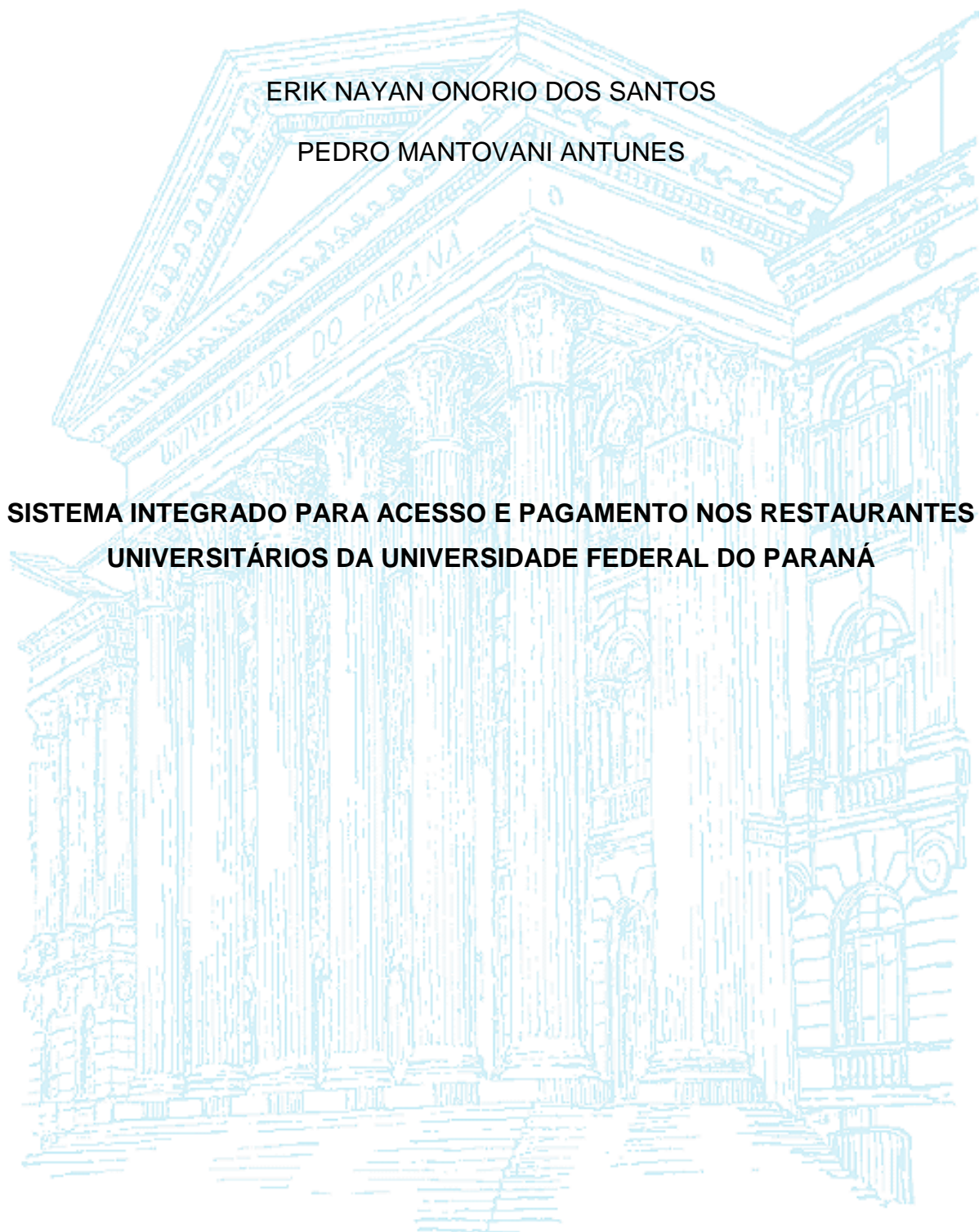


UNIVERSIDADE FEDERAL DO PARANÁ
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ERIK NAYAN ONORIO DOS SANTOS
PEDRO MANTOVANI ANTUNES

**SISTEMA INTEGRADO PARA ACESSO E PAGAMENTO NOS RESTAURANTES
UNIVERSITÁRIOS DA UNIVERSIDADE FEDERAL DO PARANÁ**



CURITIBA
2016

ERIK NAYAN ONORIO DOS SANTOS

PEDRO MANTOVANI ANTUNES

**SISTEMA INTEGRADO PARA ACESSO E PAGAMENTO NOS RESTAURANTES
UNIVERSITÁRIOS DA UNIVERSIDADE FEDERAL DO PARANÁ**

Trabalho apresentado como nota parcial à
disciplina de Projeto Integrado A do curso de
Engenharia Elétrica da Universidade Federal do
Paraná

Professor responsável: Prof. Dr. João da Silva
Dias

Orientador: Prof. Dr. Carlos Marcelo Pedroso

CURITIBA

2016

LISTA DE FIGURAS

FIGURA 1 - “MYCARD” DA UNIVERSIDADE DE LUXEMBURGO.....	10
FIGURA 2 - COMPONENTES E RECURSOS PRINCIPAIS DO SISTEMA.....	11
FIGURA 3 - SCRIPT DE AUTOMAÇÃO DO REPOSITÓRIO	17
FIGURA 4 - USE CASE DO SISTEMA	22
FIGURA 5 - AVALIAÇÃO DE DESEMPENHO (MOBILE)	24
FIGURA 6 - AVALIAÇÃO DE DESEMPENHO (DESKTOP).....	25
FIGURA 7 - SEGUNDA AVALIAÇÃO DE DESEMPENHO (COMPREENSÃO DA PERFORMANCE)	26
FIGURA 8 - SEGUNDA AVALIAÇÃO DE DESEMPENHO (TAMANHO DE ARQUIVOS).....	27
FIGURA 9 - SEGUNDA AVALIAÇÃO DE DESEMPENHO (NOTAS)	28
FIGURA 10 - FOTO DE TELA DO INÍCIO DA PÁGINA PRINCIPAL NO COMPUTADOR	29
FIGURA 11 - FOTO DE TELA DO INÍCIO DA PÁGINA PRINCIPAL NO CELULAR (PARTE 1)	30
FIGURA 12 - FOTO DE TELA DO INÍCIO DA PÁGINA PRINCIPAL NO CELULAR (PARTE 2)	31
FIGURA 13 - SPINNERS AO LADO DOS CAMPOS NUMÉRICOS	32
FIGURA 14 - CAMPO DE NÚMERO DE MATRÍCULA (EDGE).....	33
FIGURA 15 - CAMPO DE NÚMERO DE MATRÍCULA (FIREFOX)	33
FIGURA 16 - CAMPO DO NÚMERO DE MATRÍCULA (INTERNET EXPLORER)	33
FIGURA 17 - VISUALIZAÇÃO DO CONTEÚDO DA TABELA “USERS”	34
FIGURA 18 - ATUALIZAÇÃO DO CONTEÚDO DA TABELA “USERS”	34
FIGURA 19 - VISUALIZAÇÃO DO CONTEÚDO ATUALIZADO DA TABELA “USERS”	35
FIGURA 20 - SIMULAÇÃO DO FUNCIONAMENTO DO PROGRAMA DAS LEITORAS DE ACESSO	36
FIGURA 21 – PÁGINA DE CADASTRO.....	37
FIGURA 22 – PÁGINA DE SALDO E TRANSAÇÕES PÓS-LOGIN.....	38
FIGURA 23 – PÁGINA DE DADOS CADASTRAIS PÓS-LOGIN	38
FIGURA 24 – PÁGINA DE RECARGA PÓS-LOGIN	39

SUMÁRIO

1.	INTRODUÇÃO.....	6
1.1.	Objetivos	7
1.1.1.	Objetivo Geral	7
1.1.2.	Objetivos Específicos.....	7
1.2.	Estrutura do trabalho.....	8
2.	REVISÃO BIBLIOGRÁFICA E METODOLOGIA.....	9
2.1.	Visão Geral	9
2.2.	Proposta	10
2.3.	Servidor Central	12
2.3.1.	Servidor Web.....	12
2.3.2.	Servidor de Base de Dados	13
2.3.3.	Servidor de Email.....	13
2.4.	Cliente.....	13
2.5.	Aplicativo de Smartphone.....	14
2.6.	Validação.....	14
3.	DESENVOLVIMENTO	15
3.1.	Divisão do trabalho.....	15
3.2.	Servidor Linux.....	17
3.3.	Página Web	17
3.3.1.	Página principal	18
3.3.2.	Página de cadastro	19
3.3.3.	Página de login	19
3.3.4.	Página de pós-login	20
3.4.	MySQL	20
3.4.1.	Criptografia das senhas	21
3.5.	Programa das Leitoras de Acesso.....	21
4.	RESULTADOS PARCIAIS E DISCUSSÃO	22

4.1.	Página Web	23
4.1.1.	Avaliação de Desempenho	23
4.1.2.	Responsividade	28
4.1.3.	Compatibilidade entre navegadores	32
4.2.	Banco de Dados	34
4.3.	Programa das Leitoras de Acesso	35
4.4.	Visualização de páginas do site	37
5.	CONCLUSÃO	40
5.1.	Trabalhos Futuros	42
6.	REFERÊNCIAS	43
7.	APÊNDICE	46
7.1.	Script.sql	46
7.2.	Header.php	47
7.3.	Index.php	48
7.4.	Footer.php	51
7.5.	Login.php	52
7.6.	Login_page.php	53
7.7.	Register.php	54
7.8.	Register_page.php	56
7.9.	Register_sucess.php	58
7.10.	Style.css	59
7.11.	CAPTCHA_VALIDATION.PHP	62
7.12.	SEND_EMAIL.PHP	63
7.13.	CONFIRMATION.PHP	64
7.14.	DB_OPERATIONS.PHP	65
7.15.	UTILITIES.PHP	68
7.16.	WELCOME.PHP	70
7.17.	RECHARGE.PHP	72
7.18.	PERSONALINFO.PHP	76

1. INTRODUÇÃO

O Restaurante Universitário da UFPR (Universidade Federal do Paraná), mais conhecido entre a comunidade como “RU”, é um importante serviço para discentes, docentes e técnicos administrativos da universidade, pois proporciona refeições de qualidade, nutricionalmente adequadas e por um custo acessível a todos que o utilizam.

O Restaurante Universitário teve sua origem em 05/08/1966, administrado por membros do Diretório Central dos Estudantes (DCE). A partir de 1980 o Restaurante passou a ter administração da UFPR, quando recebeu, então, a denominação “Restaurante Universitário” (PRÓ-REITORIA DE ADMINISTRAÇÃO DA UFPR, 2016).

Após longo período utilizando sistema convencional de distribuição/servimento “porcionado” – onde os usuários são servidos de porções pré-estabelecidas de alimentos – em 1995, o RU Central adotou o sistema “self-service”, sendo porcionadas apenas o prato proteico (carnes) e a sobremesa, o que proporcionou maior conforto ao usuário em determinar as quantidades desejadas (PRÓ-REITORIA DE ADMINISTRAÇÃO DA UFPR, 2016).

Recentemente, o RU completou 50 anos de existência. De lá para cá, muitas mudanças ocorreram tanto na universidade, quanto no corpo discente/docente que o frequentam. Como é de se esperar, a comunidade acadêmica da UFPR aumentou e a tendência é de que continue a crescer (PROPLAN UFPR, 2014).

Atualmente, o RU conta com 4 unidades em Curitiba e outras 4 nos campi do interior (PRÓ-REITORIA DE ADMINISTRAÇÃO DA UFPR, 2016). Na conjuntura atual, é impossível não deparar-se com grandes filas e verdadeiros “congestionamentos” nos horários de pico, seja na sede do RU Politécnico, Central, Agrárias ou Jardim Botânico. Na verdade, o problema das longas filas não é recente: alunos já realizaram protestos a mais de 10 anos atrás cobrando melhorias (TRIBUNA PARANÁ, 2005). Outro problema grave e decorrente são as fraudes. Alunos que obtêm por mais de uma vez a mesma refeição, ou até

mesmo pessoas de fora da comunidade que acessam o Restaurante Universitário como membros da UFPR.

Sendo assim, o desenvolvimento de um sistema de acesso simples e ágil, mas que garanta a segurança e a comodidade dos que o utilizam torna-se pertinente. Além de gerar maior organização no acesso e diminuição no tamanho das filas, fraudes serão evitadas e, por consequência, diminuirá o custo do serviço já subsidiado pela UFPR, convertendo então em benefícios para os próprios estudantes, professores e técnicos.

1.1.OBJETIVOS

Os objetivos desse trabalho são apresentados a seguir, separados de forma a apresentar o objetivo geral e os objetivos específicos.

1.1.1. Objetivo Geral

Desenvolver um sistema para diminuir o tempo de fila de espera nos Restaurantes Universitários e aumentar a segurança contra fraudes, através de um cartão em que o usuário pode inserir créditos via plataforma digital, acessando os restaurantes de forma ágil e eficiente, sem problemas com a manipulação de dinheiro ou identificação.

1.1.2. Objetivos Específicos

Dentre os objetivos específicos destacam-se:

- desenvolver e implementar um *síte*, em servidor próprio, para que os usuários possam efetuar o seu cadastro e controlar seus créditos (Projeto Integrado A);

- criar e estruturar uma base de dados para controle e gerenciamento das informações dos usuários e suas transações (Projeto Integrado A);
- garantir a segurança das senhas armazenadas no servidor através de métodos de criptografia avançados e modernos (Projeto Integrado A);
- desenvolver um protótipo de cliente que irá atuar lendo códigos de barras e consultando a base de dados pela rede (Projeto Integrado B);
- utilizar uma plataforma segura e eficiente para inserção de créditos (Projeto Integrado C);
- disponibilizar um aplicativo para *smartphones* capaz de realizar consultas ao cadastro do usuário (Projeto Integrado D);
- enviar *emails* de confirmação automáticos após preenchimento do formulário de cadastro (Projeto Integrado A);
- ampliar a gama de utilizações do sistema para outros departamentos da universidade (empréstimo de equipamentos, biblioteca, etc) (Projeto Integrado D).

