

**FACULDADE DE TECNOLOGIA DA PARAÍBA-FATEC/PB
TECNOLOGIA EM SISTEMAS PARA INTERNET
RODRIGO GOMES SOBRAL**

**PRINTEASY: PROPOSTA PARA UMA SOLUÇÃO NO SERVIÇO DE
ATENDIMENTO PARA FOTOCOPIADORAS**

CABEDELLO

2017

RODRIGO GOMES SOBRAL

**PRINTEASY: PROPOSTA PARA UMA SOLUÇÃO NO SERVIÇO DE
ATENDIMENTO PARA FOTOCOPIADORAS**

Relatório técnico apresentado ao Curso de Sistemas para Internet na Faculdade de Tecnologia da Paraíba – FATEC/PB como requisito para obtenção do título de tecnólogo em Sistemas para Internet.

Orientador: Marcelo Fernandes de Sousa.

CABEDELO

2017

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Padre Joaquim Colaço Dourado

S677p

Sobral, Rodrigo Gomes

Printeasy: proposta para uma solução no serviço de atendimento para fotocopiadoras / Rodrigo Gomes Sobral. – Cabedelo, PB: [s.n], 2017.

34p.

Orientador. Prof. Marcelo Fernandes de Sousa. Monografia (Graduação Tecnológica em Sistemas para Internet) – Faculdade de Tecnologia da Paraíba – FATEC-PB.

1. Desenvolvimento de sistemas. 2. Sistemas de computação.
3. Levantamento de requisitos. I. Título.

CDU 004.43

RODRIGO GOMES SOBRAL

Aprovada em: 18 de Dezembro de 2017.

BANCA EXAMINADORA

Prof. Dr. Marcelo Fernandes de Sousa (orientador)
Instituto de Educação Superior da Paraíba

Prof. Fernanda Carolina Ferreira
Instituto de Educação Superior da Paraíba

Prof. Gláucio Bezerra Rocha
Instituto de Educação Superior da Paraíba

Dedico esse trabalho à minha família, amigos, à minha futura esposa: Therlen, professores e à ciência.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, que me orientou e me deu energia para fazer as pesquisas e desenvolver esse trabalho.

Agradeço aos meus pais, que me incentivaram à sempre buscar o melhor de mim todos os dias.

Aos meus professores do IESP – FATEC/PB, em especial à Gláucio Bezerra Rocha, Hercílio de Medeiros Sousa, Marcelo Fernandes de Sousa, Allana Moraes, Humberto Junior, Wellington Araujo, Jader Rodrigues, Ângelo Vidal de Negreiros e Carlos Barbosa Filho.

Sem qualquer um desses indivíduos, meu trabalho não teria sido desenvolvido da forma que foi.

“Coisas incríveis no mundo dos negócios nunca são feitas por uma única pessoa, e sim por uma equipe”.

Steve Jobs

RESUMO

No presente trabalho o autor trata de uma proposta para o desenvolvimento e aplicação de um sistema que se dispõe a organizar e facilitar o processo de imprimir e pagar por cópias de documentos em estabelecimentos de fotocopiadoras. A pesquisa se utilizou de métodos de engenharia de software, o que levou o autor à fazer pesquisas de campo para levantamento de informações tanto na literatura acadêmica quanto em visitas ao cliente do sistema proposto. O processo de levantamento de requisitos foi um dos primeiros passos no desenvolvimento do projeto e forneceu dados importantes no início do processo de esclarecimento das ideias para a definição das funcionalidades iniciais do sistema de acordo com a necessidade do cliente. O sistema proposto nesse trabalho, as pesquisas de campo e a necessidade de informatização no processo da tiragem da cópia de documento até o pagamento do serviço prestado pelo fotocopiadora levaram à conclusão que há uma carência de tecnologia para essa ação e que a utilização de um sistema informático, que organizasse esse processo, facilitaria o trabalho dos atendentes do estabelecimento. Todo o processo de descoberta e análise de informações teve cunho exploratório, uma vez que não se utilizou de um sistema modelo que propusesse a mesma perspectiva ou utilidade que o protótipo de sistema até então pelo autor proposto a ser desenvolvido. Durante a pesquisa qualitativa, depois da fase de elicitação, obteve-se uma lista de requisitos previamente definidos que vieram a se tornar regras de negócios para a criação de funcionalidades a serem utilizadas no protótipo proposto por esse trabalho. Por fim, deu-se início na programação do protótipo tendo em vista a projeção visual da ideia adquirida e aprimorada no decorrer do desenvolvimento do trabalho proposto, onde toda a codificação foi feita de acordo com as funcionalidades entendidas como necessidades do negócio do cliente.

Palavras Chaves: desenvolvimento, levantamento de requisitos, sistemas de computação.

ABSTRACT

In this present paper the author leads a proposition to a system's implementation and development that is able to organize and ease the printing process and pay for the paper copies in printing services. The research used some Software Engineering methods, what guided the author to make camp researches for the information gathering in both academic literature and in client visiting. The process of requisites gathering was the first step for the project development and provided important information at the beginning of the process of clarifying the ideas for the system firsts functionalities' definitions according to the client needs. The proposed system in this paper, the camp researches and the information needs in the process of making paper copies till the payment to the print store's services provided the author the conclusion that there is a technology need for this process and that the using of a computing system, which organizes this process, would make this workers' job easier. All the discovering process and information analysis had an exploratory way, once that there was no use of a model system that proposed or showed the same perspective or utility that the prototype till then for the author proposed to be developed. During the qualitative research, after the information gathering phase, it has gotten a list of pre defined requirements that came up to become business rules for the system functionalities to be used in the prototype proposed for this paper. Lastly, there was then the prototype-coding-start having the objective of visualize of the idea that was acquired and improved across the development of this paper, where all the coding creation was developed according to the functionalities understood as the client business rules.

Key words: development, information gathering, computing systems.

LISTA DE ILUSTRAÇÕES

Figura 1: As etapas da Engenharia de Software	18
Figura 2: Exemplo do Padrão MVC.....	20
Figura 3: Diagrama de caso de uso do <i>PrintEasy</i>	21
Figura 4: O modelo conceitual.....	22
Figura 5: Exemplo de Conversão entre o Modelo Conceitual e o modelo Lógico.....	23
Figura 6: O modelo Lógico em diagrama.....	24
Figura 7: O modelo Lógico descrito em <i>scripts</i>	24
Figura 8: O modelo Físico.....	25
Figura 9: Representação da organização do banco de dados através do phpAdmin.....	26
Figura 10: Pagina de <i>Login</i>	27
Figura 11: Divisão da pagina utilizando o conceito de grade.....	28
Figura 12: Cadastro de alunos.....	29
Figura 13: Menu do Aluno.....	30
Figura 14: Mapa de navegação do aluno no sistema.....	31

LISTA DE SIGLAS

MVC	MODEL – VIEW – CONTROLLER
HTML	<i>HIPERTEXT MARKUP LANGUAGE</i>
CSS	<i>CASCADING STYLE SHEETS</i>
UML	UNIFIED MODELING LANGUAGE
JSP	JAVA SERVER PAGES
JPA	JAVA PERSISTENCE API
UFSC	UNIVERSIDADE FEDERAL DE SANTA CATARINA
IESP	INSTITUTO DE EDUCAÇÃO SUPERIOR DA PARAIBA
FATEC/PB	FACULDADE DE TECNOLOGIA DA PARAIBA
RN _N	REGRAS DE NEGÓCIO
EX _N	EXCEÇÕES NAS REGRAS DE NEGOCIOS
E-R	ENTIDADE/RELACIONAMENTO
SGBD	SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS
SQL	STRUCTURED QUERY LANGUAGE

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVO GERAL.....	13
1.2 OBJETIVOS ESPECÍFICOS.....	13
2 TECNOLOGIAS E MÉTODOS UTILIZADOS	14
2.1 JAVA	14
2.2 BOOTSTRAP, HTML E CSS.....	15
2.3 FERRAMENTAS	16
2.4 MÉTODO DE PROTOTIPAGEM CONTINUADA.....	16
3 PRINTEASY: UMA PROPOSTA DE MELHORIA NO PROCESSO DE RELAÇÃO ENTRE CLIENTES E AMBIENTES DE FOTOCÓPIAS.....	17
3.1 LEVANTAMENTOS DE REQUISITOS.....	17
3.1.1 Entrevistando o cliente e observando in loco	17
3.1.2 Regras de negócio.....	19
Exceções.....	19
3.2 ARQUITETURA DO SISTEMA	19
3.3 DIAGRAMA DE CASO DE USO.....	21
3.4 MODELAGEM CONCEITUAL E LÓGICA DO BANCO DE DADOS.....	22
4. LAYOUT.....	26
5. CONCLUSÃO.....	32
REFERÊNCIA	33

1 INTRODUÇÃO

Esse trabalho pretende oferecer uma solução de automatização para o processo de impressão de fotocópias desenvolvendo um banco de crédito *online*. Esse sistema de banco de crédito online para impressões será implantado inicialmente na faculdade IESP-PB.

