

**UNIVERSIDADE FEDERAL DO PARANÁ**  
**SETOR LITORAL**

**LAÉRCIO JOSÉ MANIKA**

**PROJETO DE VIABILIDADE PARA FUTURA IMPLANTAÇÃO DE UM SISTEMA  
DE CONTROLE DE REFEIÇÃO NA UFPR-LITORAL**

**MATINHOS**

**2013**

**UNIVERSIDADE FEDERAL DO PARANÁ**  
**SETOR LITORAL**

LAÉRCIO JOSÉ MANIKA

**PROJETO DE VIABILIDADE PARA FUTURA IMPLANTAÇÃO DE UM SISTEMA  
DE CONTROLE DE REFEIÇÃO NA UFPR-LITORAL**

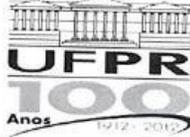
Trabalho de Conclusão de Curso apresentado  
para a obtenção do Título de Bacharel em  
Informática e Cidadania pela Universidade  
Federal do Paraná – Setor Litoral.

Orientador: Prof.º Neilor Fermino de Camargo

MATINHOS

2013

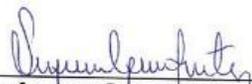
## TERMO DE APROVAÇÃO



### ATA DE AVALIAÇÃO DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

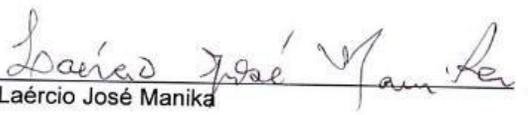
Aos vinte e três dias do mês de setembro de dois mil e treze, às nove horas, no Setor Litoral da Universidade Federal do Paraná, reuniu-se a banca avaliadora do trabalho de conclusão de curso, constituída pela professora Suzana Cini Freitas Nicolodi e pela professora Mirian Cristina Lopes sob a presidência do Orientador, Professor Neilor Fermino Camargo. O Trabalho de Conclusão do Curso de Bacharelado em Informática e Cidadania, do aluno Laércio José Manika, sob o título: "Implantação de um Sistema de Controle de Refeição na UFPR - Litoral", obteve o conceito APL. O aluno deverá efetuar as correções solicitadas pela banca e entregar a versão final em formato digital via CD-ROOM, até o dia 30 de setembro de dois mil e treze, na assessoria a Câmara do curso de Informática e Cidadania.

  
\_\_\_\_\_  
Neilor Fermino Camargo  
Professor Orientador

  
\_\_\_\_\_  
Professora Suzana Cini Freitas Nicolodi  
Membro da banca avaliadora

\_\_\_\_\_  
Suzana Cini Freitas Nicolodi  
Docente  
Matrícula 202246  
Setor Litoral - UFPR

  
\_\_\_\_\_  
Professora Mirian Cristina Lopes  
Membro da banca avaliadora

  
\_\_\_\_\_  
Laércio José Manika

## **UNIVERSIDADE FEDERAL DO PARANÁ**

Reitor: Prof. Dr. Zaki Akel Sobrinho

Vice-reitor: Prof. Dr. Rogério Andrade Mulinari

Diretor do Setor Litoral: Valdo José Cavallet

Diretor Administrativo: Renato Bochicchio

Diretor Pedagógico: Douglas Ortiz Hamermüller

Coordenador Administrativo: Jorge Luiz Lipinski

Coordenador do Curso de Informática e Cidadania: Prof. Dr. José Lannes de Melo

## AGRADECIMENTOS

Aos professores e colegas que estiveram junto nesta batalha, em fim “aos sobreviventes.”



“Se dois homens vêm andando por uma estrada, cada um com um pão, e, ao se encontrarem, trocarem os pães, cada um vai embora com um.

Se dois homens vêm andando por uma estrada, cada um com uma idéia, e, ao se encontrarem, trocarem as idéias, cada um vai embora com duas”.

Provérbio Chinês

## **RESUMO**

Neste trabalho será apresentado o desenvolvimento de um sistema para o controle diário das atividades do Restaurante Universitário visando a melhoria no atendimento dos seus clientes, bem como melhor controle das refeições, maior confiabilidade dos dados e agilidade proporcionada pelos relatórios, utilizando os conceitos de modelagem de software. Tal documentação auxiliará na implementação do sistema, vale salientar que todo o processo de modelagem bem como todos os softwares propostos neste projeto são livres, não ocasionando geração de custo.

**Palavras-chave:** Restaurante Universitário, modelagem de software, diagrama de classe, diagrama de caso de uso, diagrama sequencial.

## LISTA DE SIGLAS E ABREVIATURAS

RU		Restaurante Universitário
UML		Linguagem de Modelagem Unificada
SGBD		Sistema de Gerenciamento de Banco de Dados
PHP		Hypertext Preprocessor
HTTP		Hyper-Text Transfer Protocol
HTTPS		Hyper-Text Transfer Protocol Secure
SSL		Secure Socket Layer
FTP		File Transfer Protocol

## SUMÁRIO

### LISTA DE SIGLAS E ABREVIATURAS

INTRODUÇÃO.....	9
-----------------	---

### CAPÍTULO I - FUNDAMENTAÇÃO TEÓRICA

2.1 - DAS TECNOLOGIAS EMPREGADAS PARA O DESENVOLVIMENTO DO SISTEMA.....	10
2.2 - CICLOS DE VIDA DO SISTEMA.....	12

### CAPÍTULO II - METODOLOGIA

3.1- SISTEMA RU.....	16
3.1.3 – DIAGRAMAÇÃO UML.....	20

CAPÍTULO III - CONSIDERAÇÕES FINAIS.....	34
--	----

REFERÊNCIAS BIBLIGRÁFICAS .....	36
---------------------------------	----

## **INTRODUÇÃO**

Atualmente o uso da tecnologia da informação se faz presente tanto no mundo dos negócios, bem como em organizações públicas e privadas empresas. As informações atualizadas, seguras e rápidas implicam, muitas vezes, no sucesso ou fracasso de uma empresa.

O presente trabalho tem como objetivo estudar a viabilidade de implementação de um Sistema de Controle de Refeições para o Restaurante Universitário (RU) da Universidade Federal do Paraná - Setor Litoral, uma vez que o Restaurante atende professores, técnicos, estudantes, funcionários terceirizados, visitantes, servindo três refeições diárias aos clientes (café, almoço e jantar). Esta vasta gama de clientes e refeições levou-me a produzir um protótipo para a informatização deste setor, já que existem vários locais informatizados (biblioteca, sistema acadêmico, etc), também neste setor se faz necessário a informatização. O controle de refeições feito pelo RU é manual, estando sujeito a falhas.

## **CAPÍTULO I - FUNDAMENTAÇÃO TEÓRICA**

### **2.1 - DAS TECNOLOGIAS EMPREGADAS PARA O DESENVOLVIMENTO DO SISTEMA.**

Para o desenvolvimento do sistema usamos as seguintes tecnologias:

- Banco de dados MySql;
- Servidor Apache;
- Linguagem Php;
- SGBD Phpmyadmin;
- Editor de texto Geany;
- Editor de texto Bluefish;
- Editor de diagramação Dia;
- Sistema operacional Linux;
- Bibliotecas jquery.<sup>1</sup>

A escolha destas tecnologias se deu pelo fato de serem tecnologias de uso livre.

O MySQL foi escolhido por ser um banco de dados relacional, Multiencadeado, de código fonte aberto e nível corporativo. Nas últimas versões acrescentaram recursos tais como: transações (confirmação ou cancelamento da operação realizadas), triggers (gatilhos), stored procedure (procedimentos armazenados), views (visões). O tipo usado foi o InnoDB, por ter integridade referenciada, com implementação de constraints SET NULL, SET DEFAULT, RESTRICT E CASCADE, lock de registro, níveis de isolamento e armazenamentos de dados em tablespace. (LUKE, 2005)

A escolha pelo Servidor Apache se deu por ser um dos servidores mais conhecidos e usados no mundo, tem uma excelente performance, segurança compatível com várias plataformas de desenvolvimento e por sua enorme gama de recursos, além de possuir um módulo de PHP nativo dele. O servidor apache trabalha com diversos protocolos (HTTP, HTTPS, FTP, SSL) entre outros.

---

<sup>1</sup> [www.integartor.com.br/php](http://www.integartor.com.br/php)  
[http://ffb.virtual.ufc.br/solar/arquivos/curso/398/aula24\\_unid\\_ii\\_2.5.pdf](http://ffb.virtual.ufc.br/solar/arquivos/curso/398/aula24_unid_ii_2.5.pdf)  
<http://infomarcio.wordpress.com/2010/10/30/ciclo-de-vida-de-sistemas-de-informacao>  
<http://www.infowester.com/servapach.php>  
[http://www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2005/mono\\_leticia\\_marins.pdf](http://www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2005/mono_leticia_marins.pdf)

A linguagem PHP foi escolhida pois possui uma grande performance e estabilidade, a combinação Linux, Apache MySQL, PHP é muito forte, o PHP pode ser usado com um servidor web Apache, microsoft IIS, Personal Web Server, Netscape e vários outros . Além de SGDB (banco de dados): MySQL. SQL Server, Oracle, PostGreSQL, Informix, Dbase, ODBC, Frontbase, DBM, estes são apenas alguns bancos de dados. Pode ser usado em vários sistemas operacionais, Linux, Microsoft Windows, MAC OS, várias variantes UNIX e provavelmente outros mais, no PHP as linhas de programação não podem ser vistas por ninguém por serem executadas no servidor.