1.2. ESTRUTURA DO TRABALHO

Com a devida introdução realizada brevemente, neste capítulo mostra-se a motivação pelo projeto a ser desenvolvido. No capítulo 2 é feita uma revisão bibliográfica dos temas envolvidos bem como o detalhamento de como será desenvolvida a proposta de solução. Já no capítulo 3 é apresentado o desenvolvimento do projeto, partindo dos cronogramas de trabalho até como foi realizada a configuração de componentes do sistema. No capítulo 4 é demonstrado os resultados preliminares obtidos por meio de fluxogramas, simulações do *site*, fotos de telas das páginas desenvolvidas e da base de dados, e também através de visualizações via *prompt* de operações no cliente. Por fim, o capítulo 5 é a conclusão parcial do projeto, fazendo uma avaliação

da aplicabilidade do projeto bem como propondo evoluções e melhorias para desenvolvimentos futuros.

2. REVISÃO BIBLIOGRÁFICA E METODOLOGIA

Neste capítulo, será discutido sobre o estado da arte atual dos sistemas de acesso e pagamento em restaurantes universitários a nível global, bem como de que forma será implementada a proposta deste trabalho, detalhando os principais componentes do projeto e de que forma estes deverão atuar.

2.1. VISÃO GERAL

O procedimento atual de acesso ao RU é um tanto burocrático. Ele está devidamente descrito nos passos a seguir:

- primeiramente, na entrada do restaurante, deve-se apresentar um documento que comprove seu vínculo com a universidade e então recebe-se uma ficha indicando qual é esse vínculo. Ex: “ALUNO”, “SERVIDOR”;
- com a ficha em mãos, você deve dirigir-se ao caixa e então efetuar o pagamento da refeição;
- o indivíduo recebe um ticket fiscal que comprova o pagamento da taxa estipulada para a respectiva refeição (café da manhã, almoço ou jantar);
- o ticket deve ser entregue a um terceiro funcionário, e só então a pessoa tem acesso ao buffet.

Evidentemente, tal processo gera um certo retardo no acesso ao RU, além de ser pouco eficiente em termos de controle e agilidade. Dificuldades com o troco são igualmente comuns e o manuseio de cédulas de dinheiro e moedas antes da refeição pode ser desagradável.

Ao redor do mundo, sistemas mais modernos e versáteis já são utilizados. Na Universidade de Luxemburgo, existe um cartão denominado “mycard”, mostrado na Figura 1. O aluno pode utilizá-lo nos diversos campi da universidade, tanto para consumir as refeições do restaurante universitário, quanto para fazer lanches ou utilizar máquinas que vendem comidas e bebidas. Funciona como uma espécie de cartão de crédito universitário. A recarga pode ser feita diretamente no caixa do restaurante ou através de uma plataforma *online* (UNIVERSITY OF LUXEMBOURG, 2016).

Figura 1 - “myCard” da Universidade de Luxemburgo



Fonte:

http://www.en.uni.lu/students/useful_information_from_a_to_z/university_restaurant_university_restaurant_card (Acesso em 06/09/2016).

Não há dúvidas de que os restaurantes universitários da UFPR precisam se atualizar e optar por um sistema mais moderno e robusto. É com este intuito que será apresentado um novo modelo de sistema de acesso para os RUs da universidade.

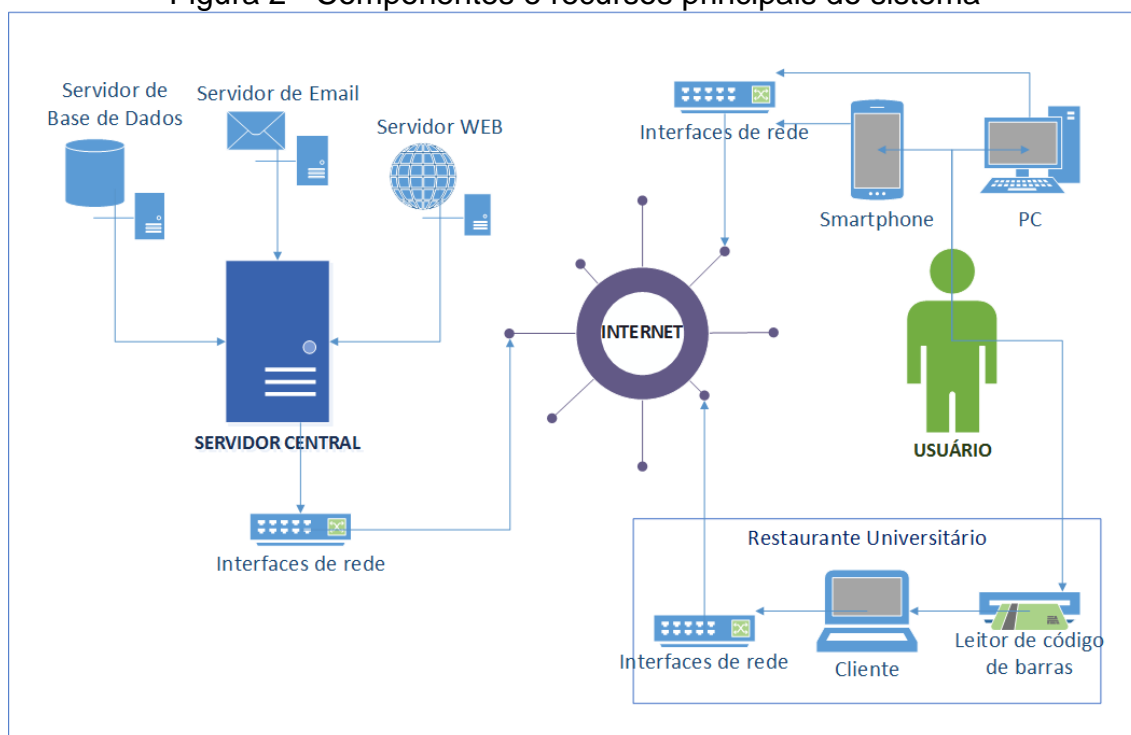
2.2. PROPOSTA

O sistema de acesso rápido ao RU irá atuar através da identificação do usuário pelo código de barras presente no seu documento de vínculo com a universidade, bem como utilizando créditos, que eliminam a necessidade de manuseio de dinheiro e acabam com as dificuldades com o troco. Futuramente, considerando a implementação prática do sistema e a completa adesão da comunidade, será eliminada a necessidade de caixas convencionais.

Inicialmente, o cadastro dos usuários será realizado por um operador do sistema na área reservada para tal no *website*, através de uma confirmação de identificação. Futuramente, este cadastro poderá ser feito de forma automática, no momento da matrícula do estudante/servidor. Contudo, este procedimento dependerá da aceitação e integração dos órgãos responsáveis da UFPR.

O usuário poderá inserir créditos na sua credencial pelo *website*. Ao fazê-lo, o usuário poderá acessar os restaurantes universitários simplesmente passando por um leitor de código de barras, que comprovará sua identidade e efetuará o desconto de uma unidade de crédito correspondente a refeição atual – preços diferentes são aplicados para café da manhã e almoço/jantar. Essa plataforma também irá limitar a quantidade de vezes que uma credencial pode ser usada por refeição (uma vez). A Figura 2 exemplifica o funcionamento geral do projeto.

Figura 2 - Componentes e recursos principais do sistema



Fonte: Dos autores.

Uma das vantagens deste modelo é o desenvolvimento do mesmo pela própria comunidade, contando com muitos recursos já existentes e portanto barateando os custos do projeto. Outro aspecto que vale ser citado é a

possibilidade de expansão fácil no futuro, podendo atuar como um sistema de controle para as mais diversas aplicações dentro da universidade. Além disso, o sistema conta com um *site* exclusivo para o projeto.

2.3. SERVIDOR CENTRAL

O servidor central deverá ser capaz de responder a todos os outros elementos do sistema pela rede, atendendo solicitações de leitura e escrita na base de dados, bem como fornecer as páginas *Web* adequadas ao usuário. Além disso, ele enviará os *emails* de confirmação assim que um usuário se cadastrar na plataforma. Uma máquina equipada com *Debian Linux*, o conjunto *LAMP server* (*Linux – Apache – MySQL – PHP*) e um servidor de *emails* atenderá a esta aplicação, reunindo todos esses recursos.

2.3.1. Servidor Web

Com a finalidade de hospedar o *site* do projeto, um servidor de páginas *Web* será empregado. O *site* tem por objetivo permitir a inscrição dos usuários via preenchimento de formulário, bem como explicar as funcionalidades e o objetivo do sistema, algumas outras informações úteis também poderão ser exibidas. Os dados inseridos no *site* serão armazenados no servidor central.

Uma ferramenta para recarga *online* dos créditos dos usuários, bem como registros de transações e histórico de utilização do sistema também deverão ser implementados nesta plataforma, tornando a experiência com o sistema mais dinâmica e facilitada.

O *Apache* é o servidor de páginas mais utilizado no mundo, possui suporte a *PHP*, mensagens de erro e criptografia (ALECRIM, 2006), por este motivo foi escolhido para atender a tal aplicação.

2.3.2. Servidor de Base de Dados

Para armazenar os dados cadastrais dos usuários do sistema, bem como seus créditos e outros dados pertinentes, uma base de dados relacional se faz necessária, pois é desejado manter controle sobre esses dados, realizar alterações de maneira rápida e garantir a integridade de tudo que está armazenado, evitando redundâncias e inconsistências (DA COSTA, 2011).

Com o objetivo de cumprir essa função, o banco de dados *MySQL* foi escolhido, por apresentar extensa documentação, milhares de referências na *internet* e principalmente pela sua fácil instalação e integração com o servidor *Web* (OFICINA DA NET, 2007).

A fim de proporcionar maior mobilidade e versatilidade, um *script* realizará a tarefa de criar a base de dados, suas tabelas, colunas e usuários. Desta forma, caso seja necessário recriar a base de dados durante a fase de testes, o *script* torna-se uma ferramenta muito útil e simples.

2.3.3. Servidor de Email

Visando o envio de *emails* de confirmação de cadastro, possíveis avisos, e também o recebimento de *emails* vindos da comunidade, uma API (*Application Programming Interface*) foi utilizada para facilitar o envio de *e-mails* através de uma conta no *gmail* criada para este fim. Aqui, foi utilizada a biblioteca *PHPMailer*, por trabalhar com a linguagem *PHP* e ser de fácil uso (BOINTON, 2016).

Posteriormente, um servidor de *emails* será instalado no servidor central. Desta forma, será possível utilizar endereços com o domínio do *website*.

2.4. CLIENTE

O cliente – que neste caso refere-se a um computador e não a um usuário do sistema – deverá ser capaz de realizar a leitura das credenciais dos

usuários dos restaurantes universitários (via leitor de código de barras), consultar remotamente a base de dados a fim de verificar o status do utilizador (saldo, validade do cadastro, etc.) e então debitar o valor da refeição, seguido da liberação de entrada. Uma máquina rodando uma aplicação escrita em *Python* desempenhará tal papel.

A escolha da linguagem de programação *Python* para desenvolver este aplicativo deve-se, sobre tudo, a sua simplicidade, disponibilidade de documentação em português e ferramentas de integração com a base de dados escolhida (PYTHON HELP, 2012).

2.5. APLICATIVO DE SMARTPHONE

Cada vez mais as pessoas se conectam a Internet através de seus smartphones (FENAINFO, 2016). Assim sendo, um aplicativo que possibilite consultar seu saldo - além de outras informações relevantes - é definitivamente um complemento muito interessante ao sistema. Portanto, será desenvolvido um aplicativo para *Android* que proporcione tal funcionalidade. A ideia é que o usuário possa exercer um controle de forma mais ágil sobre o seu balanço, consultando-o com poucos toques na tela de seu aparelho pessoal.

2.6. VALIDAÇÃO

A fim de avaliar o correto funcionamento do sistema e a devida atuação em conjunto dos diversos componentes, um procedimento de teste adequado deve ser empregado. Logo, será simulado a utilização do núcleo principal do projeto como se o mesmo já tivesse devidamente instalado e operacional. Para tal, propõem-se a seguinte sequência de procedimentos para validação da primeira etapa do projeto:

- acesso ao *website* do projeto e todas as suas páginas;
- cadastro completo de um usuário pelo *site*;
- inserção de créditos para este usuário;

- verificação da integridade dos dados inseridos na base de dados;
- verificação da segurança da criptografia das senhas armazenadas.

Com relação à inserção de créditos, tal procedimento ainda não será realizado com dinheiro real, nem através de uma transação pelo *site*, pois este recurso será melhor desenvolvido no futuro. Por enquanto, os créditos serão atribuídos diretamente no campo correspondente na base de dados.

Por último, é importante frisar que etapas adicionais e testes específicos podem ainda ser necessários para validar cada um dos procedimentos apresentados.

3. DESENVOLVIMENTO

Neste trabalho visa-se desenvolver um servidor principal, que conte com servidor de *email*, servidor de dados e servidor *Web*. Além disso, um cliente para leitura dos cartões juntamente com um aplicativo para *smartphone* deverá ser desenvolvido, contando ainda com uma página *Web* para maior facilidade de interação com o sistema de créditos e de cadastramento em geral.

3.1. DIVISÃO DO TRABALHO

O sistema proposto por este trabalho é complexo o suficiente para dividi-lo em diversos módulos independentes. Como este é também um trabalho muito longo para a execução íntegra em apenas um semestre por duas pessoas, é proposto aqui a divisão do trabalho destes diversos módulos ao longo de dois anos.

A ordem de prioridade para a programação dos módulos foi definida pelo quão essencial o módulo é para o funcionamento do sistema de forma transparente. Desta forma, a Quadro 1 relaciona a ordem de prioridade de execução dos módulos e as datas previstas de conclusão.

Quadro 1: Módulos para divisão do trabalho

Módulo	Importância	Data Prevista de Conclusão
Modelagem do Banco de Dados	Muito alta	01/12/2016
Página Web simples funcional	Muito alta	01/12/2016
Aplicativos de consulta ao BD	Muito alta	01/12/2016
Plataforma de Pagamento	Alta	01/06/2017
Página Web completa	Média	01/06/2017
Recarga Web de créditos	Média	01/12/2017
Hospedagem do servidor	Baixa	01/12/2017
Aplicativo Mobile	Baixa	01/06/2018

Fonte: Dos autores.

Dado a complexidade dos módulos e para que houvesse uma organização semanal das atividades pendentes, as tarefas desta primeira etapa foram divididas em metas menores compatíveis com a evolução do projeto. O Quadro 2 contém o cronograma atual.

Quadro 2: Cronograma do Semestre

Dia	1/Ago			22/Ago		5/Set				3/Out					7/Nov			
Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Relatório																		
Definição do Projeto																		
Setup Apache/Linux Server																		
Funcionamento num. carteirinha																		
Aprender HTML/CSS/PHP																		
Criação db MySQL																		
Página Web Funcional																		
Integração db com página Web																		
Criptografia senha																		
Aplicação Python																		
Testes e avaliação de perspectiva																		
Legenda		Cumprido					Pedro e Erik						Pedro			Erik		

Fonte: Dos autores.