Esse sistema visa facilitar a praticidade na hora de tirar impressões, ajudar na ação de dar troco em moedas, acelerar o atendimento, evitar a necessidade de andar com moedas para a impressão e reduzir o tamanho da fila de impressões.

A motivação para o desenvolvimento dessa pesquisa de campo foi a necessidade de automatização dos processos operacionais no ato de imprimir documentos e fotocópias.

Com a automatização do sistema, os usuários poderão mandar a impressão de quaisquer dispositivos que acessem internet para serem impressos no fotocopadora.

A implementação do Banco de impressões online é um negócio inovador, pois a questão da forma de pagamento se dará de forma diferente com transferência de crédito entre os seus usuários e a fotocopadora via web. O sistema oferecerá uma comodidade aos usuários e clientes das copiadoras (professores, funcionários da faculdade e alunos) que poderão mandar os seus arquivos para serem impressos de qualquer lugar, desde que tenha acesso a um dispositivo com acesso à internet.

Essa tecnologia permite que as pessoas possam imprimir cópias de documentos através de créditos em suas contas comprados anteriormente, é usada em instituições de ensino como a Universidade de Victoria, Canadá; e a faculdade *Camosun College*, Canadá.

1.1 OBJETIVO GERAL

Desenvolver um sistema *web*, que automatize o processo de imprimir e pagar pelo serviço prestado por estabelecimentos de fotocopiadoras em universidades.

1.2 OBJETIVOS ESPECÍFICOS

- Realizar uma pesquisa bibliográfica e de campo para fundamentar o desenvolvimento;
- Levantar os requisitos e regras de negócio do sistema;
- Criar um protótipo do sistema para executar as tarefas de pagamento e o processo no ato de imprimir no fotocopiadora;

2 TECNOLOGIAS E MÉTODOS UTILIZADOS

Para o desenvolvimento do protótipo, pretendido por esse trabalho, e pesquisa fundamentada na literatura técnica foram utilizadas algumas tecnologias da computação.

Linguagens como: Java, HTML 5, CSS 3 e UML; frameworks e bibliotecas como *Bootstrap*, JSP, JPA, *Servlets*; Ferramentas como: Sublime, BRModelo, Eclipse, etc.; e Métodos de desenvolvimentos e alguns padrões de projetos serão introduzidos de forma simples e breve para que possamos explanar o trabalho desde a mineração da informação até à prototipagem do sistema proposto.

2.1 - JAVA

Para o desenvolvimento da lógica e estruturação na criação do protótipo do sistema será utilizada a Linguagem de Programação Orientada a Objetos: 'JAVA'.

Em 1991, foi criado um projeto, pela empresa Sun *Microsystems*, chamado de Green. No projeto Green foi criada uma linguagem baseada em C e C++. Essa linguagem foi inicialmente chamada de Oak, em homenagem à uma árvore que seu criador, James Gosling, via da janela de seu escritório na Sun *Microsystems*. Oak passou a se chamar Java após uma visita do grupo de pesquisa Green à uma cafeteria onde o café era importado de uma cidade chamada Java. (DEITEL, 2003) p55,59

A linguagem Java começou a se popularizar em 1993 com a explosão da *World Wide Web*. "...[E] as pessoas da Sun viram o imediato potencial de utilizar Java para criar páginas da Web com o chamado *conteúdo dinâmico*" (DEITEL, 2003, p. 59).

Java é uma linguagem Orientada a Objetos, onde suas principais tecnologias são a herança e o polimorfismo.

A herança tira proveito dos relacionamentos entre classes, nos quais os objetos de certa classe - como uma classe de veículos - tem as mesmas características. [...]O polimorfismo permite-nos escrever programas de uma forma geral para tratar uma ampla variedade de classes relacionadas existentes e ainda a serem especificadas (DEITEL, 2003, p.431).

2.2 BOOTSTRAP, HTML E CSS

Para a estruturação dos elementos na página web, estilização e layout do sistema utilizaremos o framework Bootstrap e as linguagens HTML 5 e CSS 3.

HTML é uma sigla para *HiperText Markup Language* (linguagem de marcação de hipertextos) e serve para estruturar páginas webs na internet. Os componentes estruturais encontrados em uma paginam web precisam de especificidade e padrões para uso. O html trata de componentes como links, textos, imagens, botões, entre outros.

Como o HTML já trata da parte estrutural, o CSS - *Cascading Style Sheets* - trata da parte da beleza da pagina. A estilização e organização da maioria dos conteúdos encontrados online hoje em dia usa CSS. As componentes webs ganham cor, forma e estilo graças as propriedades do CSS.

O Bootstrap é uma junção de HTML, CSS e Java script. Isso abstrai parte das linguagens utilizadas por ele e facilita na criação de conteúdo para a Internet.

Para o autor, "[o] Bootstrap é um framework *front-end* para desenvolvimento de *websites* que contém um conjunto de ferramentas que auxiliam na criação de códigos HTML, CSS e Java Script" (MARTIN, 2016).

O Bootstrap é uma criação dos desenvolvedores da rede social Twitter, que buscavam inventar uma ferramenta que permitisse que o seu produto seguisse o mesmo padrão (ou pelo menos atendesse boa parte dessa padronização) na apresentação em qualquer dos navegadores. Logo, desenvolveram essa ferramenta para diminuir as divergências na interpretação entre os navegadores. Suas principais características são:

- Abordagem *Mobile First* e design responsivo - onde o desenvolvimento de paginas web - dinâmicas ou não - prioriza que sejam utilizados métodos de desenvolvimento adaptativo para dispositivos *mobile*. Essa característica oferta uma demanda maior para o crescimento na responsividade em sites e sistemas web.
- Suporte para todos os navegadores populares - o bootstrap dispõe de bibliotecas e funcionalidades que preveem e trata as diferenças de interpretação dos navegadores mais populares; isso diminui o risco de quebra de layout e favorecimento de navegadores específicos.
- Fácil aprendizagem - como o bootstrap utiliza classes predefinidas em CSS

com o acompanhamento da linguagem de script, Java Script, utilizando as *tags* da linguagem de marcação de hipertexto, HTML, o desenvolvimento e estilização fica autoexplicativa e intuitiva.

2.3 FERRAMENTAS

Para a codificação do protótipo na parte do *back-end* e na lógica do sistema de protótipo proposto pelo trabalho, utilizaremos o programa Eclipse JEE NEON. O Eclipse é uma ferramenta bastante utilizada no mercado de trabalho. Além de ter um layout amigável e uma aprendizagem bastante intuitiva, o Eclipse é gratuito. Utilizaremos o Eclipse para desenvolver as páginas JSP (páginas web para a apresentação do sistema), implementar o *hibernate* usando JPA (para o controle de acesso ao banco de dados) e os *Servlets*, que nos ajudarão a implementar a lógica do sistema.

Para a modulação dos diagramas de modelagem de banco de dados utilizamos a segunda versão de uma ferramenta Case para criar os diagramas de Modelo Entidade e Relacionamento; o BrModelo. A ferramenta BrModelo foi desenvolvida por pesquisadores da UFSC como um trabalho de pós-graduação em banco de dados. Essa ferramenta é livre para uso.

Para o desenvolvimento de desenhos explicativos, diagramas de caso de uso e diagrama de classe utilizamos a ferramenta online chamada Draw.io, um sistema web que permite criar diversos desenhos e diagramas representativos utilizados em qualquer ramo do conhecimento. Como todas as outras ferramentas comentadas anteriormente, o Draw.io também é uma ferramenta livre para uso.

2.4 MÉTODO DE PROTOTIPAGEM CONTINUADA

Para o desenvolvimento do protótipo funcional, proposto pela pesquisa, utilizamos o modelo de processo evolucionário. Esse método fala que quando não se tem o poder de informações suficientes para definir a regra de negócio do cliente ou as funcionalidades que o sistema deve oferecer, devemos evoluir um protótipo aos poucos fazendo com que as regras de negócio e as funcionalidades do sistema que estavam antes obscuras fiquem bem claras.

Frequentemente, o cliente define uma série de objetivos gerais para o software, mas não identifica, detalhadamente, os requisitos para funções e recursos. Em outros casos, o desenvolvedor encontra-se inseguro quanto à eficiência de um algoritmo, quanto à adaptabilidade de um sistema operacional ou quanto a forma em que deva ocorrer a interação homem/máquina. Em situações como essas, e em muitas outras, o paradigma de prototipação pode ser a melhor escolha de abordagem. (PRESSMAN, 2011, p. 62)

Junto com o método de Prototipagem Continuada, utilizamos o incremental e iterativo. Dividimos as etapas de desenvolvimento e visitamos o cliente para alinharmos a evolução do projeto proposto de acordo com a necessidade do seu negócio.

3 PRINTEASY: UMA PROPOSTA DE MELHORIA NO PROCESSO DE RELAÇÃO ENTRE CLIENTES E AMBIENTES DE FOTOCÓPIAS

Modelar e prototipar o *PrintEasy*, partiu da premissa de solver uma necessidade tecnológica que os estudantes da faculdade IESP e Fatec/PB carecem; um sistema eletrônico que melhorasse o serviço do fotocopadora.

