource Identifier
VCS	Version Control System
W3C	World Wide Web Consortium

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Referencial teórico</b>	<b>2</b>
2.1	W3C	2
2.2	HTML	3
2.3	CSS	3
2.4	JavaScript	3
2.5	JQuery	4
2.6	Bootstrap	5
2.7	JSTL	5
2.8	Java	6
2.9	MySQL e MariaDB	7
2.9.1	MySQL	7
2.9.2	MariaDB	8
2.10	Jelastic	9
2.11	Git e GitHub	10
2.12	Apache Tomcat e TomEE	11
2.12.1	Apache Tomcat	11
2.12.2	TomEE	12
<b>3</b>	<b>Engenharia de software</b>	<b>14</b>
3.1	Requisitos	14
3.2	Diagrama entidade relacionamento (DER)	14
3.3	Linguagem Unificada de Modelagem (UML)	15
3.4	Diagrama de pacotes	15
3.5	Métodos ágeis	15
3.5.1	Scrum	15
3.6	Arquitetura de software	17
3.6.1	Arquitetura Model-View-Controller (MVC)	17
3.6.2	Arquitetura cliente-servidor	17
3.7	Projeto do software	18
3.7.1	Business Process Model and Notation (BPMN)	18
3.8	Casos de teste - Verificação e Validação	18
3.9	Homologação	18
<b>4</b>	<b>O SGC O Electa</b>	<b>20</b>
4.1	Projeto de software	20
4.2	Apresentação das funcionalidades	35
<b>5</b>	<b>Considerações finais</b>	<b>43</b>

**REFERÊNCIAS** ..... 44

# 1 Introdução

Pode ser observado que a tecnologia está cada vez mais presente no cotidiano das pessoas. Constantemente são lançados diversos sistemas e/ou aplicativos, que facilitam de alguma forma atividades rotineiras. Como demonstrado nos exemplos de (SILVA; URSSI, 2015), tais softwares tornaram-se ferramentas que são utilizadas diariamente por inúmeros usuários com alta frequência, além de, possuir grande capacidade para auxiliar na gestão de negócios, influenciando diretamente no lucro das empresas.

O autor (LEMOS, 2005) relata que surgem várias oportunidades de negócios no desenvolvimento de software como forma de facilitar pequenos processos, agilizando tarefas do cotidianos dos usuários, tornando cômodas e práticas as atividades do dia a dia como, fazer compras, agendar consultas e chamar um táxi.

Foi encontrado uma demanda de licitação no estado de São Paulo para um software web cujo objetivo é gerenciar pequenos projetos ofertados gratuitamente por algumas prefeituras do estado. A licitação foi ganha em parceria com a empresa "ELECTA Prestação de Serviços em Tecnologia Educacional Ltda". Com o desenvolvimento do software realizado no decorrer do curso propomos elaborar o Trabalho de Conclusão de Curso (TCC) com base no mesmo. A metodologia do trabalho consistiu em definir quais tecnologias são utilizadas, como serão utilizadas e como se comunicam entre si. No decorrer do desenvolvimento a fim de gerenciar o fluxo de trabalho foi utilizado o Scrum.

O principal requisito da licitação é o gerenciamento das estruturas administrativas relacionadas as oficinas socioeducativas por prefeituras. Cada prefeitura possui unidades de ensino, cada unidade de ensino possui suas oficinas, professores e alunos. Outros requisitos como controle de frequência e geração de relatórios também estão no escopo da aplicação. A proposta de desenvolvimento deste software atendeu as necessidades da licitação, proporcionando uma aplicação totalmente focada no negócio, com grande capacidade para futuras melhorias.

O presente trabalho encontra-se dividido da seguinte forma: Referencial teórico com breve explicação de todas as tecnologias e metodologias de desenvolvimento de software empregadas; Engenharia de software com as definições de componentes, regras de negócios e arquitetura; Descrição do software web com detalhamento das principais funcionalidades; Conclusão.

## 2 Referencial teórico

Durante o decorrer do trabalho foram utilizadas várias tecnologias. Linguagens de programação e marcação, bancos de dados, ambientes de desenvolvimento integrado (IDE), frameworks, controle de versionamento, ambientes de produção e metodologias de desenvolvimento de software são algumas categorias que as tecnologias utilizadas se encaixam. A seguir são descritas as ferramentas e tecnologias utilizadas no desenvolvimento deste projeto.

### 2.1 W3C

World Wide Web Consortium (W3C) é um consócio que define especificações e padrões para a web, buscando melhorar o ambiente com foco na arquitetura, web semântica, design, tecnologias web, navegadores e ferramentas.

O W3C desenvolve especificações técnicas e orientações através de um processo projetado para maximizar a consenso sobre as recomendações, garantindo qualidades técnicas e editoriais, além de transparentemente alcançar apoio da comunidade de desenvolvedores, do consórcio e do público em geral (W3C, 2011).

Uma das especificações acima citada é a *Web Semântica* cujo objetivo é padronizar os dados para que computadores e máquinas possam entender conteúdos publicados na web, podendo assim, melhorar as buscas realizadas por Google, Yahoo, Microsoft Bing, Baidu e outros mecanismos que ajudam usuários a encontrar os conteúdos que estão procurando na internet. Estão listadas abaixo algumas das tecnologias especificadas por este consórcio.

- HTML5
- CSS
- SVG
- Ajax
- URI
- HTTP
- XML
- SOAP
- WSDL
- SPARQL

## 2.2 HTML

A Linguagem de Marcação de HiperTexto, mais conhecida como HTML, utilizada para construção de páginas web e considerada a base de toda aplicação web. Todos os novos recursos inseridos nesta linguagem são controlados pelo consórcio W3C, como é o caso da a versão mais recente HTML5. O termo *HiperTexto* se refere aos *links* que possibilitam a conexão entre páginas web, seja dentro de um único site ou até mesmo em sites diferentes.

O HTML utiliza marcações que definem comportamentos diferentes de acordo com o contexto. Existem marcações específicas para textos, imagens, links, desenhos, comportamento, tabelas, entre vários outros. Observe as seguintes marcações como exemplos: `<header></header>`, `<p></p>`, `<img src= />`; todas possuem marcações de início `<>` e fechamento `</>`, essa estrutura é conhecida como *tag* (FERREIRA; EIS, 2011).

Com a utilização do HTML definimos toda a estrutura do site a ser desenvolvido juntamente com outras tecnologias de desenvolvimento como, CSS e JavaScript.

## 2.3 CSS

Folhas de Estilo em Cascata, conhecido como CSS é uma linguagem voltado para estilos, que manipulam *tags* HTML possibilitando personalizar o visual e melhorar a apresentação do site. Além do HTML, o CSS consegue também manipular outras linguagens, tais como, SVG, MathML ou XHTML. O consórcio W3C é que mantém os padrões do CSS, como antes dito, buscando melhorar a semântica da web.

Com o CSS podemos definir propriedades de elementos HTML como, cores, altura, largura, sombra, degradê, comportamentos perante ações do usuário, fontes (tamanho, família), desenhar elementos geométricos, animações e algumas outras características (BOS et al., 2005).

## 2.4 JavaScript

JavaScript (JS) é uma linguagem de programação que possibilita realizar tarefas em páginas web com estruturas mais complexas não suportadas por HTML e CSS. Essa é a premícia que deu origem ao JS, porém, com sua evolução, possibilita a criação de páginas web com frameworks <sup>1</sup> robustos tais como, Angular, ReactJS e Vue.JS, além de aplicações desktop com Electron e até mesmo aplicativos mobile com React Native, Ionic e NativeScript. Todos os frameworks citados acima estão relacionados com o usuário, o que chamamos de front-end (FLANAGAN; NOVAK, 1998).

O desenvolvedor front-end é responsável por “dar vida” à interface. Trabalha com a parte da aplicação que interage diretamente com o usuário. Por isso, é importante que esse desenvolvedor também se preocupe com a experiência do

---

<sup>1</sup> Framework é um conjunto de abstrações de códigos que simplificam e auxiliam no desenvolvimento, provendo funcionalidades genéricas que podem ou não se tornar específicas (WILLIS, 1996).

usuário. As principais linguagens utilizadas por este profissional são: HTML, CSS e JavaScript (VIANA, 2017).

A parte do processamento dos dados é denominada back-end que atualmente (2019) é dominada pelas linguagens de programação Java, C# e PHP. O JS também entrou na parte do back-end com o surgimento do Node.JS que basicamente é um gerenciador de pacotes, semelhante ao Maven e Gradle do Java, NuGet da plataforma .Net e pip do Python. Dentro do Node.JS a *framework* mais popular atuando no back-end é o ExpressJS que faz toda parte de mapeamento objeto-relacional (ORM), manipulação de arquivos e levantamento de serviços RESTful.

Back End, como o próprio nome sugere, vem da ideia do que tem por trás de uma aplicação. Pode ficar meio abstrato em um primeiro momento, mas pense que para conseguir usar o Facebook no dia a dia, os dados do seu perfil, amigos e publicações precisam estar salvos em algum lugar, sendo esse lugar um banco de dados e processados a partir de lá. Não basta apenas o front-end em HTML e CSS! O Back End trabalha em boa partes dos casos fazendo a ponte entre os dados que vem do navegador rumo ao banco de dados e vice-versa, sempre aplicando as devidas regras de negócio, validações e garantias em um ambiente onde o usuário final não tenha acesso e possa manipular algo (SOUTO, 2017).

Restful é uma API (Interface Application Program) que utiliza requisições HTTP para extrair, inserir, postar e deletar dados. Uma API nada mais é do que um código programável que permite que dois softwares diferentes comuniquem-se entre si. A API RESTful tem como base a tecnologia REST (representational state transfer), um tipo de arquitetura e comunicação muito utilizado no desenvolvimento de serviços web. O REST utilizado pelos browsers de internet pode ser imaginado como a linguagem da internet. Com o aumento da utilização da nuvem, o REST é uma escolha lógica para a construção de APIs que permitem ao usuário conectar e interagir com aplicações na nuvem. APIs RESTful são usadas por sites como Google, Amazon, LinkedIn e Twitter (TÁVORA, 2019).

Existem algumas vertentes dentro do universo JS que buscam aperfeiçoá-lo em pontos criticados pela comunidade, como este: *comparar se "0" é igual a "null" o resultado retornado é verdadeiro*. Uma dessas vertentes é o TypeScript, que assemelha a escrita de algoritmos - que posteriormente são convertidos para o JS durante a compilação - a linguagens de programação como Java e C#, facilitam e aumentam a velocidade de entrega de produtos no caso de profissionais *Full Stack*<sup>2</sup>.

## 2.5 JQuery

É a biblioteca mais popular do universo JS, com código fonte aberto, sobre as licenças MIT e GPL. Desse modo, podemos utilizar para fins comerciais e pessoais sem desembolsar nada por isso. Sua criação tem como foco facilitar o processo de desenvolvimento, permitindo a escrita de algoritmo compactos em relação ao JS puro (NIXON, 2014).

<sup>2</sup> Programador fullstack é aquele que atua em várias partes do projeto (back-end, front-end, banco de dados) e para isso muitas vezes precisa ter conhecimento em várias tecnologias (DEV MEDIA, 2017).

A aplicação do JQuery é dada juntamente com páginas HTML no lado do usuário (*client-side*) simplificando a navegação em documentos HTML, criação de animações, desenvolvimento de aplicações AJAX, a seleção de elementos DOM, manipulação de eventos e a criação de novos *plugins* sobre ela. É considerada uma biblioteca leve, contendo aproximadamente 90 kB em sua versão de produção.

## 2.6 Bootstrap

A biblioteca Bootstrap é a biblioteca de componentes *front-end* mais popular do mundo fornecendo facilidade na criação de páginas web, contendo recursos já estilizados e totalmente responsivo (adapta a praticamente qualquer tamanho de tela). Sua orientação principal é realizar primeiro o desenvolvimento de sites para dispositivos móveis (*mobile-first*) forçando simplicidade de design e conseqüentemente melhorando a usabilidade da interface (SPURLOCK, 2013).

Esta biblioteca possui código aberto (*open source*) atuando com HTML, CSS e JS. Em sua implementação faz-se uso de *plugins* JQuery, mencionado acima, além de fornecer a tecnologia Sass que simplifica a utilização do CSS incorporado no Bootstrap através de variáveis.

## 2.7 JSTL

O JSTL é o siglônimo para Java Standard Tag Library, e é parte da plataforma Java Enterprise Edition, sendo que suas *tags* são utilizadas para substituir expressões e scripts no JSP composta por uma sintaxe específica. Para a utilização do JSTL é necessário ter em mente conceitos do HTML, Java e JSP, sendo que as Java Server Pages tem suporte para sua utilização (BAYERN, 2002).

A biblioteca provê várias *tags* customizadas em forma de um conjunto padrão, com diferentes propósitos de interação, como por exemplo acesso à URLs, formatação de valores, acesso à dados do sistema de banco de dados, manipulação de documentos XML e controlando o fluxo de dados na página JSP. Sendo assim, é possível desenvolver páginas Web dinâmicas e funcionais.

Com base em grupos de *tags*, disponíveis por meio de diferentes URIs específicas para cada finalidade, o JSTL pode ser utilizado sem ter que referenciar toda a biblioteca na página em questão. Por sua simplicidade, o desenvolvimento e implementação das páginas JSP é facilitada, economizando tempo e código em Java escrito na própria página.

A *Expression Language* (EL) da JSTL é uma linguagem de programação usada para fins específicos e é utilizada na biblioteca para inserir as expressões programáticas na página JSP. Quando uma expressão é adicionada à página, primeiramente esta é avaliada pela EL, que é responsável por seu controle literal, sendo representadas pelo símbolo \$, disponíveis por padrão na biblioteca em grupos separados:

- **XML Tag Library:** Utilizada para o acesso a elementos XML.
- **Formatting/Internationalization Tag Library:** Utilizada para a formatação de dados com base na localidade e análise.
- **Database Tag Library:** *Tags* com finalidade em acessar a base de dados.
- **Core Tag Library:** Grupo de *tags* padrões do JSTL, para executar *loops*, escrever expressões e tratar entrada e saída de dados.

## 2.8 Java

Java é uma linguagem de programação e uma plataforma lançada inicialmente em 1995 pela Sun Microsystems e atualmente pertencente a Oracle. Um ambiente Java executa em uma máquina virtual conhecida como *Java Virtual Machine (JVM)*. A JVM é responsável por pegar todos os *byte codes* gerados e transformar para linguagem de máquina possibilitando funcionar em diversos sistemas operacionais (SO). A tecnologia Java permite desenvolver aplicações tanto para desktop, quanto para dispositivos móveis, web e outros eletrônicos.