Os editores de texto foram o bluefish e o geany. O geany foi escolhido por oferecer vários snippets. (pedaços de códigos prontos que facilita na hora de programar). O editor de diagrama Dia foi utilizado para o banco de dados e UML. A biblioteca jquery por conter vários menus prontos.

Segue o ranking do PHP publicado pelo professor Mazzola<sup>2</sup>:

Figura 01: Ranking das linguagens 2013

---

<sup>2</sup>[http://professor.nizzola.com.br/item/M/texto/128/Ranking\\_das\\_Linguagens\\_de\\_Programa%C3%A7%C3%A3o\\_2013](http://professor.nizzola.com.br/item/M/texto/128/Ranking_das_Linguagens_de_Programa%C3%A7%C3%A3o_2013)

Position Apr 2013	Position Apr 2012	Delta in Position	Programming Language	Ratings Apr 2013	Delta Apr 2012	Status
1	1	=	C	17.862%	+0.31%	A
2	2	=	Java	17.681%	+0.65%	A
3	3	=	C++	9.714%	+0.82%	A
4	4	=	Objective-C	9.598%	+1.36%	A
5	5	=	C#	6.150%	-1.20%	A
6	6	=	PHP	5.428%	+0.14%	A
7	7	=	(Visual) Basic	4.699%	-0.26%	A
8	8	=	Python	4.442%	+0.78%	A
9	10	↑	Perl	2.335%	-0.05%	A
10	11	↑	Ruby	1.972%	+0.46%	A
11	9	↓↓	JavaScript	1.509%	-1.37%	A
12	14	↑↑	Visual Basic .NET	1.095%	+0.12%	A
13	15	↑↑	Lisp	0.905%	-0.05%	A
14	16	↑↑	Pascal	0.887%	+0.07%	A
15	13	↓↓	Delphi/Object Pascal	0.840%	-0.53%	A
16	32	↑↑↑↑↑↑↑↑↑↑	Bash	0.840%	+0.47%	A
17	18	↑	Transact-SQL	0.723%	-0.04%	A
18	12	↓↓↓↓↓	PL/SQL	0.715%	-0.66%	A
19	24	↑↑↑↑↑	Assembly	0.710%	+0.24%	A-
20	21	↑	Lua	0.650%	+0.08%	B

FONTE: tiobe.com

## 2.2 - CICLOS DE VIDA DO SISTEMA

Existem três formas de ciclos de vida para desenvolvimento de sistemas.

### 1 - Modelo em Cascata

É o modelo em que o sistema é feito de modo sequencial, após ser feita uma fase do projeto passa para a seguinte sem que haja interações entre as fases.

Figura 02: Sistema modelo em cascata.



FONTE: <http://infomarcio.files.wordpress.com/2010/10/modeloemcascata1.jpg>

## 2. Modelo de prototipação:

Neste modelo o desenvolvedor cria um modelo e através dele se fará a identificação dos requisitos do software, o usuário usa este protótipo e através deste protótipo o usuário irá pedir ao desenvolvedor novas atualizações do sistema.

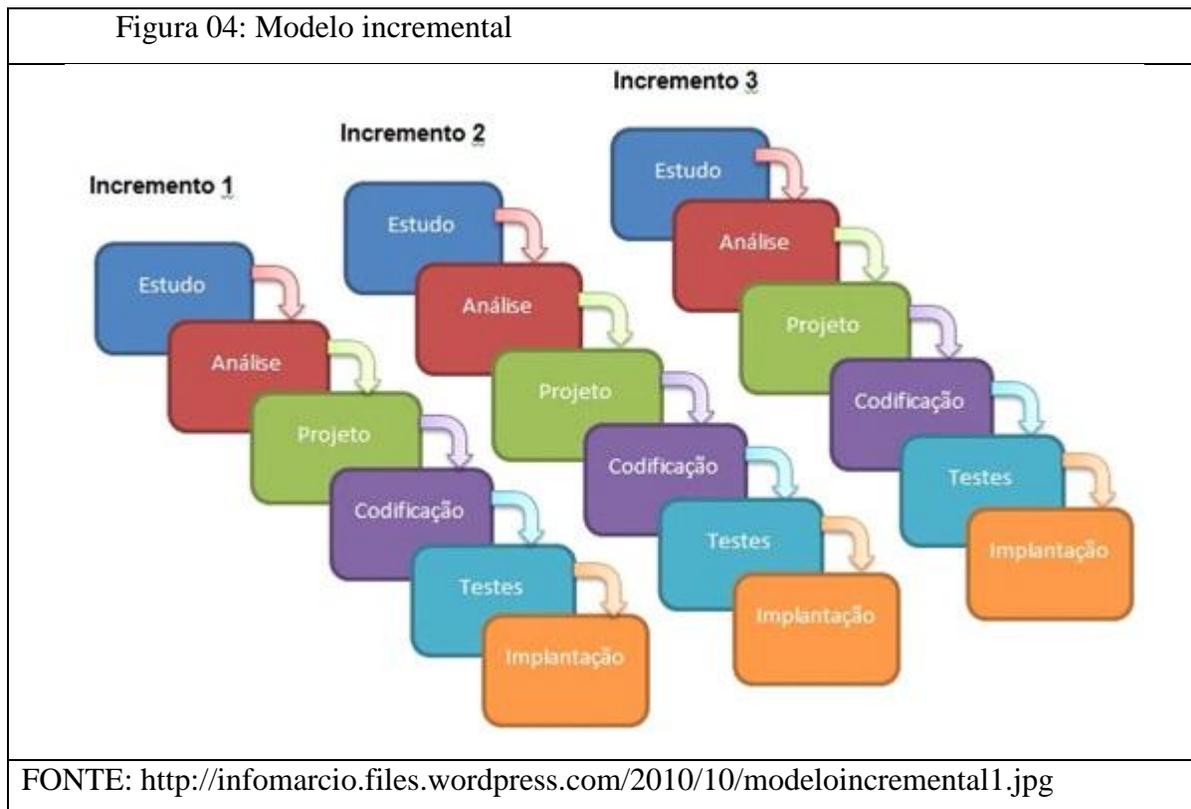
Figura 03: Modelo de Prototipação



FONTE: <http://infomarcio.files.wordpress.com/2010/10/prototipac3a7c3a3o1.jpg>

### 3. Modelo Incremental

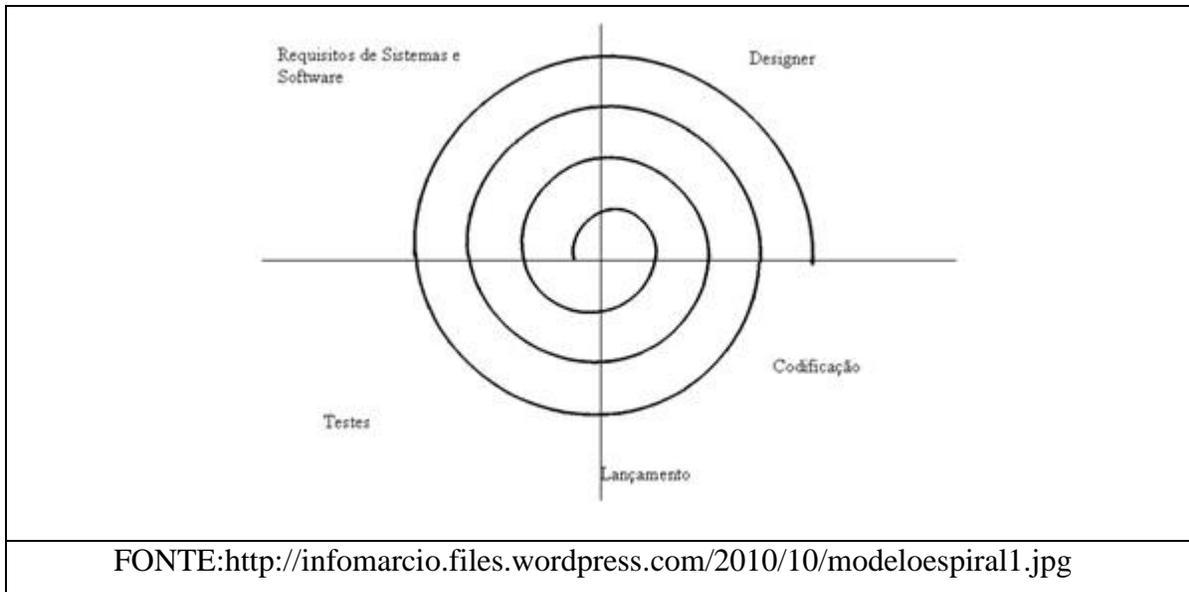
Este modelo mistura o modelo em cascata com o modelo de prototipação, como se demonstra na figura abaixo:



### 4. Modelo Espiral:

Este modelo utiliza a mistura do modelo em cascata com o de prototipação, isto é utiliza a repetição do modelo em protótipo misturada com a sequência linear do modelo em cascata. Figura abaixo.

