UNIVERSIDADE ESTADUAL DE MONTES CLAROS

Centro de Ciências Exatas e Tecnológicas

Curso de Bacharelado Sistemas de Informação

Ryan Lucas Silva Lemos

APLICAÇÃO DO EXTREME PROGRAMMING NO DESENVOLVIMENTO DE UM AMBIENTE WEB DE ENSINO E APRENDIZAGEM DA LÍNGUA INGLESA

Montes Claros/MG

Agosto/2018

**Ryan Lucas Silva Lemos**

**APLICAÇÃO DO EXTREME PROGRAMMING NO DESENVOLVIMENTO DE UM AMBIENTE WEB DE ENSINO E APRENDIZAGEM DA LÍNGUA INGLESA**

Projeto de Pesquisa apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Estadual de Montes Claros como exigência para obtenção de nota na disciplina Projeto de Trabalho de Conclusão de Curso.

Orientadora: Prof.ª. CHRISTINE MARTINS DE MATOS, MESTRA

**Montes Claros/MG**

**Agosto/2018**

**LISTA DE ABREVIATURAS E SIGLAS**

**Sumário**

# INTRODUÇÃO

# Tema e problema

Nesta seção apresentam-se o tema e o problema deste projeto.

## Tema

Desenvolvimento *web* com técnicas ágeis.

## Problema

Quais os benefícios e dificuldades de se utilizar técnica *eXtreme Programming* (XP) de desenvolvimento ágil para criação de um sistema *web* que auxilia na aprendizagem de língua inglesa?

# Justificativa

O estudo de línguas estrangeiras é disseminado globalmente, sendo requisitado em vários aspectos profissionais, educacionais, de pesquisa e de interação das pessoas nos diferentes países. Diversas escolas se prontificam a ensinar os idiomas com diferentes métodos de ensino, material e apoio informatizados. Porém, muitas escolas não contam com apoio computacional para auxiliar no processo de ensino e aprendizagem.

A escola escolhida para servir de base e apoio a confecção do ambiente é a *International* *English* *Center* (ILC), situada em Montes Claros-MG, e está inserida no mercado há 18 anos, contando com cerca de 90 alunos.

A ILC foi escolhida pelo fato de não dispor de sistema de auxílio no ensino/aprendizagem, tendo somente um *site* desenvolvido pela Empresa Júnior do curso de Sistemas de Informação da UNIMONTES, a INFOBITS, com a URL <http://www.joininit.com.br/>, com informações básicas da escola, sendo uma página estática, não tendo recursos de gerenciamento de processos gerenciais.

Então propõe-se desenvolver um ambiente *web* que apoie os processos de ensino e aprendizagem da escola, e pensa-se que ao final deste trabalho a escola possa apresentar aos seus clientes e futuros clientes o diferencial de ter um sistema exclusivo para tal fim.

Com o ambiente proposto deseja-se auxiliar alunos, professores e responsáveis (caso o aluno seja menor de idade) em aspectos específicos de ensino/aprendizagem. Um exemplo de auxílio aos professores se dá pela confecção e correção de provas e atividades que deixarão de ser manuais. A atuação humana na correção se dará em questões discursivas, com o lançamento posterior de pontuação no ambiente.

Para os alunos espera-se que haja envolvimento e aprendizado quanto aos conteúdos disponibilizados no ambiente virtual. Acredita-se que com o ambiente, a interação aluno-professor e aluno-turma transpassará da sala de aula, possibilitando interação de aprendizagem de maneira informatizada.

Em caso de alunos menores de idade, surge o papel do responsável, que no ambiente conseguirá verificar como está o processo de aprendizagem do aluno responsável, acompanhando seu desempenho e frequência.

Pretende-se utilizar para confecção do ambiente, técnicas de desenvolvimento ágil. Acredita-se que com a utilização de metodologia ágil, o processo de desenvolvimento seja feito de maneira mais profissional, econômica e rápida. Supõe-se que ao final do trabalho, uma opinião mais profunda seja formada quanto aos benefícios e dificuldades em se utilizar um método ágil para desenvolvimento. A metodologia ágil escolhida para auxiliar no processo de desenvolvimento será o *eXtreme* *Programming* (XP). Espera-se que ao final do trabalho possa-se apoiar desenvolvedores na decisão de escolha ou não do método XP.

# Objetivos

Este capítulo apresenta os objetivos deste trabalho.

## Objetivo Geral

Desenvolver ambiente *web* com técnicas ágeis para auxiliar no processo de ensino e aprendizagem de língua inglesa para a escola *International English Center* (ILC).

## Objetivos Específicos

* Disponibilizar materiais e exercícios para os alunos no ambiente desenvolvido;
* Apontar acertos e erros de provas elaboradas a partir do módulo de banco de questões;
* Identificar conteúdos de maior deficiência a partir de gráficos de desempenho.
* Apresentar benefícios e resultados do uso de técnicas de desenvolvimento ágil (metodologia XP).

# Referencial teórico

“O referencial teórico é o alicerce para o desenvolvimento de uma monografia.” (DIAS e SILVA, 2010, p. 31). Assim pode-se entender o quão importante é descrever aquilo que servirá de base para a concepção de um trabalho científico. Severino (2002) define essa etapa como sendo a busca por documentos que apresentem conteúdo relevante e que fazem sentido ao tema discutido no trabalho científico. Essa busca e escrita de conteúdo serve para que o leitor tenha uma visão inicial do que será abordado e tenha base para compreender os conceitos que serão abordados no decorrer do trabalho científico.

## Educação a distância – ambiente virtual

A maneira de se ensinar mudou-se adequando as necessidades e mudanças (sociais, tecnológicas, etc.). Um exemplo disso é o da Educação a Distância (EAD) que surgiu no Brasil por volta de 1900, por meio de cursos oferecidos por correspondência, a fim de possibilitar mais conveniência ao estudo (ALVES, NUNES, *et al.*, 2009).

Com o alcance da internet e a maior acessibilidade a computadores e smartphones a EAD se propagou pelo território nacional (ALVES, NUNES, *et al.*, 2009). E tem sido de importância tendo em vista que pessoas em localidades sem acesso direto ao ensino presencial, podem estudar e alcançar uma formação (NETO, AZEVEDO, *et al.*, 2012).

Nas secções subsequentes serão descritos alguns exemplos de sistemas que utilizam da EAD para possibilitar e automatizar o processo de ensino. Serão demonstrados no âmbito de ensino em geral, mas também para fins de idiomas, que é o foco deste trabalho.

### Sistemas de apoio ao ensino/aprendizagem – Classroom

Como descrito anteriormente existem sistemas cujo objetivo é apoiar os processos de ensino de escolas e universidades. Um exemplo deste tipo de sistema é o Google Sala de Aula que está diretamente envolvido no processo de ensino em geral. Se trata então de uma solução que se encaixa a diferentes necessidades dos professores, servindo desde turmas infantis até acadêmicos (GOOGLE, 2018). Sua definição pode ser tomada como:

O Google Sala de aula é um serviço da Web gratuito para escolas, organizações sem fins lucrativos e qualquer pessoa com uma Conta do Google pessoal. Com o Google Sala de aula, os alunos e professores se conectam facilmente, dentro e fora de escolas. (GOOGLE, 2018, p. 1).

Os alunos são divididos por turmas virtuais que podem representar por exemplo turmas do mundo real. Há a figura de um ou mais professores que fazem parte e comandam essas turmas (GOOGLE, 2018). Dentre algumas funcionalidades possíveis em uma turma tem-se a criação de atividades. Nessas atividades os alunos inseridos na turma têm ou não um prazo definido pelo professor para entrega (GOOGLE, 2018). Ainda é possível por meio do professor saber qual aluno entregou em atraso a atividade, e o tempo de atraso se houver. Além das atividades é possível estabelecer contato entre aluno e professor, compartilhar ou receber materiais, dentre outras funcionalidades (GOOGLE, 2018).

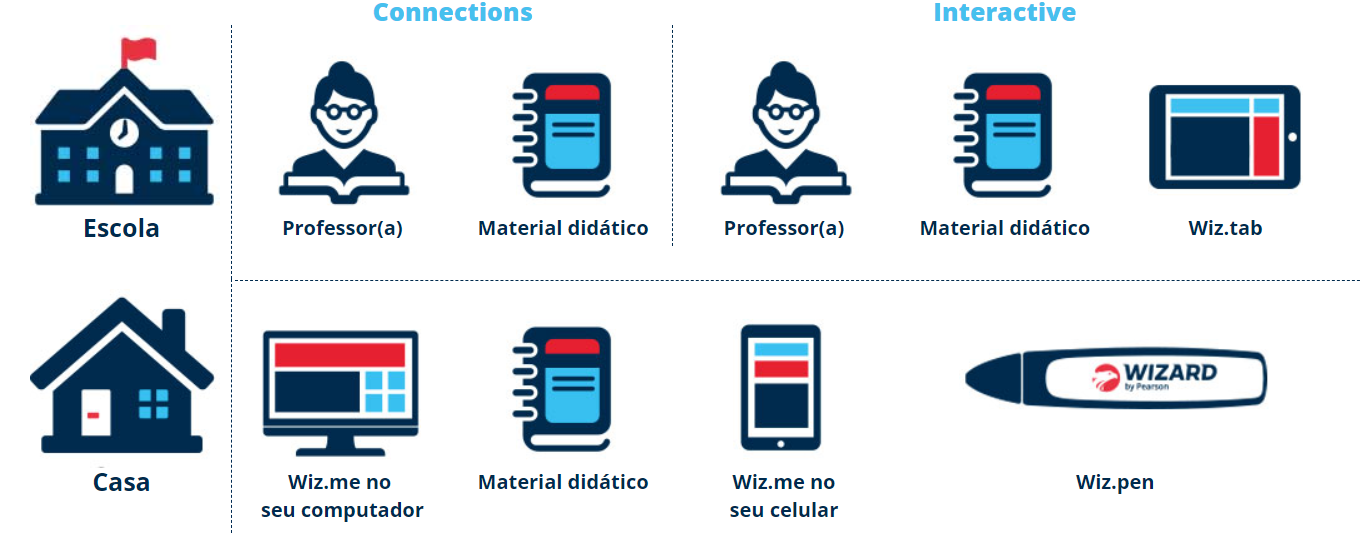
Tendo em vista algumas das funcionalidades do Google Sala de aula presume-se que sua proposta seja a de aproximar alunos e professores, já que serve de espaço para interação assim como em uma sala de aula real. Por isso servirá de base, junto com outros sistemas que serão descritos na seção 5.1.2, para criação do ambiente proposto neste trabalho.

### Metodologias/sistemas de apoio de ensino de idiomas

Como descrito anteriormente, a tecnologia influencia e molda a maneira em que o ensino é feito (ALVES, NUNES, *et al.*, 2009). Isso não é diferente para o ensino de idiomas, uma vez que as escolas têm buscado se adequar e inserir tecnologias nos seus processos de ensino, e soluções independentes tem surgido para auxiliar no aprendizado de línguas. Essas escolas e tecnologias serão descritas nos próximos parágrafos, que apresentarão suas principais características.

Como exemplo de escola que tem se moldado a se adequar a novas tecnologias toma-se a *Wizard by Pearson*, que é uma escola de idiomas internacional (possui sedes em outros países além do Brasil) (WIZARD, 2017). A instituição conta com uma metodologia de ensino que alia a clássica abordagem de sala de aula (chamada pela *Wizard* de *Connections*) ao uso de tecnologias de apoio informatizado (chamada de *Interactive*) (WIZARD, 2017). Na *connections* os alunos vão a sala de aula e interagem com outros alunos e professores, e serão guiados pelo professor (WIZARD, 2017). Já no *interactive* o aluno recebe apoio informatizado por meio de um tablet denominado Wiz.tab (WIZARD, 2017). Para auxiliar a pronúncia dos alunos dispõe-se de uma caneta (denominada Wiz.pen), que faz a leitura de palavras, frases e expressões contidas no material do aluno. A figura 1 demonstra o modelo de aprendizagem da Wizard com as características anteriormente descritas.