Para preparar um ambiente de produção, onde os desenvolvedores criam algoritmos, é necessário a instalação do *Java Development Kit (JDK)*, responsável por gerar arquivos *bytecode.class*. Já na execução do programa a JVM lê esses *bytecodes* e as bibliotecas padrões do Java que estão na *Java Runtime Environment (JRE)*. Tanto a JVM, quanto o JDK utilizam o JRE para terem acesso às bibliotecas padrões do Java. Um algoritmo Java compilado e pronto para ser utilizado por ter extensões dos seguintes tipos: **.jar**, **.war** e **.ear** (ARNOLD et al., 2000).

Java é uma linguagem orientada a objetos, compostos por atributos e métodos. As classes Java (Arquivo.class) são separadas por pacotes, fornecendo graus de organização. A orientação a objetos oferece vários recursos que facilitam a vida dos desenvolvedores, visando a reutilização, "simplicidade", códigos pequenos, e legíveis.

Existem vantagens ao escolher esta linguagem para utilização:

- **Simplicidade:** Linguagem parecida com C++, porém, muito mais fácil por não conter sobrecarga de operadores, structs, unions, ponteiros, herança múltipla, diretivas de processamento, além de ter a memória gerenciada pela própria JVM com a ajuda do *garbage collection* que desaloca regiões não mais utilizadas.
- **Processamento distribuído:** Possibilidade de realizar acesso remoto chamando funções *sockets* e protocolos da Internet (Http(s), FTP, Telnet, etc.) possibilitando a criação de aplicativos com arquitetura cliente-servidor.
- **Multithread:** Grande parte dos sistemas operacionais e processadores oferecem suporte a multitarefas que possibilita a execução de diversas tarefas ao mesmo tempo. Cada tarefa

pode ser preemptiva ou não (nível de bloqueio), com ou sem multiprocessamento. O Java possibilita a manipulação desses recursos, criando e gerenciado o fluxo dos processamentos das threads.

- **Tratamento de excessões:** A fim de evitar o travamento da aplicação por completo, o Java disponibiliza meios para trata-las, buscando processar um erro inesperado ou já esperado.

Por ter um grupo de pessoas de diferentes empresas o Java possui um comitê que define os futuros passos da linguagem da melhor forma possível. Esse comitê descentraliza o controle das mãos de uma única empresa aumentando a confiabilidade da linguagem em ambientes críticos e gerencia as especificações Java, normas que definem novas estruturas e recursos. Uma proposta de especificação é conhecida como JSR, seguidas por um número (JSR 220 - Enterprise Java Beans).

Existem alguns *frameworks* criados sobre o Java que busca solucionar algum problema recorrente. Um exemplo deles é o Hibernate, criado pela comunidade, largamente adotado por empresas e que posterior deu origem a especificação JPA. Portanto, Hibernate implementa especificação JPA. No caso do Hibernate seu papel é realizar mapeamento de objetos para banco de dados relacionais. Outro *framework* bastante utilizado é o Enterprise Java Beans (EJB) que adiciona, gerenciamento de estado por usuário, ciclo de vida aos objetos, *annotations*, injeção de dependências, manipulação de estado de objetos (*stateless ou statefull*), controle de concorrência, gerenciamento de memória, e alguns outros recursos.

## 2.9 MySQL e MariaDB

### 2.9.1 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional de código aberto baseado no Oracle Database e na *Structure Query Language* (SQL), e executa em praticamente todas as plataformas compondo o Windows, UNIX e Linux. É mais associado à aplicações Web e publicações *on-line* e também utilizado em mais uma ampla gama de aplicativos. O processamento de dados é no estilo ACID com atomicidade, consistência, isolamento e durabilidade para *transactions*.

O sistema de gerenciamento de banco de dados MySQL é um importante componente de uma pilha corporativa de código *Open Source* chamada LAMP. A plataforma tem o foco do desenvolvimento Web executado no sistema operacional Linux, envolvendo o Apache Tomcat como servidor, o MySQL como sistema de gerenciamento de banco de dados, e o PHP, Perl ou Python como linguagem de *script* orientado a objetos (WILLIAMS; LANE, 2004).

O sistema de gerenciamento de banco de dados MySQL foi originalmente criado pela empresa sueca MySQL AB, na década de 1980. Em 2008, a empresa americana Sun Microsystems adquiriu o SGDB e o administrou até 2010, quando foi comprada pela Oracle, organização

proprietária do SGDB até o momento. Para desenvolvedores sua utilização é gratuita sob a utilização da GNU *General Public License* (GPL), e para Organizações é necessário obter uma licença comercial da Oracle.

O MySQL é utilizado por alguns dos gigantes da internet como o Facebook, Twitter e Youtube, além das várias aplicações corporativas existentes. O SGDB é baseado no modelo cliente-servidor, com arquitetura composta por um núcleo (cliente) que lida com os comandos e instruções do banco de dados, e o servidor disponibilizado separadamente para a utilização em ambiente de rede, além de, uma biblioteca que pode ser vinculada a aplicativos separados.

Um dos principais recursos do MySQL é o armazenamento de dados por vários mecanismos de armazenagem, como o InnoDB, CSV e NDB e MyISAM. O SGDB é capaz de particionar tabelas e replicar dados com o intuito de acelerar o acesso aos dados e melhorar o desempenho, além de não ser necessário aprender novos comandos para se utilizar, pois são os mesmos do SQL padrão.

Na parte de segurança de dados, o MySQL gerencia com base em privilégios de acesso e um sistema de senha criptografada, permitindo uma verificação baseada em *host*. O acesso ao SGDB pode ser feito através de inúmeros protocolos, incluindo o TCP/IP em qualquer plataforma, além de suportar vários clientes e utilitários como programas de linha de comando e gerenciadores como o MySQL Workbench.

### 2.9.2 MariaDB

O MariaDB é uma ramificação do MySQL com código totalmente *Open source* para todos os fins práticos, sendo um substituto compatível com a tecnologia do SGDB pai. Foi criado em 2009 após a Oracle adquirir o MySQL, pelos mesmos desenvolvedores que conceberam o banco de dados original.

Deste modo, suas diretivas são baseadas no SQL e também suporta o gerenciamento de dados no modelo ACID, definido pela atomicidade, consistência, isolamento e durabilidade. Entre os principais recursos do MariaDB estão a utilização de APIs JSON, replicação de dados em paralelo e alguns outros recursos.

O SGDB MariaDB é fornecido como software de código aberto sob a segunda versão da GNU GPL, assim como seu mecanismo MariaDB ColumnStore, utilizado em aplicações de Big Data. Foi desenvolvido também uma implementação do MariaDB intitulada como MariaDB Galera Cluster baseada no Linux, que fornece uma opção de clusterização *multi-master* para usuários do SGDB (BARTHLOMEW, 2014).

Como em outras tecnologias RDBMS com código *Open source*, como por exemplo o PostgreSQL e Firebird, o MariaDB é uma alternativa de baixo custo comparado ao MySQL da Oracle. Cada vez mais o SGDB está sendo utilizado em aplicações Web, além da conquista de adeptos entre usuários de outros componentes que realizaram várias combinações como de

outros *softwares* de código aberto, incluindo a estrutura do OpenStack, que tem como objetivo gerenciar grandes conjuntos de recursos computacionais. Empresas como a Google, Facebook, Wikipedia e o DBS Bank utilizam o MariaDB como SGDB.

## 2.10 Jelastic

O Jelastic é uma plataforma como serviço (PaaS - *Plataform as a service*) *Multi-Cloud* (Múltiplas Nuvens) direcionada a ISVs (*Idenpendent Software Vendors*), provedores de serviço, empresas de telecomunicação ou qualquer razão que possui como objetivo aumentar a velocidade do desenvolvimento, diminuir gastos com infraestrutura da Tecnologia da Informação, otimizar a atividade e aumentar a segurança do negócio em questão (JELASTIC, 2011).

A plataforma oferece a automatização da construção, criação, dimensionamento e armazenamento, e escalonamento em *Cluster*, mantendo também atualizações constantes dos aplicativos nativos e tradicionais da nuvem. O custo é baseado na utilização (*pay-as-you-use*), sendo mensurado através de *Cloudlets*, sendo que cada *Cloudlet* equivale a 400MHz de CPU e 128MiB de memória RAM, podendo ser escalonados com base na utilização e no limite máximo definido.

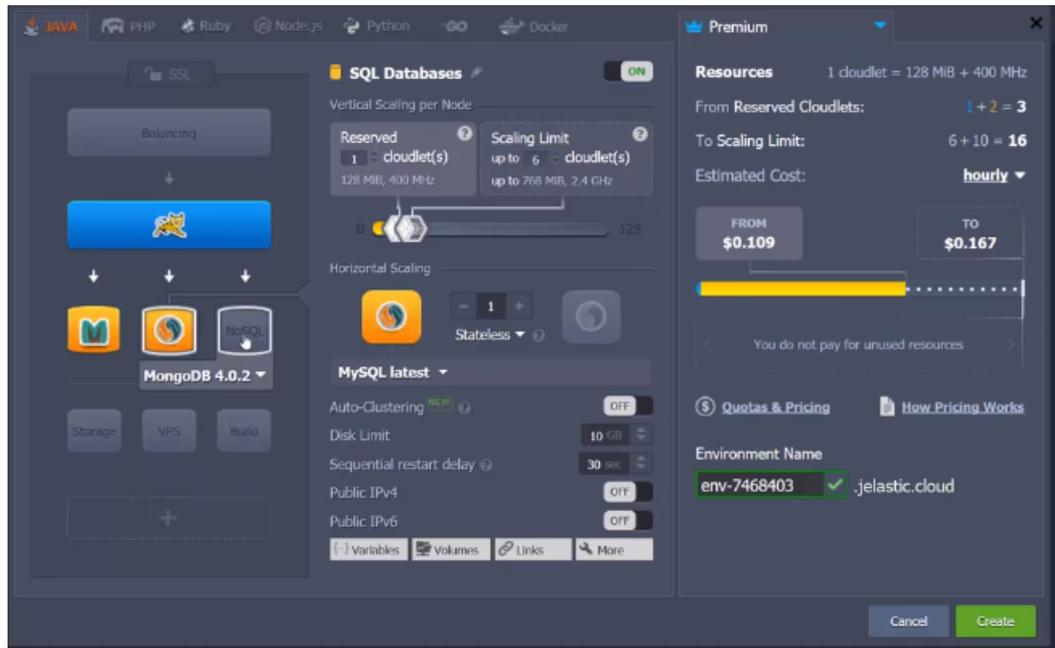
Com as ferramentas de DevOps disponibilizadas pelo serviço, é possível construir projetos que são trazidos diretamente do GitHub e automaticamente realizar o *deploy* no servidor previamente configurado na plataforma. Como todos esses recursos são integrados em um mesmo ambiente torna-se viável a entrega contínua, o gerenciamento do ciclo-de-vida e a integração.

Para colocar a aplicação em produção, é necessário fazer a configuração e estabelecer a topologia do ambiente, definindo assim toda a infraestrutura necessária para que o sistema, juntamente com seus componentes e banco de dados ofereçam escalabilidade, velocidade ao acesso à informação e gestão de dados. A imagem do painel de configuração da topologia de infraestrutura é apresentada na figura 1.

No painel é possível configurar quantos *Cloudlets* cada nó poderá operar, podendo estabelecer um limite máximo definido para ser escalado. Pode-se adicionar um *Load Balancer*, que tem como objetivo balancear a carga e quantidade de acesso em dois ou mais servidores replicados, adicionar o servidor que o sistema irá rodar em produção, definir o banco de dados, o ambiente para o *build* e algum outro nó dependendo da necessidade da aplicação.

Por sua facilidade, operabilidade, disponibilidade, escalonamento e entrega contínua, a Jelastic Cloud torna-se uma ótima opção como PaaS oferecendo disponibilidade com perfis, público, privado, híbrido e multi-nuvem em cerca de 60 *data centers* em todo o mundo. A plataforma suporta diversos ambientes de produção como, Java, PHP, Ruby, Node.js, Python, .NET, Go, bem como contêineres Docker separados ou orquestrados por Kubernetes. Há outros provedores de nuvem tais como, Google Cloud Plataform (GCP), Microsoft Azure e Amazon.

Figura 1 – Painel de configuração da topologia de infraestrutura



Fonte: Jelastic (2011)

## 2.11 Git e GitHub

Git é um sistema de controle de versão (VCS) mais utilizado atualmente (GIT, 2017). É um projeto *open source* idealizado por Linus Torvalds, o mesmo criador do kernel linux. Grande números de projetos dependem do Git, sendo eles comerciais e/ou *open source*. Para se ter ideia, até a Microsoft migrou todo o código fonte do Windows para o Git, mesmo tendo um sistema próprio de controle de versão, o *Team Foundation Server* (TFS).

Um sistema VCS possibilita que vários desenvolvedores, trabalhem em um único projeto, único código fonte e até mesmo em um único arquivo ao mesmo tempo. No entanto não fornece uma edição em tempo real como no *Google Docs*, pois poderia gerar transtornos nas várias compilações que os desenvolvedores realizam no dia a dia.

Especialmente no Git cada desenvolvedor realiza uma cópia do projeto original, dando origem ao que chamamos de ramo (*branch*). Nenhuma nova funcionalidade (*feature*) ou conserto de erros do sistema (*hotfix*) podem ser implementadas nas *branches* principais. Os erros são comumente chamados de *bugs*. Essas *branches* principais usualmente são denominadas de *master* e/ou *develop*, dependendo dos processos e formas de trabalhar com o Git.

Após ter um código escrito em uma das *branches* os desenvolvedores realizam um *commit*, que é registrar um ponto de atualização para ser mesclado com uma *branch* de nível superior. Esse processo de mesclagem é conhecido como *merge* e a chamada para tal ação é conhecida como *push*, ou empurrar no português.

O GitHub é um repositório online no qual possibilita sincronizar os projetos Git, tornando-os acessíveis a outros desenvolvedores/usuários na internet. Esses repositórios podem ser públicos ou privados, além de permitirem configurar níveis de autorização dentro do GitHub aumentando a segurança no fluxo de desenvolvimento. Existem algumas alternativas ao GitHub disponíveis no mercado, tais como, GitLab, GitBucket, dentre outras.

## 2.12 Apache Tomcat e TomEE

### 2.12.1 Apache Tomcat

O Apache TomCat é um software de código aberto com a implementação do Java Servlet, Java Server Pages, Java Expression Language e o Java WebSocket. Tais tecnologias são desenvolvidas sob o Java Community Process (JCP), que é o responsável pelo desenvolvimento contínuo do Java, garantindo um padrão de estabilidade e compatibilidade da plataforma (TOMCAT, 2013).