3.2. SERVIDOR LINUX

De forma a ter a disposição uma plataforma para trabalhar e verificar rapidamente o progresso e a aparência dos programas e páginas *Web*, foi configurado um servidor Debian provisório, o qual ficará ligado na maior parte do tempo e conectado à *Internet*. Neste servidor, serviços essenciais para o funcionamento da plataforma foram instalados, como um servidor *Web Apache*, o banco de dados *MySQL*, *PHP*, *Git*, interpretador *Python*, entre outros.

Como este servidor está fisicamente localizado na residência de um dos autores, o modem de saída para a *Internet* foi conectado em um serviço de DDNS (*Dynamic Domain Name Service*), que fornece um URL (*Uniform Resource Locator*) temporário para o modem. Então, tal modem foi configurado para realizar um encaminhamento das portas 80 e 22 para o endereço IP (*Internet Protocol*) do servidor Debian na rede interna. Isto possibilita então o acesso da página *Web* e a conexão SSH (*Secure Shell*) com o servidor de uma rede externa.

Além desta modificação na rede, também foi criado um *shell script* para que a página *Web* seja automaticamente atualizada com as mudanças realizadas no repositório do *GitHub*. A Figura 3 mostra o breve *script* descrito acima.

Figura 3 - Script de automação do repositório

```
#!/bin/bash  
  
cd /var/acesso_ufpr  
sudo git pull  
sudo cp -r website/* ../www/html  
cd ~
```

Fonte: Dos autores.

3.3. PÁGINA WEB

Para o desenvolvimento do *front-end* da página *Web*, optou-se por não utilizar nenhum CMS (*Content Management System*). Esta escolha foi feita

para que o projeto não ficasse limitado apenas às utilidades que o CMS escolhido traria, além de que é possível otimizar mais o *site* caso este seja programado diretamente, sem o intermédio de mais um programa (SUMMERS, 2014).

Como o *site* foi programado diretamente em HTML/CSS, decidiu-se utilizar o *Bootstrap* como *framework*, para que se tivesse as facilidades de um sistema de *grid* já consolidado, uma fácil adaptação para deixar o *site* responsivo, uma estrutura para formulários completa e um *layout* para utilizar como base (GIMMER, 2014). Em cima deste *layout* base, foram realizadas customizações próprias através de arquivos CSS.

Decidiu-se aproveitar a utilização do PHP para também modularizar as páginas. Como exemplo, colocando o cabeçalho e o rodapé das páginas em um arquivo separado, e apenas incluindo estes arquivos quando necessário nas páginas. A visualização final para o usuário é idêntica, porém agora o PHP é processado pelo servidor antes de enviar a página em forma de *HTML* puro para o usuário. Assim, obteve-se o mesmo resultado, mas o código com maior sustentabilidade foi deixado para futuras expansões (SANDERSON, 2009).

No momento, três páginas estão em completo funcionamento: A página principal (*homepage*), a página de cadastro, e a página de *login*. A página pós-*login*, a página pós-cadastro e página pós-confirmação de *email* encontram-se em desenvolvimento.

3.3.1. Página principal

Dentre as três páginas, a *homepage* é a mais complexa, pois exigiu um trabalho visual mais elaborado. Esta página serve para explicar de maneira interativa e fácil o sistema de acesso e pagamento proposto. Esta página foi dividida em quatro seções principais. A primeira consiste apenas de uma imagem com um descritivo básico e botões com *links* para as páginas de cadastro e *login*. Na segunda seção, é descrito o funcionamento do sistema em 4 passos. A terceira parte apresenta os desenvolvedores do sistema, enquanto a quarta seção serve para mostrar alguns testemunhos de usuários do sistema

para o usuário. Evidentemente, os testemunhos apresentados atualmente são meramente ilustrativos, já que o sistema ainda não foi implementado.

3.3.2. Página de cadastro

A página de cadastro possui o formulário base para o cadastro de um novo usuário no sistema. A partir deste cadastro, o usuário terá acesso à sua página, onde poderá consultar e inserir mais créditos, além de verificar o histórico de transações. Todos os campos do formulário serão passados para o servidor Web através do método POST, uma vez que ele é recomendado para a transferência de informações sensíveis, pois o faz através do protocolo HTTP (*Hypertext Transfer Protocol*) subjacente (W3C, 2004). A fim de preservar a privacidade e segurança dos dados, um sistema de criptografia de senha através de *hash* foi empregado, juntamente com um *captcha* para confirmar que trata-se de uma pessoa realizando o cadastro. Após realizado o cadastro, o usuário deverá confirmá-lo através de um *link* enviado automaticamente para o endereço de *email* informado. Este procedimento busca garantir que o endereço eletrônico fornecido realmente pertence a pessoa que o cadastrou. Tal *email* é enviado a partir de uma biblioteca disponível para *PHP* (BELÉM, 2009), o que permitiu a utilização de uma conta de envio do serviço *gmail*, facilitando a implementação e evitando maiores complicações com a configuração do servidor interno de *email*, que sofria com dificuldades relacionados a confiabilidade e segurança.

3.3.3. Página de *login*

A página de *login* serve apenas para a autenticação do usuário para acesso a sua página. Nela, o usuário entrará com o seu *email* e senha definidos na etapa de cadastro, e caso validados com sucesso, encaminhará para a página pós-*login*. Uma sessão para o usuário será criada através de *cookies*, que poderão ser mantidos por até 60 dias, caso o usuário deseje.

3.3.4. Página de pós-login

Após a verificação com sucesso do *email* e senha do usuário na página de *login*, o mesmo terá acesso a página pós-*login*, que em outras palavras é a página pessoal do usuário do sistema, onde ele encontrará informações sobre suas últimas transações, verificará seus dados cadastrais, poderá consultar seu saldo e realizar recargas em seu cartão. Cada uma destas opções está organizada por abas, permitindo assim uma experiência mais agradável ao usuário quando este desejar realizar operações e/ou consultas.

3.4. MYSQL

O banco de dados *MySQL* ainda está sendo estruturado conforme o progresso do projeto. Atualmente, ele possui quatro tabelas, denominadas “*Users*”, “*Tempusers*”, “*Restaurants*” e “*Transactions*”. A tabela “*Users*” manipula os dados de cadastro e saldo do usuário. A tabela “*Tempusers*” também possui um *link* direto com os usuários, porém foi desenhada com o objetivo de armazenar temporariamente os usuários que ainda não passaram pelo processo de confirmação do cadastro, tendo estes um período de três dias após o cadastro para confirmá-lo através do *link* recebido no *email* informado no ato. Caso não confirmados, o cadastro destes usuário não é efetivado e eles não são transferidos para a tabela “*Users*”. A tabela “*Restaurants*” armazena os dados dos restaurantes cadastrados. E, por último, a tabela “*Transactions*” serve para armazenar um log de todas as transações já realizadas, incluindo as operações de recarga. Alguns usuários também foram criados no *MySQL* com permissões de acesso às tabelas diferentes para aumentar a segurança. O *script* completo da criação do banco de dados e das tabelas com seus respectivos campos pode ser encontrado no Apêndice 7.1. Vale salientar novamente que mudanças na estrutura banco de dados ainda podem ser realizadas de acordo com as necessidades encontradas no decorrer do desenvolvimento do projeto.

3.4.1. Criptografia das senhas

A fim de prover um sistema seguro e confiável para todos os usuários, foi implementado um método de criptografia moderno para o armazenamento das senhas no banco de dados. O método escolhido foi o *BCRYPT*, já consolidado e muito utilizado em grandes sistemas pela Internet (BELÉM, 2012).

Neste método, a senha é criptografada utilizando o método de *hash sha-256*, que possui uma chave de 256 *bits*. Porém, caso a senha fosse armazenada simplesmente utilizando o *hash*, um invasor poderia descobrir a senha do usuário utilizando uma tabela arco-íris (O'DONNELL, 2016). A fim de resolver este problema, o *BCRYPT* implementa a técnica do *salt*, que basicamente mistura a senha original com uma *string* aleatória para só então realizar o *hash*. Desta forma, as chances desta combinação estar em uma tabela arco-íris são ínfimas.

Além disso, o *BCRYPT* é seguro contra ataques de senha temporizados e possui um mecanismo de gasto de tempo personalizável para processar a senha, buscando diminuir a efetividade de ataques de força bruta.

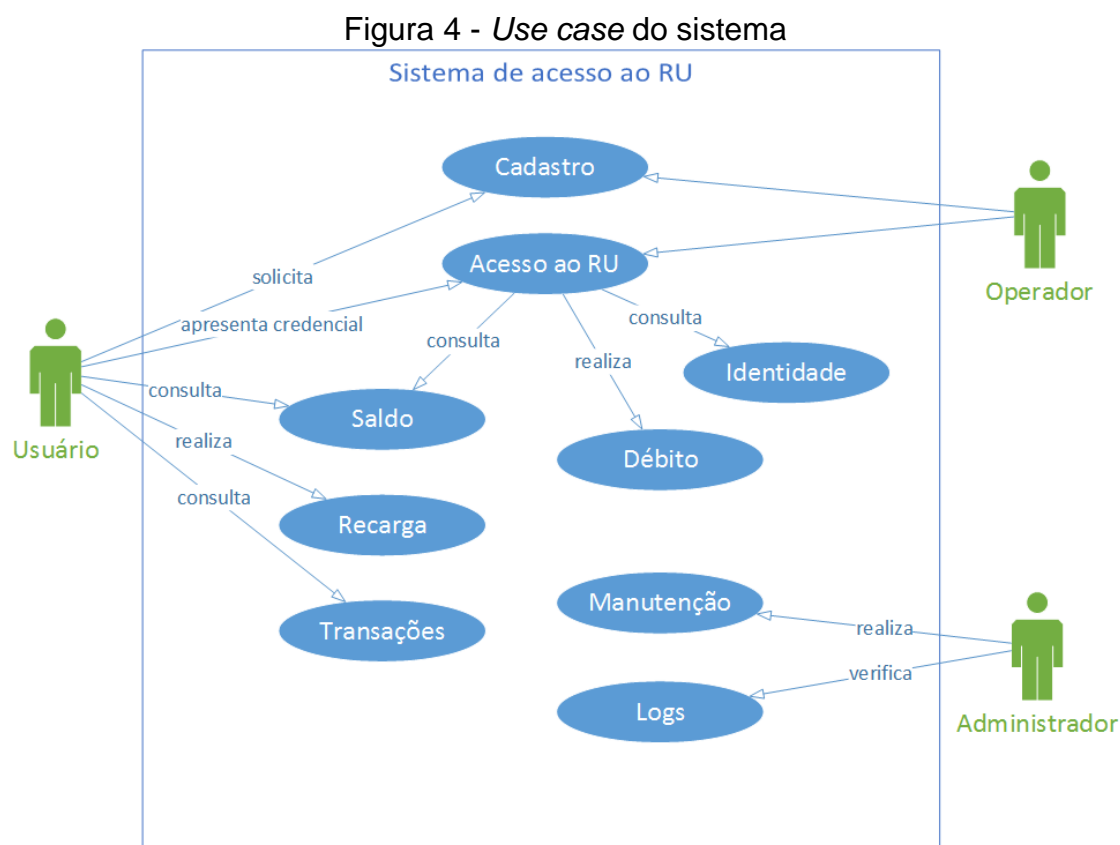
3.5. PROGRAMA DAS LEITORAS DE ACESSO

Em cada unidade do RU, haverá no mínimo um computador que se conectará com o servidor principal para consultar o banco de dados. O papel da aplicação desenvolvida em *Python* é realizar o intermédio desta operação. O computador rodará o programa em background, que a cada leitura do leitor de código de barras abrirá uma conexão TCP (*Transmission Control Protocol*) e consultará o banco de dados pela porta padrão do *MySQL* 3306. Além disso, caso o usuário seja encontrado e tenha saldo disponível para a transação, o programa em *Python* garantirá que essa transação seja executada em segurança. Conforme o resultado da consulta ao banco de dados, o programa irá retornar diversos códigos de status diferentes, que garantirão ou não o acesso ao usuário.

4. RESULTADOS PARCIAIS E DISCUSSÃO

De forma a avaliar o progresso do projeto, bem como debater os resultados até então obtidos, este capítulo exhibe alguns resultados e testes pertinentes a fim de melhor analisá-lo tecnicamente. A página *Web* foi avaliada nos quesitos desempenho, responsividade e compatibilidade entre navegadores. Tais fatores são muito importantes considerando o elevado número de possíveis usuários do sistema que irão acessá-lo de diferentes dispositivos. O banco de dados também foi testado, buscando verificar sua integridade e funcionalidade na inserção, alteração e exclusão de dados. Por fim, o cliente foi posto à prova em uma simulação de funcionamento real, onde este realiza o intermédio de transações entre usuários e a base de dados.

Para um melhor entendimento sobre os “atores” e ações por eles tomadas no sistema, um diagrama do tipo *use case* é apresentado na Figura 4, onde os principais processos são exibidos.



Fonte: Dos autores.

4.1. PÁGINA WEB

No desenvolvimento das páginas para o *website*, a experiência do usuário foi colocada em primeiro plano. Portanto, definiu-se que o *layout* da página deve ser atrativo o suficiente, e ao mesmo tempo o *site* deve responder rapidamente, ser responsivo e funcionar em diversos navegadores. Outra característica buscada foi um *layout* simples e ao mesmo tempo elegante, evitando uma poluição desnecessária da página.

Como avaliação das páginas desenvolvidas, utilizou-se três métricas diferentes. A primeira consistiu em utilizar *websites* que avaliam desempenho de outros. No segundo método, a responsividade do *site* foi verificada em aparelhos com telas menores. A terceira métrica consistiu em verificar a compatibilidade em vários navegadores diferentes. Tais métricas foram definidas de modo a abordar diferentes parâmetros que juntos tornam um *website* atrativo e funcional.

4.1.1. Avaliação de Desempenho

Para este teste, utilizou-se as páginas <https://developers.google.com/speed/pagespeed/insights> e <https://tools.pingdom.com>, que são ferramentas online gratuitas para avaliação de *websites*. As Figuras 5 e 6 mostram o resultado obtido da avaliação da Google, através do primeiro *website*.