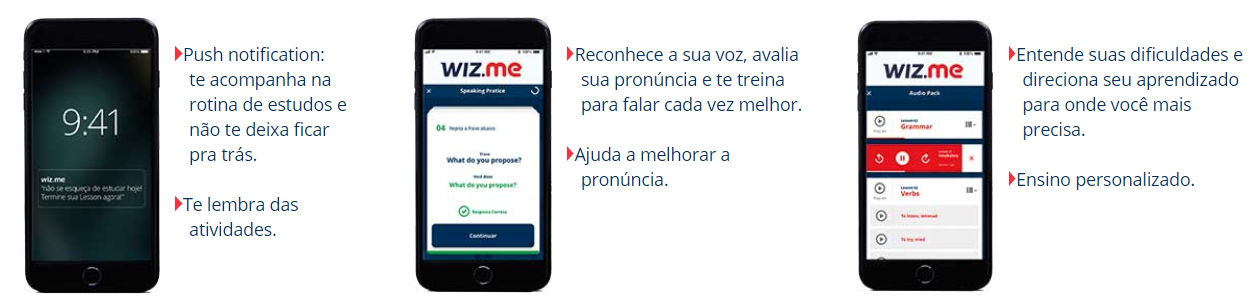
Figura 1 - Modelo de aprendizagem da Wizard



Fonte: WIZARD, 2017, p.3.

A escola ainda conta com uma aplicação *mobile* e um ambiente *Web* denominados Wiz.me, que servem de apoio aos alunos quando estão fora da sala, auxiliando-os na pronuncia e escrita (WIZARD, 2017). A figura 2 descreve algumas funcionalidades da versão móvel dessa aplicação.

Figura 2 - Funcionalidades do Wiz.me



Fonte: WIZARD, 2017, p.4.

Outra escola que interliga tecnologia e ensino de idiomas é a escola CCAA. A CCAA detém um espaço virtual denominado “Espaço CCAA Aluno” (CCAA, sd.). Nele o aluno da escola terá acesso a conteúdo para *tablet*, como textos, áudios e vídeos (CCAA, sd.). Outro recurso disponível é o *ComputerAssisted Language Learning* (CALL) que se dá por um “Software educacional que permite que o aluno realize seus exercícios escritos utilizando o computador. É uma forma rápida, fácil, interativa e agradável de fixar o conteúdo aprendido em sala de aula” (CCAA, sd., p. 1). O espaço virtual ainda conta com atividades complementares para auxílio de escrita, pronúncia e leitura.

Já em questão de aplicações independentes para auxilio de idiomas tem-se o exemplo do Babbel, que é uma aplicação disponível para dispositivos móveis e Web. Seu objetivo é o auxílio da compreensão, escrita e fala de diversos idiomas como inglês, português, espanhol, alemão, holandês, entre outros (BABBEL, 2018). Contempla diversos níveis de conhecimento, indo do básico ao avançado. Apesar da aplicação ser gratuita para ser utilizada, somente alguns níveis estão disponíveis de forma gratuita, sendo necessário assim o pagamento de planos para a liberação de todo o conteúdo disponível (BABBEL, 2018) conforme descrito na figura 3.

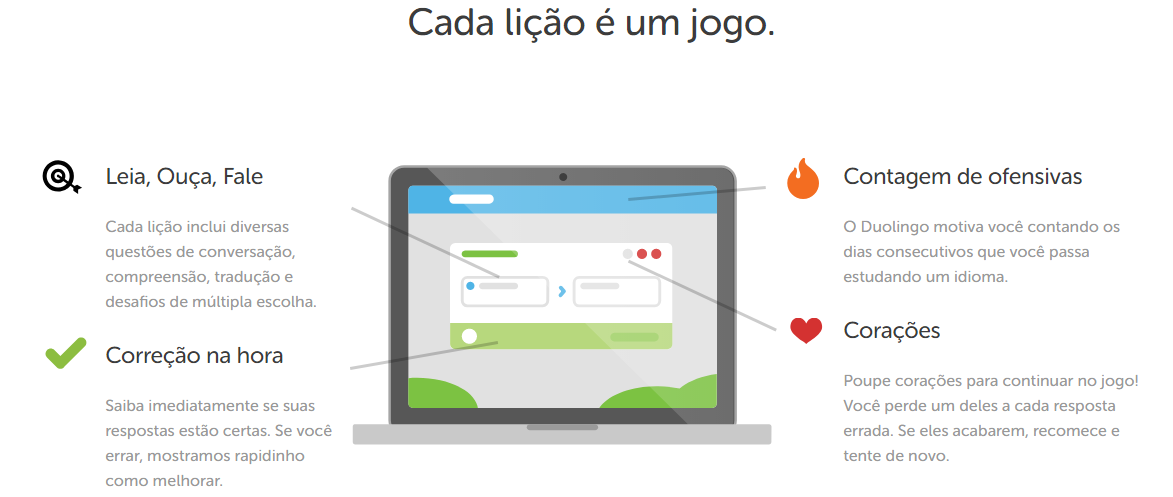
Figura 3 - Preços do Babel



Fonte: BABBEL, 2018, p.1.

Outra aplicação semelhante ao Babbel é o Duolingo que possibilita o aprendizado de idiomas de maneira gratuita. É possível o estudo de línguas como inglês, espanhol, francês, alemão, dentre outras. Apresenta-se alguns conceitos de gamificação, como o avanço de níveis e as vidas (corações) (DUOLINGO, sd.). Como nos games, que ao fazer algo de errado ocorre na penalização ou eliminação de uma vida, no Duolingo ao errar uma resposta de uma questão um coração é perdido (DUOLINGO, sd.). A figura 4 ilustra algumas características da aplicação Duolingo.

Figura 4 - Características do Duolingo



Fonte: DUOLINGO, p.1.

O ambiente proposto deseja unir algumas características dos sistemas citados acima, como a possibilidade de encontrar conteúdos de auxílio (como no espaço do CCAA) e acesso a testes de escrita e escuta como no Duolingo. Acredita-se que ao aliar essas funcionalidades já utilizadas em outros sistemas a novas funcionalidades, possa-se criar um ambiente mais adequado para apoiar o ensino de língua inglesa.

## Desenvolvimento e tecnologias de sistemas Web

Para entender-se o que é desenvolvimento pode-se tomar o contexto geral da palavra, que tem por significado ser uma “Série de etapas, acontecimentos, ações, etc. que levam ao surgimento de algo, ou à manifestação em todos os seus aspectos[...]” (FERREIRA, 2001, p. 240). Tomando isso para o meio computacional diz-se então que o desenvolvimento de um sistema se dá pela passagem por todas as etapas da sua concepção até o momento em que o sistema apresente tudo aquilo que lhe foi previsto (PRESSMAN, 2011).

Hirama (2011, p. 3) lista os tipos de atividades presentes no processo de desenvolvimento de sistemas, que são as “[...] atividades técnicas de Engenharia de Sistemas, análise, projeto, codificação e testes”. Essas atividades devem estar presentes e são responsáveis por contemplar o processo de desenvolvimento.

Já para o quesito Web, os sistemas desenvolvidos são conhecidos como *WebApps* (PRESSMAN, 2011). Esses sistemas têm por característica ser acessados e utilizados por meio da rede. Pressman (2011) cita alguns atributos que estão presentes na grande maioria dos *WebApps* que são: o uso intensivo de redes, simultaneidade, carga não previsível, desempenho, disponibilidade, orientadas a dados, sensibilidade no conteúdo, evolução contínua, imediatismo, segurança e estética. Portanto, para se desenvolver soluções web deve se levar em conta os atributos apresentados por Pressman (2011).

### Criptografia e controle de acessos

Criptografia pode ser entendido como o conjunto de métodos e técnicas capazes de modificar um texto legível transformando-o em um texto não legível (MORENO, PEREIRA e CHIARAMONTE, 2005). Isso é possível por meio de um algoritmo, que codifica a mensagem baseado em algum padrão específico. A recuperação das informações originais se dá pelo processo inverso ao da criptografia (MORENO, PEREIRA e CHIARAMONTE, 2005). A figura 5demonstra esse processo de transformação de uma mensagem e o seu retorno a mensagem original.

Figura 5 - Esquema geral para criptografia de um texto



Fonte: MORENO, PEREIRA e CHIARAMONTE, 2005, p.21.

Moreno, Pereira e Chiaramonte (2005) afirmam haver dois meios de se criptografar uma mensagem, por meio de códigos ou por meio de cifras.

Os códigos protegem as informações trocando partes destas por códigos predefinidos. Todas as pessoas autorizadas a ter acesso a uma determinada informação devem conhecer os códigos utilizados. As cifras são técnicas nas quais a informação é cifrada por meio da transposição e/ou substituição das letras da mensagem original. Assim, as pessoas autorizadas podem ter acesso às informações originais conhecendo o processo de cifragem. (MORENO, PEREIRA e CHIARAMONTE, 2005, p. 21).

Existem algoritmos que implementam processos de criptografia, cada uma gera uma *hash* contendo uma quantidade predeterminada de caracteres a partir de um determinado conteúdo (HINZ, 2000). Cada algoritmo vem aliado a uma chave, que é um valor alterável no algoritmo que pode ser modificado para se adequar as necessidades do utilizador (HINZ, 2000). Isso implica que mesmo que se conheça o funcionamento do algoritmo de criptografia uma pessoa só seria capaz de reproduzir o mesmo resultado de outra se ambas usassem a mesma chave no processo de criptografia (HINZ, 2000).

O processo de criptografia é utilizado em auxílio a segurança de senhas de usuário. Os dados sensíveis advindos do usuário, como no caso da senha, somente são salvos na base de dados após serem submetidos a um processo de criptografia, utilizando-se de algum algoritmo em específico (PHP, 2018).

Já o controle de acessos pode ser entendido como a maneira a qual se controla o que cada usuário tem acesso em um sistema, privando o usuário o que não lhe for autorizado, e permitindo somente o que deve se ter acesso de fato (ZAPATER; SUZUKI, 2005). Comumente se utiliza o modelo de controle de acessos baseado em papéis (*Role-based Access Control*) (SANDHU, 1998). Esse modelo utiliza a premissa que cada usuário cumpre um ou mais papéis, como por exemplo um professor, ou um aluno. E a esses papéis são atribuídas permissões. Essas permissões indicarão as funcionalidades que o usuário pode ter acesso no sistema (SANDHU, 1998). Por exemplo, um determinado usuário apresenta o papel de professor, como professor ele pode lançar notas e faltas aos alunos. O sistema deve ser capaz de identificar o papel do usuário e permitir somente o que aquele papel contiver como permissão (SANDHU, 1998).

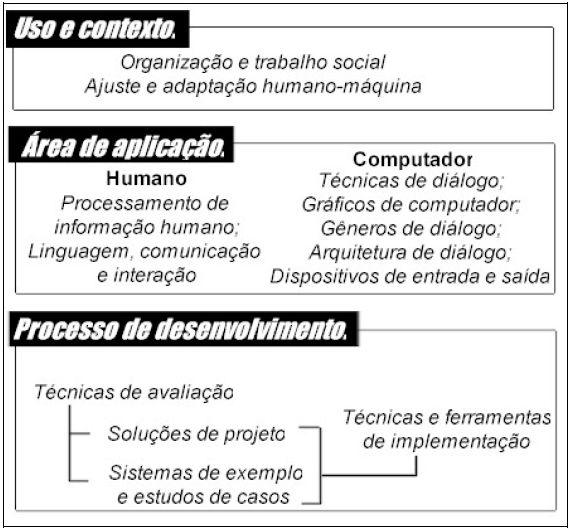
O Laravel que será discutido na seção 5.2.5.5, contém avançados algoritmos de criptografia, como AES-256 e AES-128 (OTWELL, 2018). Assim as senhas dos usuários serão submetidas a um processo de segurança. Além disso o Laravel apoia o controle de acessos por meio de papéis como descrito anteriormente, para garantir que cada usuário só acesse o que lhe for permitido. Pensa-se que com a utilização de tais mecanismos de segurança possa se conseguir uma melhora na segurança dos dados sensíveis dos usuários e dos acessos no ambiente.

### Interação humano computador (IHC)

Softwares são necessários em praticamente todas as áreas de atuação presentes no mercado (BARANAUSKAS; ROCHA, 2003). Porém não há uma solução computacional que resolva todos os problemas de todas as áreas. Então para cada problema surge uma maneira de implementar uma solução que seja utilizada por um nicho de usuários (BARANAUSKAS; ROCHA, 2003). Portanto a maneira que os sistemas interagem com os usuários tendem a ser diferentes em cada solução.

