**UNIVERSIDADE POSITIVO**

**JEAN FRANCISCO SOARES SANTINI**

**OTIMIZAÇÃO DE PROCESSOS JURIDICOS NO NPJ**

**DA FACULDADE POSITIVO LONDRINA**

**LONDRINA**

**2019**

**JEAN FRANCISCO SOARES SANTINI**

**OTIMIZAÇÃO DE PROCESSOS JURIDICOS NO NPJ**

**DA FACULDADE POSITIVO LONDRINA**

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas da Universidade Positivo.

Orientador: Prof. Me. João Gilberto Piotto

**LONDRINA**

**2019**

**TERMO DE ANUÊNCIA**

**TERMO DE APROVAÇÃO**

**RESUMO**

**ABSTRACT**

**SUMÁRIO**

**1 INTRODUÇÃO**

1.1 O NÚCLEO DE PRÁTICAS JURÍDICAS

O Núcleo de Práticas Jurídicas (NPJ) é um órgão de dentro das universidades que possuem o curso de Graduação em Direito, onde alunos, geralmente dos últimos anos ou semestres, realizam o estágio obrigatório de sua grade curricular (PROJURIS, [2017?]).

O objetivo do NPJ é solucionar conflitos, judicial ou extrajudicialmente, entre os envolvidos de maneira segura, rápida e profissional, como também dar aos alunos a possibilidade aplicar todo o conteúdo aprendido durante o curso, e desse modo são preparados para os desafios da vida prática da profissão, nas várias áreas que o Direito proporciona (PUCPR, [19--?]).

É através desse órgão que os alunos prestam atendimento ao público de forma gratuita, desde que os interessados comprovem que não possuem condições financeiras de promover uma ação, para que possam ter a assistência judiciaria gratuita (UP, [19--?]).

No curso de Graduação em Direito da Universidade Positivo (UP), campus Londrina, os alunos realizam suas atividades no NPJ durante as matérias de Estágio Supervisionado Simulado I e II e Estágio Supervisionado em Direito I, II e III (UP, [19--?]).

1.2 DA ORGANIZAÇÃO DOS DADOS NO NPJ

Baseando-se na maneira em que o NPJ da Universidade Positivo, campus Londrina trabalha, observa-se que todas as informações obtidas nas atividades do órgão devem ser armazenadas para que, no futuro, as mesmas sejam encontradas com maior facilidade, clareza e objetividade. Esse armazenamento deve auxiliar na organização dos alunos, professores e responsáveis de forma que todos os registros da negociação não se percam ou gerem alguma informação inválida. Além disso, encontrar tais dados, de maneira alguma, deve ser um processo lento e complicado.

Desde o início do projeto do NPJ na UP, há de se destacar as realidades vividas na pratica de quem está inserido no órgão. Uma delas é que o número total de atendimentos registrados vem aumentando a cada ano, outra é que em determinados períodos, ocorrem muitas demandas simultâneas.

As diferentes informações geradas nos atendimentos realizados no NPJ possuem uma relação forte, sendo assim, encontrando parte delas, as demais também devem estar visíveis ou intuitivamente fáceis para serem localizadas, o que gera uma boa experiência ao usuário.

No dia a dia do órgão, as atividades como um todo, juntando todos os seus dados necessários, não alcançaram a maneira mais prática de serem executadas, podendo atrasar ou, no pior caso, falhar em algum atendimento. O motivo não seria por alguma negligência de professores e alunos, mas pela fraca interação dos mesmos com o sistema que utilizam atualmente, dadas as limitações do mesmo.

O aumento dos dados sem um melhor método de organização, então, contribui negativamente com a avaliação dos professores e responsáveis dos alunos em relação aos seus desempenhos, por meio de relatórios mais precisos. Sem isso, o andamento das atividades de cada aluno é prejudicado, o auxílio ou reconhecimento vindo de responsáveis para os alunos se torna quase impossível, como também a completa integração do aluno ao órgão é danificada.

Delegando-se responsabilidades de um processo ou serviço para os diferentes participantes do órgão, a ausência de um controle de verificação de acessos e trâmites faz com que prazos e possíveis conclusões da negociação se tornem tarefas complexas. Um dos participantes pode abandonar o projeto ou se perder em suas demandas e não ocorrerá nenhum tipo de aviso aos responsáveis quando o prazo da tarefa esgotar.

Com isso, é fácil perceber que a gestão manual dos dados do NPJ não atende as necessidades do órgão e o uso de ferramentas específicas é necessário. O controle por meio de planilhas, anotações físicas e pastas do sistema operacional gera lentidão e desorganização para os responsáveis.

1.3 DO AUXÍLIO NA ORGANIZAÇÃO DOS DADOS

Observados os problemas, nota-se que um local ou método mais prático para organizar todos os elementos obtidos de processos, tarefas e documentos gerados vem se desenhando cada vez mais necessário.

Esse método trará mais eficácia na avaliação dos alunos, também aumentará a taxa de sucesso na conclusão dos processos. Focado no auxílio da administração, o método disponibilizará avisos de prazo, mostrará os períodos de acesso dos alunos às tarefas e a organização de prioridades e passos para o bom andamento das operações já realizadas.

É imprescindível o desenvolvimento do sistema que não mude o fluxo de trabalho que já é realizado, mas que torne todas as atividades mais práticas e ágeis. O foco de todo o esforço será na aprendizagem prática dos alunos e na resolução dos problemas que entram como demanda.

1.4 OBJETIVOS

**1.4.1 Objetivo geral**

Apresentar uma solução com foco em melhorias na organização dos dados, de forma que todos os processos ou negociações do NPJ sejam executados de forma mais proficiente.

**1.4.2 Objetivos específicos**

1. Compor um conjunto de aplicações que permitem o cadastro, a edição e a visualização, de forma rápida e intuitiva, de processos, partes, tarefas e documentos no sistema.
2. Gerar gráficos e valores comparativos para que sejam avaliados todos os processos e etapas realizados.
3. Gerar gráficos e valores de avaliação para todas as atividades e movimentações dos alunos, para que se possa ter um relatório de produtividade dos mesmos.
4. Criar um controle de acesso para que cada um possa exercer seu papel com níveis diferentes de responsabilidade, informando sempre quem foi o responsável pela última ação realizada

**2 FUNDAMENTAÇÃO TEÓRICA**

Como visto anteriormente, é impossível negar a importância de um sistema que consiga contribuir positivamente na organização de todas as tarefas do NPJ da UP. Há de se destacar, também, outros exemplos que comprovam essa necessidade. O mercado de desenvolvimento de softwares é cada vez mais forte e, com isso, novas maneiras de superar obstáculos são pensadas em implementadas (CRONAPP, 2017).