Figura 5 - Avaliação de desempenho (*Mobile*)

http://arion.ddns.net/



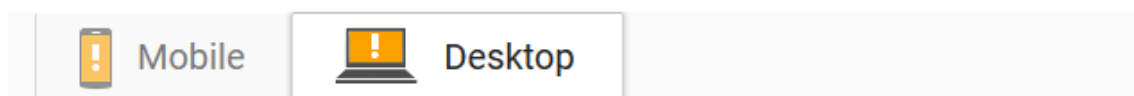
100 / 100 Experiência do usuário

✓ Parabéns! Nenhum problema encontrado.

Fonte: Dos autores.

Figura 6 - Avaliação de desempenho (*Desktop*)

http://arion.ddns.net/



82 / 100 Resumo das sugestões

! Considere corrigir:

Otimizar imagens

› [Mostrar como corrigir](#)

Eliminar JavaScript e CSS de bloqueio de renderização no conteúdo acima da borda

› [Mostrar como corrigir](#)

Aproveitar cache do navegador

› [Mostrar como corrigir](#)

Ativar compactação

› [Mostrar como corrigir](#)

✓ Aprovado em 6 regras

› [Exibir detalhes](#)

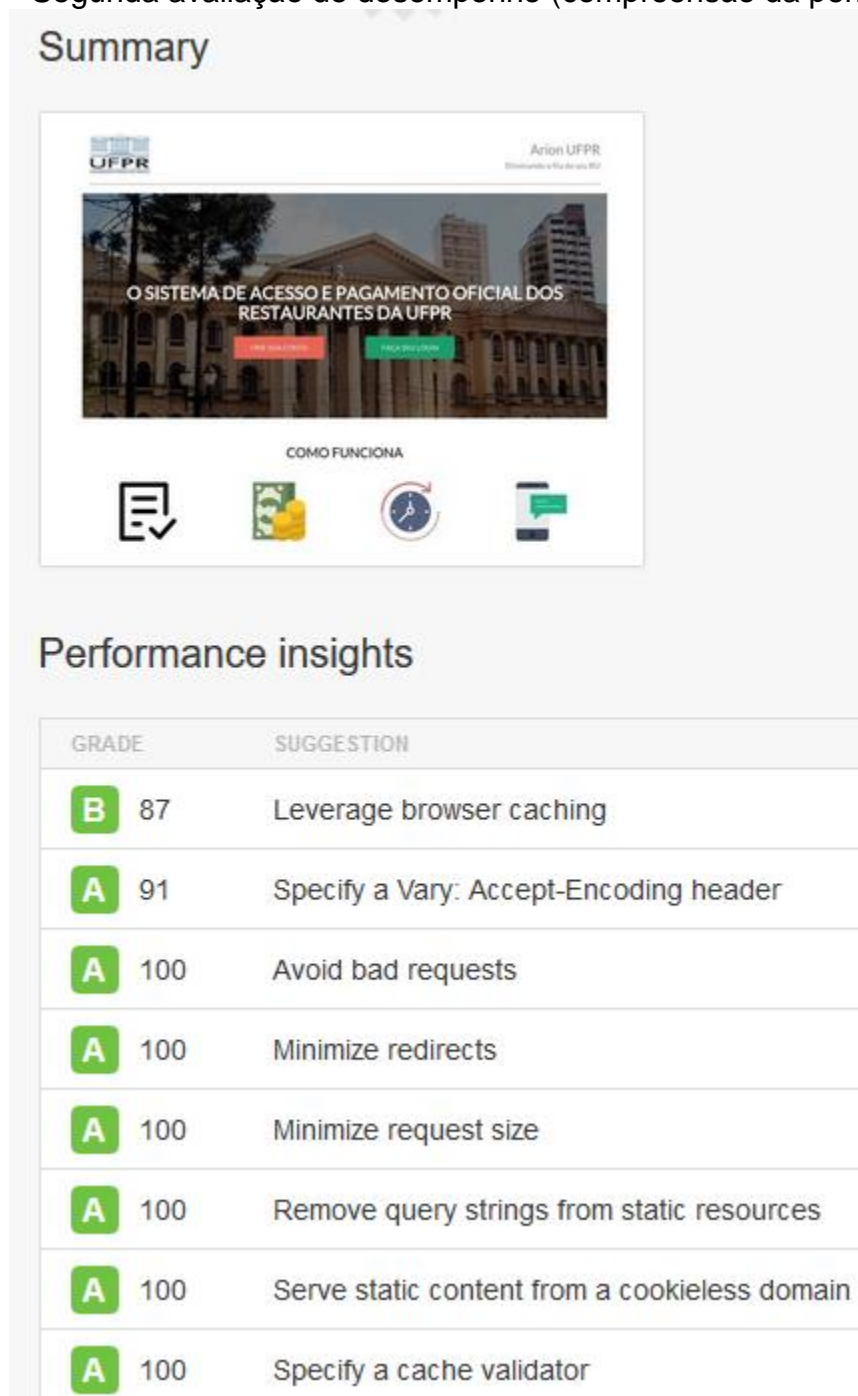
Faça o download de [recursos de imagem, JavaScript e CSS](#) otimizados para esta página.

Fonte: Dos autores.

Observando os resultados das primeiras avaliações, conclui-se que ainda há diversos aspectos nos quais há espaço para melhoras futuras. As principais melhorias que podem ser feitas é a remoção dos *JavaScripts* e *CSS* de bloqueios no cabeçalho do *HTML*, a otimização das imagens, a compactação de certos arquivos, e o aproveitamento do cache do navegador para todos os recursos. O primeiro item pode ser resolvido através do carregamento assíncrono destes dados para o navegador. A otimização das imagens pode ser feita através de programas específicos para tal, porém com o *trade-off* de uma imagem com pior qualidade. A compactação de arquivos e o aproveitamento de cache podem ser resolvidos diretamente através de configurações no *Apache*.

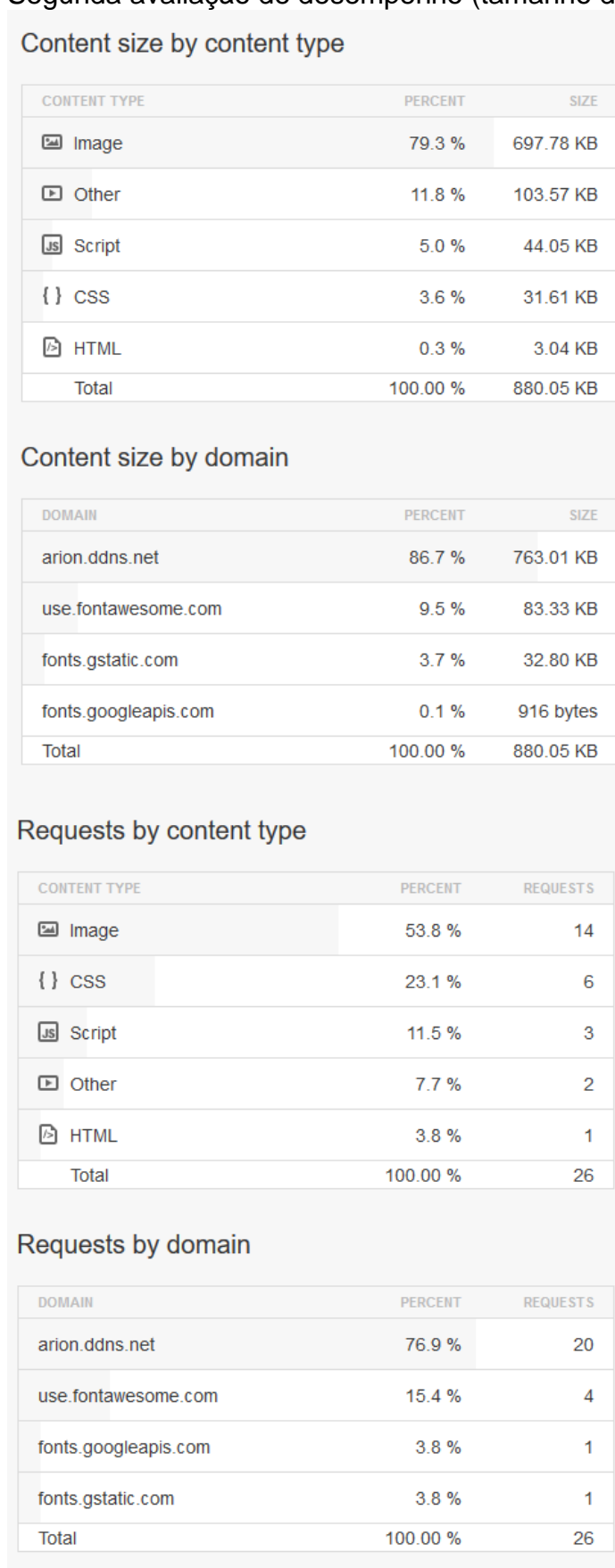
O segundo *website* para avaliação de desempenho fornece mais informações sobre os tempos de carregamento da página, além de atribuir notas e mostrar informações adicionais sobre o tamanho ocupado pelos diversos tipos de conteúdo, o que pode ser extremamente útil em um processo futuro de otimização. As Figuras 7, 8 e 9 apresentam os resultados obtidos.

Figura 7 - Segunda avaliação de desempenho (compreensão da performance)



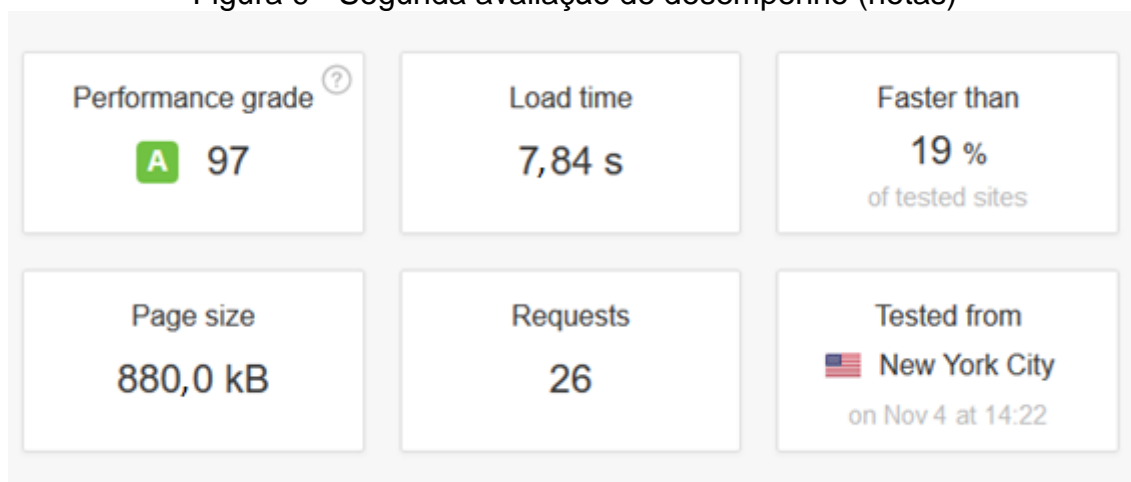
Fonte: Dos autores.

Figura 8 - Segunda avaliação de desempenho (tamanho de arquivos)



Fonte: Dos autores.

Figura 9 - Segunda avaliação de desempenho (notas)



Fonte: Dos autores.

A partir destes resultados, verifica-se que uma grande porcentagem do tamanho total dos arquivos sendo carregados na página é de imagens somente. Este seria mais um motivo para otimizar as imagens e, portanto, deixar o *site* ainda mais rápido. Vale notar, contudo, que a página está sendo atualmente hospedada em um servidor doméstico com acesso à uma rede com velocidade limitada. Portanto, os resultados obtidos nesta seção podem ser enviesados para pior, já que este *site* utilizaria um servidor dedicado em uma rede comercial uma vez que em operação.

4.1.2. Responsividade

Nos últimos anos, o uso de celular como forma primária de acesso à Internet tem subido cada vez mais (GRUBER, 2015). Por isso, é de extrema importância que os *sites* desenvolvidos agora sejam responsivos por natureza. Nos testes realizados, o *website* foi avaliado em um computador de 15 polegadas (resolução de 1920x1080px), em um celular *Nexus 5* de 5 polegadas (resolução de 1080x1920px) e de forma dinâmica variando o tamanho do navegador no mesmo computador.

A visualização em um computador é mostrada na Figura 10, enquanto a visualização em *smartphone* teve que ser dividida em duas imagens, Figuras 11 e 12, já que esta foi composta por imagens agrupadas obtidas com o

rolamento da tela do aparelho, o que a tornou excessivamente vertical para somente uma página deste documento.

Para fins de comparação, o mesmo trecho superior da página inicial foi registrado, de forma a verificar como o *site* adaptaria a exibição do mesmo conteúdo para diferentes aparelhos.

Figura 10 - Foto de tela do início da página principal no computador



Fonte: Dos autores.

Figura 11 - Foto de tela do início da página principal no celular (Parte 1)



COMO FUNCIONA

Fonte: Dos autores.

Figura 12 - Foto de tela do início da página principal no celular (Parte 2)



CADASTRO DA CARTEIRINHA

Crie sua conta e cadastre sua carteirinha da UFPR aqui. O processo leva menos de 5 minutos.



INSIRA SEUS CRÉDITOS

Use seu cartão de crédito para facilitar o pagamento do restaurante universitário. O sistema é 100% seguro e monitorado pela própria UFPR.



SEU RU SEM FILAS

Acesse o RU normalmente utilizando sua carteirinha, sem filas! O resto deixa que a gente cuida.



CONSULTE SEUS CRÉDITOS

Seja na Web ou no seu celular, consulte e insira créditos em menos de 2 minutos. Tudo isto porque sabemos que o tempo de um estudante universitário é muito valioso.

Fonte: Dos autores.

Quanto à responsividade, o *site* comportou-se da maneira esperada, redimensionando-se conforme a largura de tela. Para telas muito pequenas, os itens distribuídos em colunas tornam-se distribuídos em fileiras diferentes, tornando o acesso uma experiência visual melhor para o usuário. Esta característica de responsividade tornou-se mais simples de ser implementado graças ao sistema de *grid* do *Bootstrap*, que já possui as métricas certas para que as colunas sejam distribuídas corretamente.

4.1.3. Compatibilidade entre navegadores

Nesta seção, os navegadores *Google Chrome v53*, *Internet Explorer v11*, *Microsoft Edge 25.1* e *Mozilla Firefox 48.0.2* foram utilizados para verificar a compatibilidade. Este teste incluiu verificar a aparência e a funcionalidade de todos os elementos e *Javascripts*, além de checar a correta visualização das animações.