Figura 05: Modelo espiral.



## CAPÍTULO II - METODOLOGIA

### 3.1. SISTEMA RU

Para desenvolvimento do software RU utilizaremos o modelo em espiral, isto é, faz-se uma espiral onde se repetirá até que o sistema esteja funcionando conforme a especificação do cliente, como mostra a figura acima. Os seguintes hardwares usados para o desenvolvimento: 1 Notebook Datasul NCL50 com 4GB de Memória DDR 3 e processador Pentium Dual Core, 2 Hd externo de 500GB Samsung, Leitor de código de barras Kaiomy

#### 1 - Requisitos Funcionais:

Requisitos funcionais são as funções do sistema, ou seja, é o que o sistema pode ou não fazer, são funções requeridas entre cliente e usuário, precisam ser consistentes e não contraditórias, nos requisitos funcionais o analista não irá especificar qual a linguagem, banco de dados, servidor, sistema operacional, isto é, só se preocuparia com que o sistema a ser desenvolvido vai fazer ou não.

#### 2 - Requisitos Funcionais RU:

1. Cadastra curso
2. Cadastra clientes
3. Cadastra refeições
4. Exibe consultas
5. Faz relatórios de refeição semanal, mensal e anual.

### **3 - Requisitos não funcionais**

Requisitos não funcionais são as restrições do sistema, funcionam ligados entre si, é quando o sistema só irá funcionar em um determinado Sistema Operacional, ou que vai utilizar determinado Servidor, eles tem de ser compatíveis entre si.

O servidor pode ser operado em ambiente Linux , Windows ou outro Sistema Operacional que suporte o Apache, Mysql, PhpMyadmin. O Banco de dados MySQL pode ser instalado no Servidor ou no próprio computador local. Necessita de:

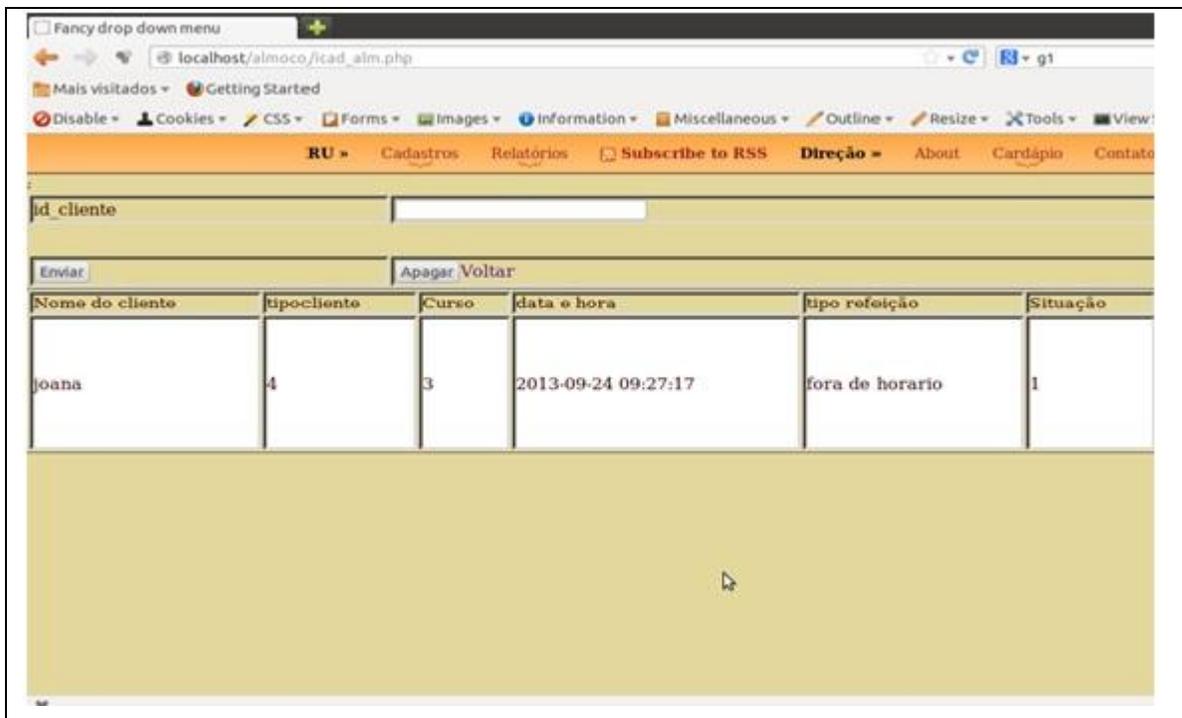
- Ter instalado Apache
- Ter instalado o banco de dados Mysql.
- Ter instalado o PhpMyadmin.

#### **3.1.1 - Processo de escolha dos pré-requisitos do sistema.**

##### **1. Primeira entrevista.**

Foi uma entrevista verbal onde o cliente queria um sistema que registrasse ao passar o código de barras do crachá, se o cliente era Professor, Técnico, Estudante e suas especificações tal como Estudante Franqueado, Estudante não Franqueado, nome do cliente, curso, tipo de refeição (conforme o horário o sistema insere automaticamente se a refeição é almoço, jantar, café ou a refeição está fora do horário), situação (1 para cliente ativo e 2 para inativo) . Ver protótipo abaixo.

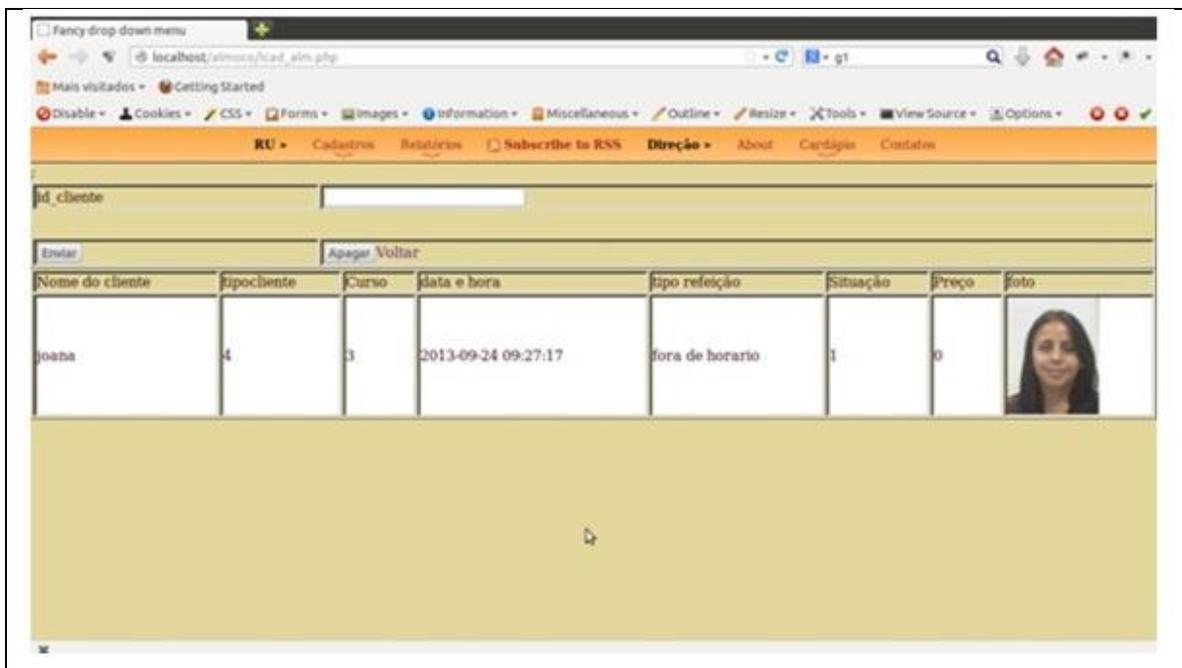
Figura 06: Protótipo 01



FONTE: O Autor

## 2. Segunda entrevista.

Na segunda entrevista feita verbalmente o cliente(Dono do RU) sugeriu que seria interessante que ao ser registrado o cliente(pessoa que faz a refeição) o sistema mostrasse a foto da pessoa que faz a refeição. Figura 07. Protótipo 02:



FONTE: O Autor.

### 3. Terceira Entrevista.

Na terceira entrevista o cliente mostrou uma foto do sistema RU do CEFET no qual ao cadastrar o cliente mostrava na tela a cor do cartão de cada cliente, laranja para técnicos, amarelo para Estudantes não Franqueado (isto é os estudantes que pagam as suas refeições), branco para Estudantes Franqueados (estudantes que não pagam suas refeições), verde para Professores, já que cada cliente paga de forma diferenciada.