2.1 TRABALHOS RELACIONADOS

**2.1.1 O armazenamento e manuseio de informações**

Para Moesch e Toffolo (2013), a maior dificuldade que alguma entidade que atua na área de direito, como um escritório de advocacia, por exemplo, está em manusear uma grande quantidade de informações que são geradas pelas demandas da entidade. Nestas demandas incluem fichas de cadastros, atendimentos, tarefas, agendamentos, processos e relatórios, todos estes documentos são muitas vezes cruciais para o sucesso de tarefas.

Uma solução citada por Moesch e Toffolo (2013), é a aquisição de um sistema de informação singular que garanta a unicidade de todos os dados da entidade. É necessário que a mesma solução garanta também maior eficiência e ganho de tempo na busca dessas informações.

De acordo com a entrevista com o advogado Miguel de Oliveira Paul ao ProJuris ([2016?]) que hoje é usuário do sistema para gestão das rotinas de seu escritório de advocacia, a maior dificuldade que ele enfrentava era encontrar um modo onde pudesse ter um controle maior de todos os processos em que seu escritório atua. Um dos seus maiores inimigos, neste caso, era o tempo.

Considerando a quantidade de colaboradores no escritório, o controle de acesso e alterações em informações salvas em planilhas ou em documentos físicos gera uma rotina de trabalho menos eficiente. Tudo isso foi solucionado com a aquisição do software que hoje contribui para gestão de todas as tarefas.

A necessidade de organização das informações de um software se torna ainda mais evidente quando comparamos as opiniões de Moesch e Toffolo (2013), a experiência vivida pelo advogado Miguel de Oliveira (PROJURIS, [2016?]), juntamente com opiniões e benefícios do site do programa SAJ ADV ([20--?]), citadas abaixo.

O software torna possível a diminuição ou até mesmo a exclusão total de planilhas de controle extras que funcionarios utilizam em suas empresas ou instituições. Isso é possível com a unificação de cadastro de atendimentos, pessoas processos, trazendo facilidade na busca global de qualquer informação em um mesmo lugar (SAJ ADV, ([20--?]).

**2.1.2 A melhor decisão: Baseado na web versus Desktop**

Quando se trata de escolher uma tecnologia para o desenvolvimento, uma grande influência na decisão vem juntamente com o domínio que o desenvolvedor já possui. Dentro do que foi utilizado por Moesch e Toffolo (2013), uma das opções foi um sistema baseado em formulários do próprio sistema operacional, ou seja, uma aplicação de desktop. No caso do software ProJuris ([2016?]) e do software SAJ ADV ([20--?]), tem-se que a aplicação foi feita baseada na web.

Tanto software em desktop quanto web são muito utilizados e requisitados. De acordo com MOMook (2018), o segredo é identificar em qual tipo de utilização se enquadra melhor para a aplicação, pois cada tipo de software apresenta vantagens e desvantagens.

Um aplicativo de desktop somente pode ser adquirido de um formato físico (CDs, DVDs, Pendrives) ou de download pela internet, isso significa que o aplicativo é executado localmente, tendo a opção de não ter uma conexão com a internet (MOMOOK, 2018).

Em contrapartida, o software baseado na web é acessado usando um navegador da web, sem a necessidade de ser adquirido ou instalado no computador. A principal diferença aqui é que há a necessidade de uma conexão com a internet, permitindo troca de informações online (MOMOOK, 2018).

Considerando todos esses pontos, para que haja acesso de vários usuários ao mesmo sistema, de forma mais simples, é descartada a opção de instalação de um programa fixo em um computador. Apenas o servidor será configurado em uma máquina fixa e a aplicação será disponibilizada na web local para que todos possam utilizar de computadores diferentes.

2.2 MODELAGEM DE PROCESSOS DE NEGÓCIO

Para um entendimento do que é a modelagem de processos de negócio, será necessário definir o que é modelo e o que é processo de negócio. Um modelo é uma representação simplificada da realidade. Um processo de negócio se define por uma sequência de atividades iniciadas a partir de demandas, com o objetivo de adicionar valor aos seus clientes (PARREIRAS, 2015).

Definidos modelo e processo de negócio, a modelagem de processos de negócio entra com o objetivo de identificar, analisar e modelar os processos, para torná-los mais eficientes, aumentando a produtividade e a lucratividade dos clientes (IDEIA CONSULTORIA, 2019).

**2.2.1 Modelo e Notação de Processos de Negócio**

O Modelo e Notação de Processos de Negócio, tradução de Business Process Model and Notation (BPMN), consiste em uma unificação das notações de diagramas que mostram as atividades de uma organização. Este trabalho permite que se tenha um maior entendimento global do negócio (TRENTIM, [20--?]).

Em BPMN, um processo de negócio é simulado por meio de uma conexão entre eventos e atividades. Essa conexão possui uma direção e demonstra a sequência em que os mesmos são realizados (SGANDERLA, 2012).

Caso haja a necessidade de divisão e unificação de fluxos paralelos no decorrer de um mesmo processo de negócio, outros elementos de controle de fluxo podem ser utilizados para o mapeamento correto do mesmo (SGANDERLA, 2012).

Um exemplo de processo mapeado utilizando BPMN, para a compra de um refrigerante, pode ser visto na Imagem 1.

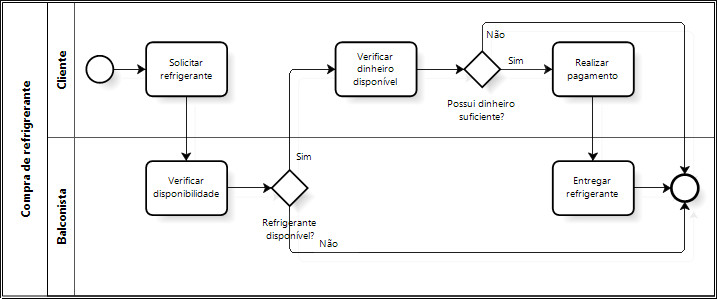


Imagem 1 – BPMN  
Fonte: Sganderla (2012)

2.3 ENGENHARIA DE REQUISITOS

Uma boa engenharia de requisitos é essencial para o desenvolvimento de um bom produto. É possível detalhar, levantar e validar todos os seus requisitos, tornando mais claro o que de fato precisa ser feito, o que tem maior valor e o que tem menor valor para quem propôs sua demanda (PAULA FILHO, 2000).