Durante este procedimento, apenas duas irregularidades foram encontradas na página de cadastro. A primeira encontra-se nos campos numéricos do formulário no *Mozilla Firefox*. Nestes campos, foram removidos os *spinners* ao lado do campo para melhor visualização do usuário no arquivo CSS. Contudo, este campo ainda aparece no *Mozilla*, conforme é apresentado na Figura 13.

Figura 13 - *Spinners* ao lado dos campos numéricos



Número de Matrícula (GRR)

XXXXXXXX

Número da carteirinha (Código de barras)

Carteirinha

Fonte: Dos autores.

Além deste pequeno problema visual, também encontrou-se outro problema nestes campos numéricos. Eles deveriam ser limitados para a digitação de somente caracteres numéricos, rejeitando a entrada de qualquer

caractere não-numérico. Isto acontece como esperado no *Google Chrome*, porém nos outros três navegadores testados, o campo permite a entrada de texto, como pode-se observar nas Figuras 14, 15 e 16. Apesar de o campo ser invalidado pelo formulário e o usuário não puder submeter o formulário desta maneira, seria mais adequado não permitir a entrada de texto. Para a resolução deste problema, é possível introduzir um pequeno *script* em *Javascript* para deletar qualquer caractere que não seja numérico nestes campos.

Figura 14 - Campo de número de matrícula (*Edge*)



Número de Matrícula (GRR)

testestesteXXXXXXXX

GRR Inválido

Fonte: Dos autores.

Figura 15 - Campo de número de matrícula (*Firefox*)



Número de Matrícula (GRR)

testesteste

GRR Inválido

Fonte: Dos autores.

Figura 16 - Campo do número de matrícula (*Internet Explorer*)



Número de Matrícula (GRR)

testeteste

GRR Inválido

Fonte: Dos autores.

Apesar destes pequenos detalhes, todas as páginas responderam como o programado, todas as animações funcionaram, e o conteúdo foi apresentado como desenhado. Além disso, todos os campos no formulário de cadastro foram validados corretamente, evitando possíveis inserções com erro na base de dados.

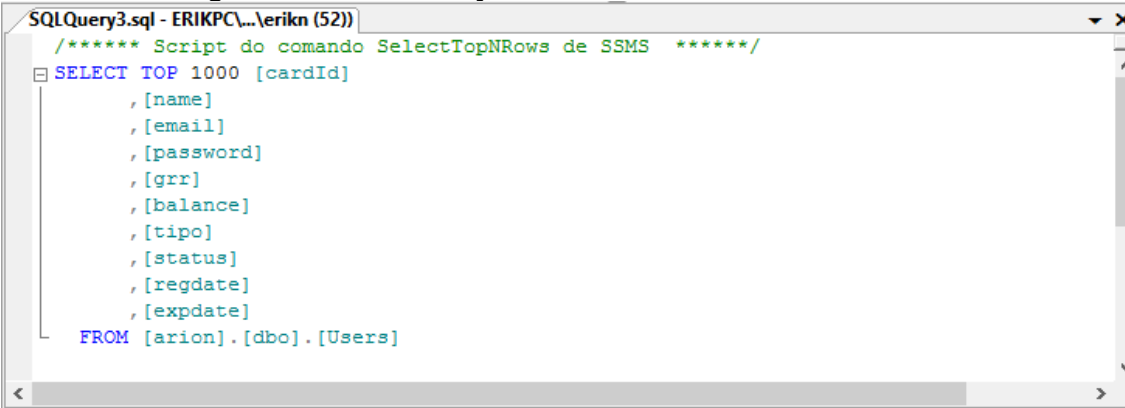
4.2. BANCO DE DADOS

Como citado na seção anterior, a estrutura atual do banco de dados consiste de cinco tabelas. Para o teste, foram inseridos dois usuários teste na tabela “users”. Depois disso, foram realizadas algumas consultas ao banco fazendo alterações no saldo de cada um, simulando descontos e créditos.

Deve ser ressaltado que nem todos os campos e tabelas tiveram seus testes demonstrados neste relatório, pois isso o tornaria demasiadamente longo e não acrescentaria novas informações, já que os testes são todos executados da mesma maneira. Sendo assim, será exemplificado o procedimento somente para o campo “balance” (saldo) da tabela “users”.

A visualização dos dados da tabela “users” encontra-se na Figura 17, os comandos para atualização dos dados de saldo encontram-se na Figura 18 e o estado da tabela após as alterações pode ser visualizado na Figura 19.

Figura 17 - Visualização do conteúdo da tabela “users”

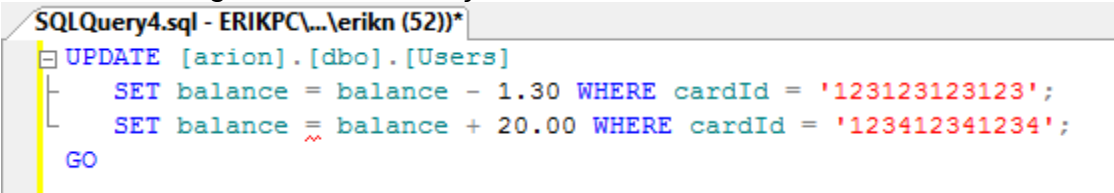


```
SQLQuery3.sql - ERIKPC\...\erikn (52))
/***** Script do comando SelectTopNRows de SSMS *****/
SELECT TOP 1000 [cardId]
, [name]
, [email]
, [password]
, [grr]
, [balance]
, [tipo]
, [status]
, [regdate]
, [expdate]
FROM [arion].[dbo].[Users]
```

	cardId	name	email	password	grr	balance	tipo	status	regdate	expdate
1	123123123123	John Lennon	john@beatles.com	R&volution9	20100000	20.00	0	1	2012-10-10	2020-10-10
2	123412341234	Paul McCartney	paul@beatles.com	H&yJude@	20101010	0.70	0	1	2012-10-10	2020-10-10

Fonte: Dos autores.

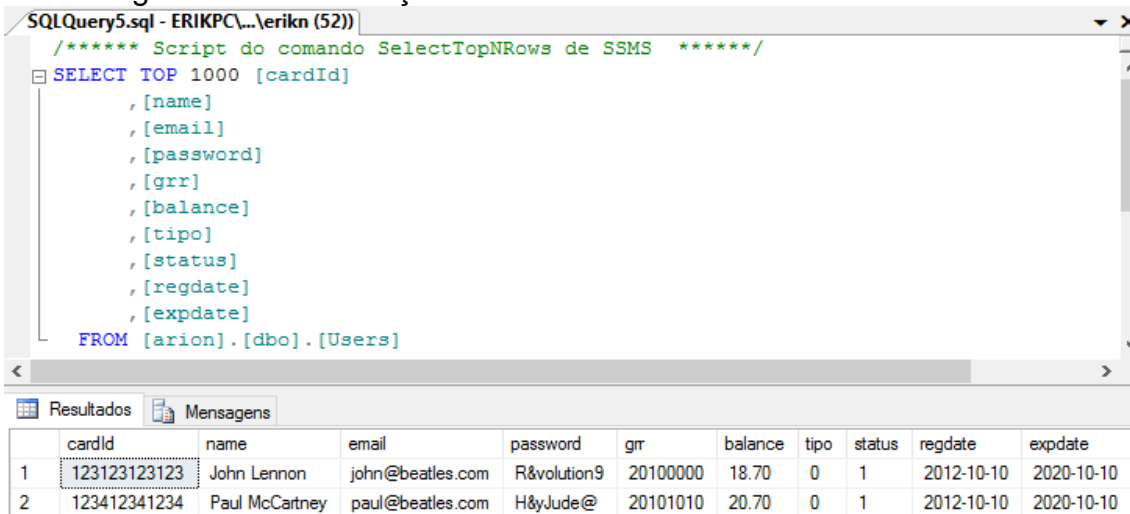
Figura 18 - Atualização do conteúdo da tabela “users”



```
SQLQuery4.sql - ERIKPC\...\erikn (52))*
UPDATE [arion].[dbo].[Users]
SET balance = balance - 1.30 WHERE cardId = '123123123123';
SET balance = balance + 20.00 WHERE cardId = '123412341234';
GO
```

Fonte: Dos autores.

Figura 19 - Visualização do conteúdo atualizado da tabela “users”



```

/***** Script do comando SelectTopNRows de SSMS *****/
SELECT TOP 1000 [cardId]
, [name]
, [email]
, [password]
, [grr]
, [balance]
, [tipo]
, [status]
, [regdate]
, [expdate]
FROM [arion].[dbo].[Users]
  
```

	cardId	name	email	password	grr	balance	tipo	status	regdate	expdate
1	123123123123	John Lennon	john@beatles.com	R&volution9	20100000	18.70	0	1	2012-10-10	2020-10-10
2	123412341234	Paul McCartney	paul@beatles.com	H&yJude@	20101010	20.70	0	1	2012-10-10	2020-10-10

Fonte: Dos autores.

Como observado, a tabelas “users” está tratando os *updates* e *selects* de forma adequada, mudando o saldo do usuário conforme o indicado pelo *update*.

O procedimento de teste acima também foi realizado para os outros campos de outras tabelas e todos mostraram-se funcionais.

4.3. PROGRAMA DAS LEITORAS DE ACESSO

O programa para as leitoras de acesso foi desenvolvido em *Python*, e no momento funciona apenas na interface de texto. Futuramente, pretende-se evoluir o programa para possuir alguma interface visual, utilizando alguma biblioteca como o *WxPython*, por exemplo. Porém, o programa já está com total funcionalidade, realizando os *queries* no banco de dados através da biblioteca *MySQLdb* em *Python*, e atualizando os campos do banco de dados com transações, alterando o saldo final do usuário. Na Figura 20, algumas simulações do programa em funcionamento são apresentadas. Nelas, o programa requer uma entrada do teclado, que tentará achar uma carteirinha com o número correspondente no banco de dados. Caso o usuário digite algum formato inválido de texto, o programa informará isto de volta ao usuário, que poderá digitar novamente o número. É mostrado também o caso em que o

número foi digitado corretamente, porém o usuário não existe no banco de dados. Desta forma, o programa realizará a *query*, porém não achará nenhum correspondente, e retornará a mensagem “*No such user found*” para o usuário. Similarmente, caso o usuário exista no banco de dados, porém não possua saldo, o programa informará a mensagem “*No balance on account*”. Finalmente, caso o usuário exista no banco de dados e possua saldo, o programa descontará o preço da refeição do saldo final do usuário no banco de dados e informará o sucesso da transação para o usuário.

Figura 20 - Simulação do funcionamento do programa das leitoras de acesso


```
pedro@pedro-VirtualBox ~ $ python client.py
Type your carteirinha number:
Invalid query format
Type your carteirinha number:
Invalid query format
Type your carteirinha number: asdf
Invalid query format
Type your carteirinha number: m456
Invalid query format
Type your carteirinha number: 1234
No such user found
Type your carteirinha number: 4321
No such user found
Type your carteirinha number: 12345678
No such user found
Type your carteirinha number: 12341234
('Pedro', 1000.0, 12341234L)
Transaction Complete
Type your carteirinha number: 12341234
('Pedro', 998.7, 12341234L)
Transaction Complete
Type your carteirinha number: 12341234
('Pedro', 997.4, 12341234L)
Transaction Complete
Type your carteirinha number: 43214321
('Guilherme', 1.1, 43214321L)
No balance on account
Type your carteirinha number: 43214321
('Guilherme', 1.1, 43214321L)
No balance on account
Type your carteirinha number: 12341234
('Pedro', 996.1, 12341234L)
Transaction Complete
Type your carteirinha number:
```

Fonte: Dos autores.

4.4. VISUALIZAÇÃO DE PÁGINAS DO SITE

De modo a demonstrar o funcionamento e o *layout* de algumas páginas do *website*, algumas fotos de tela foram tiradas e adicionadas a este documento. A Figura 21 mostra a página de cadastro, onde os dados inseridos são verificados para assegurar que nenhum dado inválido seja copiado para a base de dados. A Figura 22 exibe a aba inicial pós-*login*, onde é possível verificar o saldo atual e as últimas transações realizadas. Na Figura 23, são exibidos os dados cadastrais do usuário. Por fim, na Figura 24, o usuário encontra-se na aba de recarga, onde informações sobre o método para pagamento são requisitadas.

Figura 21 – Página de cadastro



Arion UFPR
Eliminando a fila do seu RU

Nome

Erik Nayan ✓

Endereço de Email

eriknayan@gmail.com ✓

Confirme seu Email

eriknayan@gmail.com ✓

Vínculo com a UFPR

Estudante ▼

Número de Matrícula (GRR)

20131349 ✓

Número da carteirinha (Código de barras)

121212121212 ✓


Senha

•••••••• ✓

Confirme sua senha

•••••••• ✓

✓ Não sou um robô


reCAPTCHA
Privacidade - Termos

Enviar

Fonte: Dos autores.

Figura 22 – Página de saldo e transações pós-login



Arion UFPR
Eliminando a fila do seu RU

Arion Saldo Recarga Transações Informações pessoais 

Bem-vindo Erik Nayan!

Seu saldo é de: R\$97.30
[Faça uma recarga aqui](#)


Suas últimas 5 transações foram:

ID	Data e Hora	Valor	Local
3	2016-11-22 13:20:00	1.30	RU Agrarias
2	2016-11-21 13:20:00	1.30	RU Politecnico
1	2016-11-20 16:20:00	100.00	Recarga Online


[Veja seu histórico completo aqui](#)

Fonte: Dos autores.

Figura 23 – Página de dados cadastrais pós-login



Arion UFPR
Eliminando a fila do seu RU

Arion Saldo Recarga Transações Informações pessoais 

Dados cadastrais de Erik Nayan

Nome
Erik Nayan

Endereço de Email
eriknayan@gmail.com

Vínculo com a UFPR
Estudante

Número de matrícula (GRR)
20131349

Número da carteirinha
1212121212

Fonte: Dos autores.

Figura 24 – Página de recarga pós-login



Arion UFPR
Eliminando a fila do seu RU

ArionSaldoRecargaTransaçõesInformações pessoaisSair

Para realizar uma recarga, preencha os dados do seu cartão de crédito abaixo. Todas as informações enviadas são seguras.

Detalhes de pagamento



Cartão a ser recarregado

121212121212

Valor da Recarga

R\$ 10,00

Número do cartão de crédito

Número do cartão

Nome no cartão

Nome no cartão

Data de validade

MM / AA

Cód. Verificação

CVC

Realizar pagamento

Fonte: Dos autores.