Deste modo, o Apache Tomcat fornece por padrão um conector HTTP na porta 8080 para ser utilizado como um servidor HTTP. Ele também trás consigo uma aplicação chamada Tomcat Manager, fornecendo ferramentas e funcionalidades básicas com a finalidade de gerenciar todas as aplicações Web em execução no servidor.

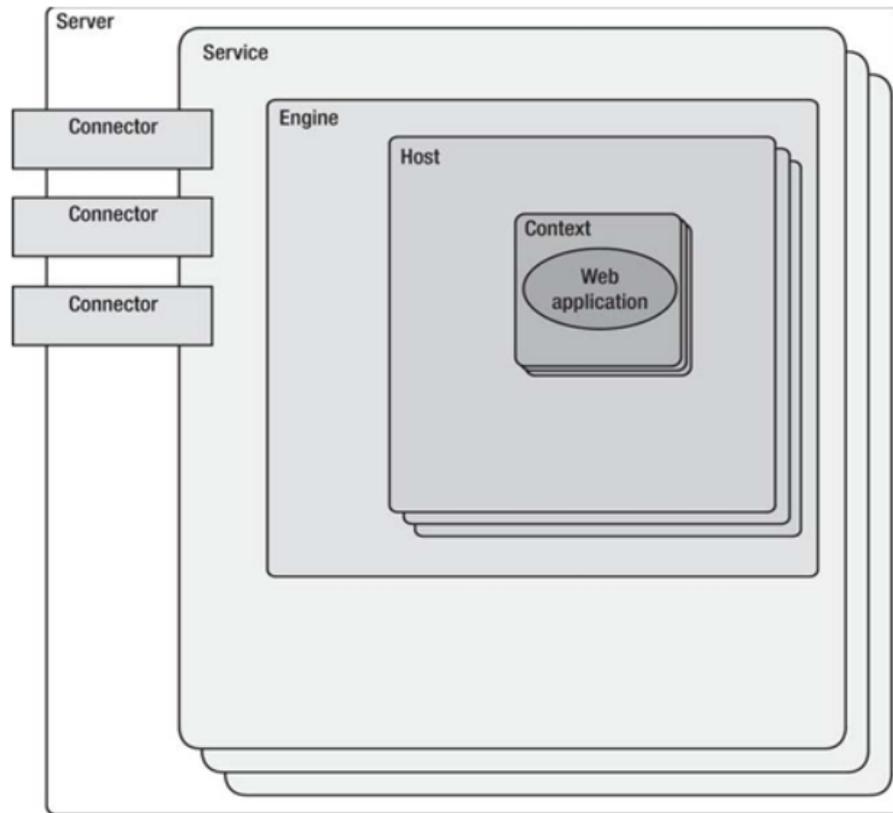
Na parte de segurança da informação, o Apache Tomcat disponibiliza métodos gerenciados pelo *container* protegendo os recursos contidos nele. Junto disso, é disponibilizado a utilização dos Realms (providos pelo Tomcat ou outros servidores de aplicações Java), que tem como objetivo gerenciar os acessos aos recursos do *container*, com base em papéis e com criação de usuários, podendo guardar tais informações no banco de dados utilizado o Java Database Connectivity (JDBC), ou em um arquivo, gravado na própria memória do servidor de execução do Apache Tomcat.

Existe também o Tomcat **Valves**, uma tecnologia agregada no Apache Tomcat 4, que foi mantida até na versão mais atual do servidor, permitindo a associação de uma instância de uma classe Java com um *container*. Uma classe Valve implementada funciona como uma preprocessora das requisições ao servidor Tomcat, mas tal tecnologia não é considerada para *containers* JSP/Servlet.

Na arquitetura, o Apache Tomcat é um grupo de aplicações de container, definida em uma hierarquia de componentes designados para efetuar tarefas diferentes. O principal componente do Apache Tomcat é o Catalina Servlet Engine, que é uma implementação do Servlet API 3.0, sendo a última especificação da SUN. A arquitetura do Apache Tomcat é apresentada pela figura 2.

O Server representa o Catalina Servlet Engine que é o nível mais alto de uma instancia do Apache Tomcat, o Service contem os conectores que compartilham a mesma *engine*, o Connector define as classes de resposta para as requisições e seu gerenciamento. A Engine gerencia todas

Figura 2 – Principais componentes da arquitetura do Tomcat



Fonte: [DevMedia \(2013\)](#)

as requisições recebidas pelos conectores, o Host define os hosts contidos em cada instancia do Catalina e o Context representa uma aplicação Web individual em execução em um Host.

### 2.12.2 TomEE

O Apache TomEE é um servidor Web para aplicações Java que combina vários projetos corporativos desenvolvidos com a tecnologia, sendo alguns deles o Apache OpenEJB, Apache OpenJPA, Apache OpemWebBeans, o Apache MyFaces dentre outros. O Apache TomEE tem como objetivo oferecer aos desenvolvedores uma série de recursos para suas aplicações contidos em um *container* simples e leve.

Por incorporar várias *features* do JavaEE, além de outras tecnologias como CDI e EJB, o TomEE resultou em um servidor Web Profile com total compatibilidade, mantendo o Tomcat como mestre. Coisas antes difíceis de se desenvolver e integrar com o Apache Tomcat, tornaram-se simples, pois os conceitos do JavaEE, EJB e Tomcat Realms são encontrados no mesmo *container* de aplicação.

Quanto a implantação dos artefatos, o TomEE trabalha com os mesmos conceitos do Apache Tomcat. Quando o pacote EAR ou WAR é copiado para a pasta `webapps/TomEE`,

automaticamente o TomEE quando em execução, realiza automaticamente o *deploy* da aplicação no servidor, criando um diretório com o a mesma nomenclatura sem as extensões antes aqui mencionadas.

A facilidade da implantação está diretamente relacionada à definição dos recursos dentro de uma aplicação, porque acaba sendo mais fácil realizá-la em um servidor que conhece todas as dependências, como o TomEE, ao invés de um que é necessário realizá-las externamente. Alguns projetos trazem a solução do problema com a combinação de métodos, por exemplo a extração de uma URL JDBC através de um pool de conexão com o banco de dados.

## 3 Engenharia de software

Engenharia de software é uma abordagem sistemática, disciplinada e quantificável aplicadas no processo de desenvolvimento, operação e manutenção do software (PRESSMAN, 2005).

Com a grande evolução da tecnologia da informação o processo de desenvolver um software também evolui e continua evoluindo. Tal tecnologia abrange computadores, *tablets*, *smartphones*, redes, computação em nuvem, entre várias outras. Tal fato reflete diretamente na complexidade de projetar *software*. Com isso foi notada a necessidade de expressar funcionalidades e técnicas em modelos que facilita a compreensão humana. Pensando nos processos de desenvolvimento várias análises são feitas com os seguintes objetivos:

- Avaliar a real necessidade do usuário;
- Avaliar a viabilidade do sistema;
- Atribuir funções e papéis aos elementos do sistema (pessoas, softwares, hardwares);
- Estimar custos, prazos;
- Definir o sistema em si;

### 3.1 Requisitos

Requisitos são funcionalidades e tarefas que se espera de um sistema, tais como, descrições dos serviços que um sistema oferece e suas restrições operacionais (SOMMERVILLE, 2011). Os requisitos podem ser funcionais e não funcionais.

Os requisitos funcionais abrange quais os serviços o sistema deve fornecer, além de ações e reações de acordo com a situação. Um exemplo seria o processamento a ser realizado de acordo com a inserção de dados pelo usuário.

Os requisitos não-funcionais não possui relação direta com as funcionalidades do sistema, mas sim com propriedades necessárias para sua execução. Estas propriedades podem ser: segurança, desempenho, escalabilidade, disponibilidade, criticidade, tolerância a falhas e outras.

### 3.2 Diagrama entidade relacionamento (DER)

O diagrama entidade relacionamento é responsável por demonstrar os dados que são realmente armazenados no banco, proporcionando um entendimento humano "facilitado". Na maioria das vezes é necessário ferramentas computacionais para conseguir entender tais diagramas devido a sua complexidade nos relacionamentos entre as tabelas. Essas ferramentas possibilitam filtrar relacionamentos por tabela, nome de campos, cardinalidades, etc. Podemos citar: MySQL Workbench, DB Visualizar, DB Navigator, SQL Server Management Studio.

### 3.3 Linguagem Unificada de Modelagem (UML)

A UML (*Unified Modeling Language*), é uma linguagem de modelagem padronizada que consiste em um conjunto integrado de diagramas, criados para ajudar desenvolvedores de sistemas e softwares a especificar, visualizar, construir e documentar os artefatos de sistemas de software, bem como modelagem de negócios e outros sistemas que não sejam de software. A UML representa uma coleção das melhores práticas de engenharia que se mostraram bem-sucedidas na modelagem de sistemas grandes e complexos.

### 3.4 Diagrama de pacotes

Um pacote agrupa elementos e/ou classes, buscando manter a estrutura do projeto organizada por funcionalidades. Significa que um pacote pode conter mais de um arquivo de diferentes tipos, porém, todos realizando tarefas com alguma similaridade, incluindo outros pacotes, criando hierarquias.

### 3.5 Métodos ágeis

A metodologia ágil é uma abordagem de desenvolvimento de software focada nas pessoas e resultados que se adapta a mudanças rapidamente. Ela se concentra no planejamento adaptativo, na auto-organização e no curto prazo de entrega. É flexível, rápida e visa melhorias contínuas na qualidade. Scrum e eXtreme Programming são algumas das metodologias ágeis do desenvolvimento de software.

#### 3.5.1 Scrum

Scrum é uma abordagem ágil para gerenciamento e planejamento de projetos de software. Os projetos são divididos em ciclos denominados de Sprints. Uma sprint representa o tempo onde determinado conjunto de atividades devem ser realizadas. O prática das sprints tornam o Scrum uma metodologia ágil iterativa e incremental. Ao final de cada Sprint deve ter parte do produto funcional finalizado naquela iteração. A próxima sprint irá incrementar novas funcionalidades.

Os recursos a serem implementados no projeto são salvos em uma lista chamada *Product Backlog*. No início de cada sprint acontece a etapa de planejamento, onde o dono do produto *Product Owner* prioriza as tarefas, e a equipe seleciona as atividades que serão executadas naquela sprint. As tarefas atribuídas na sprint são transferidas da lista de pendências de produtos *Product backlog* para a lista da *Sprint Backlog*.

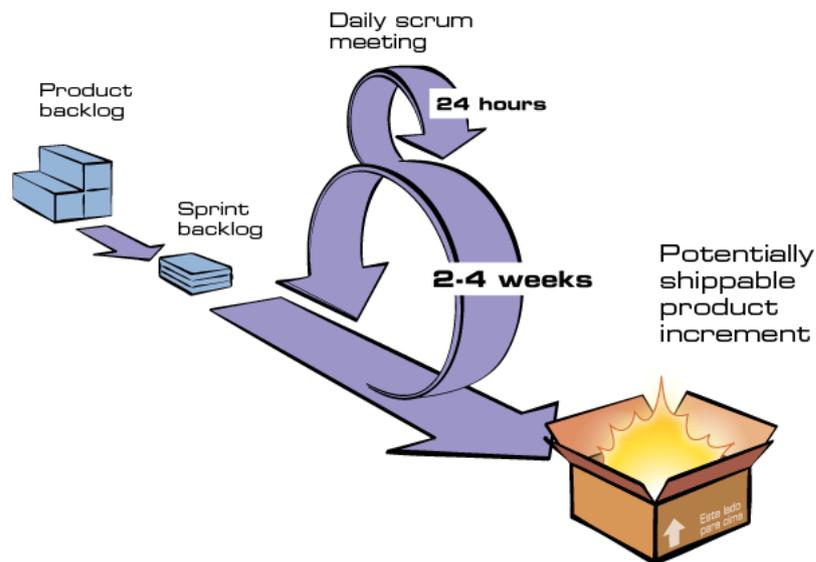
Todos os dias do sprint a equipe realiza uma breve reunião conhecida como *Daily Scrum* onde cada participante responde três perguntas:

- O que foi feito?

- O que será feito?
- Quais impedimentos atrapalham a execução da atividade?

No final da Sprint a equipe apresenta os recursos implementados, este fato é a *Sprint Review* e então a parte da avaliação e revisão determina o fim da sprint (Retrospectiva da Sprint). A partir deste ponto pode-se iniciar um novo ciclo, uma nova sprint com todas as etapas das anteriores (SCRUM, 2014). Observe a figura 3.

Figura 3 – Ciclo Scrum



Fonte: SCRUM (2014)

Segue a lista com as principais terminologias definidas pelo Scrum com uma breve explicação:

- **Product Backlog:** funcionalidades de todo o sistema para serem implementadas durante o processo de desenvolvimento. Podem ocorrer alterações durante as Sprints.
- **Release Burndown:** Gráfico 2D com a soma das horas estimadas (eixo y) pelos dias gastos (eixo x). É mostrada a linha ideal de desenvolvimento de acordo com a capacidade do time e a linha atualizada com as horas gastas.
- **Product Owner:** Pessoa responsável por levantar os requisitos do sistema junto ao cliente.
- **Scrum Master:** Pessoa responsável por retirar todos os impedimentos que impossibilitam a realização de qualquer atividade.
- **Scrum Team:** Times de desenvolvedores e arquiteto responsáveis pela implementação e sustentação do produto.

- **Sprint Backlog:** Atividades do *product backlog* alocadas para a sprint atual.
- **Sprint Planning Meeting:** Reunião de planejamento para definir as atividades do *product backlog* que serão alocadas para a atual *sprint backlog* definindo prioridades e tempo gasto.
- **Sprint Retrospective:** Apresentação das funcionalidades entregues na sprint, geralmente junto com o cliente. **Sprint Review Meeting:** Reunião de avaliação da sprint que está sendo finalizada. **Sprint Review Meeting:** Reunião de avaliação da sprint que está sendo finalizada.

## 3.6 Arquitetura de software

A arquitetura de software é o processo de conversão de características de software como flexibilidade, escalabilidade, viabilidade, reutilização e segurança em uma solução estruturada que atenda às expectativas técnicas e de negócios e mantenha o código organizado visando facilitar a manutenção.

### 3.6.1 Arquitetura Model-View-Controller (MVC)

A ideia do modelo-visão-controle (MVC) é que cada seção do código tenha uma finalidade. Parte do seu código contém os dados do seu aplicativo em si, outra parte faz com que ele fique bonito e a última controla como o aplicativo funciona.