**2.3.1 Entendendo os Requisitos**

Requisitos são as necessidades para um produto. É nessa etapa que se identifica as partes interessadas e seus diferentes pontos de vista sobre o que se tem como problema. Após isso, é definido quais desses problemas o projeto, em sua conclusão, pretende solucionar (CERRI, 2018).

**2.3.2 Tipos de Requisitos**

Definido, então, o entendimento sobre os requisitos de um produto, é possível desmembrá-los, inicialmente, em dois grupos de requisitos específicos, os requisitos de usuário e os requisitos do sistema.

Os requisitos de usuário são descritos com uma linguagem natural, de forma que se torne compreensível aos usuários, apresentando restrições, serviços e funcionalidades do sistema (COSTA, 2011).

Os requisitos de sistema são apresentados com conceitos mais técnicos e detalhados, de forma que possam ser mais úteis aos desenvolvedores do software. Devem descrever o que o sistema deve ou não fazer (SOMMERVILLE, 2012).

Dentro dos requisitos citados, evidenciando os requisitos do usuário, ainda é possível subdividi-lo em dois outros requisitos, os funcionais e não funcionais.

Os requisitos funcionais, como o próprio nome diz, delineiam as funcionalidades que o sistema trará em benefício aos seus usuários (PAULA FILHO, 2000), ou seja, o que o sistema deve entregar, como deve se comportar e o que deve responder.

Os requisitos não funcionais são aqueles ligados a restrições sobre as funções que o sistema oferece. Dentre os mais importantes, destaca-se a confiabilidade, desempenho, espaço em disco e até mesmo tempo de resposta, dada uma requisição (PAULA FILHO, 2000).

2.4 MODELAGEM DE SOFTWARE

A má compreensão de requisitos de uma demanda pode resultar deficiências em um sistema construído. Isso acontece pela falta de um amplo conhecimento na regra necessária para o desenvolvimento, causando até mesmo a criação de um produto que não atende as necessidades do cliente (CASTILHO, 2008).

São esses acontecimentos que trazem a necessidade de uma solução através da modelagem. A Modelagem de software é uma representação de algo do mundo real. A partir daí se pode projetar o que um software deverá fazer e, com isso, auxiliar analistas e programadores na compreensão de como o sistema e suas funções se comportarão (CASTILHO, 2008).

**2.4.1 Linguagem Unificada de Modelagem**

A Linguagem Unificada de Modelagem, tradução de Unified Modeling

Language (UML), é uma linguagem padrão para modelar. Utilizando modelagem orientada a objetos, ela auxilia a visualizar a comunicação entre os objetos empregando diagramas padronizados (MARTINEZ, [21--]).

Os diagramas UML são tipicamente divididos em estruturais e comportamentais. Quando é necessário especificar, construir e documentar aspectos estáticos, utilizando atributos e classes e seus relacionamentos, são utilizados diagramas estruturais (ROMBALDI, 2011). Já o diagrama comportamental, como o nome diz, detalha o comportamento ou funcionamento de partes de um sistema (GROFFE, 2013).

Alguns dos principais diagramas, que se enquadram dentro dos tipos de diagramas abordados acima estão descritos conforme a Imagem 2.

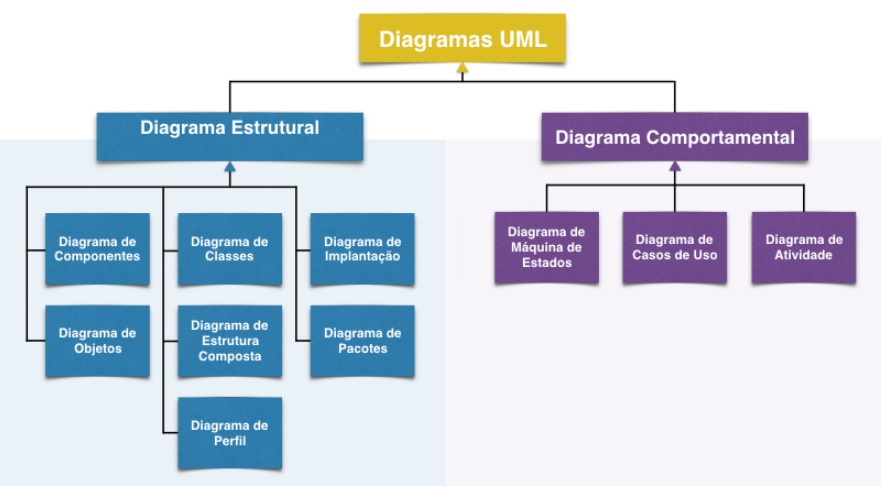


Imagem 2 – Diagramas  
Fonte: Adaptado de Vieira (2015)

**2.4.2 Diagrama de caso de uso**

O diagrama de caso de uso descreve a interação das principais funcionalidades do sistema com os seus usuários. No diagrama não há detalhes técnicos e aprofundamentos da regra de negócio, apenas os cenários, os usuários e

as tarefas possíveis de serem realizadas, bem como a comunicação entre o usuário e essas funcionalidades (RIBEIRO, 2012).

Sua utilização se dá por meio de figuras que simbolizam os artefatos citados acima, separando-as em ator como sendo o usuário, caso de uso sendo as funcionalidades ou tarefas e a comunicação, conforme a Imagem 3.

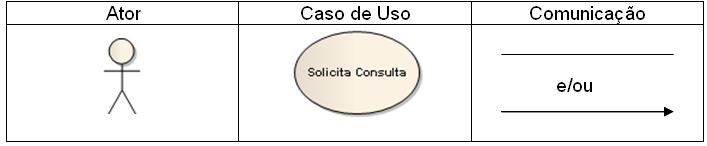


Imagem 3 – Caso de Uso  
Fonte: Ribeiro (2012)

**2.4.3 Diagrama de classes**

Diagrama de classes é uma representação que descreve a estrutura de um sistema, apresentando suas classes, operações e as relações entre os objetos. É apresentado com detalhes mais técnicos. Além disso, ele pode ser usado para a construção de outros diagramas que irão definir o tipo de comunicação, sequência e estados do sistema (SIGNIFICADOS, [2018?]).