Para buscar compreender e melhorar a maneira que usuários se interagem com sistemas surgiu o campo de estudo denominado Interação Humano Computador (IHC) (BARANAUSKAS; ROCHA, 2003). Baranauskas e Rocha (2003) afirmam que a IHC contempla todo o processo de interação do usuário com o sistema e não somente o processo de design das interfaces. Então entende-se como Interação Humano Computador a: “[...] disciplina preocupada com o design, avaliação e implementação de sistemas computacionais interativos para uso humano e com o estudo dos principais fenômenos ao redor deles.” (BARANAUSKAS e ROCHA, 2003), a figura 6 apresenta algumas informações sobre IHC.

Figura 6 - Interação humano-computador adaptada da descrição do comitê SIGCHI 1992



Fonte: BARANAUSKAS e ROCHA, 2003, p.15.

É importante que se conheça e se aprimore os processos de interação do usuário para que as pessoas consigam efetuar suas funções corretamente em um sistema, uma vez que se compreende o que cada interface significa e o que deve ser feito em cada uma delas.

### Engenharia de Software

O processo de desenvolvimento de software pode ser feito de maneiras diversas. Portanto buscou-se padronizar o processo de desenvolvimento para que se minimizasse os custos e problemas de produção e com uma melhor qualidade final do software, surgiu-se assim a Engenharia de Software (PRESSMAN, 2011). Hirama (2011, p. 7) apresenta o contexto histórico em que a definição do conceito de Engenharia de *Software* surgiu:

O conceito “Engenharia de *Software*” foi cunhado em 1969 por Fritz Bauer em uma conferência patrocinada por um Comitê de Ciência da Organização do Tratado do Atlântico Norte (Otan), no momento em que a chamada crise do software precisava de uma solução para a demanda crescente por software dentro de custo e prazo adequados.

Para entender-se o conceito do que é engenharia de *software* toma-se os significados de engenharia e *software*. A definição de engenharia se dá pela: “Aplicação de conceitos científicos e empíricos, e certas habilitações especificas, à criação de estruturas, dispositivos e processos para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas.” (FERREIRA, 2001, p. 289). A palavra *software* tem-se as seguintes definições segundo Ferreira (2001, p. 682):

1. Em um sistema computacional, o conjunto de componentes informacionais, que não faz parte do equipamento físico e inclui os programas e os dados a eles associados.
2. Qualquer programa ou conjunto de programas de computador.

A partir dos significados pode-se deduzir que engenharia de *software* se trata então da aplicação de conceitos e métodos para apoio ao processo de desenvolvimento de um sistema computacional. Sommerville(2011, p. 5) reforça essa afirmação dizendo que: “Engenharia de *software* é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de *software*, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado”. O *Institute of Eletrical and Eletronics Engineers* (IEEE)(1990, p. 67) porém define como Engenharia de software:

1. A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, a operação e manutenção do software;
2. O estudo de abordagens assim como descrito na etapa 1.

Pressman (2011) caracteriza a engenharia de software como sendo em camadas, conforme descrito na figura 7, sendo as camadas: Ferramentas, métodos, processo e foco na qualidade.

Figura 7 - Camadas da Engenharia de Software



Fonte: PRESSMAN, 2011, p.39.

Quanto ao foco na qualidade Pressman (2011), afirma que todas as disciplinas ligadas a engenharias devem ter foco na qualidade. Para isso conta-se com modelos de qualidade que apoiam a qualidade tanto no processo quanto no produto final. Já em relação ao processo Pressman (2011) ressalta que existem alguns tipos. Um deles é o de engenharia de *software*, que é a “[...] liga que mantém as camadas de tecnologia coesas e possibilita o desenvolvimento de software de forma racional e dentro do prazo.” (PRESSMAN, 2011, p. 39). Outro tipo de processo é o de software que:

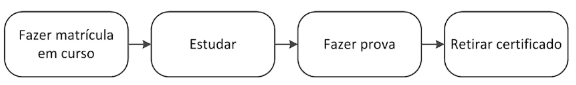
[...] constitui a base para o controle do gerenciamento de projetos de software e estabelece o contexto no qual são aplicados métodos técnicos, são produzidos produtos derivados (modelos, documentos, dados, relatórios, formulários etc), são estabelecidos marcos, a qualidade é garantida e mudanças são geridas de forma apropriada. (PRESSMAN, 2011, p. 40)

Os métodos por sua vez apresentam informações de caráter técnico para auxiliar no processo de desenvolvimento do software (PRESSMAN, 2011). “Os métodos envolvem uma ampla gama de tarefas, que incluem: comunicação, análise de requisitos, modelagem de projeto, construção de programa, testes e suporte.” (PRESSMAN, 2011, p. 40). Já as ferramentas compreendem como o apoio automatizado aos métodos e ao processo (PRESSMAN, 2011). São por exemplos: softwares para modelagem, *Integrated Development Environments* (IDES), etc.

#### Modelagem de processos com o *Business Process Model and Notation* (BPMN)

Um processo pode ser entendido como uma sequência ordenada de atividades, praticadas por um ou mais atores, com propósito de se alcançar um objetivo em específico (CAMPOS, 2014). Então, todo processo oferece algo como resultado da completitude de todas as atividades (CAMPOS, 2014). Um exemplo de processo pode ser visto na figura 8 que representa um processo cujo objetivo seja a conclusão de um determinado curso (CAMPOS, 2014).

Figura 8 - Exemplo de processo



Fonte: CAMPOS, 2014, p.18

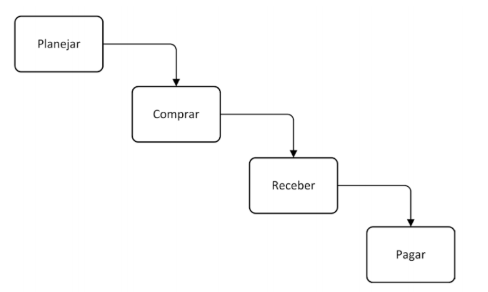
No mundo dos negócios as empresas são compostas por uma série de processos, sejam eles internos ou externos (CAMPOS, 2014). Então se faz necessário a compreensão quanto ao funcionamento desses processos (CAMPOS, 2014). Porém um processo de negócio poderia ser representado de várias maneiras, uma delas por exemplo poderia ser a maneira descrita na Figura 8.

Buscando a padronização da notação de modelagem dos processos de negócios criou-se o *Business Process Model and Notation* (BPMN) que se trata de um padrão de notação para modelagem de processos (SILVER, 2017). O BPMN auxilia nessa padronização de notação pois fornece elementos gráficos para a modelagem de fluxo de um determinado processo (SILVER, 2017). O BPMN indica por meio desses elementos gráficos quais são as atividades que compõem um processo e quem são os atores e executores de cada atividade (SILVER, 2017).

Campos (2014, p. 51) define a notação BPMN como “[...] rica em elementos de representação”. Dentre esses elementos de representação, os considerados básicos a toda modelagem são os conectores, *gateways*, eventos, *data objects*, *pools* e *lanes* (CAMPOS, 2014)*.* Cada um deles será descrito nos parágrafos subsequentes, identificando os seus significados e representações visuais.

Os conectores representam a conexão de atividades dentro de um processo (CAMPOS, 2014). Irão representar a ligação e por conseguinte sequência de atividades do processo. A representação visual dos conectores se dá por linhas indicativas (setas) conforme visto na figura 9, que relata um processo de compra de um determinado item (CAMPOS, 2014).

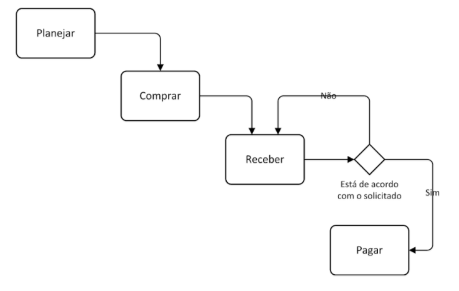
Figura 9 – Exemplo de conectores em um processo de compra



Fonte: CAMPOS, 2014, p. 53.

Já o *gateway* se trata de uma validação necessária que irá determinar o curso do fluxo do processo, alterando ou não o seu fluxo natural (CAMPOS, 2014). A notação utilizada para representar o *gateway* é a figura de um losango, conforme visto na figura 10. A figura 10 relata um processo de compra, porém na atividade de receber, ocorre uma validação de constatação de conformidade com o que foi solicitado, havendo conformidade, paga-se, não havendo retorna-se a atividade anterior (CAMPOS, 2014).

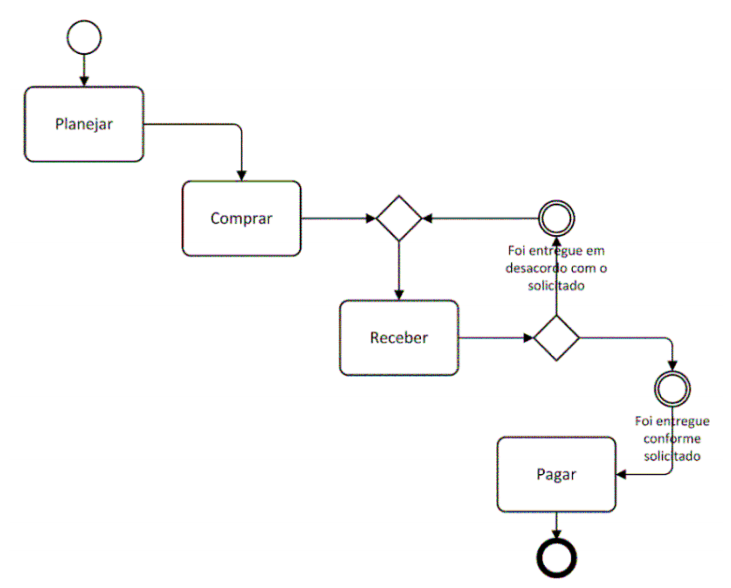
Figura 10 - Exemplo de *gateway* em um processo de compra



Fonte: CAMPOS, 2014, p. 54.

Os eventos para o BPMN servem de indicação de ações externas ao processo como por exemplo indicar o início e o término de um processo (CAMPOS, 2014). A representação gráfica dos eventos é feita por meio de círculos. Os círculos com a borda espessa indicam o fim de um processo. Enquanto os círculos que tem bordas duplas significam eventos intermediários de um processo. Já os círculos com borda menos espessa indicam o início do processo. Um exemplo de utilização dos eventos descritos é o visto na figura 11.

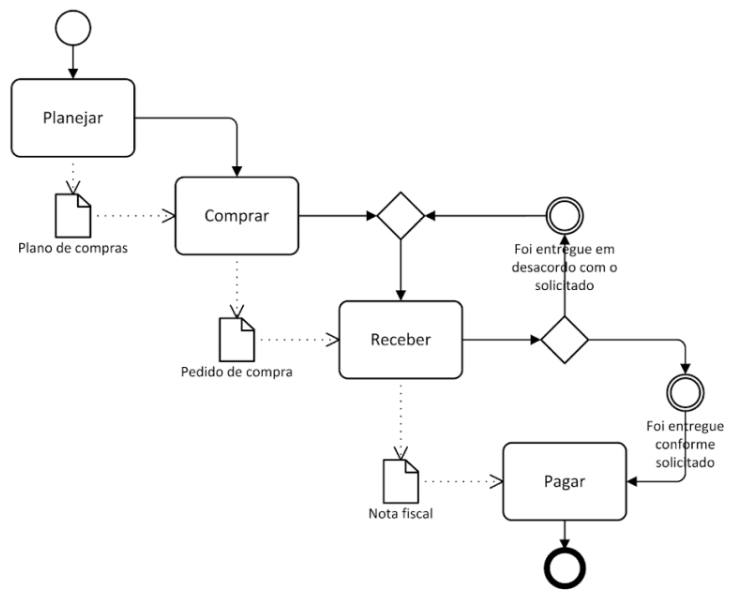
Figura 11 - Exemplo de utilização de eventos em um processo de compra



Fonte: CAMPOS, 2014, p. 56.