O código do modelo normalmente reflete coisas do mundo real. Esse código pode conter dados brutos ou definir componentes essenciais do software. O código de visão é composto por todas as funções que interagem diretamente com o usuário. Esse é o código que faz com que seu aplicativo tenha uma boa aparência e define como o usuário vê e interage com ele. O código de controle atua como uma ligação entre o modelo e a visão, recebendo a entrada do usuário e decidindo o que fazer.

### 3.6.2 Arquitetura cliente-servidor

A arquitetura cliente e servidor compartilha as tarefas de processamento de dados entre um servidor, normalmente uma estação de trabalho de alta tecnologia, e os clientes, que geralmente são computadores de usuários comuns. Os computadores têm um poder de processamento significativo, portanto, são capazes de pegar dados brutos retornados pelo servidor e formatá-los para saída, ou seja, visualização do usuário (HARRINGTON, 2009).

O tráfego de rede é reduzido a solicitações de manipulação de dados enviadas do computador do usuário para o servidor de banco de dados e os dados brutos retornados como resultado dessa solicitação. O resultado é significativamente menos tráfego de rede e, teoricamente, melhor desempenho dependendo da implementação.

## 3.7 Projeto do software

Refere-se ao ramo de gerenciamento de projetos dedicado ao planejamento, programação, alocação de recursos, execução, rastreamento e entrega de projetos de software.

### 3.7.1 Business Process Model and Notation (BPMN)

O Modelo de Processo de Negócios e Notação (BPMN) se tornou o padrão de fato para os diagramas de processos de negócios. Pode ser usado diretamente pelas partes que projetam, gerenciam e realizam processos de negócios e são precisos o suficiente para permitir que os diagramas BPMN sejam traduzidos em componentes de software. É caracterizado por possuir uma notação de fluxograma fácil de usar, independente de qualquer ambiente de implementação específico.

## 3.8 Casos de teste - Verificação e Validação

Teste de software é um processo para avaliar a funcionalidade de uma aplicação com a intenção de descobrir se o software desenvolvido atende ou não aos requisitos especificados, além de identificar os defeitos para garantir que o produto esteja livre de erros sempre visando a qualidade. Existem inúmeras formas de testar uma aplicação, tais como, testes unitário, integração, interface, performance, stress, usabilidade, e vários outros. O processo de teste possui duas macro diferenciações que são a verificação e a validação.

**Verificação:** O processo de verificação avalia fases de desenvolvimento (não o produto final) específicas para determinar se eles atendem aos requisitos especificados para tal fase.

**Validação:** Avalia durante ou no final do processo de desenvolvimento para determinar se o software atende aos requisitos de negócios especificados. Busca validar se o produto cumpre com as funcionalidades pretendidas quando colocado no ambiente de produção.

No caso do SGCO os testes foram verificados em usabilidade na interface do usuário. Dada determinadas entradas espera-se determinado comportamento. Tais testes são realizados no SGCO tanto pela parte da equipe de desenvolvimento e sustentação quanto pelos usuários finais.

## 3.9 Homologação

Homologação significa aprovar ou confirmar oficialmente. É o processo de certificação ou aprovação de um produto para indicar que ele atende aos padrões de qualidade e requisitos. As certificações de homologação podem ser concedidas por empresas, organizações, cartórios, fóruns, dependendo do contrato.

O SGCO é homologado pela supervisora do contrato que afere o funcionamento e autoriza o ambiente de produção. O sistema encontra-se em versão estável e foi avaliado por

técnicos da área para verificar as funcionalidades antes prometidas e definidas no contrato da licitação.

As principais funcionalidades que foram verificadas respectivamente e homologadas pela equipe técnica pertencente a licitação referente a prefeitura de Araçoiaba da Serra - SP:

1. Cadastramento dos locais/oficinas onde serão realizadas as oficinas;
2. Cadastramento dos alunos e instrutores das unidades onde serão realizadas as oficinas;
3. Sistematização e controle de atividades desenvolvidas pelos instrutores das oficinas;
4. Acesso rápido a informações pela secretaria da assistência social das unidades, território, instrutores, alunos, responsáveis, localização e relatórios gráficos;
5. Sistema de Informática, desenvolvido para utilização via web (Internet);
6. Sistema composto por modulo de segurança que proporcione o acesso as funcionalidades dele através de senha e níveis de acesso, tanto em nível de sistema quanto no banco de dados;
7. O sistema será alimentado diariamente (online), com as informações colhidas no momento da aplicação da aula, pelo responsável previamente cadastrado para esse fim;
8. O Sistema deverá contemplar as seguintes funcionalidades: Cadastro de usuários e níveis de acesso; Cadastro das Modalidades; Cadastro dos Alunos; Cadastro de Oficinas; Cadastro do pesos;
9. O sistema deverá gerar assinatura eletrônica por algoritmo possibilitando sequência de caracteres para autenticação em página oficial da empresa de relatórios de frequências e atividades realizadas;
10. O sistema deverá possibilitar o acesso e inserção de faltas e relatório de aula por *smartphone* na data e hora da realização das oficinas simultaneamente;
11. O Sistema também contará com um módulo de uso exclusivo da Prefeitura do Município de Caieiras para consulta as informações referentes ao desenvolvimento do projeto;
12. Os dados manipulados por meio do sistema deverão ser armazenados em servidor de Banco de Dados de responsabilidade da empresa responsável pela licitação, sendo que ao final do projeto, a empresa poderá a pedido da Prefeitura do Município de Caieiras, fornecer estes dados no formato solicitado.

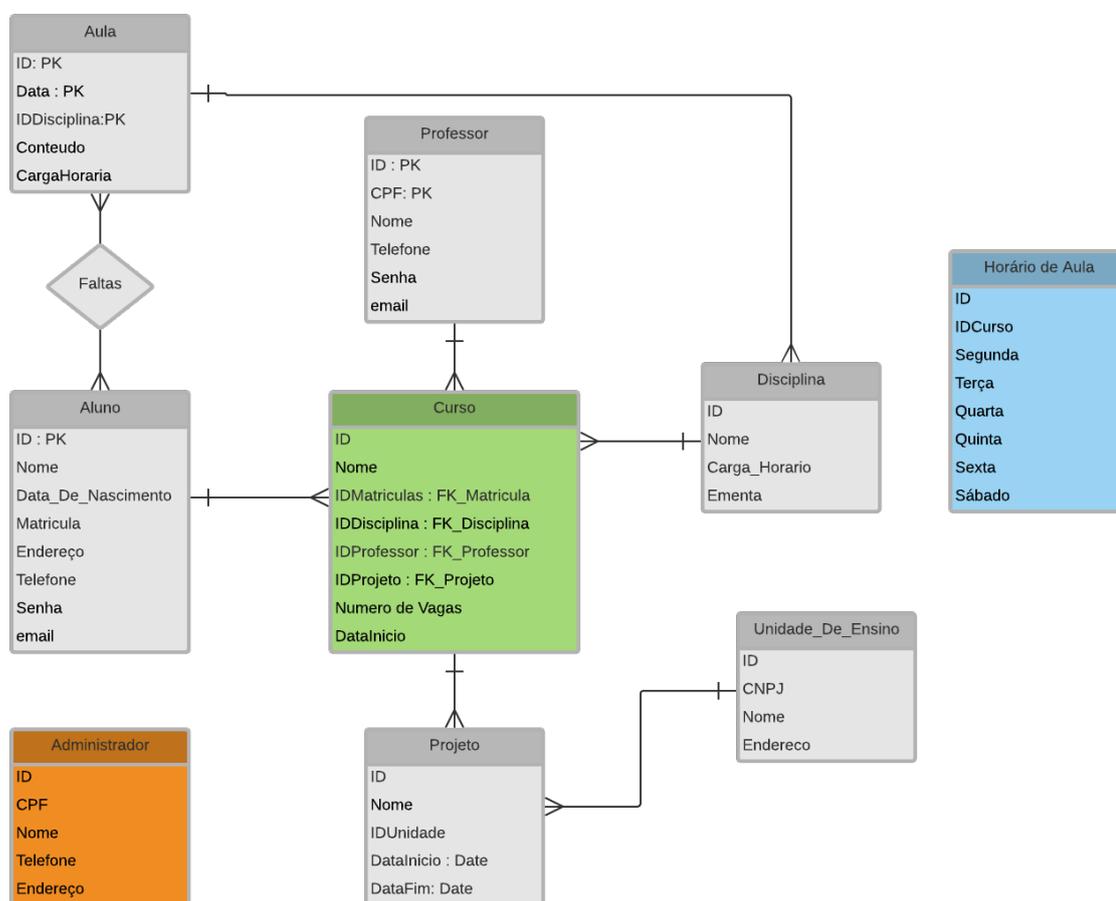
## 4 O SGCO Electa

### 4.1 Projeto de software

Para este projeto o código fonte está hospedado em um repositório no GitHub e com isso conseguimos criar uma linha de desenvolvimento (*pipeline*) automatizada. O Jelastic acessa o repositório no GitHub e carrega todo o projeto para containers Docker automaticamente. Esse processo de disponibilizar recursos é conhecido como *deploy*.

A estrutura do banco de dados inicial foi bem simplista, podendo ser considerada apenas uma macro arquitetura. Tal estrutura pode ser observada na figura 4.

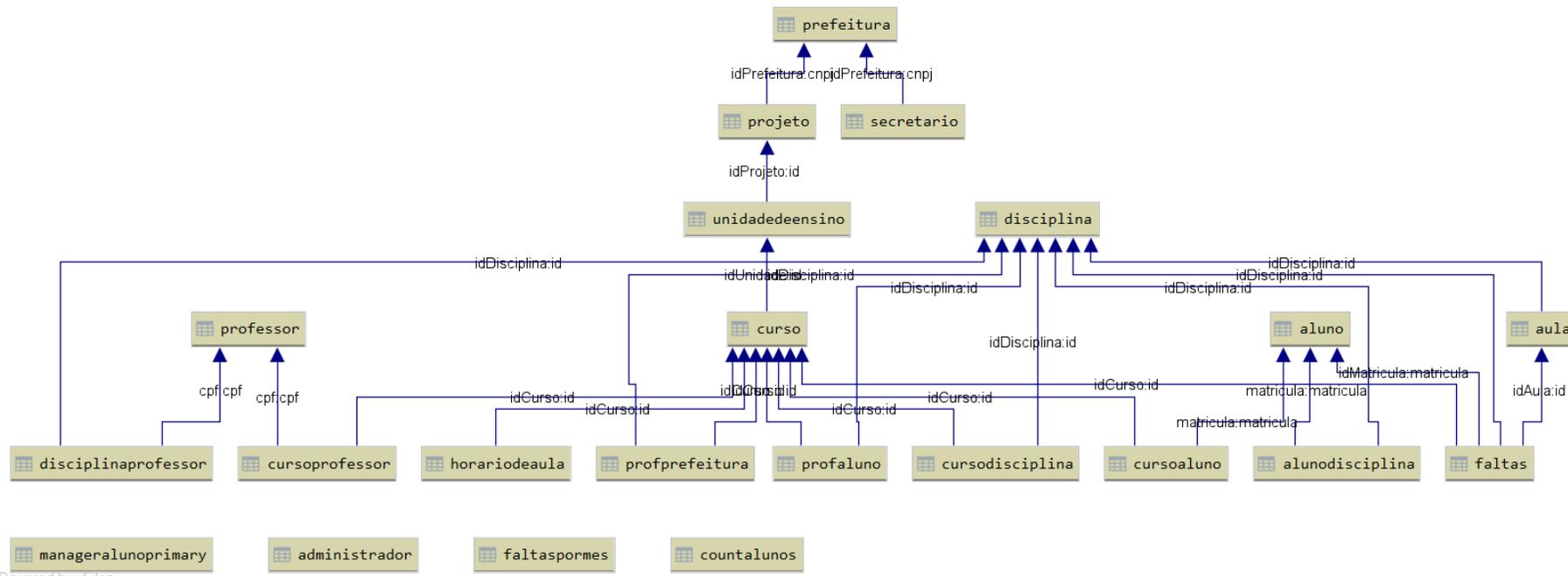
Figura 4 – Diagrama entidade relacionamento inicial



Fonte: Compilação própria

Com as futuras necessidades tal diagrama tomou proporção relativamente grande dificultando seu entendimento em apenas uma imagem. Observe a figura 5.

Figura 5 – Diagrama entidade relacionamento atual



Fonte: Compilação própria

Todo o código do backend da aplicação está separado de acordo com as funcionalidades que cada algoritmo realiza, vide figura 7. A interação entre as classes de modelo (que ligam a estrutura dos dados em Java com o banco de dados) estão organizadas como descrito na figura 6.

Como descrito na seção Referencial Teórico o Scrum possui a lista de backlog de sistema (*Product Backlog*), no caso do SGCO os seguintes requisitos foram levantados.