Dos itens da diagramação existentes em um diagrama de classe, as classes são elementos abstratos que possuem a especificação de um objeto (SIGNIFICADOS, [2018?]).

O que irá definir as características de uma classe, como nome, tipo de dados, visibilidade e propriedade são os seus atributos. Já a capacidade das classes de se relacionarem surge com a associação, que indica de onde parte e para onde irão suas informações (SIGNIFICADOS, [2018?]). As informações acima são elucidadas conforme o exemplo do diagrama de classes, referido na Imagem 4.

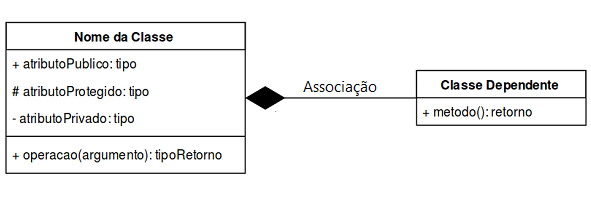


Imagem 4 – Diagrama de classes  
Fonte: Adaptado de Significados ([2018?])

2.5 TECNOLOGIAS UTILIZADAS

**2.5.1 Da organização das etapas: Kanban**

Quanto a metodologia de desenvolvimento, o Kanban, que nasceu inspirado em sistemas de organização de supermercados americanos (RUNRUN.IT, [2017?]), é responsável pelo auxílio no controle de progresso das tarefas de forma visual.

Em sua forma mais tradicional, utiliza-se um quadro no qual são colados alguns papéis (Post-it) e cada papel representará uma tarefa a ser realizada. No quadro encontram-se três etapas de desenvolvimento, as quais todos os papéis passaram por cada uma, são elas: A fazer (To Do), fazendo (In progress) e feito (Done) (BERNARDO, 2014), exemplificado na imagem 5.

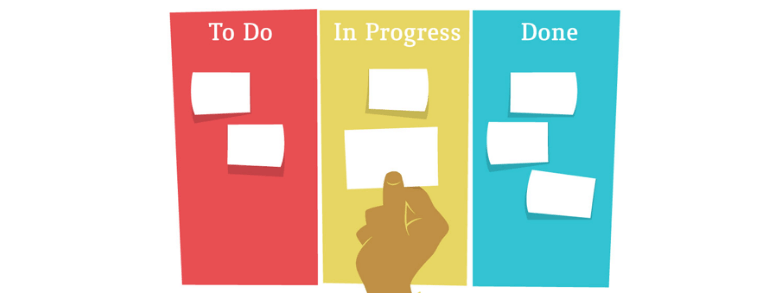


Imagem 5 – Kanban  
Fonte: Dutra (2016)

**2.5.2 Banco de dados PostgreSQL**

PostgreSQL é um sistema de gerenciamento de banco de dados que tem como principal função armazenar dados de forma segura e rápida, fazendo com que possam ser recuperados via requisições ou pedidos de uma ou mais aplicações de software (DIONISIO, 2015).

Consiste em um servidor que possui comunicação com um conjunto de programas clientes que, via instruções específicas, recebe e grava informações enviadas, garantindo também, a captura dessas informações para visualizar, editar ou removê-las, de acordo com o que é necessário (DIONISIO, 2015).

**2.5.3 Linguagem de programação: C#**

O C# é uma linguagem de programação orientada a objetos, desenvolvida pela Microsoft, criada para facilitar o processo de desenvolvimento, trazendo inúmeros recursos que proporcionam boa produtividade a quem utiliza (GUEDES, 2018).

Por ser uma linguagem multiplataforma, torna possível o desenvolvimento para plataformas web, dispositivos móveis e aplicações desktop. Desse modo, a tecnologia pode ser utilizada nos projetos mais simples até em projetos complexos (GUEDES, 2018). Tendo como parâmetro uma idade, a Imagem 6 elucida uma tomada de decisão de maioridade em C#.

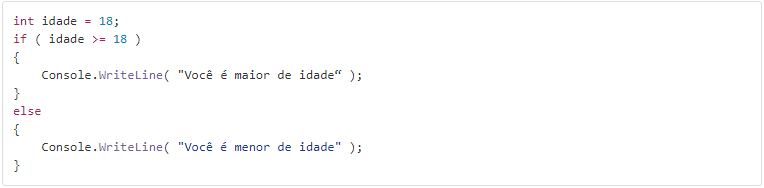


Imagem 6 – Tomada de decisão em C#  
Fonte: Guedes (2018)

**2.5.4 Entity Framework**

O Entity Framework (EF) é um framework de mapeamento objeto-relacional, do inglês Object-relational mapping (ORM), distribuído pela Microsoft (MUSSO, 2019). Uma de suas principais funções é abstrair a comunicação com um banco de dados relacional (MUSSO, 2019), de modo que o desenvolvedor possa se relacionar com a entidade do banco usando apenas objetos da linguagem C#.

Com a tecnologia é possível reduzir a incompatibilidade entre os vários bancos relacionais existentes, concentrando o código em uma única linguagem. Com isso, é possível a troca de banco de dados com pouca ou nenhuma alteração no código nas consultas do sistema (VEGA et al., 2016). Somente será reconfigurada a conexão com o novo banco.

**2.5.5 HTML**

A sigla HTML vem do inglês Hypertext Markup Language, em português Linguagem de Marcação de Hipertexto. É uma linguagem utilizada no desenvolvimento de websites (EIS, 2011).

O HTML foi criado para ser de fácil entendimento, sendo lido pelo desenvolvedor ou pela máquina. Os programas responsáveis pela leitura e interpretação das instruções em HTML são os navegadores (EIS, 2011).

Em sua estrutura, o HTML é uma linguagem baseada em marcação. O que torna possível combinar textos com informações sobre o texto, utilizando palavras-chaves diferentes (SOUZA, [20--?]). A imagem 7 exemplifica a estrutura básica de um código em HTML.

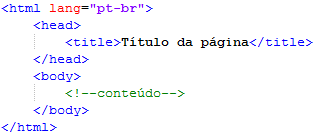


Imagem 7 – Estrutura básica HTML  
Fonte: Autoria própria

**2.5.6 CSS**

CSS é a sigla para Cascading Style Sheet, traduzido para o português, significa Folha de Estilo em Cascatas. Utilizado na decoração de páginas web, com o CSS é possível alterar cores, fontes, espaçamentos entre textos. Em suma, focado na estética, o CSS auxilia na formatação da página (GONÇALVES, 2019).