*Data objects* por sua vez, representam dados e informações que são usadas e criadas ao longo de um processo (CAMPOS, 2014). Esses dados e informações servem para utilização na atividade subsequente a atividade que criou a informação. É representado graficamente por uma folha de papel com a ponta superior dobrada, ligada por linhas seccionadas em formato de seta, que ligam a atividade que gerou a informação a atividade que irá consumir a informação (CAMPOS, 2014). Exemplos de *data objects* são vistos na figura 12, como o pedido de compra que é gerado ao comprar um produto.

Figura 12 - Exemplo da utilização de *data objects* em um processo de compra

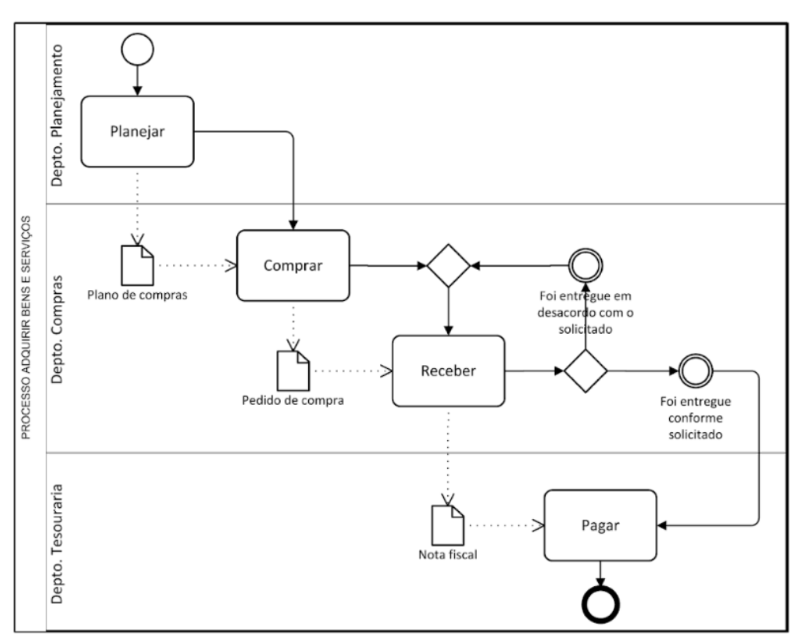


Fonte: CAMPOS, 2014, p. 58.

As *Pools*, ou piscinas em português, representam um processo como um todo, englobando todas as atividades de um processo (CAMPOS, 2014). São representados graficamente por um retângulo contendo o nome do processo.

*Lanes*, ou raias em português, representam os atores participantes de um processo, sendo esses atores pessoas, departamentos, setores, cargos, etc (CAMPOS, 2014). As raias são representadas por um retângulo contendo o nome do ator. As raias e as demais atividades do processo devem estar englobadas dentro de uma piscina, conforme visto na figura 13.

Figura 13 - Exemplo de utilização de piscinas e raias em um processo de compra



Fonte: CAMPOS, 2014, p. 59.

Acredita-se que a utilização de modelagem de processos neste trabalho servirá de documentação ao desenvolvedor e ao cliente, ajudando-os a compreender um determinado processo.

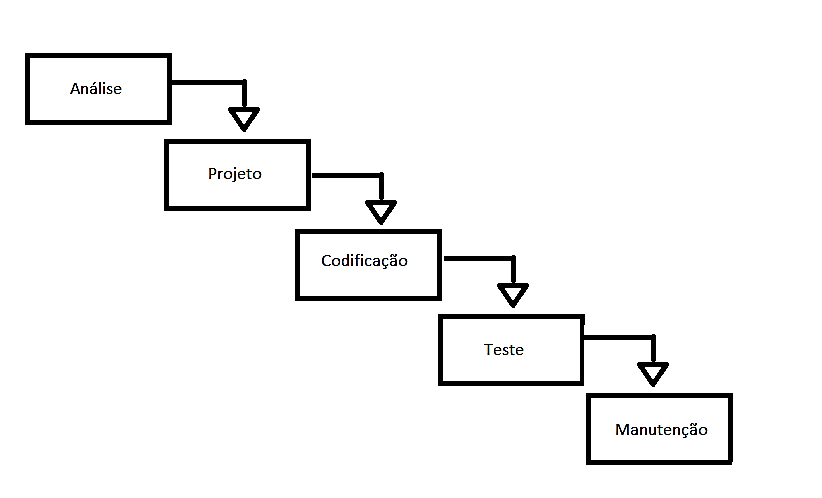
#### Metodologias de desenvolvimento

No desenvolvimento de sistemas há algumas metodologias que visam auxiliar no processo de desenvolvimento. Também chamado de processos, são importantes pois dão um norte aos participantes da equipe, demonstrando a maneira de como deve ser feito para se alcançar os objetivos do software (HIRAMA, 2011). Alguns autores como Teles (2014), classificam as metodologias de desenvolvimento em Desenvolvimento Tradicional (modelo Cascata), e Desenvolvimento Ágil.

#### Modelo Cascata

O modelo em cascata prega que o desenvolvimento do software deve ser feito por uma série de etapas lineares e que cada etapa depende do término da anterior (TELES, 2014). As etapas a serem seguidas variam de autor para autor, mas as seguintes etapas são as de base de um processo em cascata: Análise, Projeto, Codificação, Teste e Manutenção (HIRAMA, 2011). A figura 14 ilustra o modelo em cascata, e a representação de uma seta em uma atividade representa a dependência de uma atividade anterior.

Figura 14 - Representação da modelo em cascata



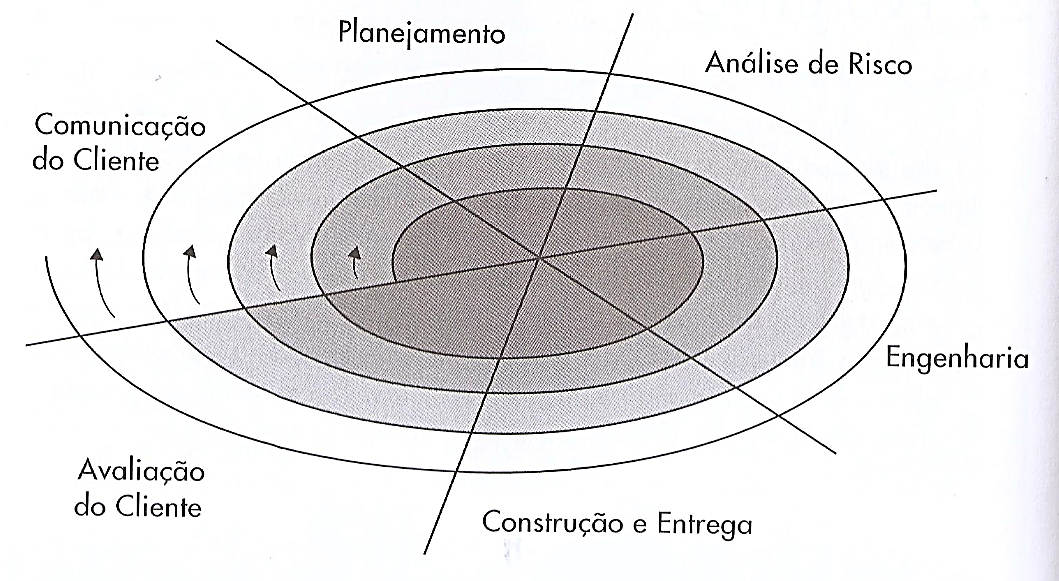
Fonte: ADAPTADA de HIRAMA, 2011, p.26.

Teles (2014, p. 35) afirma que esse modelo é “reconhecidamente ineficaz”, porém ainda é um dos mais utilizados no desenvolvimento de software. Algumas críticas acerca desse modelo são: linearidade excessiva, o foco na execução, falta de feedback do cliente e a dificuldade de se adequar a mudanças (TELES, 2014). Teles (2014) compara esse modelo de desenvolvimento ao modelo de fabricação industrial, havendo trabalhadores e cada um com um papel específico. Cada um deles sendo especialista em fazer uma determinada tarefa (TELES, 2014). Essas tarefas que devem ser simples e determinísticas a ponto de não haver necessidade de pensar no que fazer, somente executa-las (TELES, 2014). O desenvolvimento tradicional busca alcançar essa simplicidade de tarefas para que se diminua erros e o tempo de desenvolvimento (TELES, 2014). Porém esquece-se que o desenvolvimento é um processo de trabalho intelectual, ideias podem surgir a todo momento, assim como uma solução melhor para um problema (TELES, 2014). Tendo em vista o que foi apresentado, escolheu-se por não se utilizar o modelo em cascata no desenvolvimento deste trabalho.

#### Metodologia Ágil

Metodologia ágil se trata de um movimento que buscou mudar a maneira como os projetos de software eram executados (HIRAMA, 2011). Apresentou-se como premissa o aumento na velocidade de desenvolvimento de sistemas sem que houvesse uma eventual perda de qualidade (HIRAMA, 2011). E buscou-se diferenciar das metodologias tradicionais como o modelo em cascata. Uma das diferenças se dá pela filosofia iterativa (ou espiral) e não mais linear como no modelo em cascata (HIRAMA, 2011). Esse modo iterativo prega o desenvolvimento de software como algo constante. A cada iteração o cliente recebe uma parte do sistema podendo opinar sobre melhorias ou eventuais erros de modelagem (TELES, 2014). A cada iteração então recebe-se do cliente o seu *feedback* acerca do que foi implementado, além do cliente ter a possibilidade de utilizar o sistema sem que este esteja completamente terminado (HIRAMA, 2011). Ao final de todas as iterações tem-se o sistema completo e operante, uma vez que o sistema foi validado ao fim de cada iteração (HIRAMA, 2011). A figura 15 demonstra um exemplo de representação do modelo iterativo ou espiral.

Figura 15 - Modelo em espiral



Fonte: HIRAMA, 2011, p39.

O XP, que será descrito suas características e peculiaridades na seção 5.2.3.5, e utilizado no decorrer deste trabalho, é um exemplo de uma metodologia ágil.

#### *Extreme Programming* (XP)

O *Extreme Programming* (XP), se trata de uma metodologia de desenvolvimento ágil, que busca aliar agilidade no desenvolvimento, com qualidade no produto final (TELES, 2014). O XP pode então, ser descrito como “[...] um processo de desenvolvimento que busca garantir que o cliente receba o máximo de valor de cada dia de trabalho da equipe de desenvolvimento” (TELES, 2014, p. 24). Mas ao contrário do modelo em cascata, o XP é uma metodologia que apoia as mudanças (TELES, 2014). O XP entende-se que o cliente pode não ter certeza sobre o que quer logo de início, mas que vai descobrindo assim que tem contato com um protótipo do software (TELES, 2014). Assim ao ter contato direto com o software, o cliente é capaz de verificar o que deve ser melhorado ou mudado (TELES, 2014). Além disso os desenvolvedores podem identificar ainda o que o cliente considera de mais importante (TELES, 2014). Portanto é de suma importância ao XP o *feedback* do cliente acerca do que foi implementado.

O *feedback* é um dos quatro valores fundamentais do XP, junto com a comunicação, a simplicidade e a coragem (TELES, 2014). Portanto o *feedback* diz respeito a capacidade de se receber informação de volta do cliente. E através dessas informações ser capaz de entender as especificações e requisitos necessários para o desenvolvimento, conhecendo assim as necessidades e anseios do cliente, em vista que o cliente estará em contato direto com o software (TELES, 2014).

A comunicação diz respeito a equipe constantemente estar em contato direto com o cliente (TELES, 2014). Pois estabelecendo contato frente a frente, a equipe pode retirar potenciais dúvidas ou conhecer certos comportamentos requeridos no software (TELES, 2014). A diferença de se ter um contato frente a frente, para uma documentação escrita, é que a documentação não exprime emoções ou sentimentos, já na comunicação direta a linguagem corporal toma forma, e o sentimento é expressado de maneira mais eficaz como afirma Teles (2014).

A simplicidade consiste em desenvolver código o mais simples possível, somente resolvendo o necessário para o cliente (TELES, 2014). Teles (2014) explica que isso se dá, pois, um código mais simples tende a ser mais fácil de ser alterado em caso de necessidade de alteração.

A coragem diz respeito a equipe ter coragem de enfrentar algo novo. Para desenvolver usando o XP alguns padrões devem ser deixados de lado (TELES, 2014). Por exemplo, o XP prega que o desenvolvimento deve ser feito em par, ou seja, dois desenvolvedores numa mesma máquina, desenvolvendo o mesmo código. Porém pode-se acreditar que isso seja algo custoso ou dispendioso, mas Teles (2014) afirma que se deve ter coragem para esquecer-se de premissas tradicionais para acreditar e assumir no uso dessa metodologia.