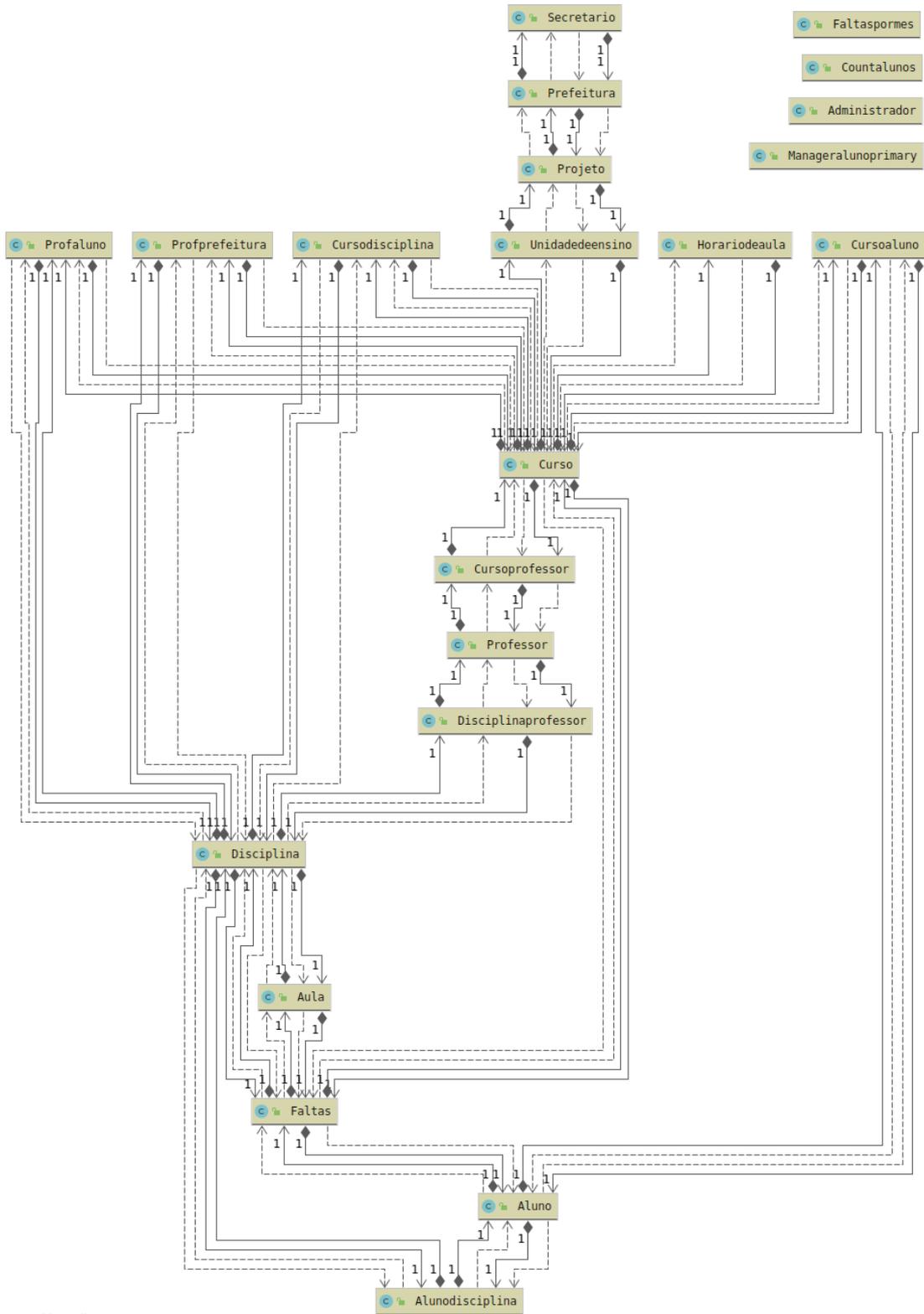
Backlog geral do sistema:

- Modelagem e criação do banco de dados;
- Criação do projeto;
- Criação do repositório git;
- Criação do ambiente de produção;
- Configuração de integração entre git e ambiente de produção;
- Implementar sistema de Login;
- Implementar controle de sessão;
- Implementação do módulo Administrador;
- Implementação do módulo Prefeitura;
- Implementação do módulo do Professor;
- Implementar recuperação de senha;
- Implementar políticas de autenticação e autorização;
- Implementar criação de relatórios das oficinas ministradas;
- Implementar a criação de relatórios da aula;
- Autenticação de documentos gerados pelo sistema;
- Adicionar funcionalidades e usuários com perfil de secretários.

Backlog dos administradores:

- CRUD de novos usuários;
- CRUD de prefeituras/secretários;
- Possui todas as permissões de Prefeitura/Secretário.

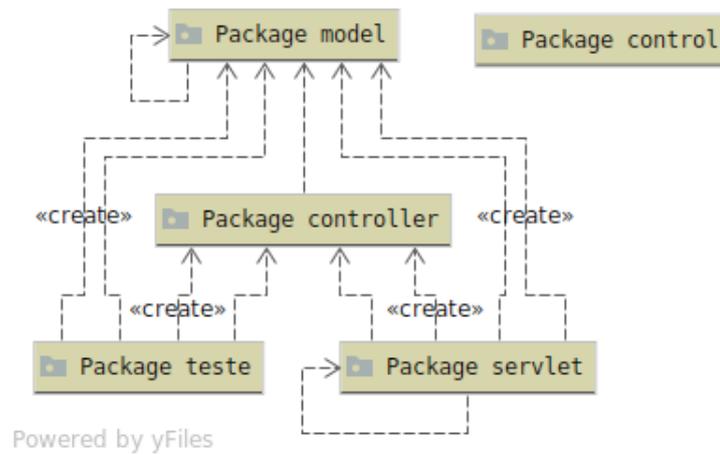
Figura 6 – Diagrama UML



Powered by yFiles

Fonte: Compilação própria

Figura 7 – Diagrama de pacotes



Fonte: Compilação própria

Backlog dos professores:

- CRUD de aulas;
- Lançamento de notas;
- Lançamento de faltas;
- Diário de frequência;
- Acesso ao horário de aula;
- Acesso às aulas ministradas.

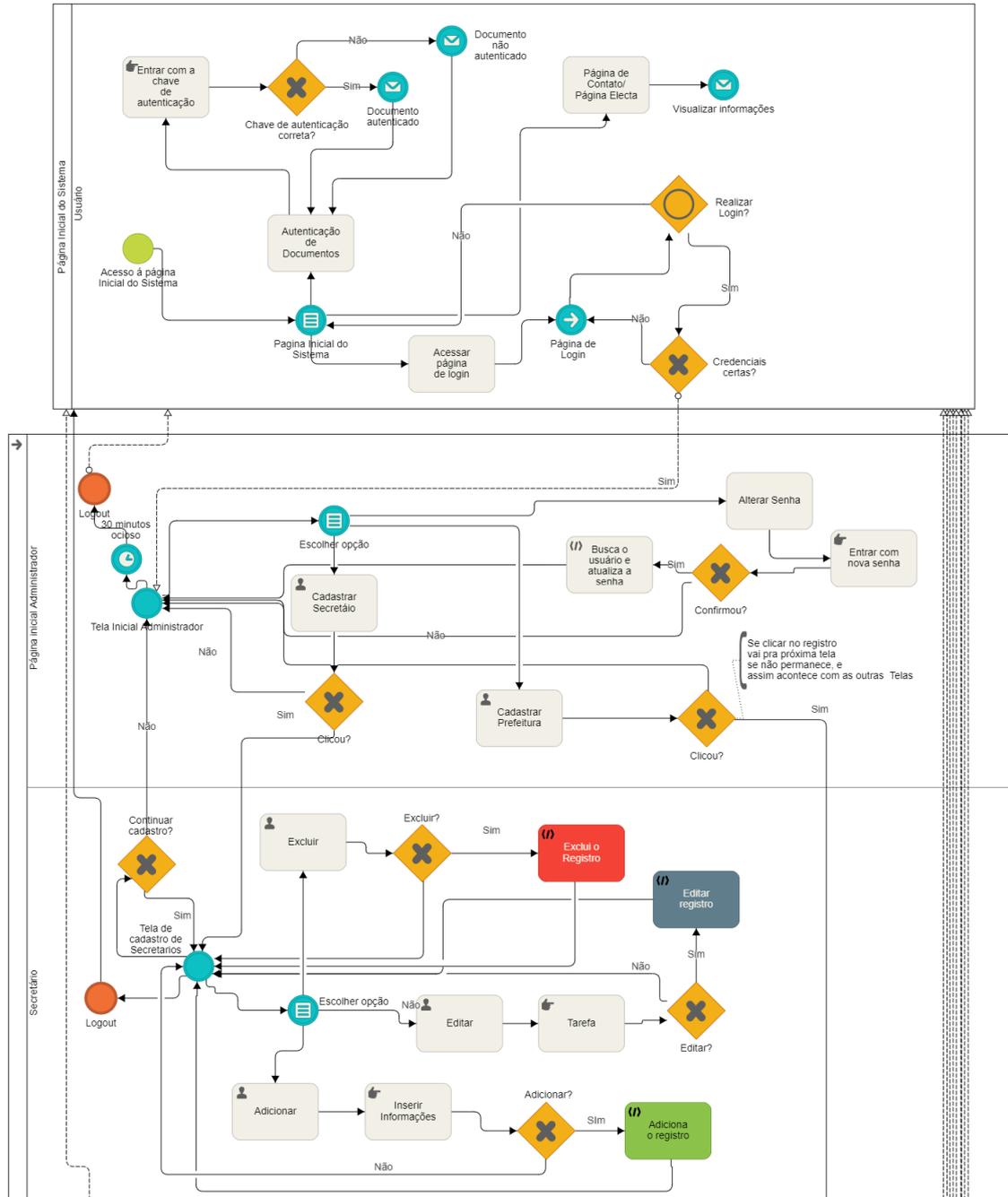
Backlog da prefeitura/secretário

- Visualizar projetos e seus cursos;
- Visualizar relatórios com gráficos;
- CRUD de professores;
- CRUD de alunos;
- CRUD de projetos;
- CRUD de localidades;
- CRUD de modalidades;

- CRUD de oficinas;
- CRUD de horário de aula.

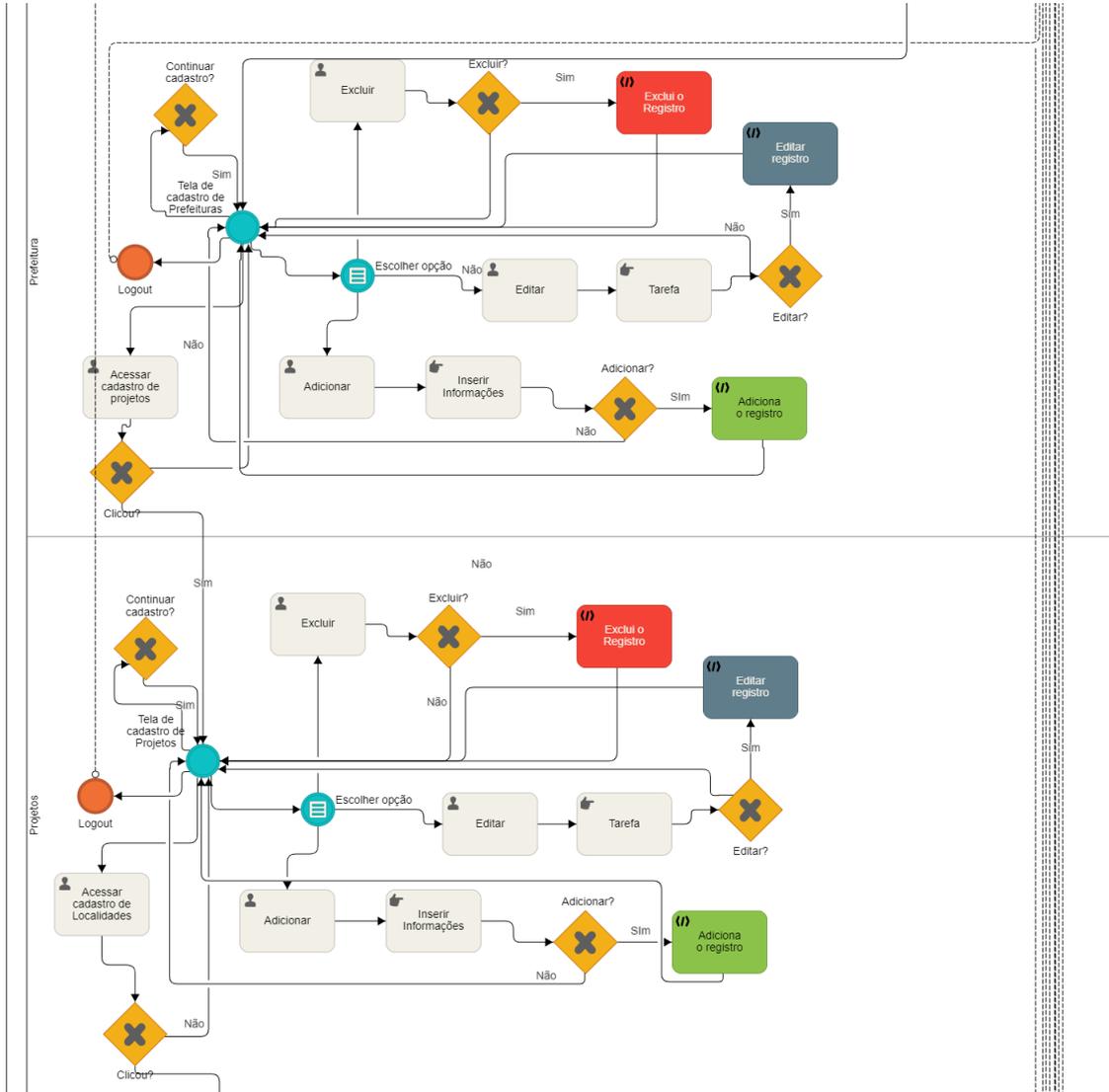
O sistema SGCO possui três tipos de usuários que podem ter acesso, são eles: administrador, prefeitura/secretário, professor. Tais usuários seguem ordem hierárquica de permissão de funcionalidades. As figuras 8, 9, 10, 11 e 12 detalham as regras negociais que um administrador pode realizar. As figuras 13 e 14 detalham as regras negociais de usuários com perfil a nível de prefeitura/secretário. As figuras 15 e 16 detalham as regras negociais de usuários que são professores.