**2.5.7 JavaScript**

JavaScript é uma linguagem de programação da web utilizada para manipular solicitações e respostas em um site, interagindo com o conteúdo da linguagem HTML e a estilização proporcionada pelo CSS (SILVA, 2015).

A linguagem permite ao desenvolvedor códigos de alto nível de complexidade em páginas web, como animações, gráficos ou informações que se atualizam em tempo real (ZAMPIERI, 2019). A Imagem 8 traz um exemplo de função que verifica se número digitado está entre 1 e 10.

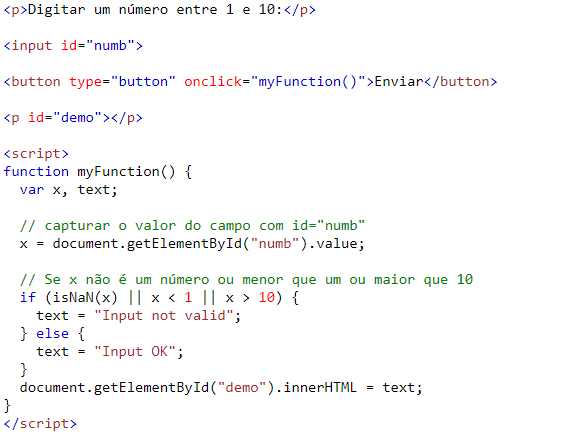


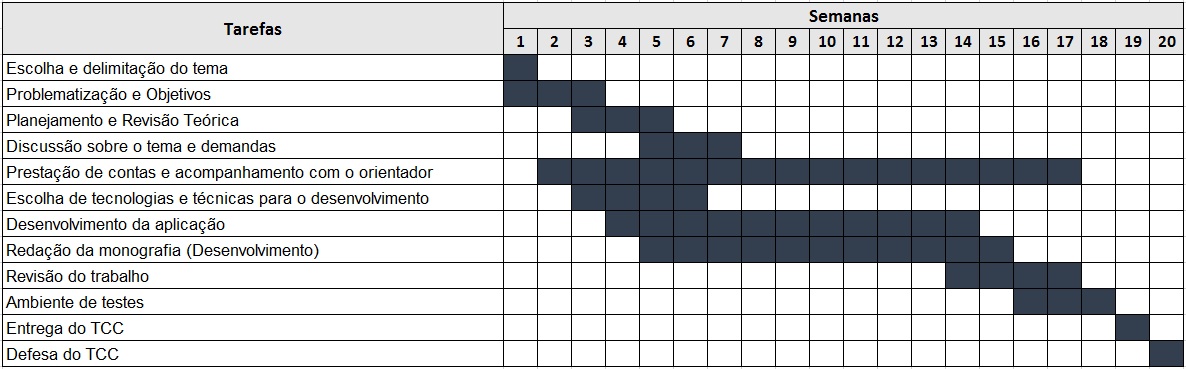
Imagem 8 – Exemplo JavaScript  
Fonte: Autoria própria

**3 DESENVOLVIMENTO**

Ao se examinar o cotidiano dos participantes do NPJ, a necessidade de organizar todas as rotinas e passos com o fim de produzir uma solução imediata se torna uma imprescindível.

3.1 PLANEJAMENTO DO DESENVOLVIMENTO

Diante da problemática acima abordada, o planejamento de todas as etapas de produção para alcançar a conclusão do projeto, na forma a qual foi definido, é indispensável. Sendo assim, de acordo com o que foi analisado a respeito do programa a ser elaborado identificou-se que é preciso seguir as orientações conforme o cronograma de tarefas anexo na Imagem 9.

Imagem 9 – Cronograma  
Fonte: Autoria própria

**3.1.1 Metodologia do desenvolvimento**

Para que haja organização no processo de criação da aplicação, é imprescindível pensar em uma engenharia de software capaz de definir todos os passos necessários para o bom andamento no seu desenvolvimento.

Dadas as informações em tópicos anteriores sobre tecnologias utilizadas, para o projeto a ser desenvolvido será escolhido o Kanban, que entrará com o auxílio no controle de progresso das tarefas de forma visual.

3.2 MODELAGEM DOS PRINCIPAIS PROCESSOS DO NEGÓCIO

**3.2.1 Atendimento**

O NPJ da UP opera por atendimentos. A partir da primeira visita do cliente é necessário um registro do atendimento realizado, com informações suficientes para dar andamento à problemática trazida até a sua resolução.

Para a melhor definição de todas as etapas da geração de um atendimento a ser realizado durante a rotina de trabalho do órgão, o fluxo foi modelado utilizando BPMN. A imagem 10 demonstra todas as sequências para cadastro de um atendimento no NPJ.

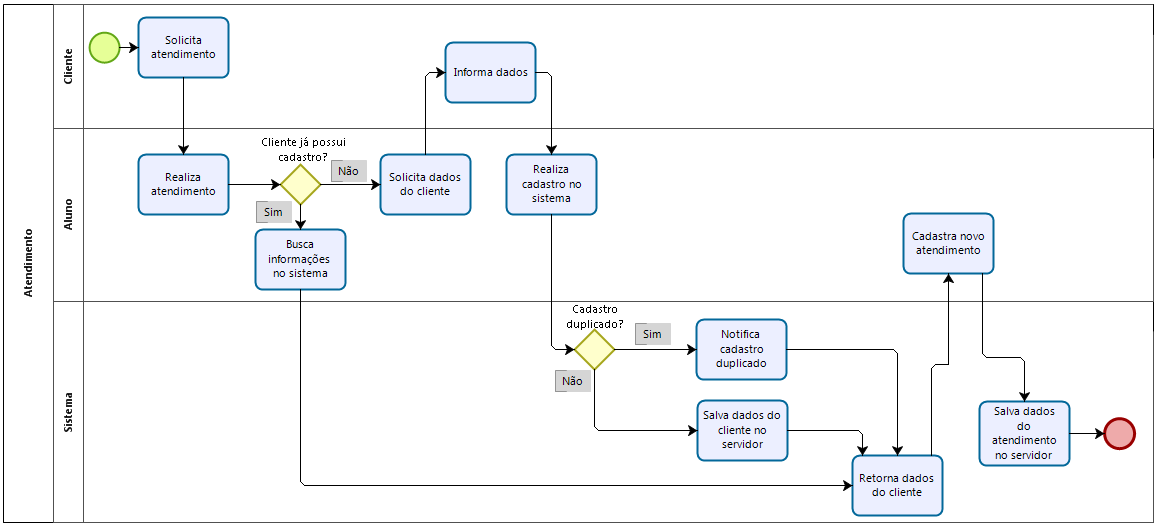


Imagem 10 – Atendimento  
Fonte: Autoria própria

**3.2.2 Tarefas**

Ao realizar um atendimento, o aluno ou responsável já é capaz de definir as

próximas etapas a serem tomadas, se necessário entrar em contato com alguma outra pessoa, ou haverá necessidade de agendar uma nova visita do cliente para atualizar a etapa do atendimento.