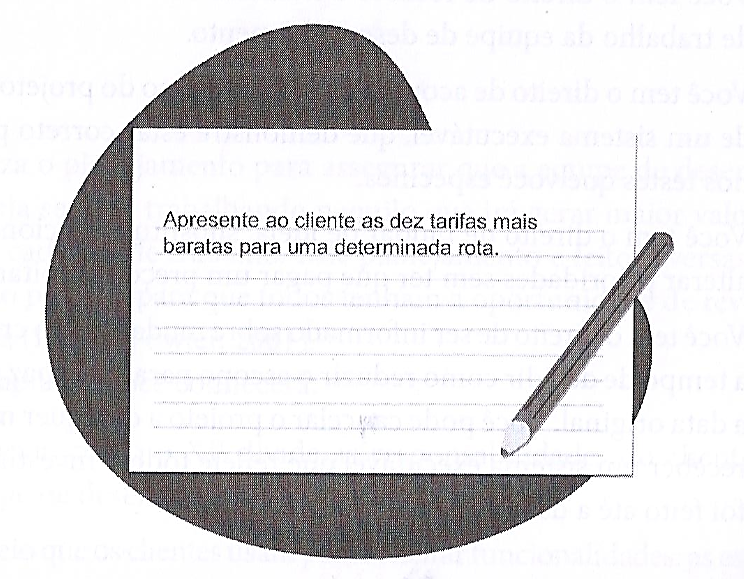
Quanto a programação em par citada anteriormente, ela é uma das práticas do XP, junto com o jogo do planejamento, o cliente presente, o *stand up meeting*, desenvolvimento guiado a testes, *refactoring*, código coletivo, código padronizado, design simples, metáfora, ritmo sustentável, integração contínua e releases curtos (TELES, 2014). A programação em par é característica que levanta mais dúvida quanto ao uso do XP (TELES, 2014). Essa dúvida ocorre pelo fato de que se pensa que com a utilização de dois programadores fazendo um único trabalho, estaria se desperdiçando um programador que poderia estar desenvolvendo em outra máquina (TELES, 2014). Porém ao se utilizar desse desenvolvimento em par espera-se que a qualidade final do código seja melhor, pois enquanto um programa o outro verifica e valida o código (TELES, 2014). Além disso pode-se aliar duas soluções distintas para um problema, cada um com uma experiencia e maneira de solucionar algo, com o intutito de encontrar uma solução considerada melhor (TELES, 2014).

Além da programação em par, destaca-se o desenvolvimento orientado a testes, que se trata de se testar primeiro antes de codificar de fato (HIRAMA, 2011). Pensa-se que ao se produzir o teste primeiro o desenvolvedor consiga compreender melhor o que será desenvolvido (TELES, 2014). Teles (2014) defende que o ato de testar deve ser uma prática constante no processo de desenvolvimento, pois assim assume-se que os erros encontrados serão corrigidos ou que através dos testes possa haver a validação do que foi requisitado.

Outro fator que se destaca no XP são as estórias de usuários que são pequenos trechos escritos em cartões ao qual o cliente descreve como quer que seja uma determinada funcionalidade do sistema (TELES, 2014). Toda estória de usuário se transforma em um requisito e, por conseguinte uma funcionalidade da aplicação desenvolvida. Um exemplo de estória de usuário é o presente na figura 16.

Como o XP se trata de uma metodologia iterativa, em certos períodos de tempo, tem-se entregas de porções do software para que o cliente possa utilizar o sistema sem que o mesmo esteja totalmente pronto. Essas entregas são conhecidas como *releases* (TELES, 2014). Ainda segundo Teles (2014) os *releases* no XP geralmente duram cerca de 2 meses, por isso são considerados *releases* curtos, pois o tempo de entrega entre os *releases* é pequeno. Após o término de um *release* faz-se uma nova entrevista com o cliente e coleta as estórias que serão desenvolvidas no novo *release* (TELES, 2014). Cada *release* é composto por iterações, uma iteração pode ser entendida como “[...] um pequeno espaço de tempo dedicado para a implementação de um conjunto de estórias” (TELES, 2014, p. 87). Segundo Teles (2014) o tempo de uma iteração pode variar cerca de uma a três semanas, mas que geralmente o tempo de duas semanas é o mais utilizado em empresas com o XP.

Figura 16 - Exemplo de uma estória de usuário



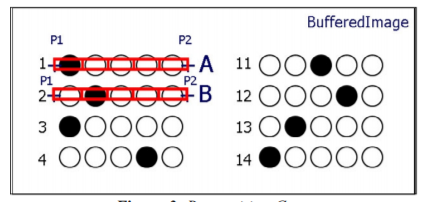
Fonte: TELES, 2014, p.78.

Pretende-se neste trabalho fazer utilização da metodologia de desenvolvimento denominada XP. Hirama (2011) afirma que o XP pode não ser recomendado a grandes projetos de software. Porém pretende-se fazer utilização desta metodologia a ponto de se chegar a uma avaliação pessoal acerca da experiência de uso do XP. Espera-se que com essa avaliação, possa-se auxiliar outros desenvolvedores na escolha, ou não, da metodologia XP. Já que eventuais dificuldades e melhorias no processo de desenvolvimento serão constatadas e relatadas ao decorrer deste trabalho. Deve-se ressaltar que não será possível a aplicação da programação em par, que é uma das práticas do XP, pois a execução do projeto será feita por uma pessoa.

### *Optical Mark Recognition* (OMR)

*Optical Mark Recognition* (OMR), se trata da capacidade de transformar uma marcação em um espaço predeterminado, em dados computacionais (SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017). É utilizado em correções de folhas de respostas em vestibulares e provas escolares. E seu processo se dá pela inserção de um formulário marcado, e através de um scanner, scaneia-se o formulário e a imagem gerada é transferida a um computador que contém o algoritmo de leitura de marcação (SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017). O algoritmo então é responsável por reconhecer os locais onde há ou não marcação e transformar aquela marcação em um dado (SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017). Por exemplo, tem-se uma folha de respostas com as alternativas de A a E, o usuário marca a letra A. O algoritmo irá interpretar que se marcou a alternativa A e não as demais conforme descrito na figura 17. A interpretação se dá de maneira que o algoritmo verifica a incidência de cor diferente da branca em uma determinada região (SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017). Caso a incidência dessa cor seja muito alta, verifica-se então que se marcou aquela alternativa (SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017). A figura 17 se trata de um exemplo de um formulário de marcação, com algumas questões marcadas, nota-se que através da posição de marcação, é possível identificar a alternativa marcada (questões 1 e 2).

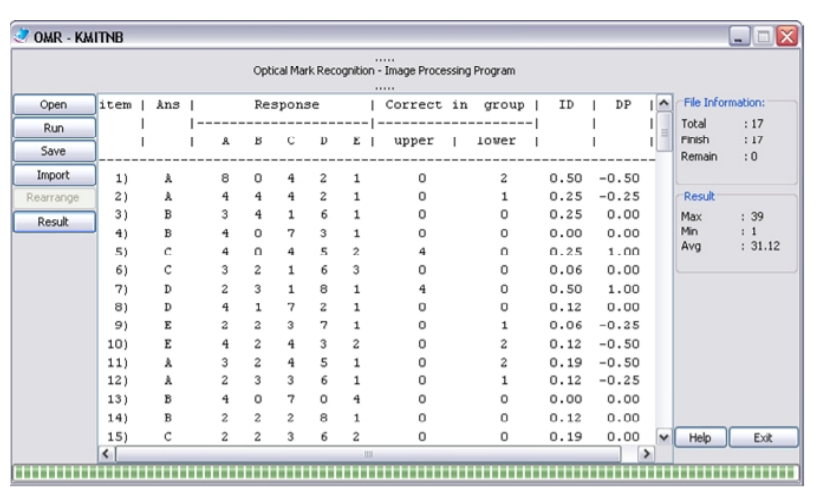
Figura 17 - Exemplo de um formulário de marcação marcado



Fonte: SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017, p.71.

Já a figura 18 se dá por um software desenvolvido por Saengtongsrikamon, Meesad e Sunantha (2017) que é responsável fazer a leitura da marcação e transforma-la em dado.

Figura 18 - Exemplo de um software que implementa OMR



Fonte: SAENGTONGSRIKAMON, MEESAD e SUNANTHA, 2017, p.73.

Pretende-se utilizar de um algoritmo de OMR neste trabalho, com o objetivo de proporcionar uma maneira automatizada de correção de provas feitas em papel. Além disso para o desenvolvimento deste trabalho, não será necessário equipamento além do scanner para leitura das folhas de resposta.

### Tecnologias para desenvolvimento WEB

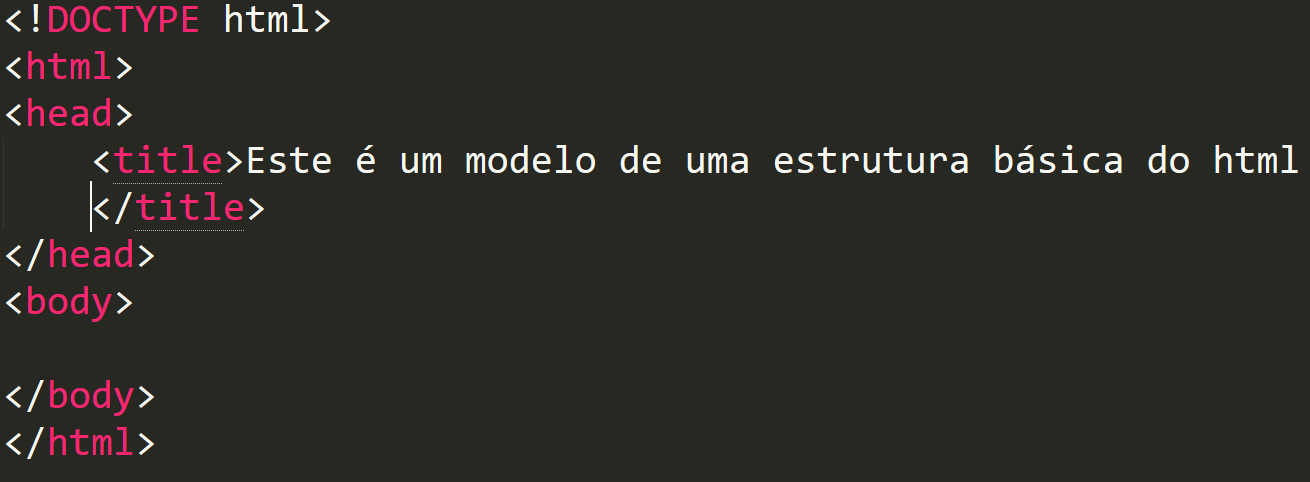
O desenvolvimento de soluções *web* envolve uma série de tecnologias. Cada uma com uma determinada função e utilidade. As tecnologias Web podem ser divididas em *Front-End* e *Back-End*. Tecnologias *Front-End* são aquelas que estão em contato direto com o usuário, como por exemplo o HTML (ROBBINS, 2013). Já as tecnologias *Back-End* são as que estão em contato direto com o servidor e não mantém vínculo direto com o usuário final, um exemplo é a linguagem de *scripting* PHP (PHP, 2018). Todas essas tecnologias trabalham em conjunto para oferecer uma melhor experiência de uso para o usuário (ROBBINS, 2013). Algumas dessas tecnologias serão abordadas nas seções seguintes deste trabalho.

#### *Hyper Text Markup Language* (HTML)

*Hyper Text Markup Language,* HTML, se trata de uma linguagem de marcação capaz de transformar documentos de texto em páginas web (ROBBINS, 2013). Seu objetivo é prover uma descrição semântica do conteúdo e formar uma estrutura para o documento (ROBBINS, 2013). A delimitação de conteúdo no HTML é feita por meio de *tags* que vão indicar onde um determinado conteúdo começa e termina. A figura 19 consiste na estrutura de *tags* básica de um documento HTML.