Figura 08. Protótipo 03.

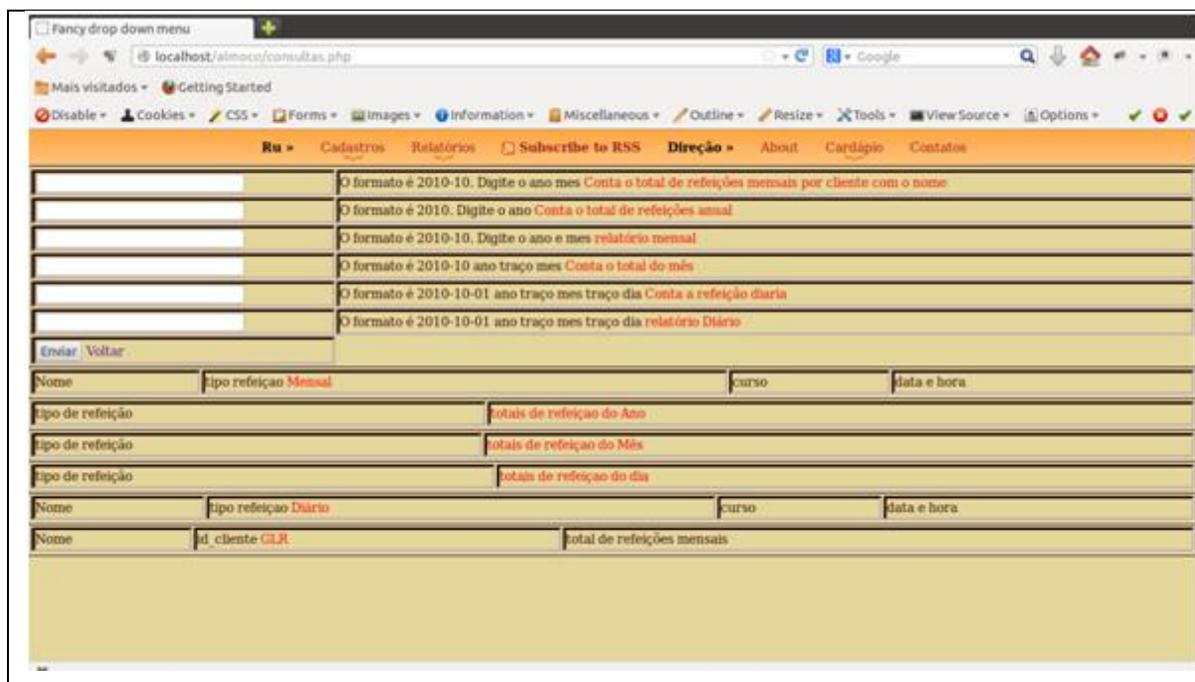
Nome do cliente	tipocliente	Curso	data e hora	tipo refeição	Situação	Preço	foto
Laércio José manika	2	2	2013-09-24 09:57:53	fora de horario	1	0	
Ricardo	1	3	2013-09-24 09:57:49	fora de horario	1	0	
Neitor	3	2	2013-09-24 09:57:45	fora de horario	1	0	

FONTE: O Autor

### 4. Quarta entrevista

Na quarta entrevista definimos quais os relatórios que seriam úteis para o Cliente (dono do Ru) foi feita através de formulários existentes. Abaixo foto e na foto os tipos de relatórios.

Figura 09: Relatório.



FONTE: O Autor

## 5. Quinta entrevista.

Depois de um tempo desenvolvendo o protótipo do Sistema Restaurante Universitário, com o sistema já funcionando, o proprietário do RU, pediu para fazer relatórios por preço diferenciado, pois a UFPR paga de forma diferente. No caso de um almoço para professor ser cobrado de forma diferente da do jantar, bem como as demais refeições feitas pelos outros clientes, isto é, o jantar é um pouco mais caro que o almoço, (esta parte ainda está para ser desenvolvida).

Como pode ser visto pelas entrevistas acima o desenvolvimento usando o modelo em espiral geram novos atributos e também muitas vezes se faz necessário a criação novas classes com novos métodos, além novas tabela e novos relacionamentos no banco de dados.

### 3.1.2 - Especificação do Sistema RU e requisitos funcionais.

Listagem de Casos de Uso RU:

- Incluir Cliente;
- Alterar Cliente;

- Incluir Curso;
- Incluir Refeição;
- Consulta - Conta o total de refeições mensais por cliente com o nome;
- Consulta - Conta o total de refeições anual;
- Consulta - Relatório mensal;
- Consulta - Conta o total de refeição do mês;
- Consulta - Conta o total de refeições diárias.
- Consulta - Relatório Diário

### **3.1.3 – DIAGRAMAÇÃO UML**

#### **3.1.4 - Como funcionam os casos de uso.**

Os casos de uso funcionam da seguinte maneira: o ator é o que pratica a ação, ele interage com o sistema e o caso de uso, o sistema por sua vez manda uma resposta para a ação efetuada pelo ator, vale lembrar que um mesmo ator pode ter vários casos de uso.

O estilo de escrita, segundo Waslawick (2011) é o tipo “ator informa/sistema informa/ evitar escrever “o sistema solicita” porque queremos representar os fluxos do sistema e não as solicitações que deram origem a seus fluxos. Usaremos [IN] para inclusão de eventos em sequência ou [OUT] para respostas do sistema em sequência, isto é [IN] para informar alguma coisa ao sistema e [OUT] para a resposta dada pelo sistema.

### **3.2 - CASOS DE USO RU**

#### **1. Caso de uso: Cadastrar Curso.**

[IN] O usuário cadastra o curso da UFPR-LITORAL.

#### **2. Caso de uso: Cadastrar Cliente.**

[IN] O usuário cadastra o cliente, com código de barras, nome, curso, situação, tipo de cliente e foto.

#### **3. Caso de uso: Cadastrar Refeição.**

[IN] O cliente passa graxa com código de barras no leitor.

[OUT] O sistema trás para a tela de cadastro os dados do cliente.

#### **4. Caso de uso: Relatórios.**

[IN] O gerente digita os dados para o tipo de relatório quer efetuar.

#### 4.1 Exceção: Curso já cadastrado.

[OUT] O sistema informa que o curso já está cadastrado.

Retorna ao menu principal.

#### Exceção 4.2: cliente já cadastrado.

[OUT] O sistema informa que o cliente já possui cadastro>Retorna ao menu principal.

#### Exceção 4.3: dados do cliente faltando ou fora dos padrões.

[OUT] O sistema informa os dados estão faltando ou que existem dados fora de padrão >

Retorna ao menu cadastrar cliente.

#### Exceção 4.4: O cliente já fez sua refeição.

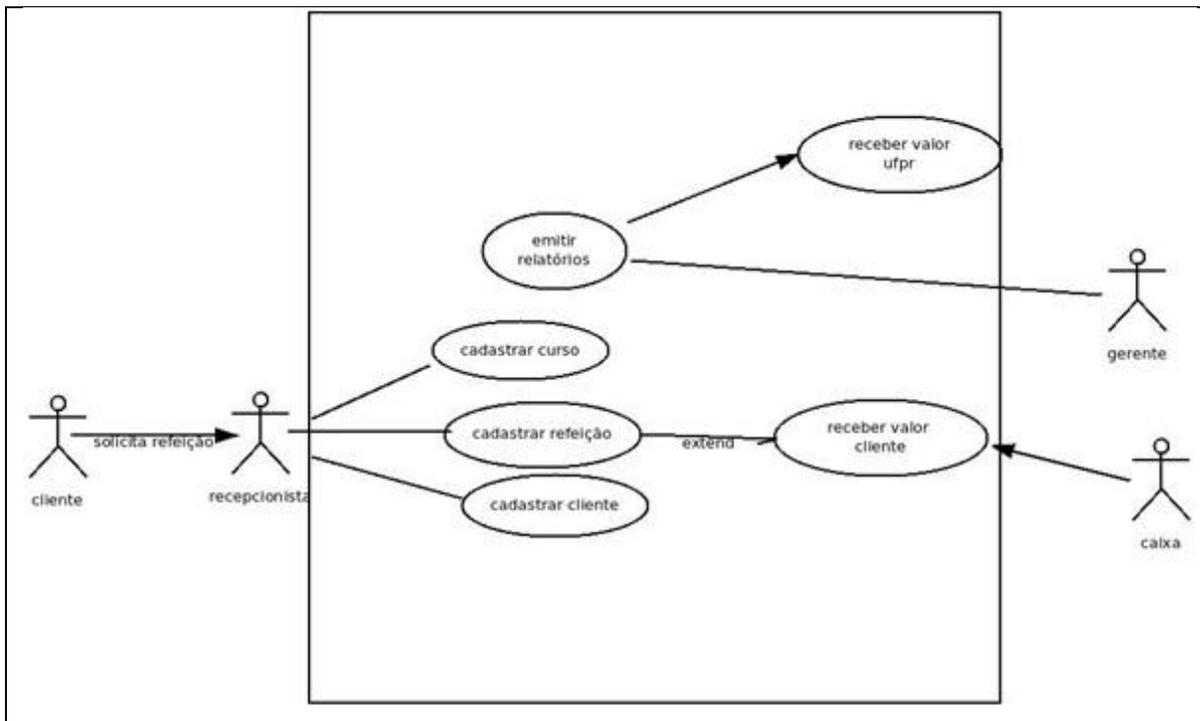
[OUT] O sistema informa que o cliente só pode fazer uma refeição por dia isto é só pode almoçar, jantar ou tomar café uma vez ao dia > O usuário do sistema cancela a refeição.