Em ambas as situações, serão necessárias várias tarefas até que o atendimento seja concluído. As tarefas criadas pelos alunos responsáveis pelo atendimento são essenciais para uma boa avaliação dos professores.

Entre as tarefas, é possível criar um registro de uma nova etapa para o atendimento, dando um novo passo até que tudo se conclua e auxiliando o aluno na organização do que precisará ser feito. A imagem 11 demonstra o fluxo de cadastro de uma tarefa no NPJ modelado utilizando BPMN.

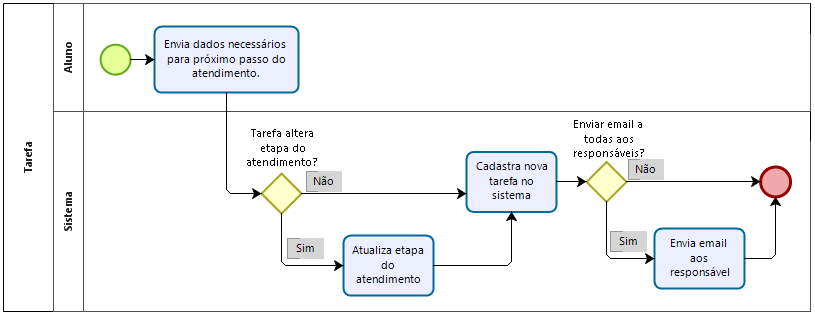


Imagem 11 – Tarefa  
Fonte: Autoria própria

3.3 ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA

**3.3.1 Requisitos de Usuário**

Com a complexidade atual do projeto, devido a vários controles de informações e de todo seu histórico, a criação de documentos que auxiliem no entendimento do que o usuário demanda será um recurso importante. Dentre os diferentes requisitos, já explicados no documento, destacam-se alguns dos requisitos de usuário que serão descritos conforme o Quadro 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Código | Requisito | Descrição | Partes interessadas | Prioridade | Funcional / Não Funcional |
| RU1 | Cadastrar usuário | Cadastro de usuário será dividido em dois tipos: Aluno ou responsável. Para ambos os casos deverá ter os campos: login, senha e papel (responsável ou aluno). Para o registro de um aluno, informações básicas como: nome, sexo, telefones, e-mail, semestre, número de documentos, matrícula, especialidades, horários e grupos. Para o usuário responsável, serão necessários: Nome, sexo, tipo de responsável, telefones, e-mail e documentos | Alunos e responsáveis. | Alta | F |
| RC1 | Cadastrar cliente | Cadastro de clientes deverá ter os campos: Nome, sexo, documentos, telefones, endereço. | Alunos e responsáveis. | Alta | F |
| RA1 | Cadastrar atendimento | Cadastro de tarefas deverá conter um cliente, um grupo, o usuário que criou o atendimento, data/Hora da criação, a etapa do atendimento, o tipo do atendimento e uma condição de concluído ou não. | Alunos e responsáveis. | Alta | F |
| RP1 | Cadastrar processo | Cadastro de processos deverá conter um vínculo com um atendimento, o número do processo, o tipo da ação, a situações atuais no Projudi e NPJ, como também a lista de situações em que passou, registradas com o usuário e data de modificação, expectativa de valor da causa, percentual de honorários, valor de honorários. | Alunos e responsáveis. | Alta | F |

Quadro 1 – Requisitos de Usuário  
Fonte: Autoria própria

**3.3.1 Requisitos de Sistema**

Listados os principais requisitos de usuário no Quadro 1, corrobora-se a necessidade de um aprofundamento maior em cada um deles a fim de auxiliar o desenvolvedor nas tarefas de trazer para o sistema exatamente o que o usuário procura. Baseando nisso, parte dos requisitos de sistema, de acordo com o Quadro 1, são listados no Quadro 2, trazendo informações para cadastrar usuário.

|  |  |
| --- | --- |
| Cadastrar usuário | |
| Função | CadastrarUsuario |
| Descrição | Realiza o cadastramento de um novo usuário no sistema a partir dos dados inseridos pelo professor ou responsável. |
| Entradas | Login, senha e papel (responsável ou aluno). Caso papel seja responsável: Nome, sexo, tipo de responsável, telefones, e-mail e documentos (RG e CPF), caso papel seja aluno: nome, sexo, telefones, e-mail, semestre, número de documentos (RG e CPF), matrícula, uma lista de especialidades e uma lista de grupos. |
| Fonte | Informações vindas do próprio aluno. |
| Saídas | Registro na tabela “Usuario” e “Aluno” ou “Responsavel” no banco de dados. |
| Ação | O cadastro de alunos informar um alerta caso tenha algum campo branco ou fora dos padrões em: login, nome, papel, e-mail, telefone, sexo. Caso o tipo de usuário seja aluno, matrícula e ao menos uma responsabilidade serão obrigatórios, no outro caso, o tipo de responsável se torna obrigatório. |
| Requisitos | Quando o responsável inicia o cadastro, o mesmo deverá ter em disponibilidade todas as informações de entradas vindas do aluno. Se não possuir uma senha de escolha, uma senha padrão pode ser inserida para que o aluno possa alterar depois. |
| Efeitos colaterais | O usuário não poderá ser cadastrado sem login, e-mail, telefone, nome válidos de acordo com os padrões corretos para os tipos de informação. |

Quadro 2 – Requisitos de Sistema (Cadastrar usuário)  
Fonte: Autoria própria

Ressalta-se, também, informações para cadastrar um novo cliente, descritas no Quadro 3.