A *tag* ‘html’ indica onde se inicia e onde se termina o documento HTML. A *tag* ‘*head*’ representa o cabeçalho do documento, dentro dela há uma outra *tag* chamada ‘*title*’ indica o título da página. Já a ‘*body’* representa o corpo do documento HTML (ROBBINS, 2013).

Figura 19 - Estrutura básica do HTML



Fonte: PRÓPRIA, utilizando o SublimeText 4.

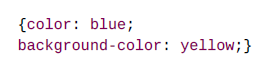
O HTML está na sua versão 5, essa versão trouxe uma série de mudanças acerca das *tags* a fim de deixar o documento HTML mais semântico (ROBBINS, 2013). Com essas mudanças deixou-se a leitura de documentos e páginas mais fácil por caracterizar melhor os elementos da página (ROBBINS, 2013).

Por acreditar que o HTML seja uma maneira simples e concisa de marcação, além do Laravel reconhecer HTML, optou-se então pelo seu uso no desenvolvimento deste trabalho.

#### *Cascading Style Sheets* (CSS)

*Cascading Style Sheets,* com acrônimo (CSS), é uma linguagem criada com o objetivo de cuidar do layout e do estilo de uma página (CAELUM, 2018). Além de estilização dos elementos, com o CSS também é possível acrescentar efeitos nos elementos como desaparecer da tela por exemplo (MCFARLAND, 2013). E apesar de o CSS colaborar com o HTML, ele não se trata de HTML. Enquanto o HTML provê uma organização estrutural de um documento, o CSS trabalha em colaboração com o *browser* com a finalidade de fazer com que uma página HTML contenha estilo próprio (MCFARLAND, 2013). A sintaxe do CSS compreende-se em: ‘propriedade: valor;’ (CAELUM, 2018) conforme descrito na figura 20.

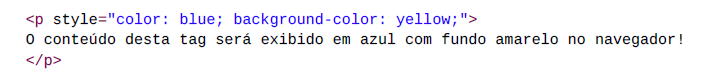
Figura 20 - Sintaxe CSS



Fonte: CAELUM, 2018, p.21.

Quanto a inclusão do CSS em elementos HTML, ela pode ser feita de três maneiras. A primeira delas é inserindo o código CSS diretamente na *tag* HTML do elemento, utilizando-se do atributo ‘*style*’ da *tag* em questão conforme descrito na figura 21.

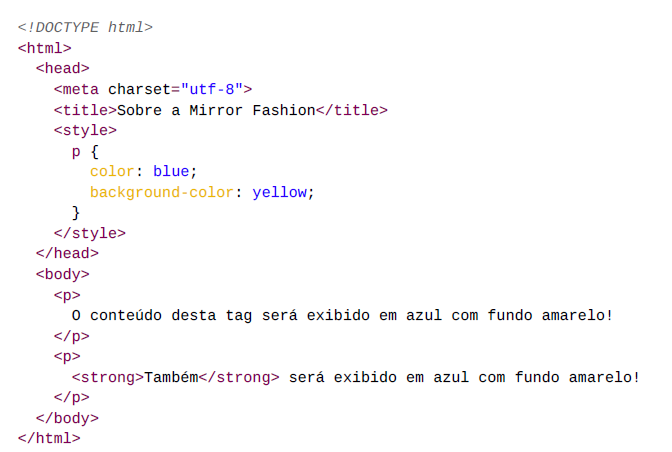
Figura 21 - CSS inserido diretamente na tag HTML



Fonte: CAELUM, 2018, p.21.

Outra maneira de se inserir o CSS é por meio de uma *tag* HTML denominada ‘*style*’, onde o seletor do elemento deve ser referenciado como na figura 22. O seletor pode ser entendido como uma referência aos elementos de uma página (CAELUM, 2018). As principais maneiras de se referenciar elementos são pelo atributo id (identificado com o símbolo #), pelo atributo *class* (identificado com o ponto final), ou pela *tag* de um elemento (CAELUM, 2018). Na sintaxe CSS dentro da *tag* ‘*style’* deve-se indicar o seletor, no caso da figura 22 o seletor é uma *tag* de parágrafo (p), e o que vier dentro das chaves serão seus estilos (CAELUM, 2018).

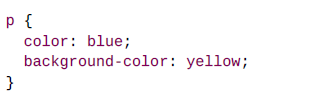
Figura 22 - CSS inserido através da *tag style*



Fonte: CAELUM, 2018, p.22.

A terceira maneira de se estilizar os elementos HTML se dá por meio de um arquivo separado como visto na figura 23, de maneira que o código CSS fique separado da codificação feita em HTML (CAELUM, 2018). Assim como no exemplo da figura 22 a sintaxe CSS deve se indicar o seletor, no caso da figura o seletor é uma *tag* de parágrafo (p), e o que vier dentro das chaves serão seus estilos (CAELUM, 2018).

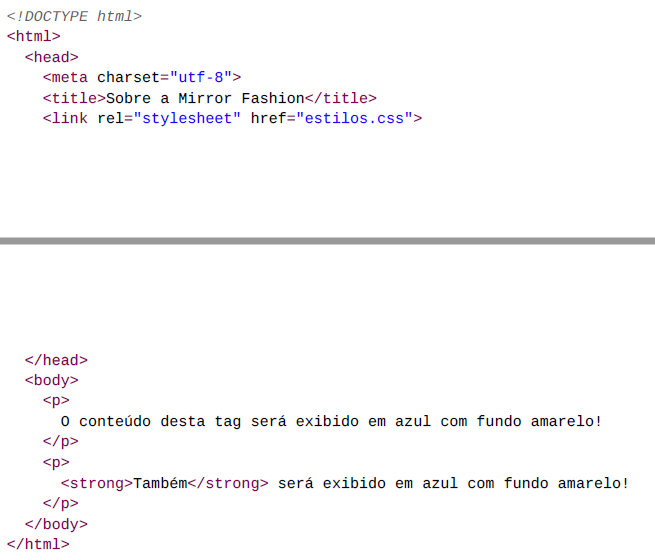
Figura 23 - CSS contido no arquivo estilos.css



Fonte: CAELUM, 2018, p.23.

A única diferença é que para utilizar a estilização contida no arquivo CSS no documento HTML deve-se utilizar no código HTML uma *tag* chamada ‘*link*’ (CAELUM, 2018). A *tag* ‘*link’* é responsável por carregar e possibilitar o uso dos estilos a partir de um arquivo externo (CAELUM, 2018). Para indicar a localização do arquivo CSS, deve-se utilizar um atributo da *tag* ‘*link’* chamado ‘href’, e nesse atributo indicar o caminho até o arquivo de estilos, conforme demonstrado na figura 24 (CAELUM, 2018). O estilo criado na figura 23 é utilizado na *tag* <p> do documento HTML da figura 24.

Figura 24 - CSS inserido através de um arquivo externo



Fonte: CAELUM, 2018, p.22.

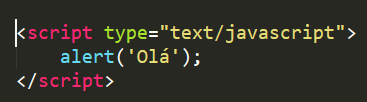
Com o uso do CSS no ambiente proposto espera-se contemplar os usuários com páginas mais bonitas e agradáveis para que a experiencia de uso seja a melhor possível.

#### JavaScript

O JavaScript se trata de uma linguagem de *scripting* (CAELUM, 2018). Que tem por característica ser uma linguagem interpretada, ou seja, não depende de compilação para sua execução (CAELUM, 2018). Além disso se trata de uma linguagem de tipagem dinâmica, ou seja, as variáveis não estão diretamente ligadas a somente um tipo, como por exemplo int, *double*, mas variam de acordo com o dado inserido na variável (CROCKFORD, 2008).

O JavaScript é comumente utilizado em *web* browsers, que tem o papel de interpretar o código em JavaScript e gerar uma saída (CAELUM, 2018). O intuito da linguagem é prover dinamicidade aos conteúdos, uma vez que o HTML e o CSS provêm conteúdo estático a um documento web (CAELUM, 2018). Para se utilizar o JavaScript em páginas HTML é necessário que se faça uso de uma *tag* específica chamada script conforme descrito na Figura 25 e na Figura 26. É possível ainda escrever os scripts de duas maneiras. A primeira delas é inserindo o script diretamente na página HTML (Figura 25).

Figura 25 - Exemplo de uso do JavaScript diretamente no HTML



Fonte: PRÓPRIA, utilizando o SublimeText 4.

A outra maneira de se utilizar o JavaScript em páginas HTML é escrevendo o script em um arquivo separado e anexá-lo ao documento HTML como visto na Figura 26. Acredita-se que assim possa separar melhor HTML de JavaScript e deixando mais coesa a leitura de um documento HTML (CAELUM, 2018). Porém em algumas ocasiões é necessário inserir o script diretamente no documento HTML.

Figura 26 - Exemplo de uso do JavaScript por meio de um arquivo externo



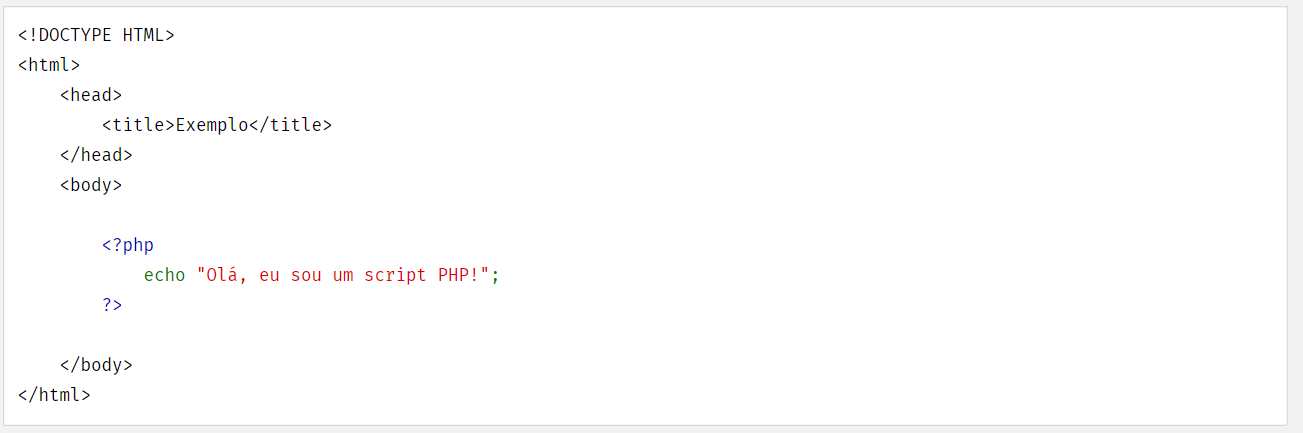
Fonte: PRÓPRIA, utilizando o SublimeText 4.

Portanto, visando alcançar dinamicidade nos documentos HTML, optou-se pela utilização do JavaScript. Pensa-se que assim a experiência final do usuário com o sistema possa ser mais dinâmica.

#### *Hypertext PreProcessor* (PHP)

O *Hypertext* *PreProcessor* (PHP) é uma linguagem de script usada principalmente no contexto web para fazer a manipulação de páginas HTML (PHP, 2018). Para seu uso é necessário a abertura e fechamento de uma *tag* PHP conforme descrito na figura 27.

Figura 27 - Exemplo de código PHP em página HTML



Fonte: PHP, 2018, p.1.

O PHP tem por característica ser uma linguagem interpretada *Server-Side*, ou seja,executada no lado do servidor (SKLAR, 2016). Assim, uma aplicação utilizando o PHP pode ser disponibilizada para uma série de pessoas através de uma *Uniform Resource Locator* (URL) (SKLAR, 2016). Sklar (2016) explica que se detendo da URL o usuário faz uma solicitação ao servidor que interpreta o documento PHP e retorna o resultado HTML que é entendível pelo Browser. Os servidores web em que geralmente o PHP é utilizado são os servidores Apache ou Ngix (LOCKHART, 2015).

A linguagem apresenta uma série de recursos e aditivos, como por exemplo manipulação de imagens, recursos de abstração de banco de dados, além de pacotes dos mais variados fins criados pela comunidade e que auxiliam na popularização da linguagem e seu uso para diversos fins (SKLAR, 2016).

A utilização do PHP neste trabalho se dará por meio do *Framework* Laravel que será discutido na seção seguinte.