5. CONCLUSÃO

Ao fim dessa primeira etapa de pesquisa e realização do projeto, alguns objetivos específicos puderam ser alcançados e outros encontram-se ainda em desenvolvimento. O servidor central provisório está operacional e os recursos necessários ao projeto já se encontram instalados, mas podem evidentemente ser alterados de acordo com necessidades futuras. Hospedado neste servidor, o *site* do projeto está no ar e a página inicial, bem como a de preenchimento do formulário de inscrição, já estão implementadas e devidamente elaboradas para oferecer um *layout* simples, moderno e objetivo. Na sequência, a meta é finalizar a estrutura principal do sistema, ou seja, permitir que uma credencial da UFPR seja cadastrada, receba a inserção de créditos e, ao ser apresentada ao cliente equipado com leitor de código de barras, este então consulte sua situação na base de dados e efetue o débito de uma unidade de crédito, atualizando o saldo do usuário.

Um dos primeiros requerimentos para este projeto sair do papel era configurar um servidor e disponibilizá-lo para acesso externo pela Internet. Assim sendo, esta foi a primeira prioridade do projeto, e foi rapidamente realizada através da configuração de um computador rodando Debian na rede local de um dos integrantes do grupo, e disponibilizando seu acesso por uma combinação de *port forwarding* e *dynamic DNS*.

Como o sistema proposto é muito orientado a dados, o próximo passo foi instalar um banco de dados e definir as tabelas a serem utilizadas no sistema. Foi utilizado então o banco de dados MySQL, por apresentar fácil integração com o PHP e Apache. Apesar de a estrutura das tabelas ter sido constantemente alterada ao longo do desenvolvimento do projeto, ela foi sempre sendo aprimorada a fim de representar os dados de uma maneira confiável e atualmente está em um estado satisfatório para os autores.

Durante o desenvolvimento do módulo para armazenamento de senhas no banco de dados, foi tomado um cuidado extra para a segurança destes dados. Neste quesito, foi utilizado o método de BCrypt para *hash* das senhas, que se provou seguro contra vários tipos diferentes de ataque ao banco de dados.

Na implementação final do projeto, um computador cliente estará recebendo leituras dos cartões e precisará de um programa que realizará a comunicação com o banco de dados. Apesar de o desenvolvimento deste programa não estar previsto no cronograma deste semestre, este já foi iniciado e cumpre seu papel principal, comunicar-se com o banco de dados e atualizar as tabelas com novas transações. Futuramente, uma ênfase maior será dada a este programa, aumentando os seus *features*.

Outra meta estipulada no início do projeto seria o envio de *emails* no momento do cadastro de novos usuários. Este objetivo foi atingido através utilização de uma biblioteca desenvolvida para este fim. Para os próximos trabalhos, é visada a instalação de um servidor de *emails* na máquina utilizada.

Alguns outros objetivos já foram projetados para os próximos semestres e, portanto, ainda não foram alcançados. Entre estes, está incluído uma plataforma segura de pagamento online para os usuários. Neste quesito, as partes mais desafiadoras incluem prover um website que utilize um protocolo de criptografia fim a fim, como o SSL (*Secure Socket Layer*) por exemplo, e a utilização de uma API para realizar as transações do cartão de uma forma confiável.

Similarmente, o desenvolvimento de um aplicativo para smartphones foi projetado para os próximos semestres, uma vez que não está no topo da lista de prioridades do projeto e trata-se de uma tecnologia mais complexa.

Quanto da ampliação da gama de utilizações do sistema, este é um tópico que é mais interessante ser tratado futuramente, pois o fundamental no momento é o funcionamento deste sistema base por completo, e só então a expansão para outras áreas.

Durante o período que compreendeu o desenvolvimento apresentado neste documento, houveram grandes avanços como também novos problemas a serem solucionados. Um dos desafios a ser superado é o de assegurar, no momento do cadastro, que o documento de vínculo com a universidade pertence a quem o está cadastrando. Dentre os progressos, vale citar que a operação do servidor *Web* vem sendo extremamente satisfatória para o objetivo proposto.

5.1. TRABALHOS FUTUROS

Este projeto tem como algumas de suas características principais ser expansível e utilizar sistema operacional *open source* nos computadores que o sustentam. Entusiastas e pesquisadores interessados têm a oportunidade de desenvolver trabalhos que atinjam outras esferas dentro e fora da universidade, bem como optar por um recurso de validação de acesso e credenciamento mais avançado, substituindo o código de barras por biometria, por exemplo.

No que diz respeito a ampliação do nível de integração do sistema, existe muito a ser explorado, pois o projeto permite realizar um controle eficiente através da carteira de identificação da universidade. Assim sendo, incentiva-se o uso do conceito para solucionar outras questões, como gerenciamento de equipamentos de almoxarifado, utilização de recursos internos que demandem identificação, acesso a plataformas exclusivas, entre outros.

Por último, anseia-se que o projeto inspire o desenvolvimento de soluções criativas, inovadoras e eficazes para o uso dos acadêmicos como um todo.

6. REFERÊNCIAS

NORMAS TÉCNICAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724:** Informação e documentação — Trabalhos acadêmicos — Apresentação. 2011.

DOCUMENTOS CONSULTADOS ONLINE

PRÓ-REITORIA DE ADMINISTRAÇÃO DA UFPR. **Breve Histórico** (2016). Disponível em: <<http://www.pra.ufpr.br/portal/ru/historico/>> Acesso em: 08 set. 2016.

PRÓ-REITORIA DE PLANEJAMENTO, FINANÇAS E ORÇAMENTOS (PROPLAN). **Relatório de Atividades UFPR 2014**. Disponível em: <http://acervodigital.ufpr.br/bitstream/handle/1884/40654/relatorio_de_atividades_2014.pdf?sequence=1&isAllowed=y> Acesso em 08 set. 2016.

FURLAN, NÁJIA. **Alunos da UFPR cobram investimentos em infraestrutura**. Tribuna Paraná, 2005. Disponível em: <<http://www.tribunapr.com.br/noticias/parana/alunos-da-ufpr-cobram-investimentos-em-infra-estrutura/>> Acesso em: 08 set. 2016.

DA COSTA, ELISÂNGELA ROCHA. **Bancos de dados relacionais** (2011). Disponível em: <<http://www.fatecsp.br/dti/tcc/tcc0025.pdf>> Acesso em 09 set. 2016.

OFICINA DA NET. **Conheça um pouco mais sobre o MySQL** (2007). Disponível em: <https://www.oficinadanet.com.br/artigo/390/conheca_um_pouco_sobre_o_mysql> Acesso em 09 set. 2016.

ALECRIM, EMERSON. **Conhecendo o Servidor Apache (HTTP Server Project)** (2006). Disponível em: <<http://www.infowester.com/servapach.php>> Acesso em 02 set. 2016

PYTHON HELP. **Por que Python?** (2012). Disponível em: <<https://pythonhelp.wordpress.com/por-que-python/>> Acesso em 04 set. 2016.

UNIVERSITY OF LUXEMBOURG. **University Restaurant/University Restaurant Card.** Disponível em: <http://www.en.uni.lu/students/useful_information_from_a_to_z/university_restaurant_university_restaurant_card> Acesso em 06 set. 2016.

FENAINFO. **Com aumento do uso de smartphones e tablets, gestão de dispositivos vai decolar.** Disponível em: <http://fenainfo.org.br/info_ler.php?id=43556> Acesso em 04 set. 2016.

SUMMERS, BRENT. **Do you really need a CMS?** (2014). Disponível em: <<http://www.dtelepathy.com/blog/philosophy/do-you-really-need-a-cms>> Acesso em 12 set. 2016.

GIMMER, CHRISTOPHER. **Top 5 Reasons to use Bootstrap** (2014). Disponível em: <<https://bootstrapbay.com/blog/reasons-to-use-bootstrap/>> Acesso em 05 set. 2016.

SANDERSON, GREG. **How to automatically include your header, navigation, and footer on every page** (2009). Disponível em: <<http://www.apaddedcell.com/how-automatically-include-your-header-navigation-and-footer-every-page>> Acesso em 07 set. 2016.

W3C. **URIs, Addressability, and the use of HTTP GET and POST** (2004). Disponível em: <<https://www.w3.org/2001/tag/doc/whenToUseGet.html>> Acesso em 01 set. 2016.

GRUBER, JOHN. **Which is the most important device you use to connect to the Internet?** (2015). Disponível em: <https://daringfireball.net/2015/08/most_important_device> Acesso em 29 ago. 2016.

BELÉM, THIAGO. **Enviar e-mails pelo PHP usando o PHPMailer** (2009). Disponível em: <<http://blog.thiagobelem.net/enviar-e-mails-pelo-php-usando-o-phpmailer>> Acesso em 27 out. 2016.

BELÉM, THIAGO. **Criptografando senhas no PHP usando bcrypt (Blowfish)** (2012). Disponível em: <<http://blog.thiagobelem.net/criptografando-senhas-no-php-usando-bcrypt-blowfish>> Acesso em 22 nov. 2016.

BOINTON, MARCUS. **PHPMailer** (2016). Disponível em: <<https://github.com/PHPMailer/PHPMailer>> Acesso em 09 dez. 2016.

O'DONNELL, ANDY. **Rainbow Tables: Your Password's Worst Nightmare** (2016). Disponível em: <<https://www.lifewire.com/rainbow-tables-your-passwords-worst-nightmare-2487288>> Acesso em 24 nov. 2016.

7. APÊNDICE

Nesta última seção encontram-se os apêndices do projeto, que se compõem basicamente dos códigos de programação envolvidos no desenvolvimento do sistema.

7.1.SCRIP.T.SQL

```
/*
Script de criação de Base de Dados - Acesso UFPR
Erik Nayan & Pedro Mantovani
*/

DROP DATABASE arion; -- USE WITH EXTRA CAREFUL
CREATE DATABASE IF NOT EXISTS arion;

USE arion;

CREATE TABLE IF NOT EXISTS Users (
    cardId      BIGINT(12) UNSIGNED NOT NULL,
    name        VARCHAR(50) NOT NULL,
    email       VARCHAR(50) NOT NULL UNIQUE,
    password    VARCHAR(70) NOT NULL,
    grr         INT(8) UNSIGNED NOT NULL UNIQUE,
    type        TINYINT UNSIGNED NOT NULL, -- 0: Estudante, 1:
Professor, 2: Servidor
    status      BIT DEFAULT 1 NOT NULL, -- 1=Ativo, 0=Inativo;
    regdate     DATE NOT NULL,
    expdate     DATE NOT NULL,
    balance     DECIMAL(6,2) DEFAULT 0.00 NOT NULL,
    PRIMARY KEY (cardId)
);

CREATE TABLE IF NOT EXISTS Tempusers (
    cardId      BIGINT(12) UNSIGNED NOT NULL,
    name        VARCHAR(50) NOT NULL,
    email       VARCHAR(50) NOT NULL UNIQUE,
    password    VARCHAR(70) NOT NULL,
    grr         INT(8) UNSIGNED NOT NULL UNIQUE,
    type        TINYINT UNSIGNED NOT NULL, -- 0: Estudante, 1:
Professor, 2: Servidor
    regdate     DATE NOT NULL,
    confirmkey  VARCHAR(32) NOT NULL,
    PRIMARY KEY (cardId)
);

CREATE TABLE IF NOT EXISTS Restaurants (
    restId      INT AUTO INCREMENT,
    restName    VARCHAR(50) NOT NULL,
    restAddr    VARCHAR(250) NOT NULL,
    PRIMARY KEY (restId)
);

CREATE TABLE IF NOT EXISTS Transactions (
    tranId      BIGINT(12) UNSIGNED AUTO INCREMENT,
    cardId      BIGINT(12) UNSIGNED NOT NULL,
```

```

value          DECIMAL(6,2) NOT NULL,
type           BIT DEFAULT 1 NOT NULL, -- 0: Recarga, 1: Descarga
tranTime       DATETIME NOT NULL,
restId         INT,
PRIMARY KEY (tranId),
FOREIGN KEY (restId)
    REFERENCES Restaurants(restId)
    ON DELETE SET NULL
    ON UPDATE CASCADE,
FOREIGN KEY (cardId)
    REFERENCES Users(cardId)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

/* Create event to auto-delete temporary users after 3 days of
registration */
CREATE EVENT IF NOT EXISTS arion.DeleteTemp
ON SCHEDULE
EVERY 1 DAY
COMMENT 'Deletes temporary users if they dont activate their accounts'
DO
    DELETE FROM arion.Tempusers WHERE DATEDIFF(CURDATE(), regdate) >
3;

/* Enables events of database */
SET GLOBAL event_scheduler = ON;

/*
CREATE USER IF NOT EXISTS 'read'@'%' IDENTIFIED BY '***PASSWD***'; --
read access only, '%' guarantees access from any computer
GRANT SELECT ON arion.Users to 'read'@'%';
GRANT SELECT ON arion.Transactions to 'read'@'%';
GRANT SELECT ON arion.Recharges to 'read'@'%';
GRANT SELECT ON arion.Restaurants to 'read'@'%';

CREATE USER IF NOT EXISTS 'form'@'%' IDENTIFIED BY '***PASSWD***'; --
db access for new user manipulation
GRANT SELECT, INSERT, UPDATE ON arion.Users to 'form'@'%';
GRANT SELECT, INSERT, UPDATE ON arion.Tempusers to 'form'@'%';

CREATE USER IF NOT EXISTS 'scanner'@'%' IDENTIFIED BY '***PASSWD***';
-- inserts new transactions in db
GRANT SELECT, INSERT ON arion.Transactions to 'scanner'@'%';
GRANT SELECT ON arion.Users to 'scanner'@'%';
GRANT SELECT ON arion.Restaurants to 'scanner'@'%';

FLUSH PRIVILEGES;
*/

```

7.2. HEADER.PHP

```

<div class="row header">
    <div class="col-xs-6">
        <a href="/index.php"></a>
    </div>
    <div class="col-xs-6 text-right">
        <h2>Arion UFPR</h2>
    </div>
</div>

```

```

        <h4>Eliminando a fila do seu RU</h4>
    </div>
</div>
<hr style="width: 100%; height: 1px; background-color:#868686;">