|  |  |
| --- | --- |
| Cadastrar cliente | |
| Função | CadastrarCliente |
| Descrição | Realiza o cadastramento de um novo cliente no sistema a partir dos dados inseridos pelo responsável ou aluno. |
| Entradas | Nome, sexo, data de nascimento, documentos (RG e CPF), telefones, endereço. |
| Fonte | Informações vindas do próprio cliente. |
| Saídas | Registro de um novo cliente na tabela “Cliente” do banco de dados. |
| Ação | O cadastro de clientes deverá informar caso tenha algum formulário em branco ou inconsistente. O sistema possuirá validações para nome, sexo, telefone, endereço e data de nascimento. Clientes duplicados não deverão ser salvos. |
| Requisitos | Quando o aluno ou responsável iniciam o cadastro, os mesmos deverão ter em disponibilidade todas as informações de entradas vindas do cliente. |
| Efeitos colaterais | O cliente não poderá ser cadastrado sem algum telefone, endereço e data de nascimento válidos de acordo com os padrões corretos para os tipos de informação. Nomes vazios também não serão aceitos. |

Quadro 3 – Requisitos de Sistema (Cadastrar cliente)  
Fonte: Autoria própria

Trazendo foco nas informações para cadastro de atendimento, segue as suas descrições de sistema, vide Quadro 4.

|  |  |
| --- | --- |
| Cadastrar atendimento | |
| Função | CadastrarAtendimento |
| Descrição | Cadastro de tarefas deverá conter um cliente, um grupo, o usuário que criou o atendimento, data/Hora da criação, a etapa do atendimento, o tipo do atendimento e uma condição de concluído ou não. |
| Entradas | Id do cliente, Id do grupo, Id do usuário de criação, Id do usuário responsável, Data/Hora da Criação, Id da Etapa de Atendimento Atual, Id do Tipo de Atendimento, status de concluído como verdadeiro ou falso. |
| Fonte | Informações vindas dos próprios responsáveis ou alunos. |
| Saídas | Registro de um responsável na tabela “Atendimento” e possíveis registros na tabela “TarefasAtendimento” do banco de dados. |
| Ação | O sistema possuirá validações para os Ids vazios de Cliente, Id do usuário de criação, Id do usuário responsável e Id da etapa de atendimento atual. |
| Requisitos | Quando o responsável ou iniciam o cadastro, os mesmos deverão ter em disponibilidade todas as informações de entradas vindas dos mesmos. |
| Efeitos colaterais | O atendimento não poderá ser cadastrado sem um cliente vinculado, sem um usuário responsável ou sem uma etapa de atendimento atual. |

Quadro 4 – Requisitos de Sistema (Cadastrar atendimento)  
Fonte: Autoria própria

Os requisitos de sistema para cadastro de um novo processo são listados no Quadro 5.

|  |  |
| --- | --- |
| Cadastrar processo | |
| Função | CadastrarProcesso |
| Descrição | Cadastro de processos deverá conter um vínculo com um atendimento, o número do processo, o tipo da ação, a situações atuais no Projudi e NPJ, como também a lista de situações em que passou, registradas com o usuário e data de modificação, expectativa de valor da causa, percentual de honorários, valor de honorários. |
| Entradas | Id do atendimento, número do processo, Id to dipo da ação, Data/Hora da criação, Id do polo, Id da situação Projudi atual, Id da situação NPJ atual, Expectativa do valor da causa, percentual de honorários, valor de honorários. |
| Fonte | Informações vindas dos próprios professores, alunos ou responsáveis, juntamente com informações dos clientes. |
| Saídas | Registro de cadastro de um processo na tabela “Processo”, possibilidade de registro na tabela “TipoProcesso”, “RegistroSituacaoprojudi” e “RegistroSituacaoNPJ” do banco de dados. |
| Ação | O sistema possuirá validações para status, tipo da ação, Id da situação NPJ atual. |
| Requisitos | Quando o professor, aluno ou responsável iniciam o cadastro, os mesmos deverão ter em disponibilidade todas as informações de entradas vindas do cliente. |
| Efeitos colaterais | O processo não poderá ser cadastrado sem as informações do status, sem um vínculo com um atendimento, sem uma situação NPJ ou sem um tipo da ação. |

Quadro 5 – Requisitos de Sistema (Cadastrar processo)  
Fonte: Autoria própria

**REFERÊNCIAS**

BERNARDO, Kleber. **Kanban: Do início ao fim!**. [*S. l.*], 8 dez. 2014. Disponível em: https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/. Acesso em: 25 mar. 2019.

CASTILHO, Marcelo. **Modelagem de Software**. [*S. l.*], 27 ago. 2008. Disponível em: https://www.dimensaotech.com/2008/08/modelagem-de-software/. Acesso em: 4 abr. 2019.

CERRI, Fernando. **Engenharia de Requisitos: conheça todas as etapas do processo**. São Paulo, 2 jul. 2018. Disponível em: https://www.opencadd.com.br/engenharia-de-requisitos-conheca-todas-as-etapas-do-processo/. Acesso em: 31 mar. 2019.

COSTA, Leandro. **Engenharia de Requisitos**. [*S. l.*], 15 jul. 2011. Disponível em: http://www.semeru.com.br/blog/engenharia-de-requisitos/. Acesso em: 31 mar. 2019.

CRONAPP. **Os maiores desafios e barreiras no desenvolvimento de softwares**. [*S. l.*], 9 out. 2017. Disponível em: https://www.cronapp.io/pt-br/os-maiores-desafios-e-barreiras-no-desenvolvimento-de-softwares/. Acesso em: 31 mar. 2019.

DIONISIO, Edson José. **PostgreSQL Tutorial**. [*S. l.*], 2015. Disponível em: https://www.devmedia.com.br/postgresql-tutorial/33025. Acesso em: 14 abr. 2019.

DUTRA, Cecília. **OTIMIZE PROCESSOS COM A METODOLOGIA KANBAN**. [*S. l.*], 1 ago. 2016. Disponível em: https://www.siteware.com.br/metodologias/metodologia-kanban/. Acesso em: 14 abr. 2019.

EIS, Diego. **O básico: O que é HTML?**. [*S. l.*], 21 jan. 2011. Disponível em: https://tableless.com.br/o-que-html-basico/. Acesso em: 29 abr. 2019.