#### *Framework* Laravel

O Laravel é um *Framework* baseado na estratégia de desenvolvimento *Model*, *View* e *Controller* (MVC) e feito sobre linguagem PHP, que tem como objetivo facilitar e melhorar o processo de desenvolvimento (STAUFFER, 2017). Stauffer (2017, p. 22) apresenta o porquê da utilização de frameworks para desenvolvimento dizendo:

Frameworks como o Laravel – e o Symfony, Sliex, Lumen e Slim – pré-empacotam um conjunto de componentes de terceiros com características personalizadas do framework, como arquivos de configuração, provedores de serviço, estruturas de diretório prescritas e carregadores de aplicativos. Logo, o benefício de usar um framework em geral é o de que alguém tomou decisões por você não só sobre componentes individuas, mas também sobre como eles devem ser integrados.

Portanto pode-se afirmar que um *framework* como o Laravel simplifica alguns processos comuns a sistemas (como por exemplo o processo de autenticação) por meio de pacotes prontos. Isso faz com que o desenvolvedor se concentre em processos específicos da lógica de negócio ao qual seu sistema pertence, agilizando o desenvolvimento (STAUFFER, 2017).

O Laravel demonstra como valores o aumento da velocidade de desenvolvimento e a satisfação do desenvolvedor (STAUFFER, 2017). Para isso dispõe de uma série de ferramentas que auxiliam em diversos processos no desenvolvimento, como por exemplo o processo de autenticação, o envio de e-mails, processos de banco de dados (criação de tabelas, transações e consultas), entre outros (STAUFFER, 2017). Tendo em vista a quantidade de recursos disponíveis pelo Laravel, uma comunidade que auxilia em momentos de dúvida, além do conhecimento prévio, decidiu-se pela sua utilização neste trabalho de conclusão de curso.

### Sistema de Gerenciamento de Banco de Dados (MySQL)

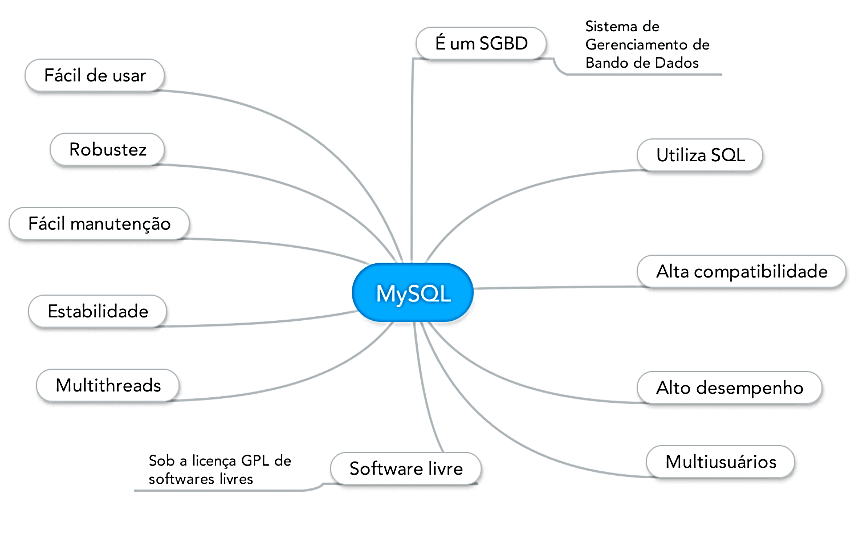
Bancos de dados estão presentes em todos os lugares em que se tem uso de computadores, desde a gestão de dados de uma multinacional, até numa farmácia de esquina (ELMASRI; NAVATHE, 2011). Mas o que de fato são bancos de dados? Elmasri e Navathe (2011, p. 3) definem como banco de dados “[...] uma coleção de dados relacionados” que possuem as seguintes características (ELMASRI; NAVATHE, 2011):

* Apresentar algum aspecto do mundo real;
* Ser um conjunto de dados que apresentem algum sentindo inerente;
* Projetado, construído e populado com dados com um fim em específico.

Conhecendo sobre o que se trata um banco de dados toma-se o conceito de Sistemas de Gerenciamento de Banco de Dados (SGBD) como “um conjunto de dados associados a um conjunto de programas para acesso a esses dados” (SILBERCHATZ, KORTH e SUDARSHAN, 1999). Portanto, para se gerir os dados que estão presentes em um banco de dados é necessário a atuação de um SGBD. O SGBD que será responsável então pela “[...] definição das estruturas de armazenamento das informações e a definição dos mecanismos para manipulação dessas informações.” (SILBERCHATZ, KORTH e SUDARSHAN, 1999, p. 1).

Dentre os SGBS, um que se destaca é o MySQL (CARVALHO, 2015). O MySQL é um SGBD gratuito e com base de licença *open source*, com ferramentas de desenvolvimento robustas e de qualidade (CARVALHO, 2015). A figura 28 representa um mapa mental que contém as principais características do MySQL.

Figura 28 - Características do MySQL



Fonte: CARVALHO, 2015, p.3.

Com o MySQL é possível criar, editar e excluir dados através de sentenças na linguagem SQL. Além disso como visto na figura 28, apresenta características como possibilidade de uso de mais de um usuário, estabilidade, etc. Diversas linguagens de programação oferecem integração com o MySQL, e uma delas é o PHP e, por conseguinte o Laravel (STAUFFER, 2017). Um dos motivos da escolha do MySQL como SGBD deste trabalho foi o fato de ser gratuito e a fácil integração com o Laravel, sem se esquecer das características que auxiliam em qualidade e que foram descritas na Figura 28.

# PROCEDIMENTOs METODOLÓGICOs

Marconi e Lakatos (2003, p. 83) definem método como sendo “o conjunto de atividades sistemáticas e racionais que, com maior segurança e economia, permite alcançar o objetivo [...], traçando o caminho a ser seguido, detectando erros e auxiliando as decisões do cientista”. Metodologia, porém, tem um significado diferente de método e pode ser entendida como “o estudo dos métodos” (WAZLAWICK, 2009, p. 39). Assim Wazlawick (2009) afirma que um trabalho científico deve ser embasado em um método e não em uma metodologia. Ou seja, a pesquisa cientifica então deve ser concebida através de métodos, por servirem de base e de auxílio no alcance dos resultados, ou objetivos, desejados.

## CLASSIFICAÇÃO DA PESQUISA/MÉTODO

A pesquisa se refere ao “procedimento formal, com método de pensamento reflexivo, que requer um tratamento científico e se constitui no caminho para conhecer a realidade ou para descobrir verdades parciais” (MARCONI e LAKATOS, 2003, p. 155). Segundo (WAZLAWICK, 2009) as pesquisas realizadas em subáreas da computação podem ser classificadas em três maneiras distintas sendo: Pesquisas formais, pesquisas empíricas, ou pesquisas exploratórias. Para a confecção deste trabalho optou-se pela utilização da pesquisa exploratória, os motivos dessa escolha, bem como o que se define como pesquisa exploratória serão abordados na próxima subseção.

### Pesquisa Exploratória

Uma pesquisa exploratória é aquela que tem por objetivo “proporcionar maior familiaridade com o problema, como vistas a torna-lo mais explícito ou a constituir hipóteses. [...] estas pesquisas têm por objetivo principal o aprimoramento de idéias ou a descoberta de intuições.” (GIL, 2002, p. 41). Esse tipo de pesquisa então, serve como apoio ao entendimento do problema a ser resolvido e intuições a cerca de uma solução. Wazlawick (2009, p. 14) reforça isso dizendo que nesse tipo de pesquisa “não se consegue provar uma teoria nem apresentar resultados estatisticamente aceitos.”

Acredita-se que este projeto científico seja classificado como uma pesquisa exploratória pelo fato de que não propõe uma solução considerada a melhor possível em termos estatísticos ou de caráter inovador. Porém se trata de uma solução que une características de outras soluções já existentes, mas fazendo-o de maneira diferente a se adequar a realidade da empresa que será usada como base (no caso a escola de inglês).

## TÉCNICAS DE PESQUISA

Técnica se refere a “um conjunto de preceitos ou processos de que se serve uma ciência ou arte; é a habilidade para usar esses preceitos ou normas, a parte prática.” (MARCONI e LAKATOS, 2003, p. 174). Ou seja, a maneira que se dará a aplicação prática da pesquisa. Para este trabalho escolheu-se como técnicas de pesquisa a pesquisa bibliográfica, a observação e a entrevista.

### Pesquisa Bibliográfica

Gil (2002, p. 44) define a pesquisa bibliográfica como aquela que é “[...] desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos”. Já Marconi e Lakatos (2003, p. 183) estabelecem como pesquisa bibliográfica aquela que:

[...] abrange toda bibliografia já tornada pública em relação ao tema de estudo, desde publicações avulsas, boletins, jornais, revistas, livros, pesquisas, monografias, teses, material cartográfico etc., até meios de comunicação orais: rádio, gravações em fita magnética e audiovisuais; filmes e televisão.

Todos esses meios de informação que contenham informações relevantes devem ser checados e utilizados no trabalho a fim de aumentar os conhecimentos do pesquisador, tendo em vista que outros autores já escreveram sobre algo semelhante. Ou seja, é necessário pois além do conhecimento obtido, evita retrabalho caso se encontre uma solução semelhante a solução proposta pelo pesquisador, assim só se baseia ou se modifica a ideia de pesquisa proposta. Como afirma Marconi e Lakatos (2003) a pesquisa bibliográfica também pode servir para a constatação de uma ideia nova, já que consultando-se de materiais pode-se verificar ou não a existência de uma solução semelhante à proposta pelo pesquisador.

Gil (2002, p. 45) apresenta como sendo a principal vantagem da pesquisa bibliográfica o fato de que ela permite “[...] ao investigador a cobertura de uma gama de fenômenos muito mais ampla do que aquela que poderia pesquisar diretamente”. Porém Gil (2002) afirma que em contrapartida essa vantagem pode diminuir a qualidade final da pesquisa, pois podem apresentar informações erradas ou equivocadas e que podem levar o pesquisador a cometer os mesmos erros apresentados nas fontes bibliográficas. Portanto é necessário que se filtre e identifique a fonte de pesquisa e quanto a veracidade das informações dispostas.

Para a pesquisa bibliográfica deste trabalho, pretende-se fazer uso de livros, artigos, apostilas e periódicos, sejam eles em meio digital (dispostos na *web*), ou o material físico.

### Observação

A observação pode ser entendida como uma “[...] técnica de coleta de dados para conseguir informações e utiliza os sentidos na obtenção de determinados aspectos da realidade” (MARCONI e LAKATOS, 2003, p. 190). Assim o pesquisador é capaz de adquirir dados e informações acerca de algo analisando a maneira que é feito, como por exemplo observando um determinado processo de uma organização para fazer a sua modelagem.

Marconi e Lakatos (2003) citam algumas vantagens e limitações da observação, dentre as vantagens há o fato de possibilitar de maneira satisfatória o estudo de uma ampla gama de fenômenos (como o exemplo do processo), exigir menos esforço do observador se comparada a outras técnicas, permitir que se adquira informações através da observação comportamental de atitudes típicas. Dentre as limitações, essa técnica pode levar o observador a ser influenciado pelo observado, criando no observador impressões favoráveis ou não. Há também como limitação o fato de que fatores inesperados poderem influenciar e até atrapalhar o processo de observação do pesquisador.

A observação neste trabalho será utilizada principalmente para se conhecer os processos a qual a escola escolhida está submetida. Colaborando assim no processo de modelagem e auxílio na melhora dos processos.

### Entrevista

A entrevista pode ser entendida como “um encontro entre duas pessoas, a fim de que uma delas obtenha informações a respeito de um determinado assunto, mediante uma conversação de natureza profissional” (MARCONI e LAKATOS, 2003, p. 195). O intuito dessa pesquisa é que o pesquisador possa retirar informações necessárias sobre um determinado assunto ou competência por meio de uma conversa direta e contato face-a-face com outra pessoa.

Marconi e Lakatos (2003) afirmam que uma entrevista pode ser classificada em três tipos: Padronizada ou Estruturada, não padronizada ou não-estruturada e Painel. Segundo Marconi e Lakatos (2003) uma entrevista estruturada é uma entrevista a qual se tem um roteiro com perguntas pré-determinadas e esse roteiro deve ser seguido. Já a entrevista não estruturada é aquela que apresenta maior liberdade acerca do rumo da entrevista. A entrevista em painel por sua vez se trata de uma entrevista em que se contenha repetição de perguntas em períodos curtos de tempo, isso se dá, pois, esse tipo de entrevista é utilizada para estudo da evolução e mudança de opiniões em pequenos espaços de tempo.