```

7.3.INDEX.PHP

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery-3.1.0.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="https://use.fontawesome.com/f56e36ff4f.js"></script>
    <link
        rel="stylesheet"
        href="https://fonts.googleapis.com/css?family=Lato"
    >
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/hover-style.css">
    <link rel="shortcut icon" href="http://arion.ddns.net/images/favicon.ico"
    type="image/x-icon">
    <link rel="icon" href="http://arion.ddns.net/images/favicon.ico"
    type="image/x-icon">
    <title>Arion - Eliminando a fila do seu RU</title>
    <meta name="description" content="O sistema de acesso oficial da UFPR">
</head>
<body>
    <div class="container">
        <!-- HEADER SECTION -->
        <?php include("header.php"); ?>
        <!-- IMAGE AND SHORT DESCRIPTION SECTION -->
        <div class="row title-section text-center">
            <div class="col-md-12 text-uppercase">
                <h1 class="title">O sistema de acesso e pagamento oficial dos
                restaurantes da UFPR</h1>
            </div>
            <!-- BUTTONS SECTION -->
            <div class="text-center buttons">
                <div class="col-md-offset-3 col-md-3">
                    <a href="register.php" role="button" class="btn btn-primary
                    button-margin button-register text-uppercase">Crie sua conta</a>
                </div>
                <div class="col-md-3">
                    <a href="login.php" role="button" class="btn btn-primary
                    button-margin button-login text-uppercase">Faça seu login</a>
                </div>
            </div>
        </div>

        <!-- FEATURES TITLE SECTION -->
        <div class="row text-center title-margin">
            <div class="col-md-12 text-uppercase sub-title"><h2>Como
            funciona</h2></div>
        </div>
        <!-- FEATURES SECTION -->
        <div class="row features text-center">
            <div class="col-sm-3 feature">
                
                <h3 class="text-uppercase sub-title">Cadastro da
                Carteirainha</h3><hr style="width: 50%; height: 3px; background-color:red;">
                <p>Crie sua conta e cadastre sua carteirainha da UFPR aqui. O
                processo leva menos de 5 minutos.</p></div>

```



```

        <div class="col-sm-3 feature">
            
            <h3 class="text-uppercase sub-title">Insira seus créditos</h3><hr
style="width: 50%; height: 3px; background-color:green;">
            <p>Use seu cartão de crédito para facilitar o pagamento do
restaurante universitário. O sistema é 100% seguro e monitorado pela própria
UFPR.</p></div>
        <div class="col-sm-3 feature">
            
            <h3 class="text-uppercase sub-title">Seu RU sem filas</h3><hr
style="width: 50%; height: 3px; background-color:yellow;">
            <p>Acesse o RU normalmente utilizando sua carteirinha, sem filas!
O resto deixa que a gente cuida.</p></div>
        <div class="col-sm-3 feature">
            
            <h3 class="text-uppercase sub-title">Consulte seus
créditos</h3><hr style="width: 50%; height: 3px; background-color:blue;">
            <p>Seja na Web ou no seu celular, consulte e insira créditos em
menos de 2 minutos. Tudo isto porque sabemos que o tempo de um estudante
universitário é muito valioso.</p></div>
    </div>
    <!-- DEVELOPERS TITLE SECTION -->
    <div class="row text-center title-margin">
        <div class="col-md-12 text-uppercase sub-
title"><h2>Desenvolvedores</h2></div>
    </div>
    <div class="row text-center title-margin">
        <div class="col-md-12"><p>Como estudantes, entendemos a importância da
praticidade e segurança no nosso cotidiano. Por isso, trabalhamos
incansavelmente por uma plataforma fácil e segura para todos os estudantes da
UFPR.</p></div>
    </div>
    <!-- DEVELOPERS SECTION -->
    <div class="row developers text-center">
        <div class="col-sm-6 developer">
            <div class="hovereffect">
                
                <div class="overlay">
                    <p class="info">Estudante de Engenharia Elétrica na UFPR
desde 2013. Após longas horas de tempo perdido na fila do RU, Pedro começou a
desenvolver o sistema de acesso para que outros estudantes não percam este
mesmo tempo.</p>
                </div>
                <h3>Pedro Mantovani Antunes</h3><hr style="width: 50%; height:
3px; background-color:red;">
                <h4 class="dev-desc-margin">Engenharia Elétrica - UFPR<br />Back-
end e front-end developer</h4>
            </div>
            <div class="row developer-margin">
                <div class="col-xs-3 col-xs-offset-3">
                    <a href="https://www.facebook.com/pedro.mantovani.12"
target="_blank"><i class="fa fa-facebook fa-2x social-net-btn" aria-
hidden="true"></i></a>
                </div>
                <div class="col-xs-3">
                    <a href="https://www.linkedin.com/in/pedro-mantovani"
target="_blank"><i class="fa fa-linkedin fa-2x social-net-btn" aria-
hidden="true"></i></a>
                </div>
            </div>
        </div>
        <div class="col-sm-6 developer">
            <div class="hovereffect">

```

```

        
        <div class="overlay">
            <p class="info">Estudante de Engenharia Elétrica na UFPR
desde 2013. Cansado de carregar moedas e de problemas com o troco, Erik
começou a desenvolver o sistema de acesso para que tudo seja mais simples e
ágil, utilizando o sistema de créditos.</p>
        </div>
        <h3>Erik Nayan</h3><hr style="width: 50%; height: 3px;
background-color:green;">
        <h4 class="dev-desc-margin">Engenharia Elétrica - UFPR<br
/>Back-end e front-end developer</h4>
        </div>
        <div class="row developer-margin">
            <div class="col-xs-3 col-xs-offset-3">
                <a href="https://www.facebook.com/eriknayan"
target="_blank"><i class="fa fa-facebook fa-2x social-net-btn" aria-
hidden="true"></i></a>
            </div>
            <div class="col-xs-3">
                <a href="https://br.linkedin.com/in/eriknayan/en"
target="_blank"><i class="fa fa-linkedin fa-2x social-net-btn" aria-
hidden="true"></i></a>
            </div>
        </div>
    </div>
    <!-- TESTIMONY TITLE SECTION -->
    <div class="row testimonial-section">
        <div class="text-center title-margin">
            <div class="col-md-12 text-uppercase sub-title testimonial-
title"><h2>Testemunhos</h2></div>
        </div>
        <div class="text-center">
            <div class="col-md-12 testimonial-subtitle">Veja o que os outros
estudantes estão dizendo da plataforma.</div>
        </div>
    <!-- TESTIMONY SECTION -->

    <div class="text-center margin-bottom-30">
        <div class="col-md-4 testimonial">
            <div class="col-xs-offset-1 col-xs-10 margin-20">
                <div class="row white-bg img-rounded padding-25">
                    <div class="col-xs-12">
                        <p>Com o novo sistema de acesso, economizo pelo
menos 30 minutos todos os dias. Mais tempo para estudar cálculo e reprovar da
mesma forma!</p>
                    </div>
                    <div class="col-xs-6">
                        <p class="margin-top-20">- Engenheiro</p>
                    </div>
                    <div class="col-xs-6">
                        
                    </div>
                </div>
            </div>
        </div>
        <div class="col-md-4 testimonial">
            <div class="col-xs-offset-1 col-xs-10 margin-20">
                <div class="row white-bg img-rounded padding-25">
                    <div class="col-xs-12">
                        <p>Acredito que o sistema de cotas para alunos que
usem a plataforma de acesso ao RU seja uma proposta muito interessante. Viva a
igualdade!</p>
                    </div>
                    <div class="col-xs-6">

```

```

        <p class="margin-top-20"> - Historiadora</p>
    </div>
    <div class="col-xs-6">
        
    </div>
</div>
</div>
</div>
<div class="col-md-4 testimonial">
    <div class="col-xs-offset-1 col-xs-10 margin-20">
        <div class="row white-bg img-rounded padding-25">
            <div class="col-xs-12">
                <p>O sistema me ajuda a tornar meu almoço mais
ágil para que eu possa voltar a programar meu hack do GTA.</p>
            </div>
            <div class="col-xs-6">
                <p class="margin-top-20"> - Programador</p>
            </div>
            <div class="col-xs-6">
                
            </div>
        </div>
    </div>
</div>
</div>
</div>
<!-- FOOTER SECTION -->
<?php include("footer.php") ?>
</div>
</body>
</html>

```

7.4.FOOTER.PHP

```

<!-- FOOTER SECTION -->
    <div class="row footer text-center">
        <div class="col-md-4">
            <a href="http://maps.google.com/?q=-25.449753,-49.232942"
target="_blank">
                
                <address>
                    <strong>Universidade Federal do
Paraná</strong><br>
                    Centro Politécnico<br>
                    Curitiba, PR
                </address>
            </a>
        </div>
        <div class="col-md-4">
            <a href="mailto:pmantovani94@gmail.com">
                
                <p>pmantovani94@gmail.com</p>
            </a>
        </div>
        <div class="col-md-4">
            <a href="tel:+554191477772">

```

```

                
                <p>+55 41 9147-7772</p>
            </a>
        </div>
    </div>

```

7.5.LOGIN.PHP

```

<?php

require_once("utilities.php");

function showErrorMessage($msg) {
    $_SESSION['Error'] = $msg;
    include('login_page.php');
    die();
}

if($_SERVER['REQUEST_METHOD'] == "GET") {

    //Check if logout was requested
    if (isset($_GET["logout"])) {
        // Delete cookie
        deleteCookie();
    }

    // Checks if cookie is set for this connection
    else if (isset($_COOKIE["session"])) {
        if (validateCookie($_COOKIE["session"])) {
            // If cookie is valid, redirect to home page
            header("Location: welcome.php");
            die();
        }
        else {
            // If cookie is invalid, erase it
            deleteCookie();
        }
    }
    // Shows login page in case no data was posted to server and no
    cookie was set
    include("login_page.php");
    die();
}
// Executed when email and passwd are sent to login
else {

    // In case post to site didn't contain email, password or cookie,
    show error message
    if ((empty($_POST["email"]) || empty($_POST["passwd"])) &&
!isset($_COOKIE["session"])) {
        showErrorMessage("Email ou senha não foram fornecidos.");
    }

    // Validate by cookie
    if (isset($_COOKIE["session"])) {
        if (validateCookie($_COOKIE["session"])) {
            // Valid cookie, go to welcome page
            header("Location: welcome.php");

```

```

        die();
    }
    else {
        // Invalid cookie, reset it and output error message
        deleteCookie();
        showErrorMessage("Erro no login. Por favor tente
novamente.");
    }
}

require_once("db_operations.php");
$db = new DBOperator();

if ($db->isPasswordValid($_POST["email"], $_POST["passwd"])) {
    // Validated! Redirect to welcome page. Check how to validate
    email and passwd again after redirect

    if (isset($_POST["remember"])) {
        // Expires in 60 days, httponly
        // TODO: Change secure flag to true after HTTPS is
        implemented
        setcookie("session", createSecureCookie($_POST["email"]),
time()+60*60*24*60, "/", "arion.ddns.net", false, true);
    }
    else {
        // Expires at the end of session (when browser is closed),
httponly
        // TODO: Change secure flag to true after HTTPS is
        implemented
        setcookie("session", createSecureCookie($_POST["email"]),
0, "/", "arion.ddns.net", false, true);
    }
    // Redirects to welcome page
    header("Location: welcome.php");
}
else {
    showErrorMessage("Login ou senha inválidos. Por favor tente
novamente.");
}

}
?>

```

7.6.LOGIN_PAGE.PHP

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery-3.1.0.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet">
    <link rel="stylesheet" href="css/style.css">
    <title>Acesso RU UFPR - Cadastro</title>

```

```

        <meta name="description" content="O sistema de acesso oficial da
UFPR">
</head>
<body>
<div class="container">
    <!-- HEADER SECTION -->
    <?php include("header.php"); ?>

    <!-- CHECKS IF ANY ERROR OCCURED IN LOGIN.PHP -->
    <?php
    if (isset($_SESSION['Error'])) {
        echo '
        <div class="row">
            <div class="col-sm-4 col-sm-offset-4 text-center">
                <div class="alert alert-danger">
                    ' . $_SESSION['Error'] . '
                </div>
            </div>
        </div>';
    }
    ?>

    <!-- LOGIN FIELDS -->
    <div class="row form-margin">
        <div class="col-sm-4 col-sm-offset-4">
            <form role="form" method="post">
                <div class="form-group">
                    <label for="emailInput">Endereço de Email</label>
                    <input type="email" name="email" class="form-
control" id="emailInput" placeholder="Digite seu email" autofocus>
                </div>
                <div class="form-group">
                    <label for="passwordInput">Senha</label>
                    <input type="password" name="passwd" class="form-
control" id="passwordInput" placeholder="Senha">
                </div>
                <div class="checkbox">
                    <label><input type="checkbox" name="remember"
value="checked">Lembre-se do usuário</label>
                </div>
                <button type="submit" class="btn btn-
default">Login</button>
            </form>
        </div>
    </div>
    <!-- FOOTER SECTION -->
    <?php include("footer.php") ?>
</body>

```

7.7.REGISTER.PHP

```

<?php
if ($_SERVER["REQUEST_METHOD"] === 'GET') {
    // Shows registration form in case no data was posted to server
    include("register_page.php");
    die();
}
// If data was posted to server, validate and add to db
else {

```

```

function showErrorMessage($msg) {
    $_SESSION['Error'] = $msg;
    include('register_page.php');
    die();
}

// Checks if all fields were filled
if (empty($_POST["name"]) || empty($_POST["email"]) ||
empty($_POST["grr"])
    || empty($_POST["barcode"]) || empty($_POST["passwd"]) ||
empty($_POST["role"])) {
    // Shows error message
    showErrorMessage("Um ou mais campos preenchidos são inválidos.
Por favor tente novamente.");
}

require("captcha_validation.php");
if (!validateCaptcha($_POST["g-recaptcha-response"])) {
    // Captcha failed to validate, show error page
    showErrorMessage("Erro na validação do captcha. Por favor
tente novamente.");
}

require_once("db_operations.php");
$db = new DBOperator();

// Validate input from POST parameters
// TODO: fix regex of preg_match call
if (/*!preg_match("[A-Za-z\x20áâãäåêëéöøô]", $_POST["name"]) ||*/
!ctype_digit($_POST["grr"]) ||
    !ctype_digit($_POST["barcode"]) || ($_POST["role"] !=
"Estudante" && $_POST["role"] != "Professor" &&
    $_POST["role"] != "Servidor") || strlen($_POST["name"]) >
50 || strlen($_POST["email"]) > 50 ||
    strlen($_POST["passwd"]) > 35 || strlen($_POST["grr"]) > 8 ||
strlen($_POST["barcode"]) > 12) {
    showErrorMessage("Um ou mais campos preenchidos são inválidos.
Por favor tente novamente.");
}

if ($db->isUserInDb($_POST["email"], $_POST["barcode"])) {
    showErrorMessage("O usuário que você está tentando cadastrar
já existe.");
}

$key = $db->insertUserInTemporaryTable($_POST["barcode"],
$_POST["name"], $_POST["email"], $_POST["passwd"], $_POST["grr"],
$_POST["role"]);
if (!$key) {
    showErrorMessage("Tivemos um erro ao cadastrá-lo. Por favor
tente novamente mais tarde");
}

require("send_email.php");
if (!sendEmail($_POST["name"], $_POST["email"], $key)) {
    showErrorMessage("Tivemos um erro ao enviar seu email. Tente
novamente em 72 horas.");
}

// TODO: Implement front-end for register_success.php