GONÇALVES, Ariane. **O que é CSS? Guia Básico para Iniciantes**. [*S. l.*], 29 jan. 2019. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/. Acesso em: 30 abr. 2019.

GROFFE, Renato Jose. **Modelagem de sistemas através de UML: uma visão geral**. São Paulo, 2013. Disponível em: https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913. Acesso em: 7 abr. 2019.

GUEDES, Marylene. **Como Começar Com C# (C Sharp)?**. [*S. l.*], 24 jan. 2018. Disponível em: https://www.treinaweb.com.br/blog/como-comecar-com-c-sharp/. Acesso em: 14 abr. 2019.

IDEIA CONSULTORIA. **O que é Modelagem de Processos e como Fazer?**. [*S. l.*], 26 jan. 2019. Disponível em: https://ideiaconsultoria.com.br/gestao-empresarial/o-que-e-modelagem-de-processos-e-como-fazer/. Acesso em: 1 maio 2019.

MARTINEZ, Marina. **UML**. [*S. l.*], [21--]. Disponível em: https://www.infoescola.com/engenharia-de-software/uml/. Acesso em: 5 abr. 2019.

MOESCH, Beatriz; TOFFOLO, Geferson. **SISTEMA PARA GERENCIAMENTO DE PROCESSOS JURÍDICOS PARA O SAJUG**. Santa Catarina, 2013. Disponível em: http://revista.faifaculdades.edu.br/index.php/conexao/article/view/23/9. Acesso em: 25 mar. 2019.

MOMOOK. **WEB-BASED VS. DESKTOP SOFTWARE: WHICH IS BETTER?**. Lituânia, 9 out. 2018. Disponível em: https://momook.com/web-based-vs-desktop-software-which-is-better/. Acesso em: 25 mar. 2019.

MUSSO, Emiliano. **Entity Framework Introduction Using C# - Part One**. [*S. l.*], 7 abr. 2019. Disponível em: https://www.c-sharpcorner.com/article/entity-framework-introduction-using-c-sharp-part-one/. Acesso em: 29 abr. 2019.

PARREIRAS, Pedro. **Modelagem de Processos de Negócios: O que é, para que serve?**. [*S. l.*], 17 out. 2015. Disponível em: https://fluxoconsultoria.poli.ufrj.br/blog/gestao-empresarial/modelagem-de-processos-de-negocios/. Acesso em: 1 maio 2019.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software**: fundamentos, métodos e padrões. São Paulo: LTC Editora, 2000.

PROJURIS. **Aprenda em 1 minuto: o que é NPJ? Como funciona? Quem pode participar?**. [*S. l.*], [2017?]. Disponível em: https://www.projuris.com.br/como-funciona-nucleo-de-pratica-juridica-npj/. Acesso em: 25 mar. 2019.

PROJURIS. **“Com ProJuris, acabamos com a necessidade de alimentar planilhas em Excel, relatórios e agenda física”**. [*S. l.*], [2016?]. Disponível em: https://www.projuris.com.br/case-software-juridico-projuris-paul-gouveia-horta-advogados/. Acesso em: 25 mar. 2019.

PUCPR. **Núcleo de Prática Jurídica**. [*S. l.*], [19--?]. Disponível em: https://www.pucpr.br/escola-de-direito/nucleo-de-pratica-juridica/. Acesso em: 25 mar. 2019.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. São Paulo, 2012. Disponível em: https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408. Acesso em: 7 abr. 2019.

ROMBALDI, Lucas. **UML e os Diagramas Estruturais**. [*S. l.*], 2011. Disponível em: http://micreiros.com/uml-e-os-diagramas-estruturais/. Acesso em: 5 abr. 2019.

RUNRUN.IT. **O que é kanban e como ele pode ajudar na organização da rotina de trabalho**. [*S. l.*], [2017?]. Disponível em: https://blog.runrun.it/o-que-e-kanban/. Acesso em: 25 mar. 2019.

SAJ ADV. **SAJ ADV**. Santa Catarina, [20--?]. Disponível em: https://www.sajadv.com.br/. Acesso em: 25 mar. 2019.

SGANDERLA, Kelly. **Um guia para iniciar estudos em BPMN (I): Atividades e sequência**. [*S. l.*], 19 nov. 2012. Disponível em: http://blog.iprocess.com.br/2012/11/um-guia-para-iniciar-estudos-em-bpmn-i-atividades-e-sequencia/. Acesso em: 1 maio 2019.

SIGNIFICADOS. **Significado de Diagrama de classes**. [*S. l.*], [2018?]. Disponível em: https://www.significados.com.br/diagrama-de-classes/. Acesso em: 7 abr. 2019.

SILVA, Giancarlo. **O que é e como funciona a linguagem JavaScript?**. [*S. l.*], 28 jan. 2015. Disponível em: https://canaltech.com.br/internet/O-que-e-e-como-funciona-a-linguagem-JavaScript/. Acesso em: 29 abr. 2019.

SOMMERVILLE, Ian. **Engenharia de software**. [*S. l.*]: Pearson, 2007.

SOUZA, Tiago Daniel. **Guia Prático de HTML - Parte 1**. [*S. l.*], [20--?]. Disponível em: http://www.linhadecodigo.com.br/artigo/1184/guia-pratico-de-html-parte-1.aspx. Acesso em: 29 abr. 2019.

UP. **Núcleo de Prática Jurídica (NPJ)**. Paraná, [19--?]. Disponível em: https://www.up.edu.br/londrina/nucleo-de-pratica-juridica--npj-. Acesso em: 25 mar. 2019.

TRENTIM, Mario Henrique. **Modelagem de Processos de Negócio**. [*S. l.*], [20--?]. Disponível em: http://www.techoje.com.br/site/techoje/categoria/detalhe\_artigo/1142. Acesso em: 1 maio 2019.

VEGA, Diego *et al*. **Entity Framework 6**. [*S. l.*], 22 out. 2016. Disponível em: https://docs.microsoft.com/en-us/ef/ef6/index. Acesso em: 29 abr. 2019.

VIEIRA, Rodrigo. **UML — Unified Modeling Language**. [*S. l.*], 6 dez. 2015. Disponível em: https://medium.com/operacionalti/uml-1f7b99dd15bb. Acesso em: 7 abr. 2019.

ZAMPIERI, Gabriel. **O que é JavaScript**. [*S. l.*], 22 jan. 2019. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-javascript/. Acesso em: 29 abr. 2019.