De acordo com Marconi e Lakatos (2003), essa técnica tem como vantagens: poder ser utilizada em todos os nichos da população, obtenção de dados que não podem ser encontrados em fontes como documentos mas que são de relevância para o trabalho, entre outras. Dentre as limitações dessa técnica tem-se o fato de poder haver dificuldade de expressão e comunicação, possibilidade de influência do entrevistador para as respostas do entrevistado, além de que pode ser demorada e dificil de se fazer, entre outras limitações.

A técnica de entrevista será utilizada neste trabalho pois acredita-se que assim se possa conhecer melhor algumas características da escola em que o trabalho será direcionado, além de conhecer os clientes mais a fundo. Será utilizada neste trabalho a entrevista não-estruturada, porque como foi descrito anteriormente ela é uma entrevista com mais liberdade, semelhante a uma conversa informal. Presume-se então que os clientes (no caso os administradores da escola) possam ficar mais a vontade para responder as perguntas se forem feitas como uma conversa informal.

## MÉTODOS

Para o desenvolvimento do ambiente proposto, será seguido a metodologia XP que é uma metodologia iterativa e com o desenvolvimento dividido em *releases* (ou entregas constantes como visto na subseção 5.2.3.5). Como o período de execução do trabalho será de janeiro a setembro de 2019, sendo um total de nove meses, decidiu-se utilizar para desenvolvimento seis desses nove meses. E como descrito no nono parágrafo da subseção 5.2.3.5, os *releases* são divididos de dois em dois meses, totalizando então para o desenvolvimento três *releases*.

Sendo assim a etapa inicial será o planejamento e entrevistas ao cliente, para a coleta de estórias de usuário que serão utilizadas no primeiro *release*. Com as estórias de usuários coletadas, partirá para a modelagem dos processos, bem como modelagem do banco de dados. Uma vez modelados e bem definidos os processos e o banco de dados, se iniciará o processo de desenvolvimento que será aliado com os testes afim de oferecer uma melhor qualidade no código final. Para validar as funcionalidades, bem como a retirada de possíveis dúvidas, o XP prega que o cliente deve estar presente no processo de desenvolvimento, portanto reuniões frequentes com o cliente serão feitas no decorrer do desenvolvimento para sanar possíveis dúvidas e validar funcionalidades implementadas. Ao final do desenvolvimento do primeiro *release* pretende-se reunir-se com o cliente e entregar o que foi desenvolvido no primeiro *release*.

Esses processos de desenvolvimento se repetirão ao longo de cada um dos três *releases*. Ao final do terceiro *release* pretende-se disponibilizar o sistema completo para testes de aceitação do cliente. Esses testes servirão para o cliente constatar e validar a completude das funcionalidades requisitadas, bem como testar o sistema como um todo. Caso necessite, correções serão feitas nessa etapa a fim de que se alcance o que foi requisitado.

Após o sistema ter sido entregue ao cliente, o processo de avaliação da metodologia XP será iniciado, e deve ser terminado em cerca de duas semanas. Ao final deste processo pretende-se entregar uma avaliação pessoal acerca do uso do XP, apresentando potenciais dificuldades bem como facilidades no uso dessa metodologia.

# Cronograma

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ATIVIDADES | 1º SEMESTRE 2019 | | | | | 2º SEMESTRE 2019 | | | | |
| JAN/2019 | FEV/2019 | MAR/2019 | ABR/2019 | MAIO/2019 | JUN/2019 | JUL/2019 | AGO/2019 | SET/2019 |  |
| Coleta de Dados para o primeiro *release* (Pesquisa, Observação e Entrevista) |  |  |  |  |  |  |  |  |  |  |
| Análise dos Requisitos do primeiro *release* |  |  |  |  |  |  |  |  |  |  |
| Modelagem dos processos e do banco de dados do primeiro *release* |  |  |  |  |  |  |  |  |  |  |
| Desenvolvimento (codificação) das funcionalidades do primeiro *release* |  |  |  |  |  |  |  |  |  |  |
| Testes e correções das funcionalidades do primeiro *release* |  |  |  |  |  |  |  |  |  |  |
| Reuniões com o cliente para retiradas de dúvidas acerca das funcionalidades do primeiro *release* |  |  |  |  |  |  |  |  |  |  |
| Coleta de Dados para o segundo *release* (Pesquisa, Observação e Entrevista) e entrega do primeiro *release* |  |  |  |  |  |  |  |  |  |  |
| Análise dos Requisitos do segundo *release* |  |  |  |  |  |  |  |  |  |  |
| Modelagem dos processos do segundo *release* |  |  |  |  |  |  |  |  |  |  |
| Desenvolvimento (codificação) das funcionalidades do segundo *release* |  |  |  |  |  |  |  |  |  |  |
| Testes e correções das funcionalidades do segundo *release* |  |  |  |  |  |  |  |  |  |  |
| Reuniões com o cliente para retiradas de dúvidas acerca das funcionalidades do segundo *release* |  |  |  |  |  |  |  |  |  |  |
| Coleta de Dados para o terceiro *release* (Pesquisa, Observação e Entrevista) e entrega do segundo release |  |  |  |  |  |  |  |  |  |  |
| Análise dos Requisitos do terceiro *release* |  |  |  |  |  |  |  |  |  |  |
| Modelagem dos processos do terceiro *release* |  |  |  |  |  |  |  |  |  |  |
| Desenvolvimento (codificação) das funcionalidades do terceiro *release* |  |  |  |  |  |  |  |  |  |  |
| Testes e correções das funcionalidades do terceiro *release* |  |  |  |  |  |  |  |  |  |  |
| Reuniões com o cliente para retiradas de dúvidas acerca das funcionalidades do terceiro *release* |  |  |  |  |  |  |  |  |  |  |
| Entrega do terceiro *release* |  |  |  |  |  |  |  |  |  |  |
| Aplicação do teste de aceitação do sistema |  |  |  |  |  |  |  |  |  |  |
| Em caso de correções serem necessárias |  |  |  |  |  |  |  |  |  |  |
| Avaliação do uso da Metodologia XP |  |  |  |  |  |  |  |  |  |  |
| Escrita da Monografia |  |  |  |  |  |  |  |  |  |  |

# Referências

ALVES, J. R. M. et al. **Educação a Distância:** o estado da arte. São Paulo: Pearson Education do Brasil, v. 1, 2009.

BABBEL. **Preços**, 2018. Disponível em: <https://home.babbel.com/prices>. Acesso em: 23 ago. 2018.

BABBEL. **Fale um idioma como sempre sonhou**, 2018. Disponível em: <https://pt.babbel.com/>. Acesso em: 23 ago. 2018.

BARANAUSKAS, M. C. C.; ROCHA, H. V. D. **Design e Avaliação de Interfaces Humano-Computador**. Campinas: UNIVERSIDADE ESTADUAL DE CAMPINAS, 2003.

BENTO, E. J. **Desenvolvimento Web com PHP e MySQL**. São Paulo: Casa do Código, 2013.

CAELUM. **Desenvolvimento Web com HTML, CSS e JavaScript**. São Paulo: Caelum ensino e inovação, 2018. Disponível em: <https://www.caelum.com.br/download/caelum-html-css-javascript.pdf>. Acesso em: 07 out. 2018.

CAMPOS, A. L. N. **Modelagem de Processos com BPMN**. 2. ed. Rio de Janeiro: Brasport, 2014.

CARVALHO, V. **MySQL:** Comece com o principal banco de dados open source do mercado. São Paulo: Casa do Código, 2015.

CCAA. **Espaço CCAA Aluno**, sd. Disponível em: <https://www.ccaa.com.br/espacoccaa/conteudos/>. Acesso em: 23 ago. 2018.

CROCKFORD, D. **JavaScript:** The Good Parts. Sebastopol: O'Reilly, 2008.

DIAS, D. D. S.; SILVA, M. F. D. **Como escrever uma monografia:** Manual de elaboração com exemplos e exercícios. Rio de Janeiro: Atlas, 2010.

DUOLINGO. **Aprenda idiomas de graça. Para sempre**, sd. Disponível em: <https://pt.duolingo.com/>. Acesso em: 23 ago. 2018.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Education, 2011.

FERREIRA, A. B. D. H. **Mini Aurélio Século XXI:** O minidicionário da língua portuguesa. 5. ed. Rio de Janeiro: Nova Fronteira S.A, 2001.

GOOGLE. **Sobre o Google Sala de aula**, 2018. Disponível em: <https://support.google.com/edu/classroom/answer/6020279?hl=pt-BR#>. Acesso em: 23 ago. 2018.

HINZ, M. A. M. **Um estudo descritivo de novos algoritmos de criptografia**. Dissertação (Bacharel em Informática) - UFP. Pelotas, p. 58. 2000.

HIRAMA, K. **Engenharia de Software:** Qualidade e Produtividade com Tecnologia. Rio de Janeiro: Elsevier, 2011.

INSTITUTE OF ELETRICAL AND ELETRONICS ENGINEERS. **IEE Std 610.12-1990:** IEEE Standard Glossary of Software Engineering Terminology. New York: [s.n.], 1990. 84 p.

LOCKHART, J. **PHP Moderno**. São Paulo: Novatec, 2015.

MCFARLAND, D. S. **CSS3:** the missing manual. 3. ed. Sebastopol: O'Reilly, 2013.

MORENO, E. D.; PEREIRA, F. D.; CHIARAMONTE, R. B. **Criptografia em Hardware e Software**. São Paulo: Novatec, 2005.

NETO, J. A. D. M. et al. **Educação a distância:** o estado da arte. São Paulo: Pearson Education do Brasil, v. 2, 2012.

OTWELL, T. Encryption. **Laravel Docs**, 2018. Disponível em: <https://laravel.com/docs/5.7/encryption>. Acesso em: 05 out. 2018.

PHP. **O que é o PHP?**, 2018. Disponível em: <https://secure.php.net/manual/pt\_BR/intro-whatis.php>. Acesso em: 30 set. 2018.

PHP. Modelo de Armazenamento Criptografado. **php.net**, 2018. Disponível em: <https://secure.php.net/manual/pt\_BR/security.database.storage.php>. Acesso em: 05 out. 2018.

PRESSMAN, R. S. **Engenharia de Software:** Uma abordagem Profissional. 7. ed. Porto Alegre: Bookman, 2011.

ROBBINS, J. N. **HTML5:** Pocket Reference. 5. ed. Sebastopol: O'Reilly, 2013.

SAENGTONGSRIKAMON, C.; MEESAD, P.; SUNANTHA, S. Scanner-Based Optical Mark Recognition. **International Conference on Computing and Information Technology**, Bangkok, 2017. 5.

SANDHU, R. S. Role-based Access Control. In: SANDHU, R. S. **Advances in Computers**. Fairfax: Academic Press, v. 46, 1998. p. 237-286.

SEVERINO, A. J. **Metodologia de trabalho científico**. 22. ed. São Paulo: Cortez, 2002.

SILBERCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 3. ed. São Paulo: Pearson Education, 1999.

SILVER, B. **BPMN Method and Style:** with Bpmn Implementer's Guide. 2. ed. Altadena: Cody-Cassidy Press, 2017.

SKLAR, D. **Aprendendo PHP:** Introdução amigável à linguagem mais popular da WEB. São Paulo: Novatec, 2016.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

STAUFFER, M. **Desenvolvendo com Laravel:** Um Framework para construção de aplicativos PHP modernos. 1. ed. São Paulo: Novatec, 2017.

TELES, V. M. **Extreme Programming:** Aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. 2. ed. São Paulo: Novatec, 2014.

WIZARD. **Experiências Wizard**, 2017. Disponível em: <http://www.wizard.com.br/experiencias-wizard/>. Acesso em: 23 ago. 2018.

WIZARD. **Sobre a Wizard**, 2017. Disponível em: <http://www.wizard.com.br/sobre-wizard/>. Acesso em: 23 ago. 2018.

ZAPATER, M.; SUZUKI, R. **Segurança da Informação: Um diferencial determinante na competitividade das corporações**. Promon Business & Tecnology Review. Rio de Janeiro, p. 28. 2005.