Figura 8 – BPMN - Administrador parte 01



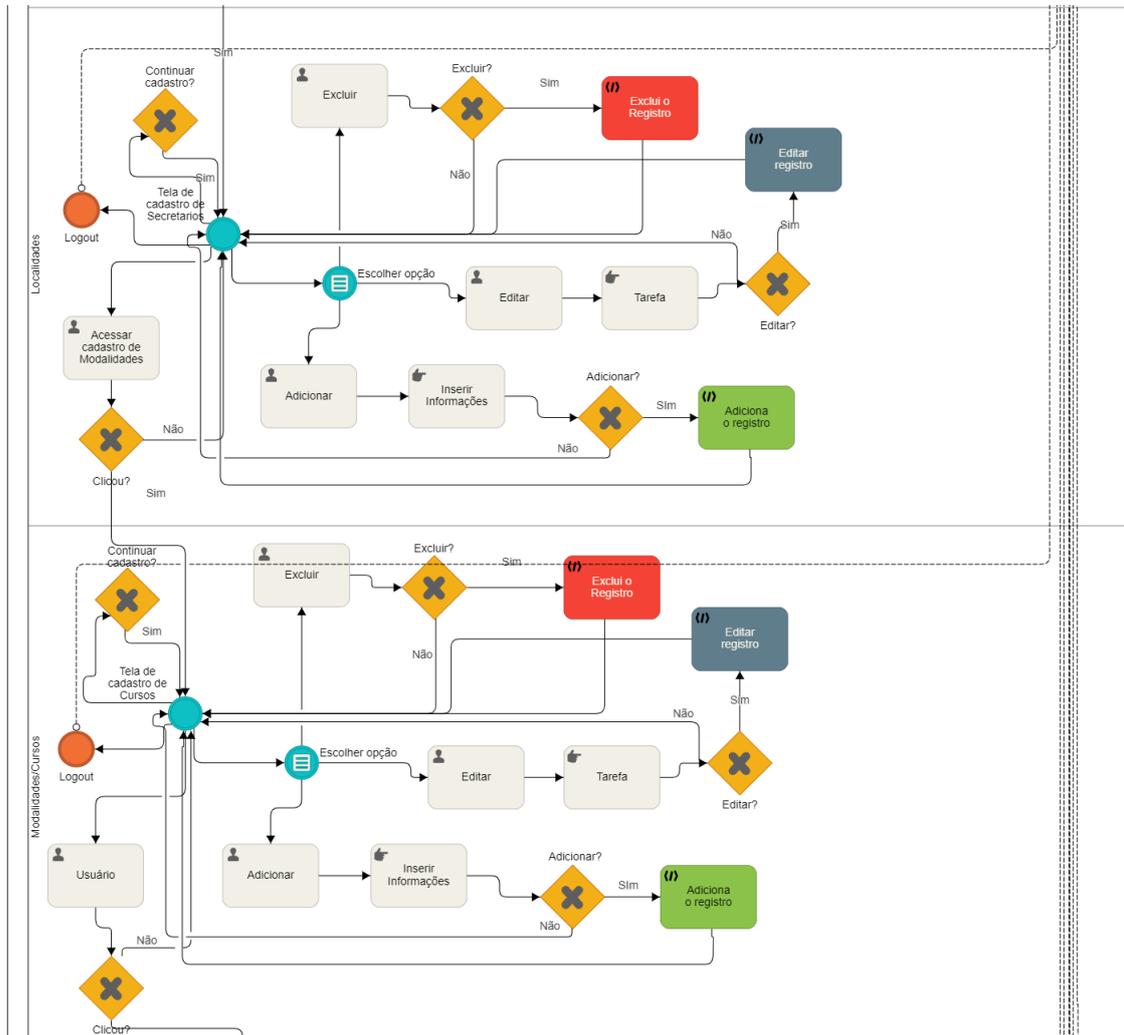
Fonte: Compilação própria

Figura 9 – BPMN - Administrador parte 02



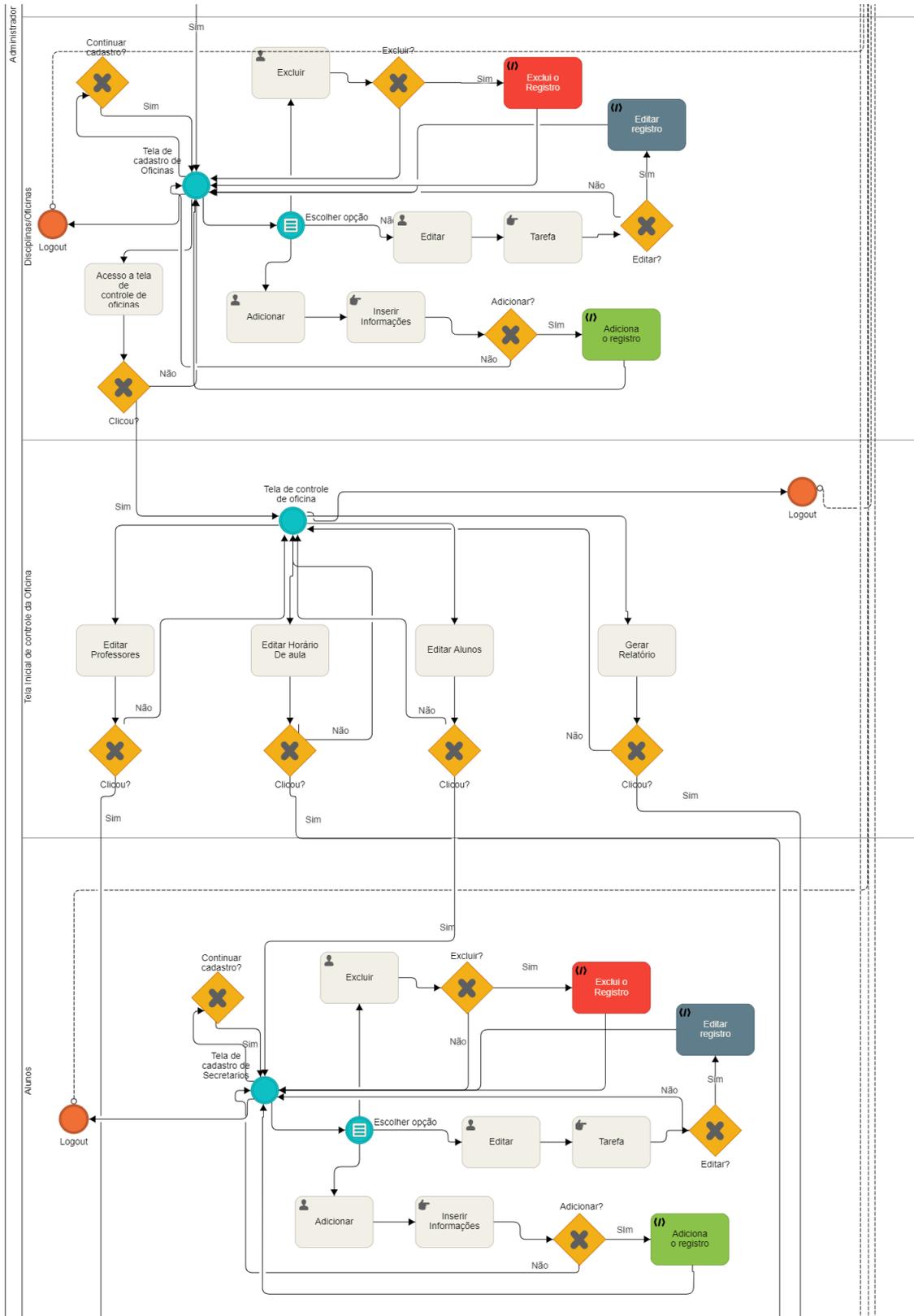
Fonte: Compilação própria

Figura 10 – BPMN - Administrador parte 03



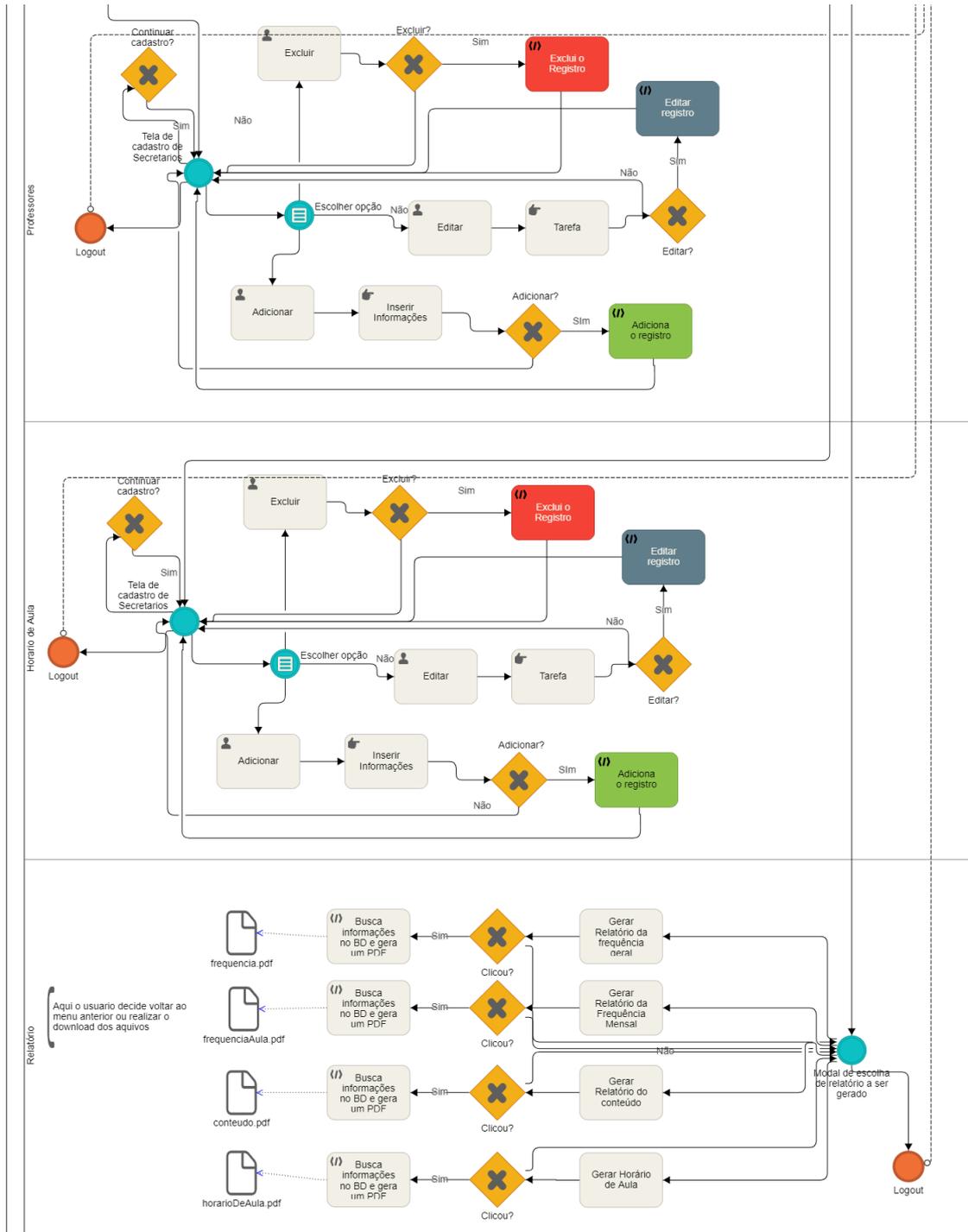
Fonte: Compilação própria

Figura 11 – BPMN - Administrador parte 04



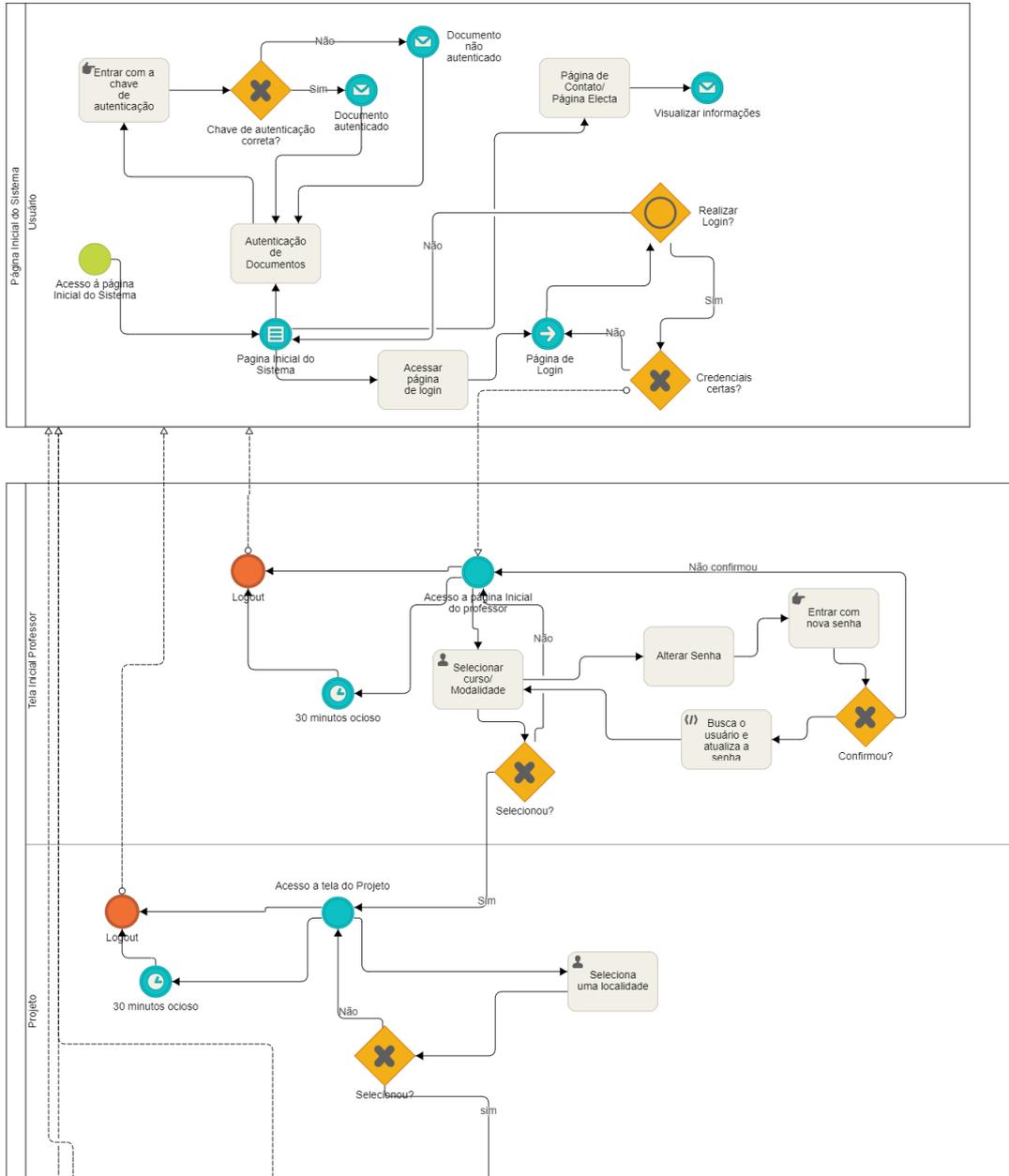
Fonte: Compilação própria

Figura 12 – BPMN - Administrador parte 05



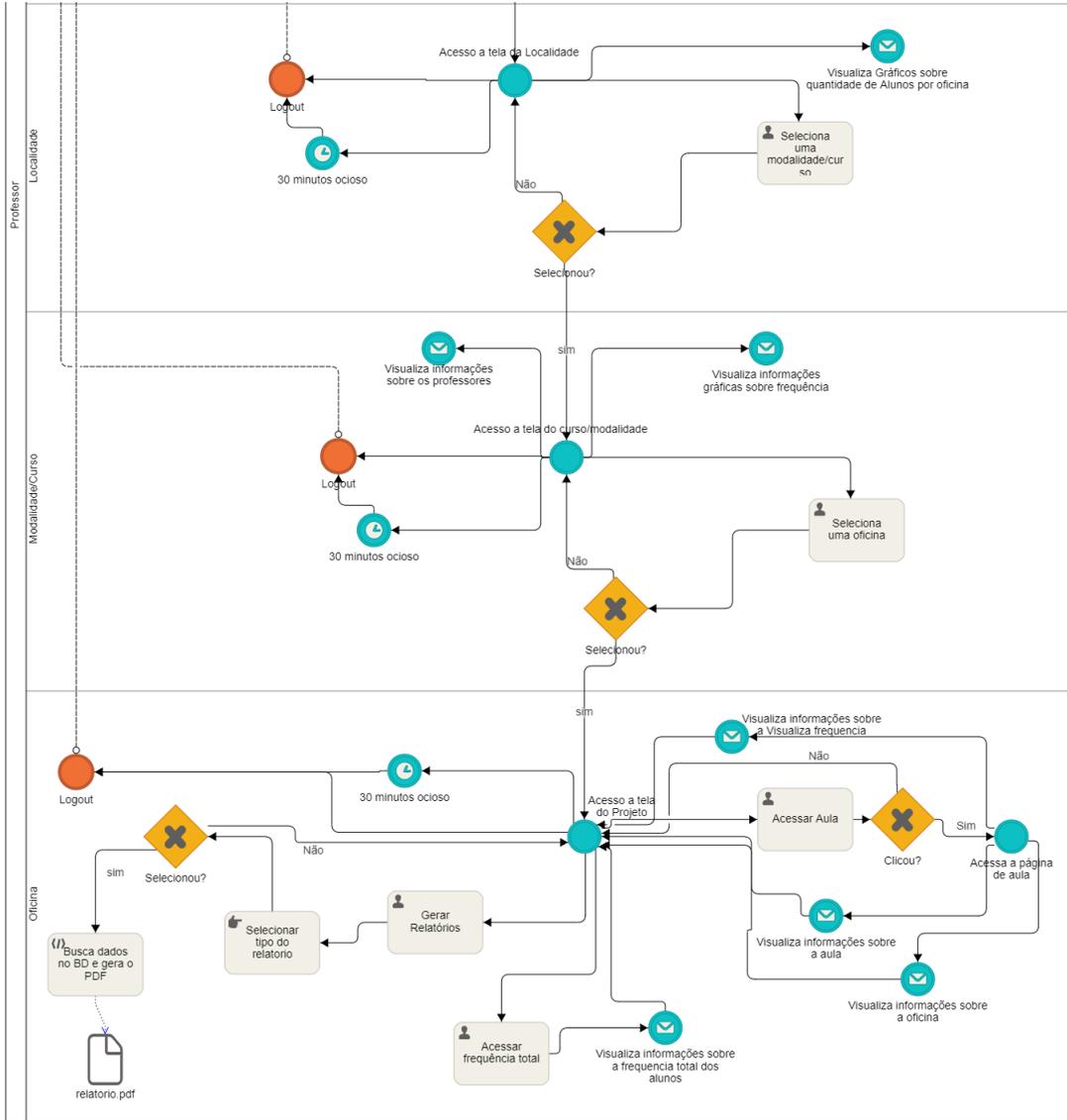
Fonte: Compilação própria

Figura 13 – BPMN - Prefeitura / Secretário parte 01



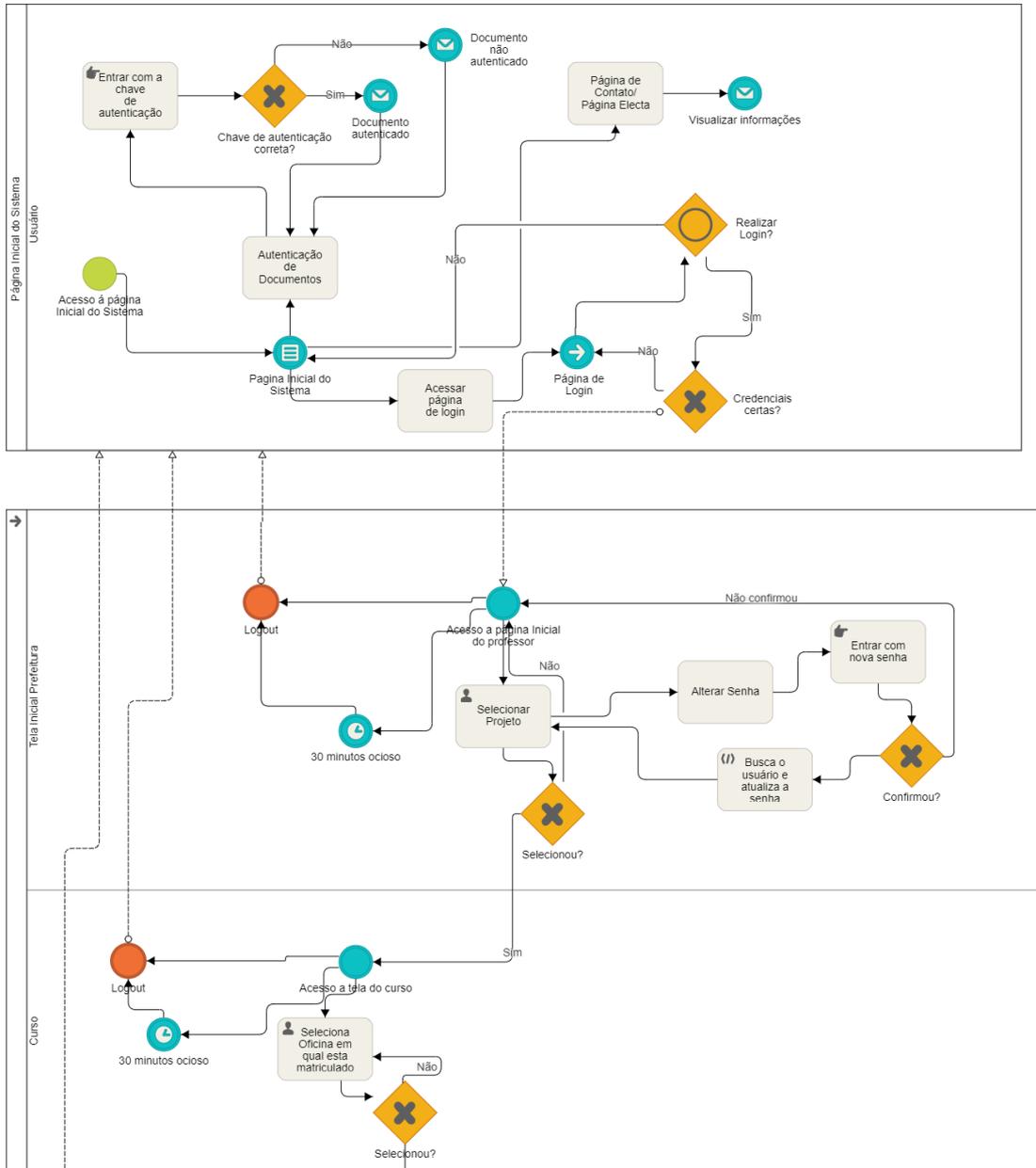
Fonte: Compilação própria

Figura 14 – BPMN - Prefeitura / Secretário parte 02



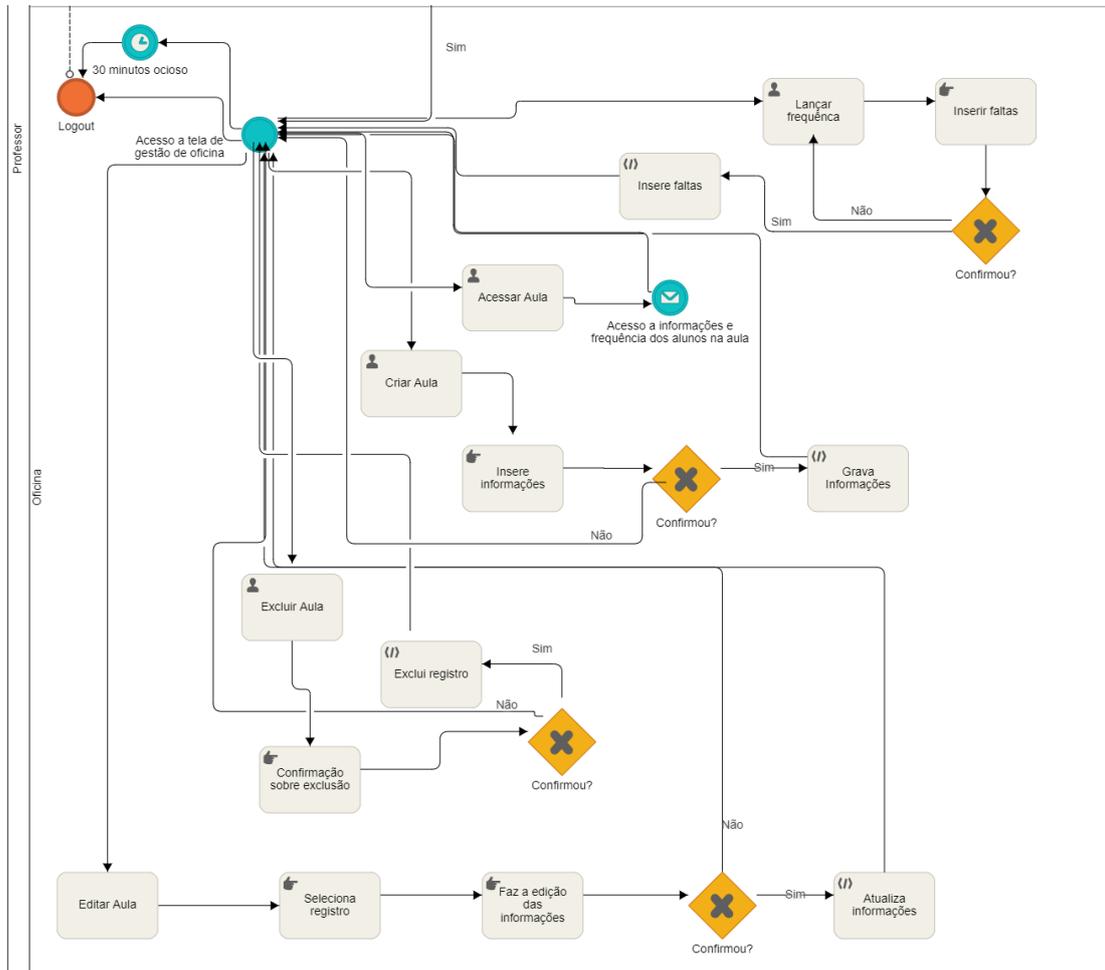
Fonte: Compilação própria

Figura 15 – BPMN - Professores parte 01



Fonte: Compilação própria

Figura 16 – BPMN - Professores parte 02



Fonte: Compilação própria

## 4.2 Apresentação das funcionalidades

Este projeto tem como objetivo facilitar o gerenciamento de informações relacionadas a atividades na área educacional, auxiliando principalmente os professores no controle das turmas, aulas, alunos, faltas, e a parte administrativa gerenciada pelas prefeituras. Nas próximas linhas serão descritas as principais funcionalidades do sistema seguidas por imagens.