[IN] O Usuário do sistema verifica se a foto que está na tela do computador corresponde ao cliente que passou o crachá se não corresponder > O Usuário do sistema cancela a refeição e subtrai do Cliente o Crachá.

#### Exceção 4.5: O gerente digitou os dados para relatório fora do padrão.

[OUT] o Sistema informa que os dados são incompatíveis para este relatório > Retorna a página de relatórios.

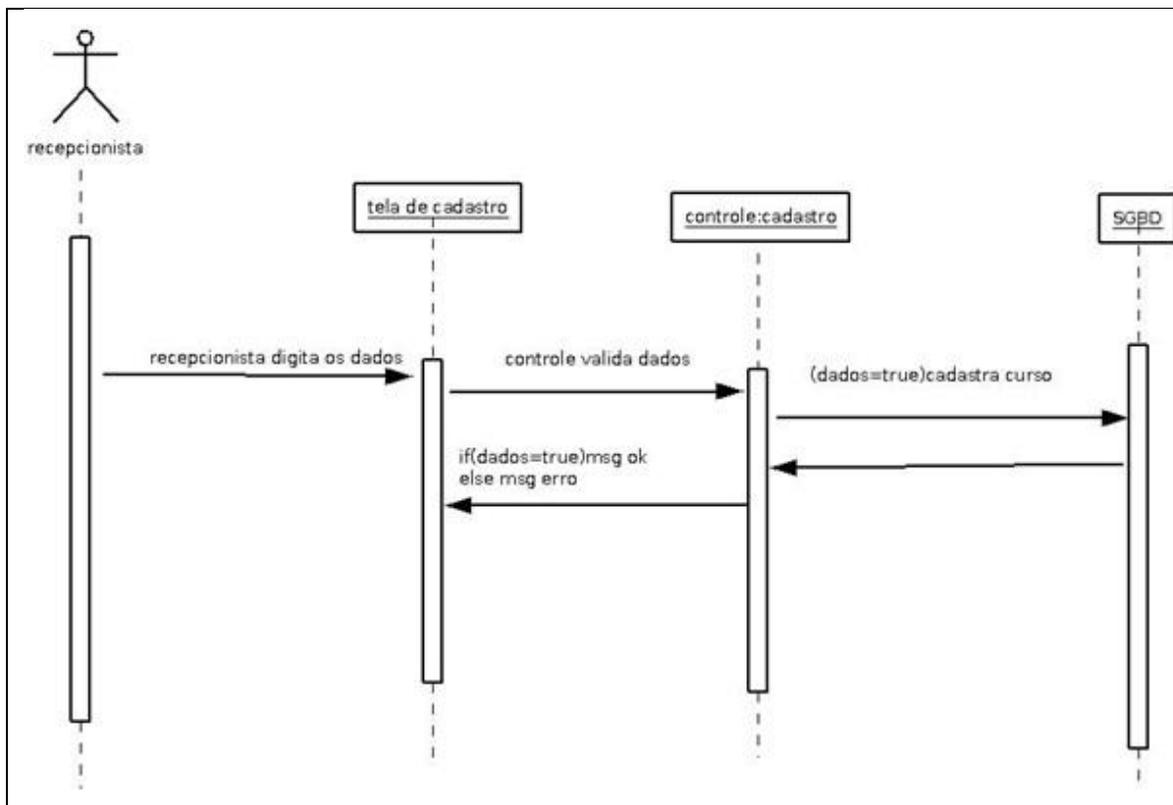
Figura 10: Diagrama de caso Sistema RU:



FONTE: O Autor

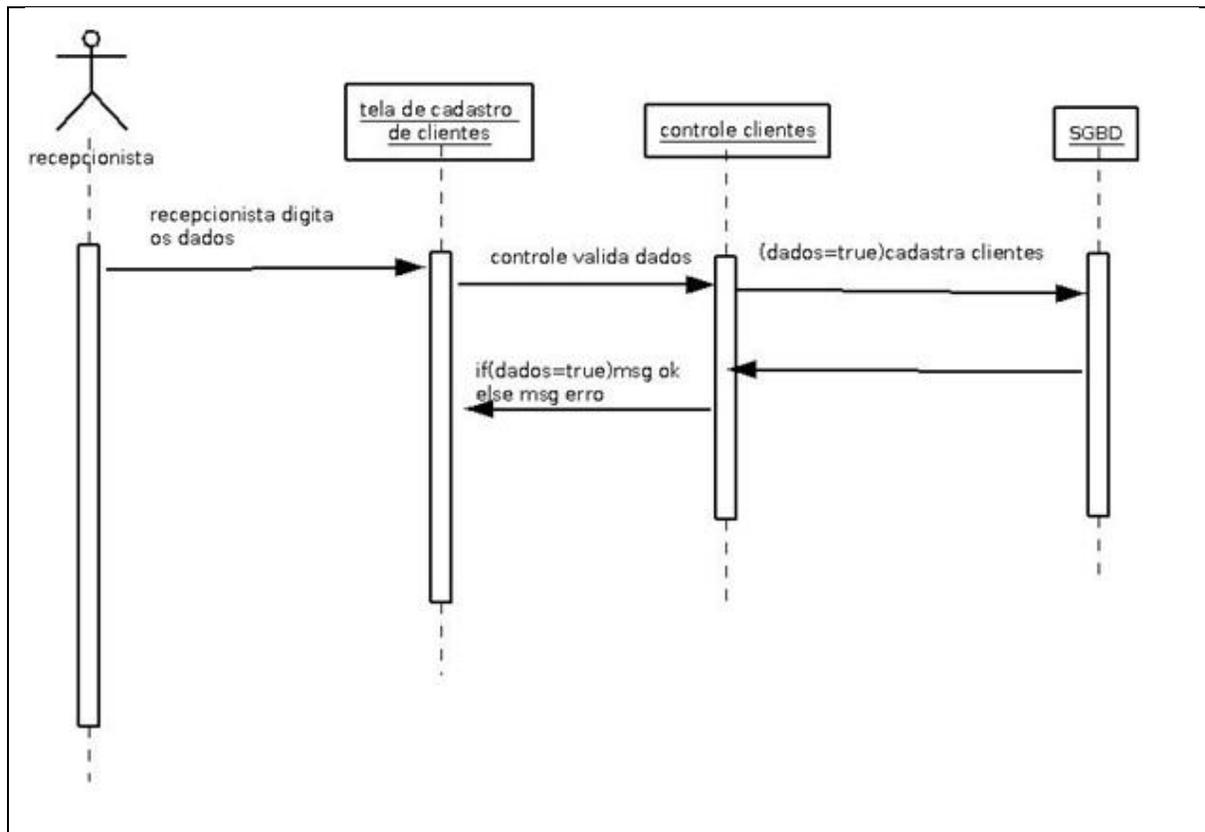
O diagrama de sequência para modelar a interação entre os objetos e normalmente estão associados a apenas um caso de uso, nos mostra a colaboração de cada objeto e é feito por setas horizontais que mostra o fluxo da aplicação, já as linhas verticais mostra o tempo de vida desta sequência.

Figura 11: Diagrama de sequencia de cadastro de curso.



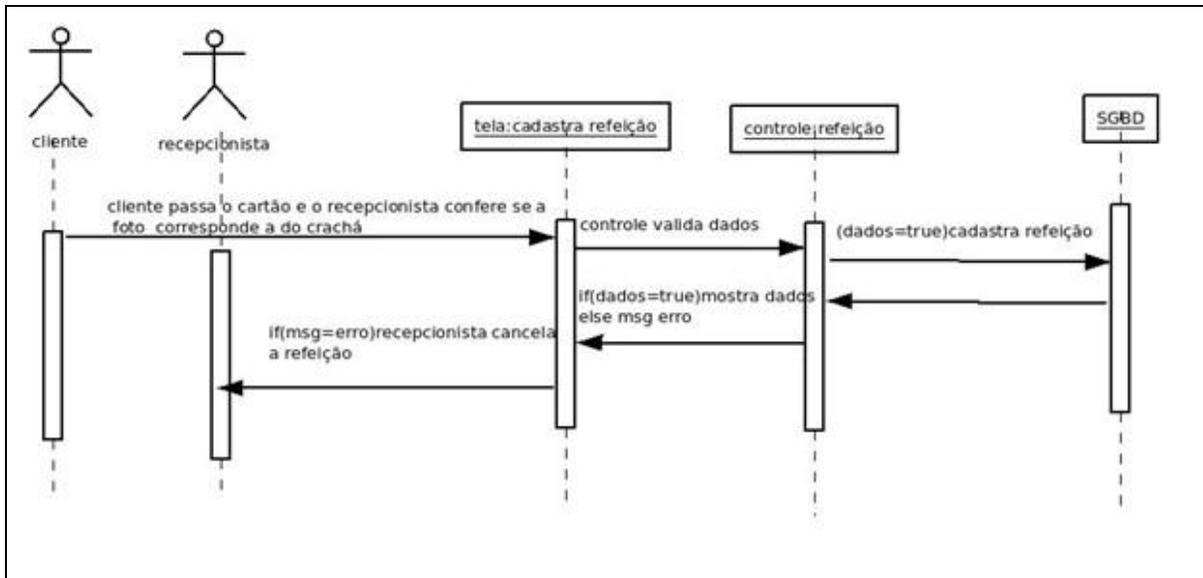
FONTE: O Autor.