3.1 LEVANTAMENTOS DE REQUISITOS

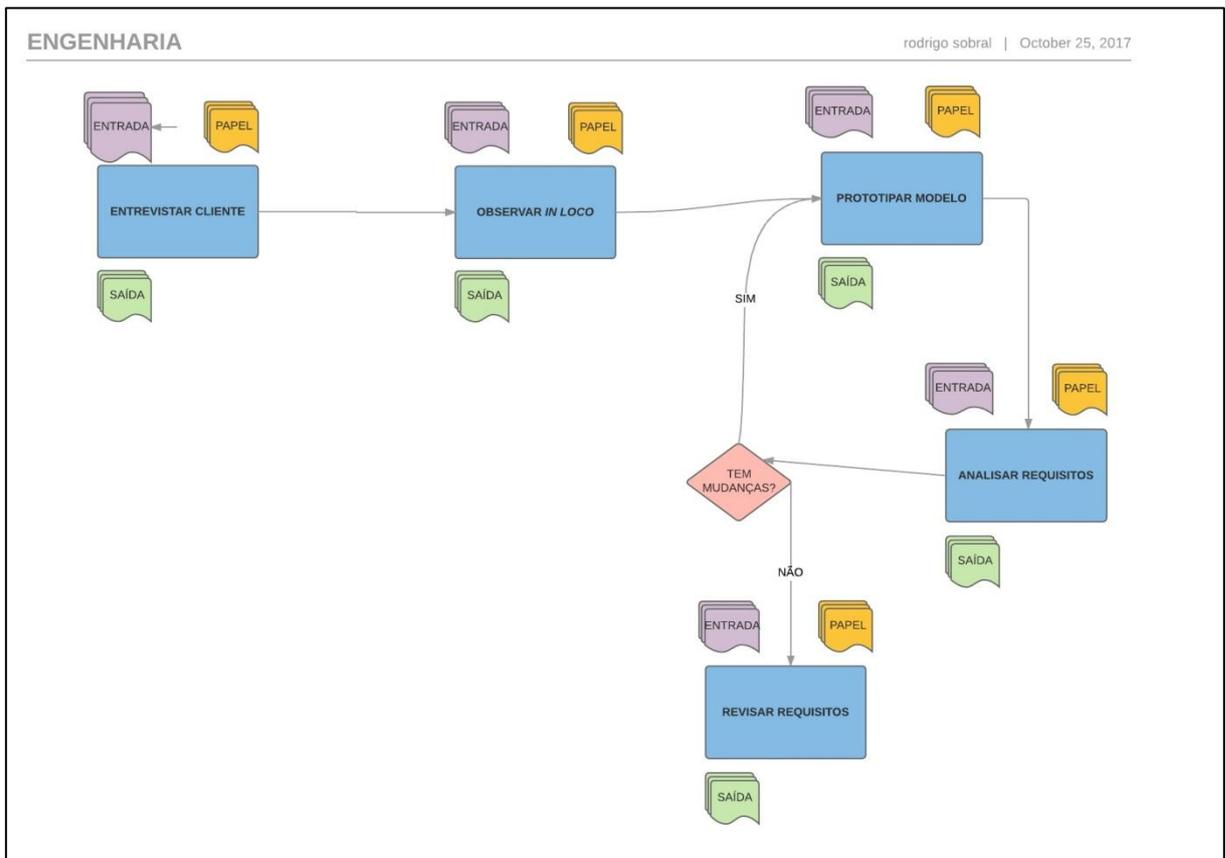
Segundo Bezerra (2007), o levantamento ou elicitación de requisitos é uma fase do projeto de desenvolvimento de software muito importante. É nessa fase do projeto, onde serão coletados dados iniciais para a análise de requisitos. Essa análise resultará em uma listagem de requisitos definidos, que guiará o grupo envolvido na pesquisa no desenvolvimento do sistema ou software.

3.1.1 Entrevistando o cliente e observando *in loco*

No dia 21 de setembro de 2017, fizemos uma entrevista com o cliente, e funcionário do fotocopadora, para saber como funcionavam seus serviços e quais eram suas regras de negócio.

Foi feito um levantamento sobre quais dados eram necessários para a criação de uma conta para esse futuro usuário do sistema e quais seriam os problemas futuros. A figura mostra os passos para o desenvolvimento do protótipo. Cada passo ou etapa no desenvolvimento do software agregou uma parte ou uma regra na evolução do protótipo.

Figura 1: As etapas da engenharia de Software



Fonte: o próprio autor, 2017.

Foram discutidas questões como: o controle da demanda de impressões, a devolução do dinheiro caso o aluno não quisesse mais o seu crédito, os planos para oferecer créditos aos alunos.

Utilizamos métodos de questionamento como: preenchimento de formulário conversa informal e uma observação do fluxo de serviço.

Depois da coleta de dados, pudemos listar as possíveis funcionalidades que o sistema provavelmente viesse a ter.

3.1.2 Regras de negócio

Graças às observações feitas na entrevista, aos formulários e à conversa gravada na visita ao cliente, foi possível obter um detalhamento aprofundado sobre o negócio. A seguinte lista de regras do negócio foi construída junto ao funcionário entrevistado:

- [Rn1] - O aluno precisa criar uma conta e estar logado no sistema para utilizá-lo.
- [Rn2] - O aluno precisa ter crédito em sua conta para imprimir ou transferir crédito de seu domínio para outro usuário do sistema.
- [Rn3] - O aluno deve enviar no um arquivo pelo sistema, podendo esse ser um documento de texto ou uma imagem, para ser impresso no fotocopadora.
- [Rn4] - Caso o aluno imprima um arquivo, será descontado o valor da quantidade de páginas impressas vezes o valor da impressão.
- [Rn5] - Para ter acesso aos créditos no sistema, o aluno deve comprar um cartão com um código de crédito ou receber uma transferência de valores de outro aluno que já tem crédito em sua conta.
- [Rn6] - o sistema deve validar a autenticidade e o valor do crédito inserido pelo aluno.

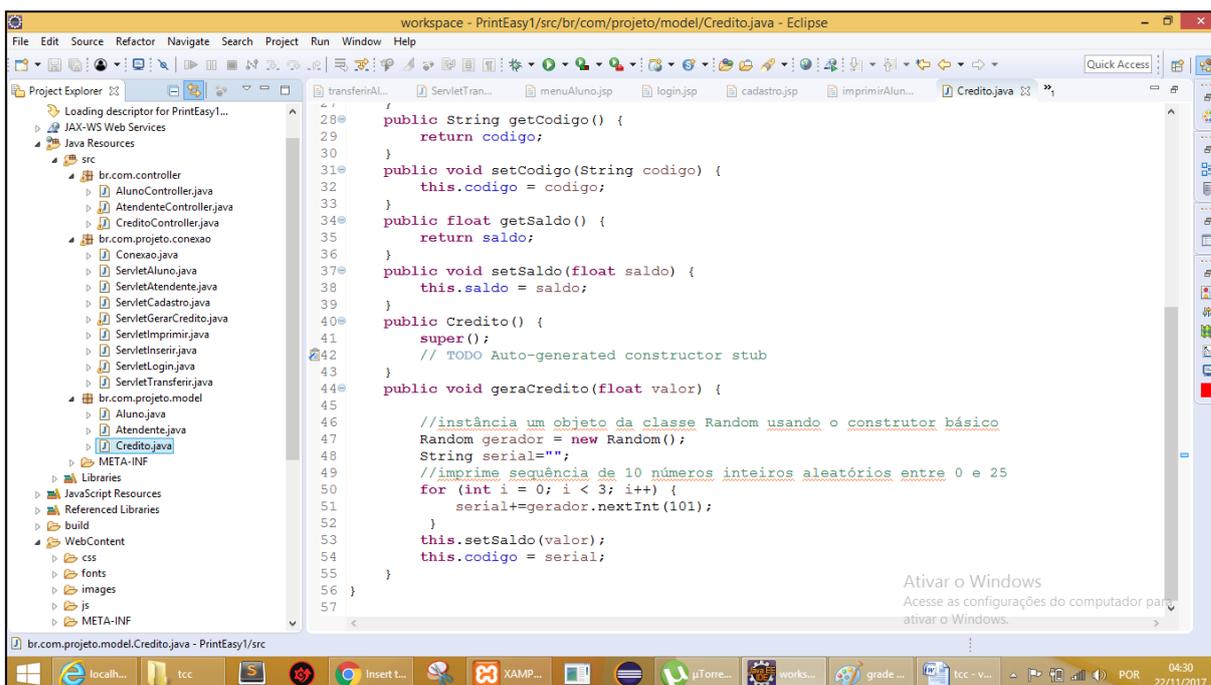
Exceções

- [Ex1] - Se o aluno não possuir crédito, ele não poderá enviar o arquivo para ser impresso.
- [Ex2] - Se o administrador do sistema excluir o arquivo enviado pelo aluno para ser impresso, não será descontado crédito dele.
- [Ex3] - Caso o aluno saia da faculdade com crédito, ele poderá negociar a devolução de seus créditos e o resgate de parte do seu dinheiro.

3.2 ARQUITETURA DO SISTEMA

O desenvolvimento do protótipo está sendo construído em uma perspectiva simplória, apenas para concretizar a ideia do sistema funcional. Esperasse resultados reais com os critérios de qualidade solicitados pelo cliente.