O sistema possui telas que realizam operações CRUD (*create, read, update, delete*) que seguem a mesma estrutura para cadastro de prefeituras, localidades, modalidades, oficinas, secretários, professores e alunos. Observe uma dessas telas na figura 17. Operações de adicionar, editar e remover são realizadas em modais, como na figura 18.

Figura 17 – Operações CRUD

SGOElect Inicio Electa Contato Autenticação de Documentos Raphael Policena

Projetos / Localidades / Modalidades

### Modalidades

Prefeitura: Prefeitura de Araçoiaba da Serra  
Projeto: Oficinas para o CIE "Profa Célia Miguel Nottolini"  
Localidade: CIE "Profa Célia Miguel Nottolini"

Adicionar Editar Remover

Show 3 entries Search:

Nome	Vagas	Data início	Localidade
TEATRO DE BONECOS E CONTAÇÃO DE HISTÓRIAS	180	12/08/2019	8
EXPRESSÃO CORPORAL & JOGOS TEATRAIS	180	12/08/2019	8
ARTES PLÁSTICAS	180	12/08/2019	8

Showing 1 to 3 of 3 entries Previous 1 Next

Início Sobre

© Copyright 2019 - Sistema de Gerenciamento de Curso Online

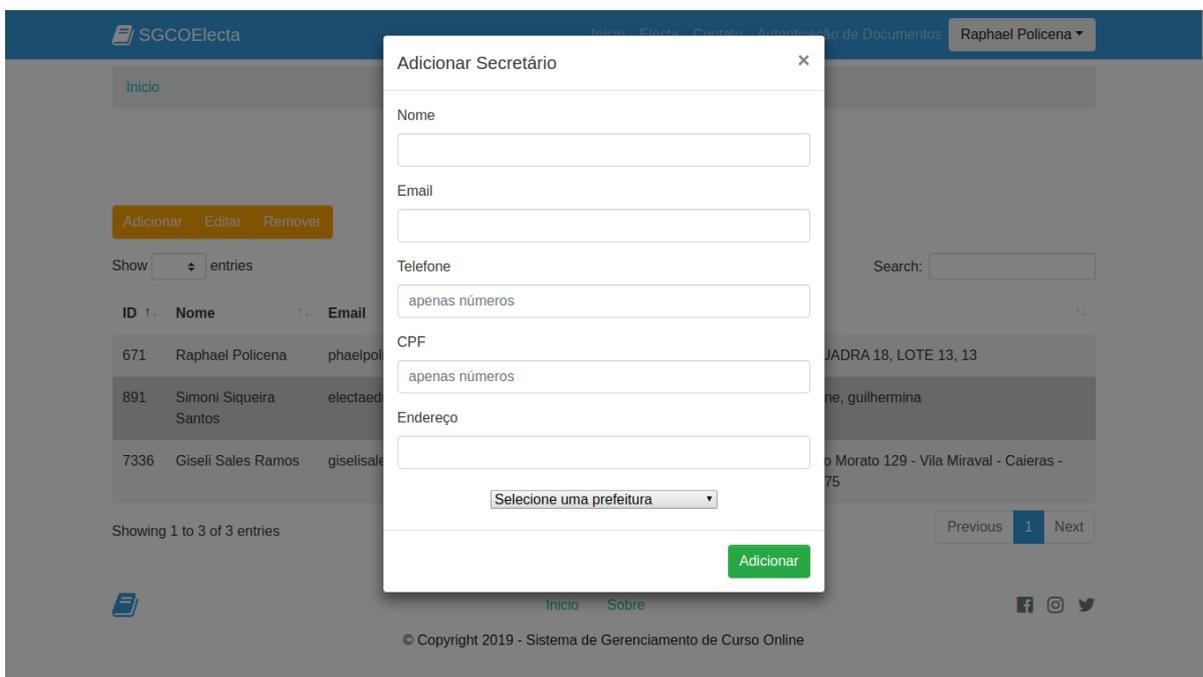
Fonte: Compilação própria

Usuário administrador é o único que tem permissões para gerenciar secretários de uma prefeitura e gerenciar prefeituras. Administradores possuem, além desta permissão, todas as permissões de um secretário. A figura 19 mostra a tela inicial do sistema para um administrador.