Figura 12: Diagrama de sequencia de cadastro de clientes.



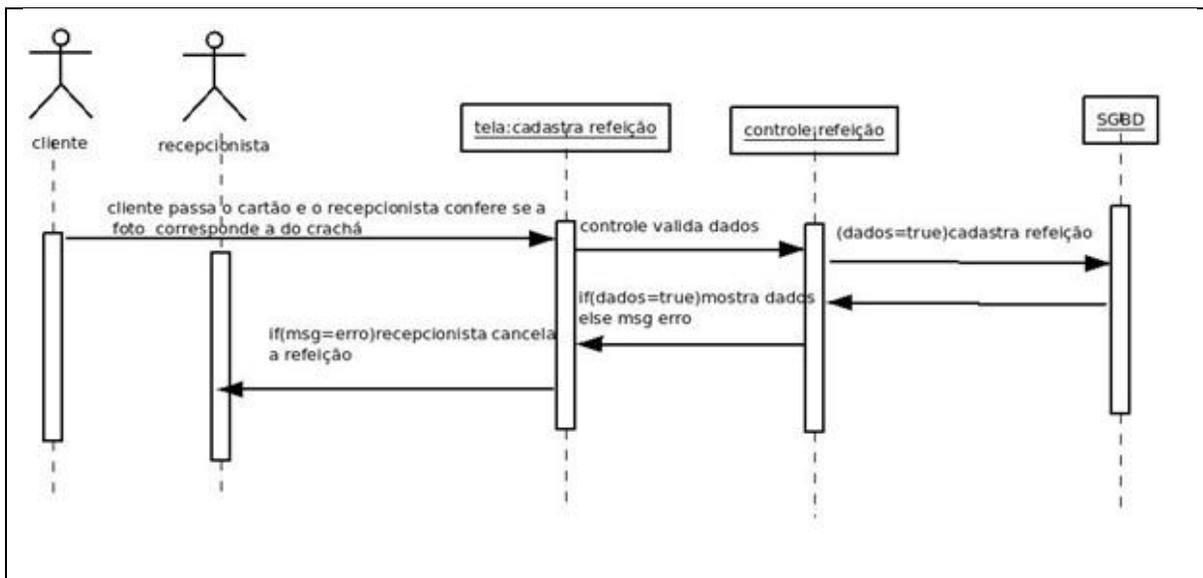
FONTE: O Autor.

Figura 13: Diagrama de sequencia de cadastro de refeição.



FONTE: O Autor.

Figura 14: Diagrama de sequencia de Relatórios.

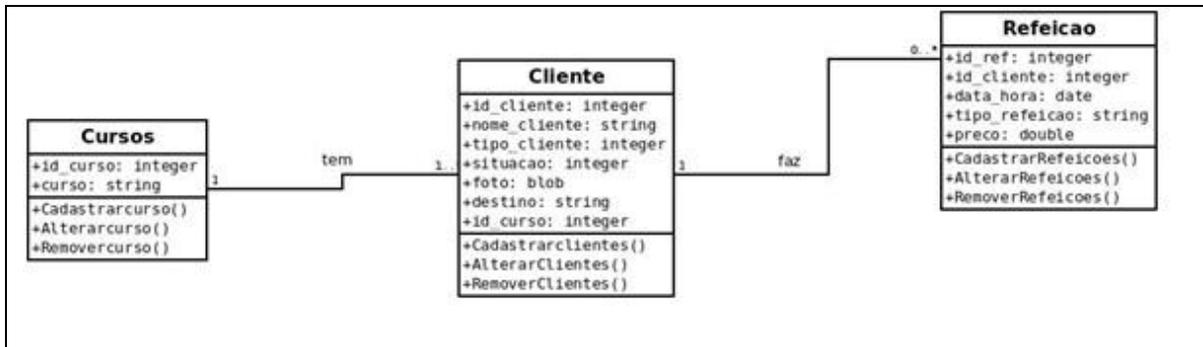


FONTE: O Autor.

### 3.2.1 - Diagrama de Classes RU

O diagrama de classes mostra a multiplicidade entre as classes assim como seus métodos.

Figura 15: Diagrama de classes



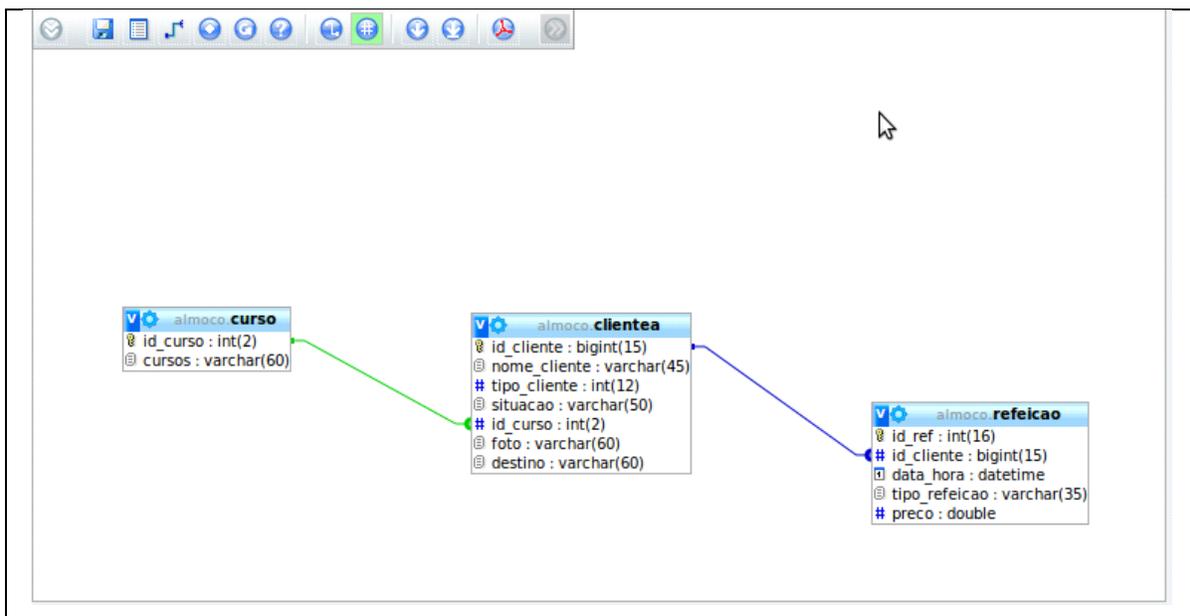
FONTE: O Autor.

A multiplicidade é representada por uma expressão numérica.

- “\*” representa o infinito,
- ponto-ponto (..) significa “até”,
- 0..\* zero até o infinito,
- 1..\* um até o infinito

Isto é um curso tem de um a vários clientes e um cliente pertence a apenas um curso, assim como um cliente pode fazer zero ou muitas refeições.

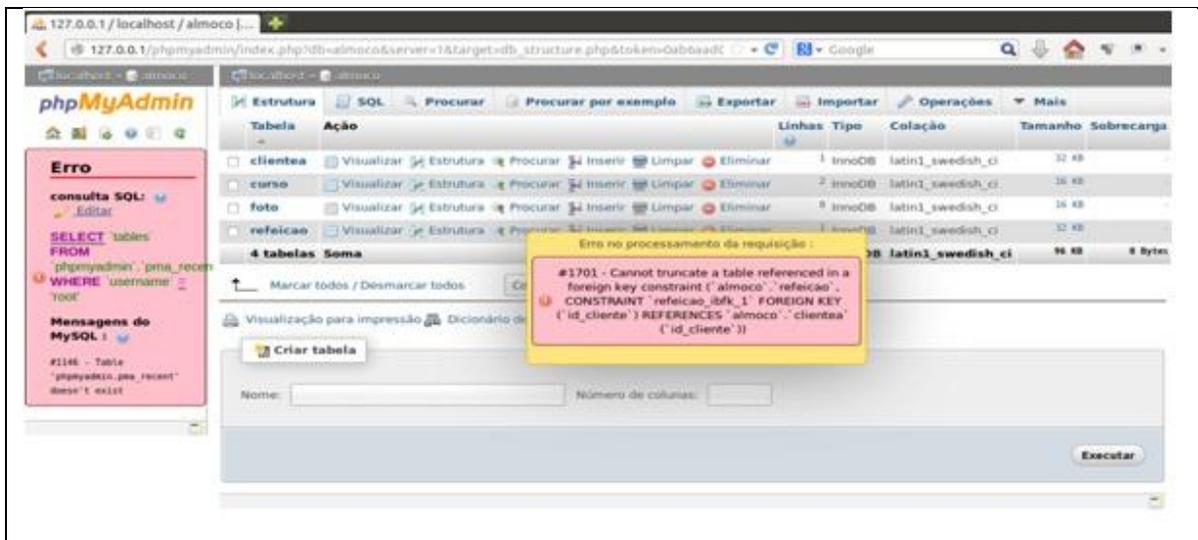
Abaixo o diagrama do banco de dado e seus relacionamentos. Figura 16.