Figura 2: Exemplo do Padrão MVC



Fonte: o próprio autor, 2017.

O método de engenharia de software utilizado na construção do sistema será o padrão *Model View Controller*. A figura mostra a divisão na organização dos tipos de classes e arquivos dentro da estrutura do diretório do projeto na ferramenta Eclipse. Esse padrão visa separar as classes de modelos, que serão as entidades palpáveis ou personificadas que fazem parte do processo de impressão, como: aluno, arquivos ou crédito (dinheiro), as classes *View* que são os arquivos que determinam o *layout* do sistema e o *Controller*, que guarda toda a lógica do sistema. Separar os arquivos de classes e arquitetar a construção nessa forma facilitará a manutenção do sistema em possíveis futuras versões.

O modelo é a estrutura de lógica dos dados, a visão o suporte da *interface* com o usuário e o controle realiza a comunicação entre o modelo e a visão. (BRITO MACHADO; ZANATTA, 2008).

Esse método avançado de programação nos permitiu organizar o código fonte, deixando-o de fácil manutenção em possíveis refatorações.

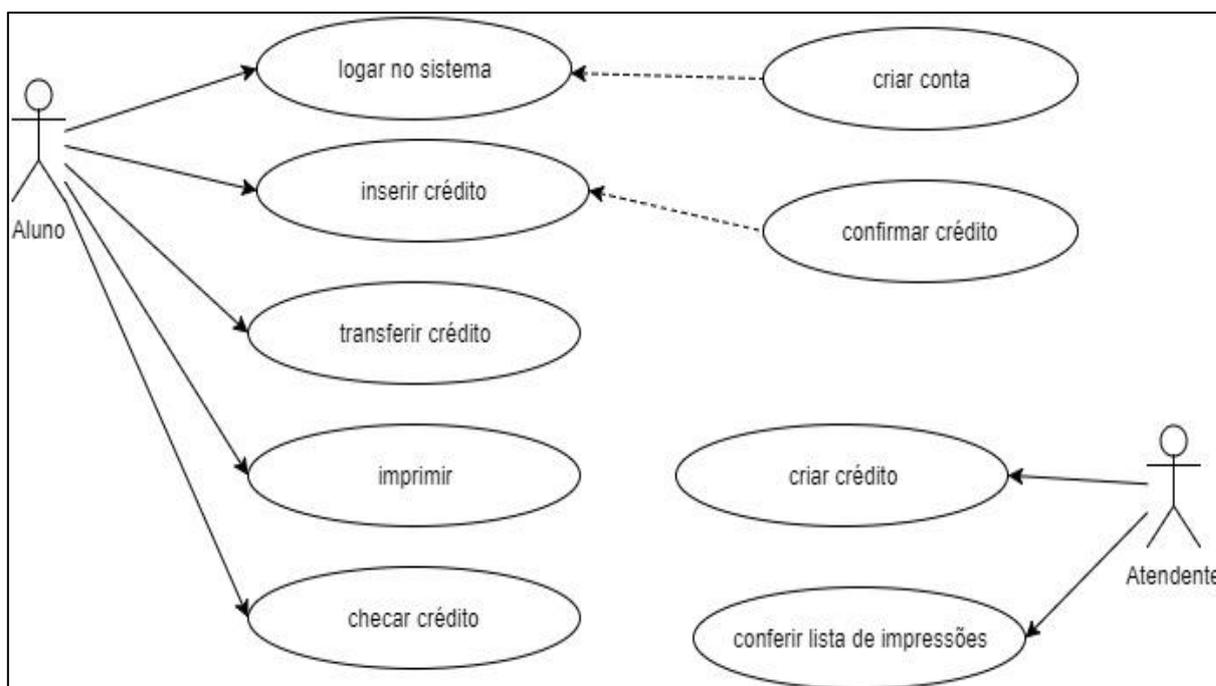
3.3 :DIAGRAMA DE CASO DE USO

O diagrama de caso de uso é uma representação gráfica do sistema. Os componentes utilizados representam os responsáveis por executar tarefas diversas no sistema, as funcionalidades levantadas através da pesquisa de campo na fase de elicitação de requisitos e as relações entre eles.

As formas elípticas, chamadas de casos de uso, representam as ações no sistema e podem se tornarem métodos das classes no desenvolvimento de sistemas orientados a objeto.

Como podemos observar na figura 3, as setas e linhas representam as relações entre os atores e os casos de uso. Os bonecos em forma de palitos representam os responsáveis por executar alguma tarefa no sistema. Os atores podem tomar papel do usuário, do próprio sistema ou de outros sistemas.

Figura 3: Diagrama de caso de uso do *PrintEasy*



Fonte: próprio autor, 2017.

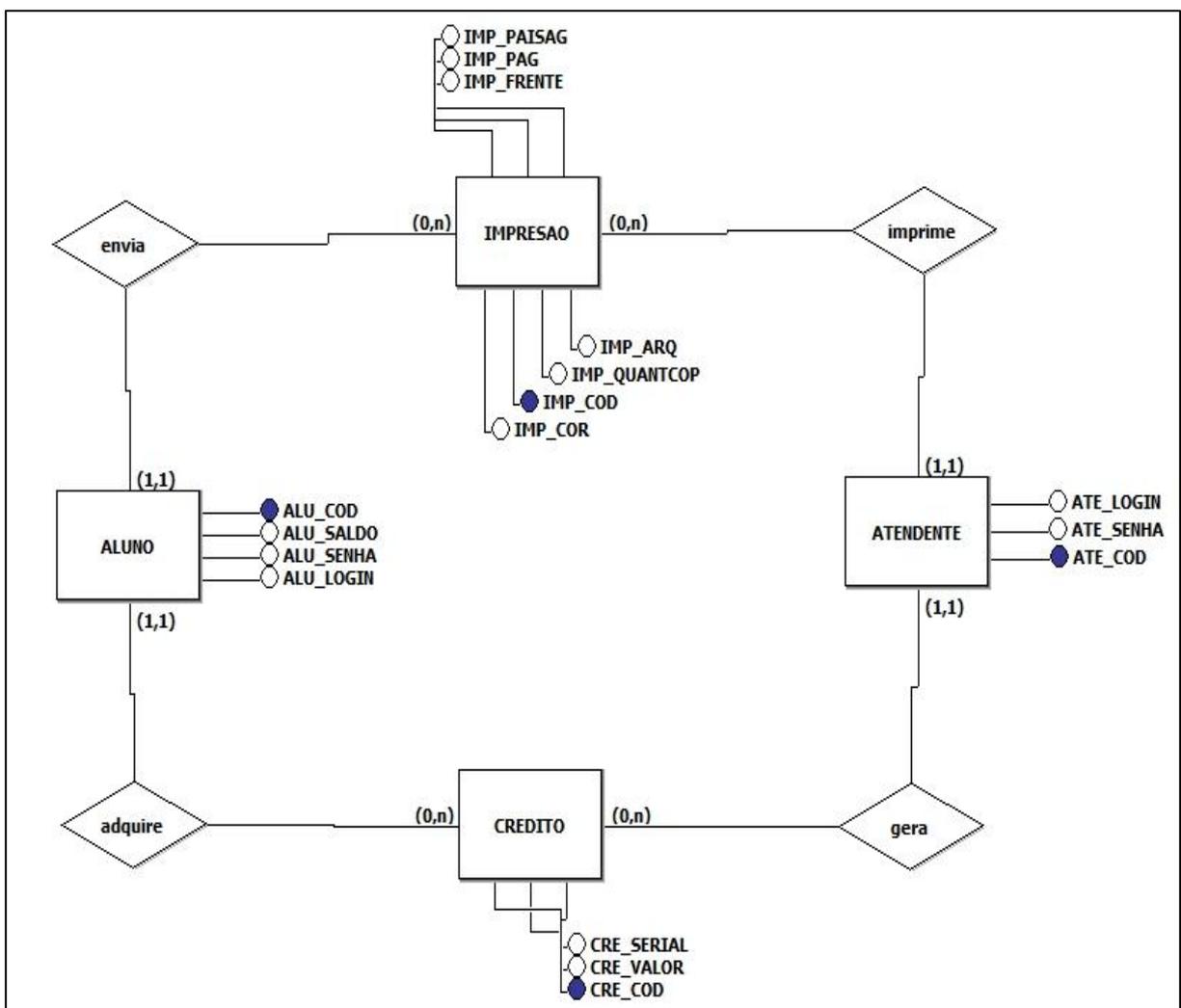
O diagrama de caso de uso é um dos principais diagramas da linguagem UML. E mostra de forma visual como o sistema será organizado.

3.4 – MODELAGEM CONCEITUAL E LÓGICA DO BANCO DE DADOS

O modelo de diagrama de Entidade-relacionamento é uma representação simples e clara de como está organizado o banco de dados.