Secretários das prefeituras realizam o cadastro de novos projetos, localidades, modalidades e oficinas, vinculando professores e alunos a cada oficina. Os secretários possuem um painel administrativo para cada oficina (Figuras 20 e 21), podendo vincular professores e alunos a tais oficinas, gerar relatórios PDF (Figura 22) e gerenciar o horário das aulas (Figura 23).

Usuários professores, ao realizar login, tem acesso a lista das modalidades vinculadas a ele (Figura 24), onde cada modalidade possui oficinas. Dentro de determinada oficina professores

Figura 18 – Modal para operações, adicionar, editar e remover



Fonte: Compilação própria

Figura 19 – Tela inicial para usuários administradores



Fonte: Compilação própria

Figura 20 – Painel administrativo das oficinas - Visão geral

**Oficina**

Prefeitura: Prefeitura de Araçoiaba da Serra  
Projeto: Oficinas para o CIE "Profª Célia Miguel Nottolini"  
Localidade: CIE "Profª Célia Miguel Nottolini"  
Modalidade: TEATRO DE BONECOS E CONTAÇÃO DE HISTÓRIAS  
Oficina: 1º ano A

**Informações**

**Nome:** 1º ano A [Editar horários](#)

**Carga horária:** 54,0 hrs  
**Data início:** 19/08/2019  
**Data fim:** 15/12/2019

**Ementa:** A oficina propõe através da prática da "Confeção de Bonecos" e da semanal "Contação de Histórias", utilizar-se das diversas lendas, parlendas, contos e causos da cultura popular brasileira para contribuir na ampliação do repertório cultural das crianças, no estímulo à criatividade e imaginação, não só ouvindo, mas também interagindo com as histórias propostas a cada contação. Já os momentos relacionados à criação e manipulação de bonecos de diversos tipos (marionetes, títeres, entre outros) terão como objetivo estimular o desenvolvimento motor das crianças, além de proporcionar momentos de expressão e troca entre as crianças ao criar os personagens e histórias. [Gerar Relatórios](#)

**Professores** [Editar professores](#)

1. Maria Caroline Nakazone
2. Raphael

Fonte: Compilação própria

Figura 21 – Painel administrativo das oficinas - Alunos matriculados

**Alunos matriculados** [Editar alunos](#)

1. Ana Beatriz Gregório
2. Ana Luiza de Oliveira Vieira
3. Arthur Ohata Rodrigues
4. Davi Kauan Batista
5. Gabriela Beatriz Vieira do Amaral Leme
6. Igor Camargo Moraes
7. Isabelly Fernandes de Souza
8. Kaila Gabriela de Souza Camargo
9. Kaique Arruda Domingues
10. Kemili Alves da Silva
11. Laura da Costa Dutra Angerami
12. Lucas Gabriel de Goes Macedo Adão
13. Maicon Henrique Cardoso Rodrigues Barbosa
14. Mathias de Oliveira Cardoso Assunção
15. Mavily Isabelly Santana Peixoto
16. Nicolly dos Santos Medeiros de Almeida
17. Rebeca Alves da Silva
18. Samuel Lucas Ferreira
19. Thayssa Isabella Rodrigues
20. Thiago Henrique Leoterio da Silva



[Início](#) [Sobre](#)



© Copyright 2019 - Sistema de Gerenciamento de Curso Online

Fonte: Compilação própria

Figura 22 – Geração de relatórios

The image shows a web interface with two report generation forms. The first form, titled "Gerar os Relatórios", contains the text "Este relatório é para ser usado quando não há todas as informações dos alunos cadastrados." Below this text are two dropdown menus, one showing "11" and the other showing "2019", followed by a blue button labeled "Gerar Relatório". The second form, titled "Frequência dos alunos nas aulas", contains the text "Aqui você pode gerar um arquivo em PDF contendo todas as faltas de todos os alunos em todas as aulas." It also features two dropdown menus with "11" and "2019" and a blue "Gerar Relatório" button.

Fonte: Compilação própria

Figura 23 – Gerenciamento dos horários de aula para cada oficina

The image displays a "Cadastrar o Horário De Aula" interface. At the top, there is a blue header with the title and a subtitle: "Aqui você pode cadastrar os horários de aula. Cada modalidade pode ter somente um horário cadastrado." Below the header is an orange button labeled "+ Adicionar coluna". The main area contains a table with the following structure:

Horarios de Aula					
Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira	Horario
Português	Matemática	Geografia	Inglês	Espanhol	07:00 as 08:50
Português	Matemática	Geografia	Inglês	Espanhol	07:00 as 08:50
Português	Matemática	Geografia	Inglês	Espanhol	07:00 as 08:50

Below the table is a green button labeled "Criar Horário". At the bottom of the interface, there is a blue header for "Horários" and a text area containing "TEATRO DE BONECOS E CONTAÇÃO DE HISTÓRIAS".

Fonte: Compilação própria

Figura 24 – Tela inicial para professores

Informações Cadastrais		
Informações	Dados Cadastrados	Opções
Nome	Raphael Policena	
Endereço	RUA 13 A, QUADRA 18, LOTE 13, 13	
email	phaelpolicena@gmail.com	

Fonte: Compilação própria

podem, criar aulas (Figura 25), acessar o diário de frequência dos alunos (Figura 26), lançar faltas (Figura 27), acessar aula (Figura 28), editar aula e excluir aula.

Há no SGC O a verificação de autenticidade dos relatórios e documentos gerados pelo sistema. Para utilizar tal funcionalidade o usuário não precisa estar logado, basta ir na página *Autenticação de Documentos* no menu principal da página inicial, vide figura 29.

Figura 25 – Adicionar aula em uma oficina

**Criar Aula**

Data

Conteúdo / Atividades Realizadas  
  
 Máx 500 caracteres.

Materiais Utilizados  
  
 Máx 500 caracteres.

Observações  
  
 Máx 500 caracteres.

Carga Horária  Disciplina

**Criar Aula**

Fonte: Compilação própria

Figura 26 – Diário de frequência

**1º ano A**

Aqui você tem acesso aos dados dos alunos matriculados no curso, bem como as faltas deles desta disciplina / oficina!

Matricula	Nome	Telefone	E-mail	Filiação	Faltas
202371	Ana Beatriz Gregório				11
202373	Ana Luiza de Oliveira Vieira				1
202374	Arthur Ohata Rodrigues				5
202375	Davi Kauan Batista				3
202376	Gabriela Beatriz Vieira do Amaral Leme				9
202377	Igor Camargo Moraes				1
202378	Isabelly Fernandes de souza				6
202370	Kaia Gabriela da Souza Camargo				6

Fonte: Compilação própria

Figura 27 – Lançar faltas

# Aula da data de 21/11/2019

Carga Horária: 1.0

Projeto Oficinas para o CIE "Profª Célia Miguel Nottolini"

Localidade CIE "Profª Célia Miguel Nottolini"

Alunos	
Nome do Aluno	Quantidade de Faltas
Ana Beatriz Gregório	<input type="text" value="0"/>
Ana Luiza de Oliveira Vieira	<input type="text" value="0"/>
Arthur Ohata Rodrigues	<input type="text" value="0"/>
Davi Kauan Batista	<input type="text" value="0"/>
Gabriela Beatriz Vieira do Amaral Leme	<input type="text" value="0"/>
Igor Camargo Moraes	<input type="text" value="0"/>

Fonte: Compilação própria

Figura 28 – Acessar aula

Informações da Aula								
Informações da Oficina								
Dados dos Alunos								
Matricula	Nome	RG/NIS	Telefone	E-mail	Filiação	Faltas		
202371	Ana Beatriz Gregório					0		
202373	Ana Luiza de Oliveira Vieira					0		
202374	Arthur Ohata Rodrigues					0		
202375	Davi Kauan Batista					0		
202376	Gabriela Beatriz Vieira do Amaral Leme					0		
202377	Igor Camargo Moraes					0		
202378	Isabelly Fernandes de souza					0		
202379	Kaila Gabriela de Souza Camargo					0		
202389	Kaique Arruda Domingues					0		
202380	Kemili Alves da Silva					0		
202381	Laura da Costa Dutra Angerami					0		
.....	.....					-		

Fonte: Compilação própria

Figura 29 – Autenticação de documentos

The screenshot shows a web page for document authentication. At the top, there is a blue navigation bar with the logo 'SGOElecta' on the left and links for 'Início', 'Electa', 'Contato', and 'Autenticação de Documentos' in the center. On the right of the bar, the name 'Raphael Policena' is displayed with a dropdown arrow. The main heading is 'Autenticação' in a large, bold font, with the subtext 'Verifique a autenticidade dos documentos!' below it. The form consists of two main sections: 'Caracteres de autenticação' with a text input field containing the alphanumeric string 'CH7AS9GES7DGH6SA3DE26ASAS1KLOMMNHIPVJNGN' and a small instruction 'Coloque todos os caracteres da autenticação presentes no documento'; and 'Coordenador do Projeto' with a dropdown menu showing 'Selecione o coordenador do Projeto' and a blue 'Enviar' button. At the bottom, there is a small logo on the left, links for 'Início' and 'Sobre' in the center, and social media icons for Facebook, Instagram, and Twitter on the right.

Fonte: Compilação própria

## 5 Considerações finais

Este trabalho apresenta o processo de desenvolvimento do software web que gerencia cursos, projeto e oficinas ofertadas por algumas prefeituras do estado de São Paulo. O sistema visa facilitar o trabalho de gerenciamento de turmas possibilitando acesso do usuário ao sistema em qualquer dispositivo com acesso a internet por meio de um navegador. O sistema é escalável possibilitando o aumento na quantidade de acessos em momentos inesperados graças a plataforma de implantação (Jelastic) auto gerenciável.

O sistema pode ser incrementado com novas funcionalidades de acordo com futuras necessidades. Melhorias de interface gráfica, performance e alocação de recursos computacionais, além de padronização nos dados. Outro ponto significativo de melhoria seria a atualização das tecnologias utilizadas, consideradas ultrapassadas. O que motivou a utilização dessas tecnologias foi o conhecimento obtido no decorrer do curso e a falta de domínio sobre frameworks atuais.

O projeto atendeu aos requisitos gerais e específicos, segundo a licitação. A principal dificuldade encontrada foi devido a falta de experiência no desenvolvimento de projetos, podemos citar a forma de negociar os requisitos do sistema com o cliente.

## Referências

- ARNOLD, K. et al. *The Java programming language*. [S.l.]: Addison-wesley Reading, 2000. v. 2. Citado na página 6.
- BARTHOLOMEW, D. *MariaDB cookbook*. [S.l.]: Packt Publishing Ltd, 2014. Citado na página 8.
- BAYERN, S. *JSTL in Action*. [S.l.]: Manning Publications Co., 2002. Citado na página 5.
- BOS, B. et al. Cascading style sheets level 2 revision 1 (css 2.1) specification. *W3C working draft, W3C, June*, 2005. Citado na página 3.
- DEVMEDIA. *Introduzindo o servidor de aplicação Apache Tomcat*. 2013. Disponível em: <<https://www.devmedia.com.br/introduzindo-o-servidor-de-aplicacao-apache-tomcat/27939>>. Acesso em: 12 nov. 2019. Citado na página 12.
- DEVMEDIA. *O que é Full Stack? Veja se você é um programador fullstack*. DevMedia, 2017. Disponível em: <<https://www.devmedia.com.br/quem-quer-ser-um-programador-fullstack/38786>>. Citado na página 4.
- FERREIRA, E.; EIS, D. *Html5: Curso w3c escritório brasil. São Paulo*, 2011. Citado na página 3.
- FLANAGAN, D.; NOVAK, G. M. *Java-Script: The Definitive Guide*. [S.l.]: American Institute of Physics, 1998. Citado na página 3.
- GIT. *About Git*. [S.l.]: GIT, 2017. <<https://git-scm.com/about>>. Acesso em: 11 fev. 2020. Citado na página 10.
- HARRINGTON, J. L. *Relational database design*. 2009. Citado na página 17.
- JELASTIC. *Jelastic, Ruslan Synytsky*. 2011. Disponível em: <<https://jelastic.com/>>. Acesso em: 07 nov. 2019. Citado 2 vezes nas páginas 9 e 10.
- LEMONS, A. *Cibercultura e mobilidade. a era da conexão. intercom*. In: *XXVIII Congresso Brasileiro de Ciências da Comunicação–Uerj*. [S.l.: s.n.], 2005. Citado na página 1.
- NIXON, R. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. [S.l.]: "O'Reilly Media, Inc.", 2014. Citado na página 4.
- PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave Macmillan, 2005. Citado na página 14.
- SCRUM. 2014. <<https://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 20 nov. 2019. Citado na página 16.
- SILVA, R. J. da; URSSI, N. J. *Urbx—como os aplicativos móveis potencializam a vida urbana. Revista de Iniciação Científica, Tecnológica e Artística, São Paulo*, v. 5, n. 1, 2015. Citado na página 1.
- SOMMERVILLE, I. *Software engineering*. [S.l.]: Addison-Wesley/Pearson, 2011. Citado na página 14.

SOUTO, M. *O que é front-end e back-end?* [S.l.]: Mario Souto, 2017. <<https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end/>>. Acesso em: 11 fev. 2020. Citado na página 4.

SPURLOCK, J. *Bootstrap: Responsive Web Development*. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado na página 5.

TOMCAT, A. *Apache Tomcat*. [S.l.]: Apache Tomcat, 2013. <<http://tomcat.apache.org/>>. Acesso em: 11 fev. 2020. Citado na página 11.

TÁVORA, T. *O que é API RESTful? Entenda aqui!* [S.l.]: Thiago Távora, 2019. <<https://www.iset.com.br/blog/o-que-e-api-restful-entenda-aqui/>>. Acesso em: 11 fev. 2020. Citado na página 4.

VIANA, D. *O que é front-end e back-end?* [S.l.]: Daniel Viana, 2017. <<https://www.treinaweb.com.br/blog/o-que-e-front-end-e-back-end/>>. Acesso em: 11 fev. 2020. Citado na página 4.

W3C. *W3C - World Wide Web Consortium*. 2011. Disponível em: <<http://www.w3c.br/Padroes/>>. Acesso em: 06 nov. 2019. Citado na página 2.

WILLIAMS, H. E.; LANE, D. *Web Database Applications with PHP and MySQL: Building Effective Database-Driven Web Sites*. [S.l.]: "O'Reilly Media, Inc.", 2004. Citado na página 7.

WILLIS, J. *A framework for task-based learning*. [S.l.]: Longman Harlow, 1996. v. 60. Citado na página 3.