FONTE: O Autor.

Acima estão os relacionamentos; as chaves primárias representadas por uma pequena chave, também conhecidas como *primary key* (PK) tem como função tornar um registro de uma tabela como único, geralmente são campos auto-incremento (auto-increment), uma chave primaria pode ser chave estrangeira de outra tabela, as chaves estrangeiras ou *foreign key* (FK) são usadas para relacionamentos entre tabelas no banco de dados muito útil para as consultas SQL, além de manterem a integridade referencial estas chaves são chamadas de *CONSTRAINTS*, ou seja, restrições que se estabelecem para uma determinada coluna de uma tabela do banco de dados podem ser CASCADE, RESTRICT, SET NULL E SET DEFAULT, quando definimos como CASCADE, os registros que pertencem as duas tabelas são deletadas automaticamente, quando definimos com RESTRICT, os registros de uma tabela não poderão ser deletados antes de se removerem os registros que se subordinam a chave de outra tabela, por exemplo, a tabela curso tem uma chave primária que é chave estrangeira da tabela cliente, isto é, não existe cliente sem curso se tentarmos excluir um curso o banco não deixará pois existe clientes cadastrados neste curso, teríamos que excluir primeiramente os clientes pertencentes da tabela clientes referentes a este curso, já em CASCADE quando excluirmos o registro do curso todos os registros que tem referência a tabela clientes serão deletados.

Figura 17. Como funciona as restrições ao tentar excluir um cliente que já fez alguma refeição.



FONTE: O Autor.







FONTE: O Autor.

Os dados que contém o fechamento semanal são importados para a planilha abaixo onde será feito o fechamento mensal para posterior cobrança do RU-Litoral junto a UFPR.

Figura 21.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
7																		
8	SEMANA 01		SEMANA 02		SEMANA 03		SEMANA 04		SEMANA 05		TOTAL	TOTAL	VALOR	VALOR	TOTAL	TOTAL		
9	ALM	JAN	ALM	JAN	ALM	JAN	ALM	JAN	ALM	JAN	ALMOÇO	JANTA	ALMOÇO	JANTA	ALMOÇO	JANTA		
10	ALUNO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
11	SERVIDOR TÉCNICO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
12	SERVIDOR DOCENTE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
13	FUNC. TERCEIRIZADO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
14	VISITANTE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
15	FRANQUEADO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
16	ESTAGIARIO/BOLSISTA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
17	ALUNO EXTRA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
18	FUNAI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
19	AUGM																RS -	RS -
20	EXTRAS T.A																RS -	RS -
21	EXTRAS T.D																RS -	RS -
22	ESPEC. S. SOCIAL																RS -	RS -
23	PROJETOS																RS -	RS -
24																	RS -	RS -
25																	RS -	RS -
26																	RS -	RS -
27																		
28	TOTAL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RS -	RS -
29																		
30																	TOTAL GERAL	RS -
31																		
32																		
33																		

FONTE: O Autor.

Segue abaixo como funcionara o RU com a automatização:

## 1. Página inicial

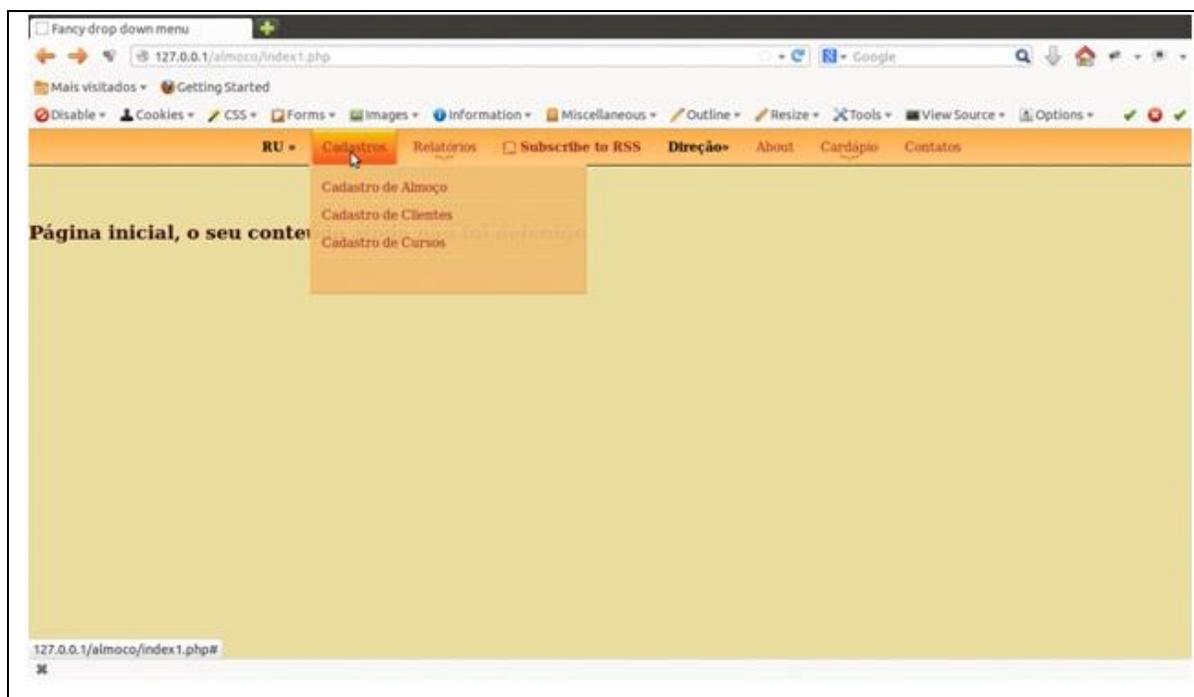
Contém na realidade o template da aplicação, pois todas as páginas são iguais diferenciando apenas no corpo da página, isto é, o menu cadastros contém o formulários para cadastro de cursos, refeições e clientes poderá futuramente conter outros formulários como o de cardápio se assim o desejarem. Em outro menu temos a página de relatórios que nos trás uma infinidade de consultas, tais como relatório mensal, anual e diário de refeições realizadas. Os links Direção, Contato e Cardápio serão definidos pelos usuários para verificar a sua utilização ou não. Abaixo segue as imagens das páginas.

Figura 22. Página Inicial.



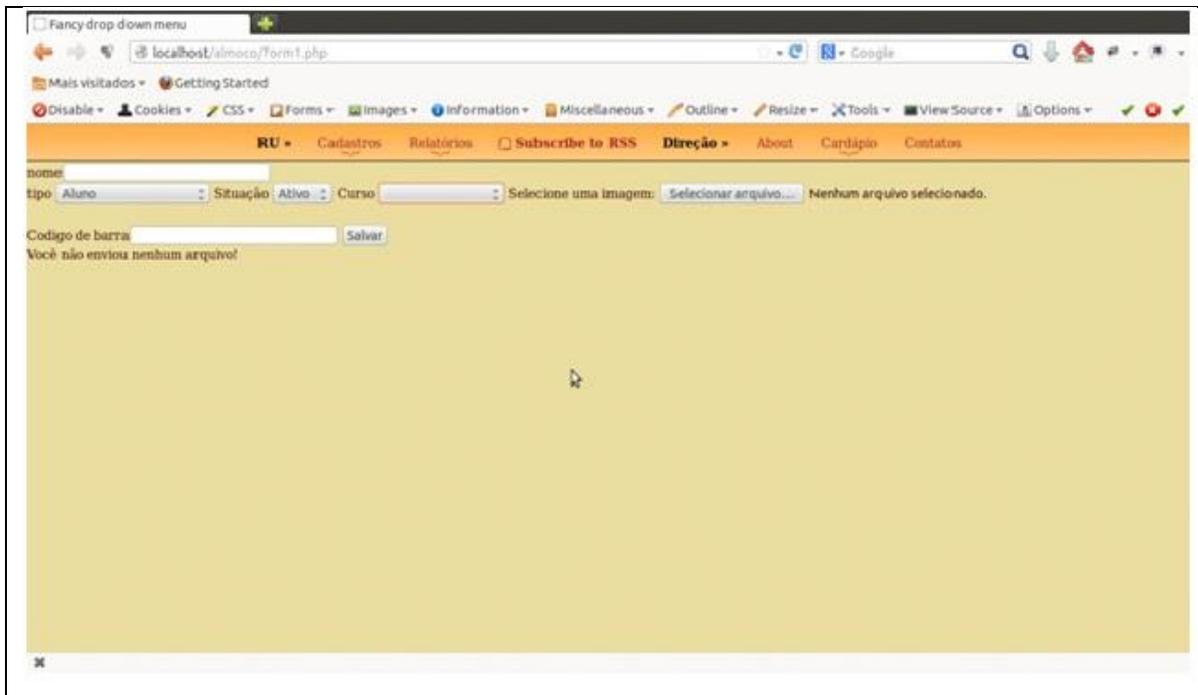
FONTE: O Autor.