O modelo entidade/relacionamento (E-R) é baseado em uma percepção de um mundo real que consiste em uma coleção de objetos básicos, chamados entidades, e as relações entre esses objetos. Uma entidade é uma 'coisa' ou 'objeto' no mundo real que é distinguível dos outros objetos. (Silberschatz, Korth, Sudarshan, 2006, p. 5)

Figura 4: O modelo conceitual



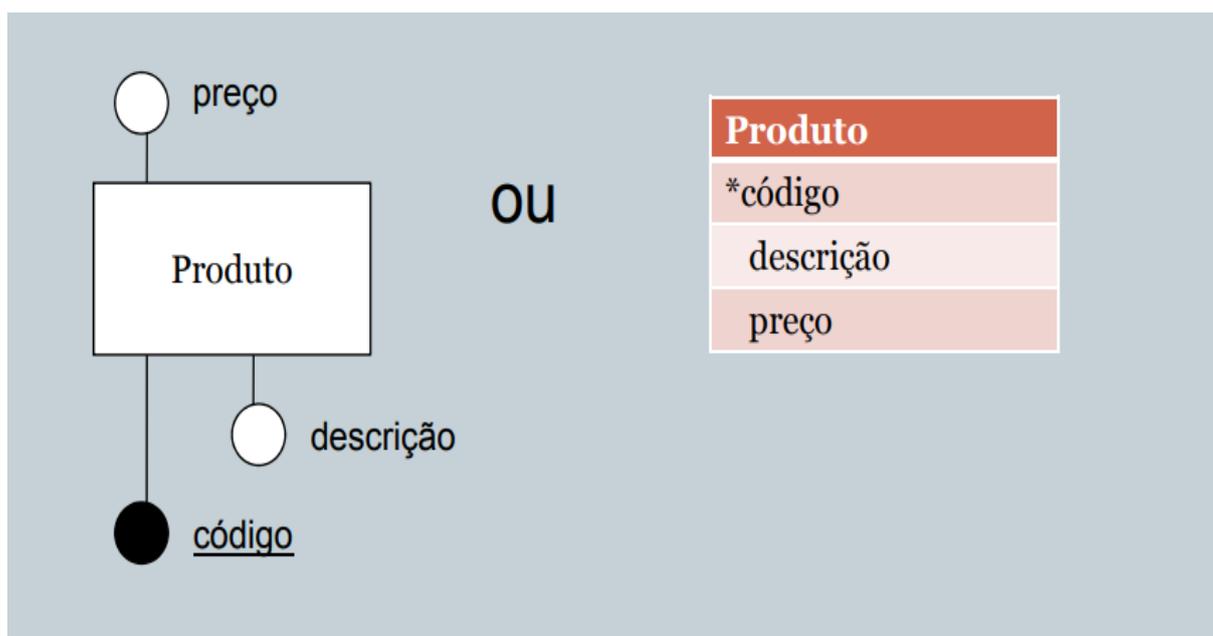
Fonte: O próprio autor, 2017.

A figura 4 mostra um exemplo de um modelo entidade-relacionamento. Esse diagrama foi feito para orientar e guiar a construção e organização do projeto proposto por esse trabalho.

As formas representadas nesse diagrama são a explicação visual dos componentes do diagrama, onde:

- Os retângulos – representam um conjunto de entidades.
- Os círculos com traços – representam os atributos ou características de cada entidade. Os atributos preto-preenchidos são considerados atributos identificadores. Os atributos identificadores são características únicas de cada entidade.
- As linhas – representam as ligações entre os componentes. Essas ligações representam os relacionamentos entre as entidades. Os relacionamentos podem ser binários, ternários ou multirrelacionais.
- Os losangos – representam os relacionamentos entre as entidades.
- Os números – representam a cardinalidade do relacionamento entre as entidades, ou seja, como e de que forma as entidades irão se relacionar consigo mesmas ou com outra entidade e/ou conjunto de entidades.

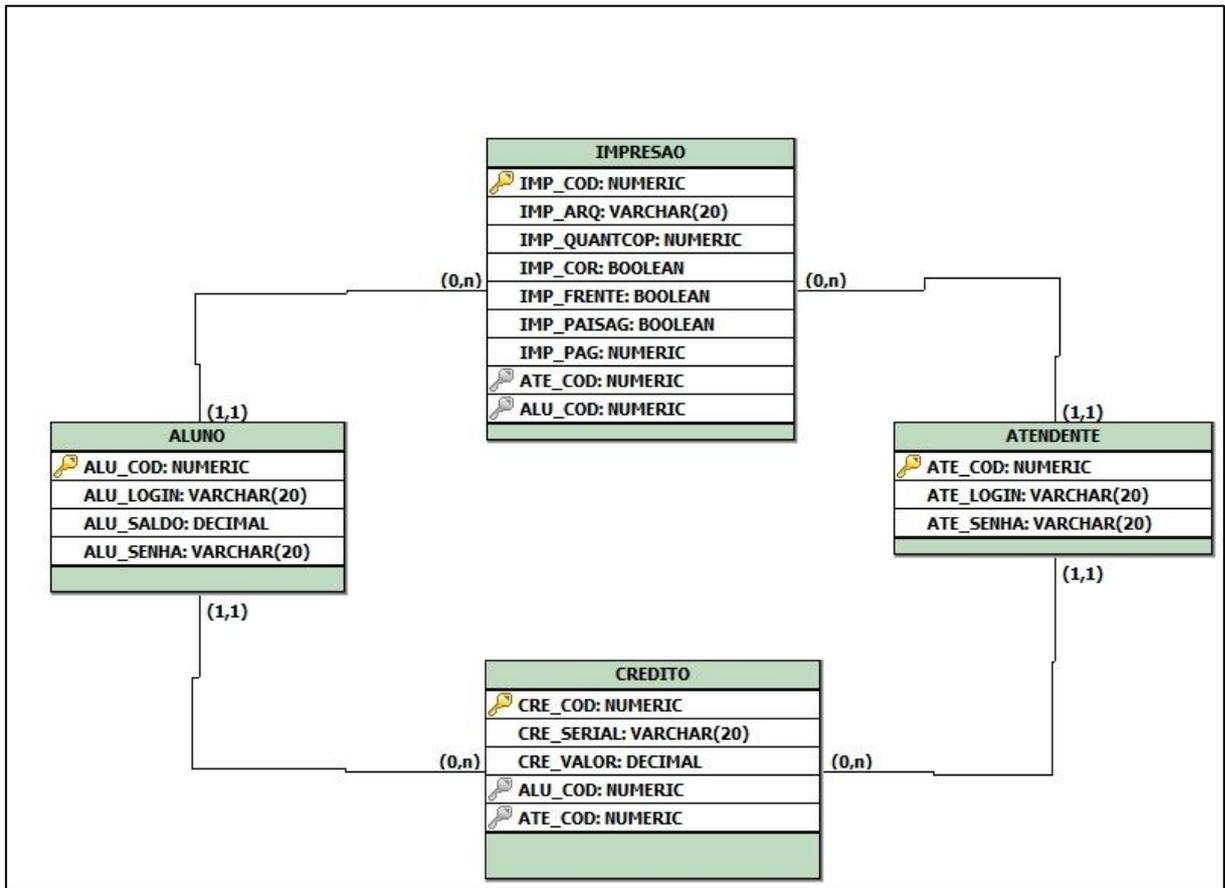
Figura 5: Exemplo de Conversão entre o Modelo Conceitual e o modelo Lógico



Fonte: Abrahao Lopes , 2017

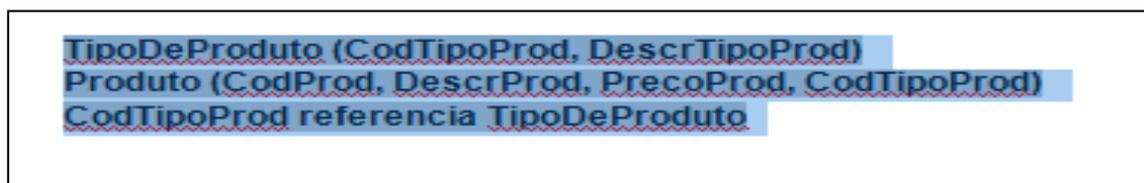
A figura 6 representa a forma seguinte do modelo entidade-relacionamento; que é a fase lógica. Nessa fase, os atributos viram campos e podem se transformarem em chaves (característica única para identificar a unidade da entidade e criar um conceito de relacionamento no banco de dados).

Figura 6: O modelo Lógico em diagrama



Fonte: O próprio autor, 2017.

“A etapa de projeto lógico objetiva transformar o modelo conceitual obtido na primeira fase em um modelo lógico. O modelo lógico define como o banco de dados será implementado em um SGBD específico.”(HEUSER, 2004)

Figura 7: O modelo Lógico descrito em *scripts*

Fonte: O próprio autor, 2017.

A modelagem lógica do banco serve para compreender a descrição da estrutura que será armazenada no banco.

Segundo HEUSER, o modelo lógico também pode ser representado como na figura 7.

Figura 8: O modelo Físico

```

CREATE TABLE ALUNO (
  ALU_COD NUMERIC PRIMARY KEY,
  ALU_LOGIN VARCHAR(20),
  ALU_SALDO DECIMAL,
  ALU_SENHA VARCHAR(20)
)

CREATE TABLE IMPRESAO (
  IMP_COD NUMERIC PRIMARY KEY,
  IMP_ARQ VARCHAR(20),
  IMP_QUANTCOP NUMERIC,
  IMP_COR BOOLEAN,
  IMP_FRENTE BOOLEAN,
  IMP_PAISAG BOOLEAN,
  IMP_PAG NUMERIC,
  ATE_COD NUMERIC,
  ALU_COD NUMERIC,
  FOREIGN KEY(ALU_COD) REFERENCES ALUNO (ALU_COD)
)

CREATE TABLE ATENDENTE (
  ATE_COD NUMERIC PRIMARY KEY,
  ATE_LOGIN VARCHAR(20),
  ATE_SENHA VARCHAR(20)
)

CREATE TABLE CREDITO (
  CRE_COD NUMERIC PRIMARY KEY,
  CRE_SERIAL VARCHAR(20),
  CRE_VALOR DECIMAL,
  ALU_COD NUMERIC,
  ATE_COD NUMERIC,
  FOREIGN KEY(ALU_COD) REFERENCES ALUNO (ALU_COD),
  FOREIGN KEY(ATE_COD) REFERENCES ATENDENTE (ATE_COD)
)

ALTER TABLE IMPRESAO ADD FOREIGN KEY(ATE_COD) REFERENCES ATENDENTE (ATE_COD)

```

Fonte: O próprio autor, 2017.

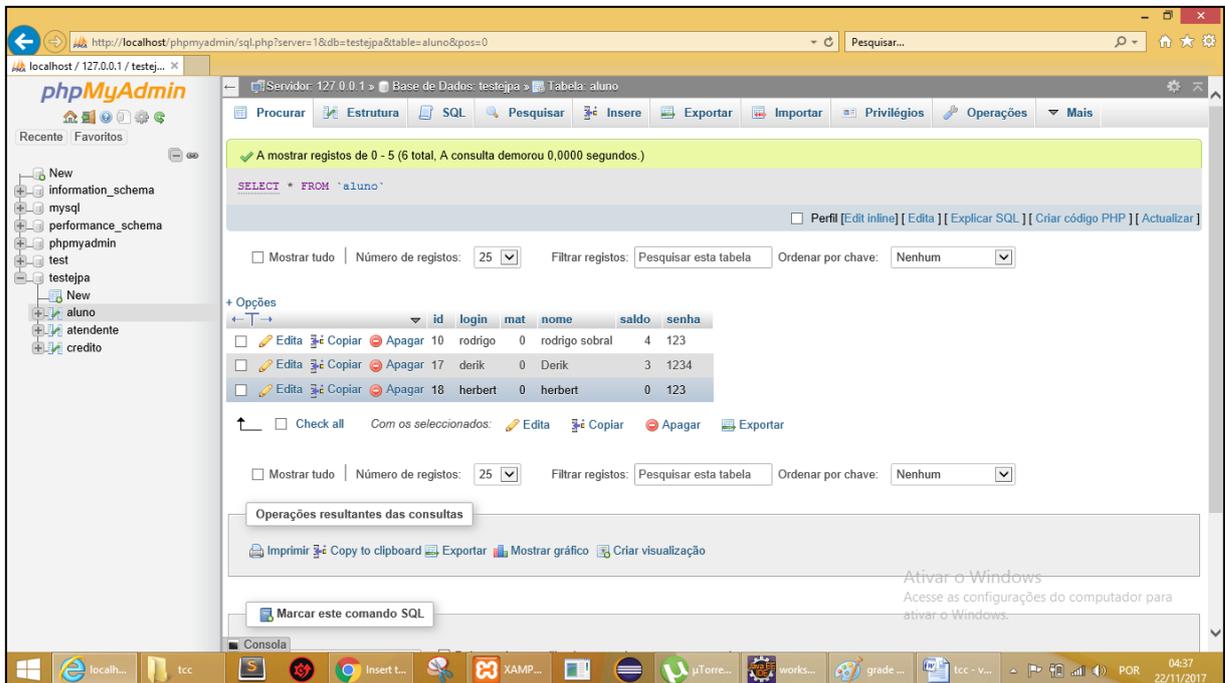
De acordo com Lopes, o modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD. Assim, esse modelo depende do SGBD que está sendo usado.

No modelo lógico detalha-se a estrutura física do banco, como tabelas, campos, tipos de valores, índices, etc. A figura 8 mostra a criação da estrutura do banco de dados em linguagem SQL.

Abaixo, segue a figura tirada de um *printscren* do sistema phpAdmin que é

uma ferramenta utilizada para simular um administrador de banco como se estivesse no servidor, com alguns dados que exemplificam a organização de dados no banco.

Figura 9: Representação da organização do banco de dados através do phpAdmin.

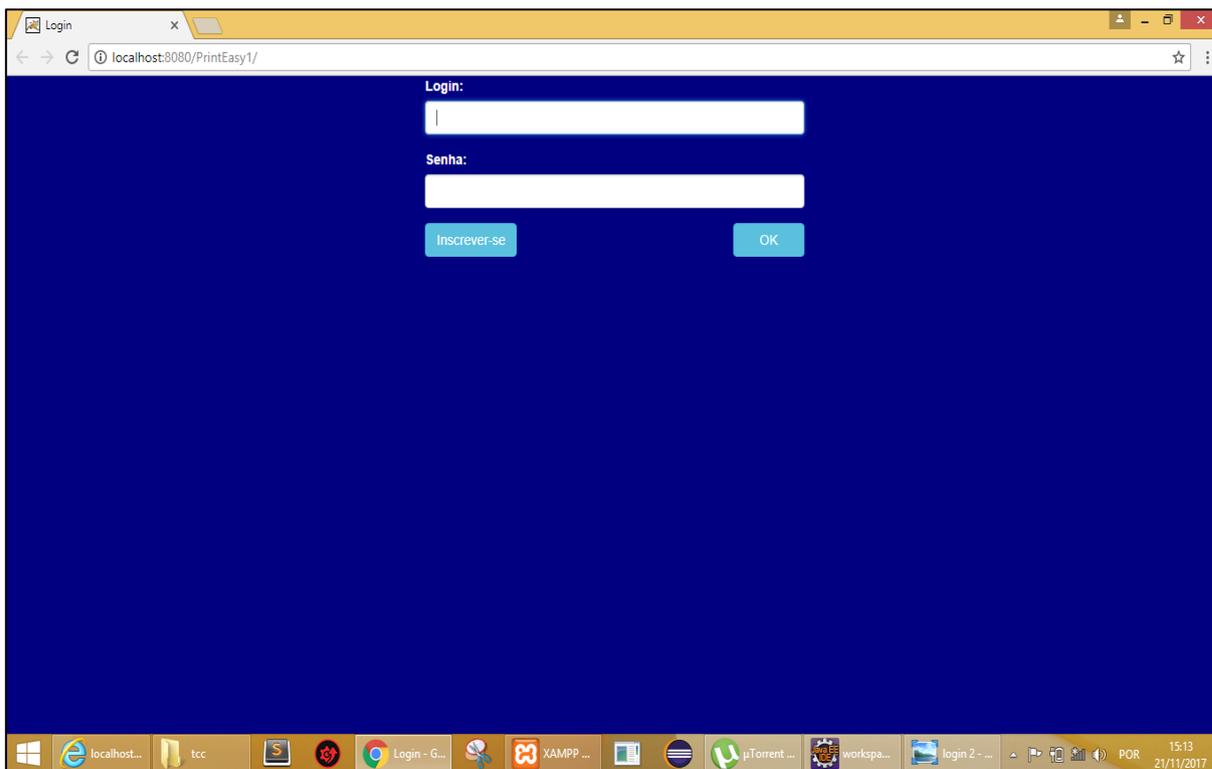


Fonte: o próprio autor, 2017.

4. LAYOUT

O projeto que nós propomos tem a característica de ser um controle de crédito para que os seus usuários tirem impressões, a partir de suas contas.

Por isso que fizemos uma primeira pagina para o *login* do usuário, como mostra a figura 10.