Figura 23. Página de Cadastros.



FONTE: O Autor.

Figura 24. Página de cadastro de Clientes.

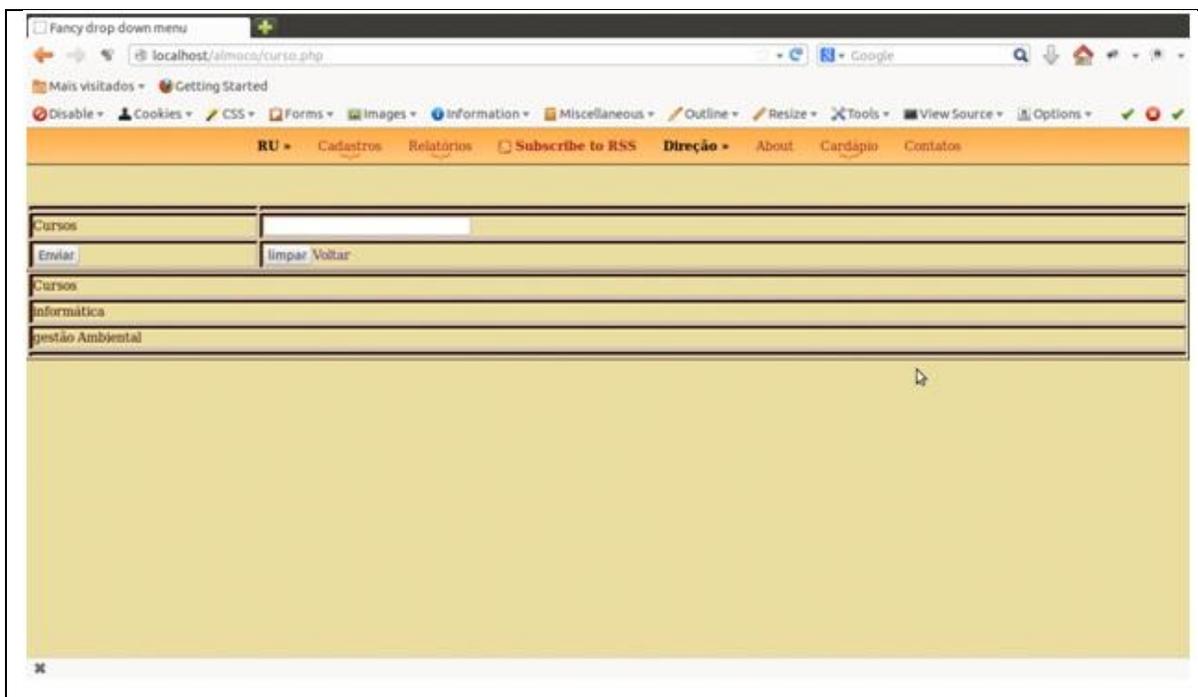


The screenshot shows a web browser window with the address bar displaying "localhost/almoco/Form1.php". The browser's menu bar includes "Mais visitados", "Getting Started", "Disable", "Cookies", "CSS", "Forms", "Images", "Information", "Miscellaneous", "Outline", "Resize", "Tools", "View Source", and "Options". The website's navigation bar is orange and contains "RU", "Cadastros", "Relatórios", "Subscribe to RSS", "Direção", "About", "Cardápio", and "Contatos". The main content area is a registration form with the following elements:

- A text input field for "nome".
- A dropdown menu for "tipo" with "Aluno" selected.
- A dropdown menu for "Situação" with "Ativo" selected.
- A dropdown menu for "Curso".
- An image selection area with the text "Selecione uma imagem:" and a button "Selecionar arquivo...". Below it, it says "Nenhum arquivo selecionado."
- A text input field for "Codigo de barra" and a "Salvar" button.
- A message: "Você não enviou nenhum arquivo!"

FONTE: O Autor.

Figura 25. Página de Cadastro de Cursos.



The screenshot shows a web browser window with the address bar displaying "localhost/almoco/curso.php". The browser's menu bar is the same as in Figure 24. The website's navigation bar is orange and contains "RU", "Cadastros", "Relatórios", "Subscribe to RSS", "Direção", "About", "Cardápio", and "Contatos". The main content area is a registration form with the following elements:

- A text input field for "Cursos".
- An "Enviar" button and a "limpar Voltar" button.
- A list of course names: "Curso", "informática", and "gestão Ambiental".

FONTE: O Autor.

Figura 26. Página de Cadastro de Refeições.

id\_cliente

Enviar Apagar Voltar

Nome do cliente	tipocliente	Curso	data e hora	tipo refeição	Situação	Preço	foto
Laércio José manika	2	2	2013-09-24 09:57:53	fora de horario	1	0	
Ricardo	1	3	2013-09-24 09:57:49	fora de horario	1	0	
Neilor	3	2	2013-09-24 09:57:45	fora de horario	1	0	

FONTE: O Autor.

Figura 27. Página de Relatórios.

O formato é 2010-10. Digite o ano mes Conta o total de refeições mensais por cliente com o nome

O formato é 2010. Digite o ano Conta o total de refeições anual

O formato é 2010-10. Digite o ano e mes relatório mensal

O formato é 2010-10 ano traço mes Conta o total do mês

O formato é 2010-10-01 ano traço mes traço dia Conta a refeição diaria

O formato é 2010-10-01 ano traço mes traço dia relatório Diário

Enviar Voltar

Nome	tipo refeição Mensal	curso	data e hora	total de refeição do Ano	total de refeição do Mês	total de refeição do dia
Nome	tipo refeição Diário	curso	data e hora	total de refeições mensais		

FONTE: O Autor.

As demais páginas podem ser definidas pelos usuários.

### **CAPÍTULO III - CONSIDERAÇÕES FINAIS**

A realização desse trabalho levou quatro anos, ou seja, todo o período da graduação. O primeiro contato para o desenvolvimento deste protótipo foi através do Diretor do Restaurante Central da UFPR – Curitiba, Lineu D'Lago que autorizou a execução, visto que o CCE se comprometeu em desenvolver tal projeto, porém o projeto não foi realizado. Uma das dificuldades encontradas foi a dificuldade de acesso ao banco de dados da PROGRADE que possui todos os dados necessários para o desenvolvimento tais como código de barras, foto, curso etc. Ao pedir acesso a estes dados no CCE enfrentei algumas exigências tal como o sistema teria que funcionar com catraca, já o dono do RU- LITORAL não quer o uso de catraca no seu Restaurante, as catracas existentes na UFPR vem com um software embarcado que é o ACESS do pacote OFICCE da Microsoft que é um pacote pago. Este projeto foi totalmente desenvolvido com software Free, sem custo algum. Os relatórios gerados para o RU- LITORAL seriam diferentes do RU-CENTRAL, este projeto seria específico ao RU-LITORAL, pois o RU-CENTRAL é administrado pela UFPR (administração pública) e o RU-LITORAL é de administração privada concedida através de concorrência.

O trabalho foi elaborado no modelo conceitual, apenas com quatro diagramas, diagrama de caso de uso, diagrama sequencial, diagrama de classe e o diagrama para modelagem do banco de dados. Os objetivos da modelagem do sistema visou compreender melhor o desenvolvimento do sistema, visualizando seus atores e os cenários onde atuariam. Também nos mostra o fluxo em que o sistema ocorrerá, bem como o comportamento dos objetos que atuaram e a sequência deste fluxo, já no diagrama de classe definimos seus atributos e seus métodos, devido ao pouco tempo para a modelagem e desenvolvimento, o projeto foi produtivo, pois ao examinarmos os diagramas temos a idéia de como o sistema funcionará e se irá atingir seus objetivos. Constata-se que o sistema não

está completo, pois objetiva-se transformar em um modelo MVC (*model, view, control*) orientado a objetos, esse é o próximo passo, visto que a implementação é viável.

## **REFERÊNCIAS BIBLIGRÁFICAS**

NIEDERAUER, Juliano. Desenvolvendo websites com PHP. 2.ed. São Paulo: Novatec Editora, 2011.

WAZLAWICK, Raul, Sidnei. Análise e projeto de sistemas de informação orientados a objetos. 2.º ed. Rio de Janeiro: Elsevier, 2011.

WELLING, Luke. PHP e MySQL desenvolvimento web. Rio de Janeiro: Elsevier, 2005.