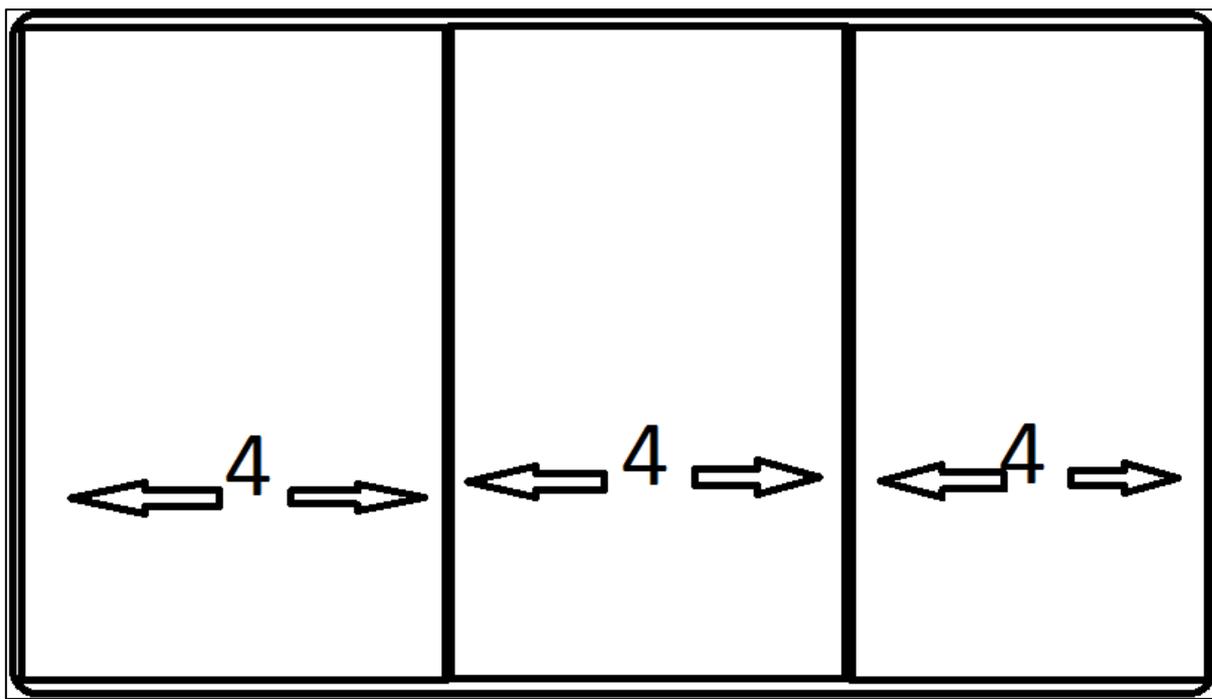
Figura 10: Pagina de *Login*

Fonte: o próprio autor, 2017.

Todo o Layout do sistema foi feito utilizando Bootstrap. Usamos a técnica de grade ou '*table*' para a centralização do conteúdo no navegador.

Essa divisão de conteúdo em grade tem a forma de uma tabela com três colunas com tamanhos iguais. Como a divisão de Bootstrap trabalha com a idealização relativa do número 12, dividimos o espaço visual do sistema de acordo com a figura 11.

Figura 11: Divisão da página utilizando o conceito de grade

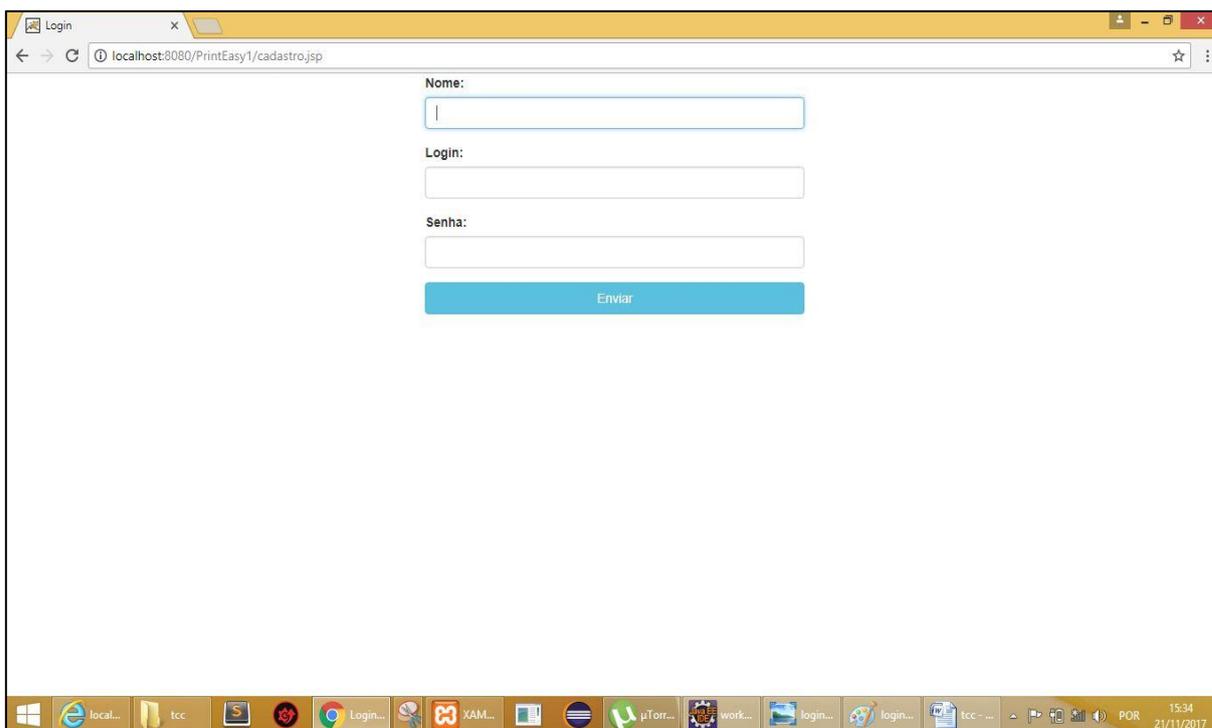


Fonte – o próprio autor, 2017.

Somando os espaços da página temos o tamanho 12 que é referente à extensão de toda a página. Para a organização e centralização do conteúdo, utilizamos apenas a grade do meio deixando espaço branco nas laterais.

Na figura 12, temos o cadastro de usuário. A página contém as informações necessárias para cadastrar o aluno; permitindo que ele utilize normalmente o sistema.

Figura 12: Cadastro de alunos



The image shows a web browser window with a single tab titled "Login". The address bar displays "localhost:8080/PrintEasy1/cadastro.jsp". The main content area contains a registration form with the following fields and elements:

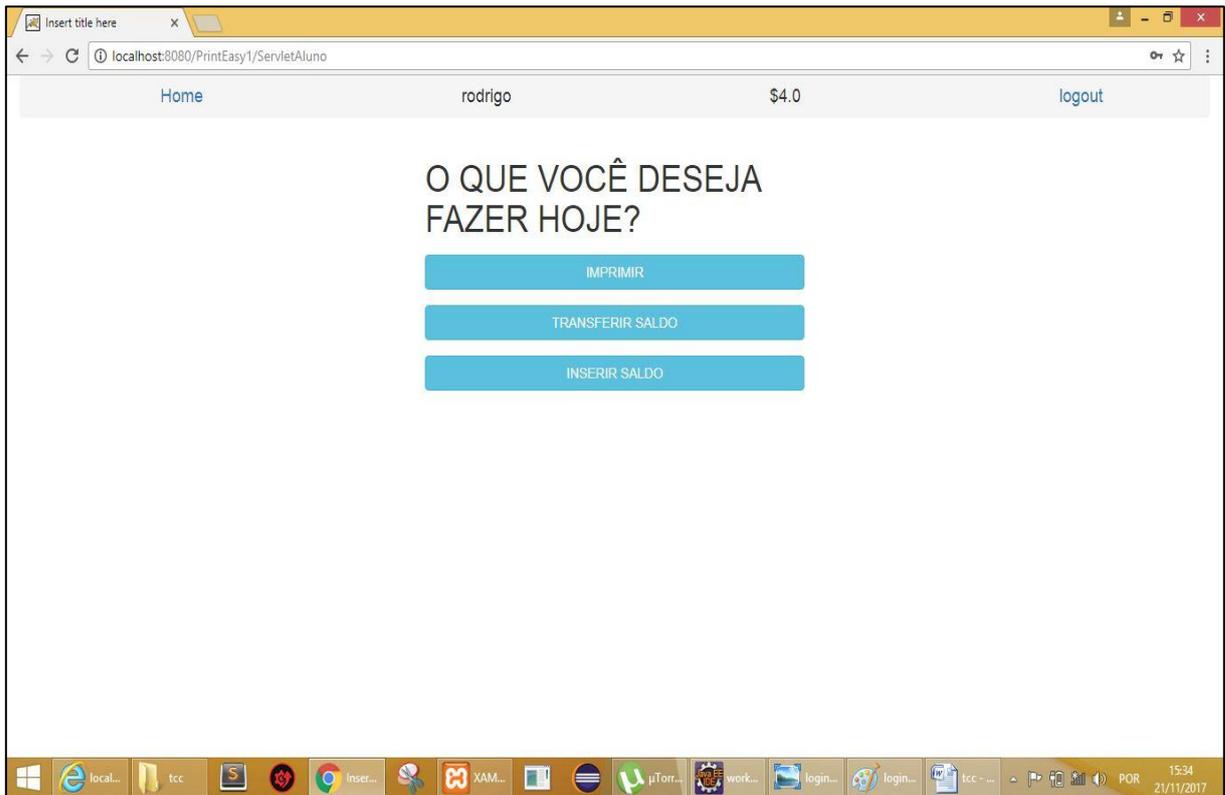
- Nome:** A text input field with a vertical cursor.
- Login:** A text input field.
- Senha:** A text input field.
- Enviar:** A blue button with white text.

The Windows taskbar is visible at the bottom, showing various application icons and the system tray with the date and time (15:34, 21/11/2017).

Fonte – o próprio autor, 2017.

Para a utilização do sistema, ao logar, o usuário tem informações básicas da sua conta no menu inicial que fica na parte superior da tela, como saldo, login e o botão de sair do sistema, como mostra na figura 13

Figura 13: Menu do Aluno

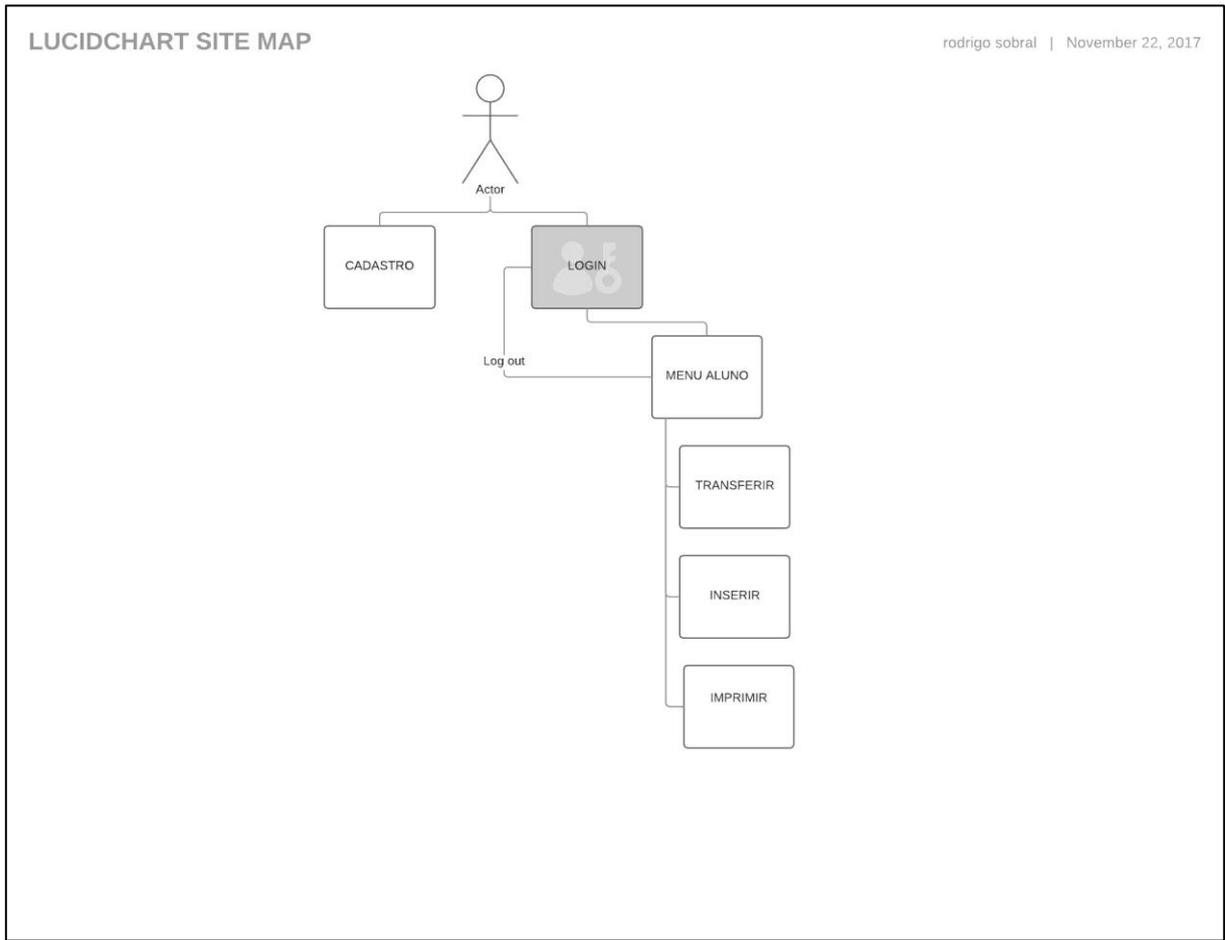


Fonte – o próprio autor, 2017.

As telas para que o aluno imprima o arquivo, insira o crédito e transfira o saldo para outro aluno também são organizadas com o conceito de grade, e também mostram o menu de dados do usuário.

Todas as demais telas do sistema utilizam o conceito de centralização de grade 4x4x4. Essa padronização permite que o usuário se acostume com as funcionalidades do sistema e crie uma memória visual para as futuras utilizações do sistema.

Figura 14: Mapa de navegação do aluno no sistema



Fonte: o próprio autor, 2017.

A figura 14 mostra a navegação do usuário no sistema. Assim que ele acessa o sistema, ele está na página de login. O usuário pode ou não logar direto no sistema.

Caso o aluno não tenha usuário cadastrado, ele terá que fazer uma conta no sistema acessando a página de cadastro para se inscrever.

Após fazer o login, o aluno entra na página do menu do aluno e escolhe imprimir, transferir saldo para outro aluno ou inserir crédito em sua conta.

5 CONCLUSÃO

O Projeto mostrou que é possível criar um sistema para aprimorar o processo de acompanhamento do ato de imprimir documentos em fotocopiadoras.

O levantamento de requisitos nos forneceu todas as premissas e necessidades para a execução e organização, de forma procedimental, durante a fase de desenvolvimento do sistema da *PrintEasy* para que o projeto de prototipação do sistema de banco de impressões evoluísse.

A documentação criada serviu para a orientação no processo de evolução do projeto e nos permitiu tomar decisões assertivas em tempo hábil; isso preveniu que o projeto atrasasse as suas entregas. Além de orientar a equipe, a documentação também poderia vir a servir como contrato com o cliente, pois o mesmo poderia mudar de ideia e constantemente atrapalhando a evolução do protótipo; o que não veio a acontecer.

Como o projeto foi um protótipo de um sistema para um nicho de mercado que geralmente não utiliza softwares, tecnologia web ou afins, o desenvolvimento de prototipação continuada nos ajudou a entender melhor as necessidades mais complexas do cliente, uma vez que podíamos fazer apresentações visuais do sistema para os alinhamentos junto ao cliente.

O protótipo buscou ter nível acadêmico, o cliente foi real e a intenção foi de prover uma solução para uma demanda de esforço que já existia foi verdadeira. Concluímos então que toda a tecnologia e métodos acadêmicos utilizados neste projeto foram de suma importância para a evolução do mesmo.

REFERÊNCIA

ALVES, W. P. **Fundamentos de Bancos de Dados**. Érica, 2004

Booch, Grady. **UML: guia do usuário** / Grady Booch, James Rumbaugh, Ivar Jacobson; tradução de Fábio Freitas da Silva e Cristina de Amorim Machado. – Rio de Janeiro: Elsevier, 2012. – 12º impressão

BEZERRA, Eduardo. **Princípios de análise e projeto de sistema com UML**. Rio de JANEIRO: Elsevier, 2007. 6ª reimpressão.

Deitel, H. M. **Java, como programar** / H.M. Deitel e P. J. Deitel; trad. Carlos Arthur Lang Lisbôa. - 4.ed. - Porto Alegre : Bookman, 2003.

Guedes, Gilleanes T. A. **UML 2** : uma nova abordagem prática / Gilleanes T. A. Guedes. – 2.ed. – São Paulo : Novatec Editora, 2011.

Gonçalves, Edson. **Dominando Java Server Faces e Facelets Utilizando Spring 2.5**, Hibernate e JPA. Rio de Janeiro : Editora Ciência Moderna Ltda, 2008.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. Sagra Luzzatto, 2004.

Lopes, A. **AULA 11-12 Modelo Conceitual, Lógico e Físico, Entidade-Relacionamento**. IFRN, 2017. Acesso em: <<http://docente.ifrn.edu.br/abrahamlopes/2017.1-integrado/3.02401.1m-prog-bd/1o-bimestre/slide-modelos-diagramas-e-cardinalidade>>

Luckow, Décio Heinzeimann. **Programação Java para a Web** / Décio Heinzeimann Luckow e Alexandre Altair de Melo. – São Paulo : Novatec Editora, 2010.

Martins, C. A. **Java Server Faces 2 e BOOTSTRAP: design responsivo com o melhor de Java e Java Script - Web design responsivo com Bootstrap e JSF 2** - 148.ed. - Grajaú - RJ: Devmedia - Java Magazine, 2016.

NETO, Antônio Gonçalves dos Santos. **Java na Web**. Rio de Janeiro : Editora Ciência Moderna Ltda. 2011.

Nice Brito Machado Pereira, Tuani Zanatta. ARTIGO **SOBRE MVC (MODEL VIEW CONTROLLER)**. Disponível em:

<http://anacarol.blog.br/old/aulas/artigos_uteis/modelo_visualizacao_controle.pdf>

Acesso em: 03/10/ 2008.

Pressman, R. S. - **Engenharia de software**: uma abordagem profissional / Roger S. Pressman; tradução Ariovaldo Griesi, Mario Moro Fecchio ; revisão técnica Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade. - 7. ed. - Porto Alegre : AMGH, 2011.

Silberschatz, Abraham – **Sistemas de Banco de Dados** / Abraham Silberschatz, Henry F. Korth, S. Sudarschan ; tradução de Daniel Vieira – 5. Ed - . Rio de Janeiro : Elsevier, 2006.