```

```

        header("Location: register_success.php");
    }
    ?>

```

7.8. REGISTER_PAGE.PHP

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery-3.1.0.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet">
    <link rel="stylesheet" href="css/style.css">
    <script src="js/validator.min.js"></script>
    <script src="js/jquery-3.1.0.min.js"></script>
    <script src='https://www.google.com/recaptcha/api.js'
async></script>
    <title>Acesso RU UFPR - Cadastro</title>
    <meta name="description" content="O sistema de acesso oficial da
UFPR">
</head>
<body>
<div class="container">
    <!-- HEADER SECTION -->
    <?php include("header.php"); ?>

    <!-- CHECKS IF ANY ERROR OCCURED IN REGISTER.PHP -->
    <?php
    if (isset($_SESSION['Error'])) {
        echo '
        <div class="row">
            <div class="col-sm-4 col-sm-offset-4 text-center">
                <div class="alert alert-danger">
                    ' . $_SESSION['Error'] . '
                </div>
            </div>
        </div>';
    }
    ?>

    <!-- REGISTRATION FIELDS -->
    <div class="row">
        <div class="col-sm-4 col-sm-offset-4">
            <form method="post" data-toggle="validator" role="form">
                <!-- NOME -->
                <div class="form-group has-feedback">
                    <label for="nameInput">Nome</label>
                    <input type="text" name="name" class="form-
control" pattern="[A-Za-z\x20áâãäåæçèéêëìíîïð] {1,}" id="nameInput" data-
error="Nome não pode possuir números ou símbolos." placeholder="Digite
seu nome" required autofocus>
                    <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>

```



```

        <div class="help-block with-errors"></div>
    </div>
    <!-- EMAIL -->
    <div class="form-group has-feedback">
        <label for="emailInput">Endereço de Email</label>
        <input type="email" name="email" class="form-
control" id="emailInput" data-error="Email inválido"
placeholder="Digite seu email" required>
        <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>
        <div class="help-block with-errors"></div>
    </div>
    <!-- CONFIRME EMAIL -->
    <div class="form-group has-feedback">
        <label for="emailInput">Confirme seu Email</label>
        <input type="email" class="form-control"
id="emailInput" data-match="#emailInput" data-match-error="Emails não
conferem" placeholder="Confirme seu email" required>
        <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>
        <div class="help-block with-errors"></div>
    </div>
    <!-- VINCULO -->
    <div class="form-group">
        <label>Vínculo com a UFPR</label>
        <select class="form-control" name="role">
            <option>Estudante</option>
            <option>Professor</option>
            <option>Servidor</option>
        </select>
    </div>
    <!-- GRR -->
    <div class="form-group has-feedback">
        <label for="grrInput">Número de Matrícula
(GRR)</label>
        <input type="number" name="grr" class="form-
control" id="grrInput" placeholder="XXXXXXXX" min="10000000"
max="99999999" maxlength="8" oninput="maxLengthCheck(this)" data-
error="GRR Inválido" required/>
        <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>
        <div class="help-block with-errors"></div>
    </div>
    <!-- NUM. CARTEIRINHA -->
    <div class="form-group has-feedback">
        <label for="grrInput">Número da carteirinha
(Código de barras)</label>
        <input type="number" name="barcode" class="form-
control" placeholder="Carteirinha" min="1000000000000"
max="999999999999" maxlength="12" oninput="maxLengthCheck(this)" data-
error="Número de carteirinha inválido" required/>
        <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>
        <div class="help-block with-errors"></div>
    </div>
    <!-- SENHA -->
    <div class="form-group has-feedback">
        <label for="passwordInput">Senha</label>
        <input type="password" name="passwd" class="form-
control" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{6,}" data-

```

```

error="Senha deve conter no mínimo um número, uma letra maiúscula e
uma minúscula, e 6 ou mais caracteres." id="passwordInput"
placeholder="Senha" required>
    <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>
    <div class="help-block with-errors"></div>
</div>
<!-- CONFIRMAÇÃO DE SENHA -->
<div class="form-group has-feedback">
    <label for="passwdConfirmInput">Confirme sua
senha</label>
    <input type="password" class="form-control" data-
match="#passwordInput" data-match-error="Senhas não conferem"
placeholder="Confirme sua senha" required>
    <span class="glyphicon form-control-feedback"
aria-hidden="true"></span>
    <div class="help-block with-errors"></div>
</div>
<!-- BOTÃO ENVIAR -->
<div class="form-group g-recaptcha" data-
sitekey="6Lc_bikTAAAAAP2BKGGqMG1jKeYRwsJi-SLWT2yL"></div>
    <button type="submit" class="btn btn-default"
id="submitButton">Enviar</button>
</form>
</div>
</div>
<!-- FOOTER SECTION -->
<?php include("footer.php") ?>
</body>
<script type="text/javascript">
    function maxLengthCheck(object) {
        if(object.value.length > object.maxLength)
            object.value = object.value.slice(0, object.maxLength)
    }
</script>

<script type="text/javascript">
    $(document).ready(function() {
        $("#submitButton").click(function() {
            if (grecaptcha.getResponse() == ""){
                alert("Por favor, valide o captcha para continuar.");
                return false;
            }
        })
    })
</script>
</html>

```

7.9. REGISTER_SUCESS.PHP

```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">

```

```

        <title>Acesso RU UFPR - Cadastro</title>
        <meta name="description" content="O sistema de acesso oficial da
UFPR">
    </head>
    <body>
    <div class="container">
        <div class="row" style="margin-top: 40px">
            <div class="col-xs-6 col-xs-offset-3">
                <div class="alert alert-success">
                    Seu cadastro foi realizado com sucesso! Verifique seu
email para ativar sua conta.
                </div>
            </div>
        </div>
    </div>
    </body>

```

7.10. STYLE.CSS

```

* {
    font-family: 'Lato', sans-serif;
    color: #848484;
}

.title-section {
    background: linear-gradient( rgba(0, 0, 0, 0.5), rgba(0, 0, 0,
0.5) ), url(../images/ufpr2.jpg) no-repeat center center fixed;
    -webkit-background-size: cover;
    -moz-background-size: cover;
    -o-background-size: cover;
    background-size: cover;
    min-height: 500px;
}

.title {
    margin-top: 200px;
    color: #FFFFFF;
    font-size: 40px;
    font-weight: 900;
}

.buttons {
    margin-top: 80px;
    margin-bottom: 40px;
}

.button-margin {
    margin-top: 20px;
    padding-right: 35px;
    padding-left: 35px;
    padding-top: 13px;
    padding-bottom: 13px;
    font-size: 16px;
}

.title-margin {
    margin-top: 40px;
    margin-bottom: 40px;
}

```

```

.sub-title {
    color: #000000;
}

.img-feature {
    max-height: 130px;
}

.footer {
    margin-top: 40px;
    background-color: #111111;
}

a {
    color: #848484;
}

.margin-20 {
    margin-top: 20px;
    margin-bottom: 20px;
}

.padding-25 {
    padding: 25px;
}

.margin-top-20 {
    margin-top: 20px;
}

.margin-bottom-30 {
    margin-bottom: 30px;
}

.button-register {
    background-color: #e96656;
    border-color: #e96656;
    -webkit-transition: all 0.3s ease-in-out;
    transition: all 0.3s ease-in-out;
    transition-property: all;
    transition-duration: 0.3s;
    transition-timing-function: ease-in-out;
    transition-delay: initial;
    font-weight: lighter;
}

/* Changes background color when focused */
.button-register:hover, .button-register:focus {
    background-color: #9B4439;
    border-color: #9B4439;
}

.button-login {
    background-color: #1e9e6b;
    border-color: #1e9e6b;
    -webkit-transition: all 0.3s ease-in-out;
    transition: all 0.3s ease-in-out;
    transition-property: all;
    transition-duration: 0.3s;
}

```

```

        transition-timing-function: ease-in-out;
        transition-delay: initial;
        font-weight: lighter;
    }

    .button-login:hover, .button-login:focus {
        background-color: #146947;
        border-color: #146947;
    }

    .testimonial-section {
        background: #8c7620;
    }

    .testimonial-title {
        padding-top: 40px;
        color: white;
    }

    .testimonial-subtitle {
        color: white;
        margin-bottom: 40px;
    }

    .img-testimonial {
        max-width: 70px;
    }

    .developer-margin {
        margin-bottom: 40px;
    }

    .dev-desc-margin {
        margin-bottom: 20px;
    }

    .social-net-btn {
        color: #848484;
        border-color: #848484;
        -webkit-transition: all 0.3s ease-in-out;
        transition: all 0.3s ease-in-out;
        transition-property: all;
        transition-duration: 0.3s;
        transition-timing-function: ease-in-out;
        transition-delay: initial;
    }

    .social-net-btn:hover, .social-net-btn:focus {
        color: #337ab7;
        border-color: #337ab7;
    }

    .white-bg {
        background-color: white;
    }

    /* Removes spinner for number fields on registration */
    input::-webkit-outer-spin-button,
    input::-webkit-inner-spin-button {
        /* display: none; <- Crashes Chrome on hover */
    }

```

```

        -webkit-appearance: none;
        margin: 0; /* <-- Apparently some margin are still there even
though it's hidden */
    }

```

7.11. CAPTCHA_VALIDATION.PHP

```

<?php
    // Checks captcha validations
    function validateCaptcha($response) {
        require_once('utilities.php');

        $recaptchaUrl =
'https://www.google.com/recaptcha/api/siteverify';
        // Data to be sent via POST method to recaptcha server
        $data = array('secret' => Keys::getCaptchaKey(),
'response' => $response);
        $options = array(
            'http' => array(
                'header' => "Content-type: application/x-www-
form-urlencoded\r\n",
                'method' => 'POST',
                'content' => http_build_query($data)
            )
        );
        $context = stream_context_create($options);
        // Sends request to server and gets the response
        $result = file_get_contents($recaptchaUrl, false,
$context);
        // If result is false, there was an error getting the
response
        if ($result === FALSE) {
            /* Handle error */
            //echo "Error getting response from catpcha server
<br>";
            return false;
        }
        else {
            $json_res = json_decode($result, true);
            if (!$json_res["success"]) {
                // Recaptcha failed!
                //echo "Error in captcha validation: <br>" .
var_dump($json_res["error-codes"]);
                return false;
            }
            else {
                // Recaptcha was successful
                return true;
            }
        }
    }

?>

```

7.12. SEND_EMAIL.PHP

```
<?php
function sendEmail($name, $email, $key) {
    // Includes phpmailer library
    require_once("class/PHPMailerAutoload.php");
    require_once("utilities.php");

    // Instantiates PHPMailer class
    $mail = new PHPMailer(true);

    // SMTP message type
    $mail->IsSMTP();

    try {
        $mail->Host = 'smtp.gmail.com'; // SMTP server address
        $mail->SMTPAuth = true; // Use SMTP authentication
        $mail->Port = 587; // Use SMTP port 587
        $mail->Username = 'arion.ufpr'; // Username at SMTP server
        $mail->Password = Keys::getEmailPasswd(); // Password at
SMTP server

        // Sender info
        $mail->SetFrom('arion.ufpr@gmail.com', 'Pedro Mantovani
Antunes'); // Email and name
        $mail->Subject = 'Bem-vindo ao sistema Arion'; // Email
subject

        // Receiver info
        $mail->AddAddress($email, $name);

        // Optional fields
        // $mail->AddCC('destinatario@dominio.com.br',
'Destinatario'); // CC
        // $mail->AddBCC('destinatario_oculto@dominio.com.br',
'Destinatario2'); // BCC (hidden copy)
        // $mail->AddAttachment('images/phpmailer.gif'); // Add an
attachment

        // Email body (html)
        $mail->MsgHTML("Olá " . $name . "! <br><br>
        Seja bem-vindo ao sistema Arion. Trabalhamos duro para
diminuir a fila do seu Restaurante Universitário.<br><br>
        Mas antes, precisamos que ative sua conta com o link
abaixo:

        <a href='arion.ddns.net/confirmation.php?k=" . $key .
"'> Ative sua conta aqui!</a><br><br>
        Obrigado!
        Equipe Arion");

        // To include an html file instead
        // $mail->MsgHTML(file_get_contents('arquivo.html'));

        // Returns true if successful
        return $mail->Send();

    } catch (phpmailerException $e) {
        return false; // PHPMailer error message
    }
}
```

```

    }
}
?>

```

7.13. CONFIRMATION.PHP

```

<?php

    require_once('utilities.php');

    if (empty($_GET["k"])) {
        die ("Missing confirmation key");
    }

    $key = $_GET["k"];

    require_once("db_operations.php");
    $db = new DBOperator();

    if (!$db->insertUserInPermanentTable($key)) {
        die("Tivemos um erro ao confirmar seu usuário. Tente confirmar
ou criar sua conta novamente.");
    }

    header("refresh:5;url=login.php");
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script src="js/jquery-3.1.0.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet">
    <link rel="stylesheet" href="css/style.css">
    <title>Acesso RU UFPR - Confirmação</title>
    <meta name="description" content="O sistema de acesso oficial da
UFPR">
</head>
<body>
    <div class="container">
        <!-- HEADER SECTION -->
        <?php include("header.php"); ?>

        <div class="row">
            <div class="col-xs-12">
                <div class="alert alert-success text-center">
                    Obrigado! Sua conta foi ativada com sucesso!<br>
                    Você será redirecionado em 5 segundos.
                </div>
            </div>
        </div>
    </div>
</div>

```


</body>

7.14. DB_OPERATIONS.PHP

```
<?php

require_once('utilities.php');

class DBOperator {
    private $conn;

    function __construct() {

        $dbhost = 'localhost';
        $dbuser = Keys::getDbUser();
        $dbpass = Keys::getDbPasswd();
        $dbname = 'arion';
        $this->conn = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

        // Checks if successfully connected to db
        if($this->conn->connect_errno) {
            error_log("db_operations: Failed to connect to db.");
            throw new Exception("Unable to connect do db.");
        }
    }

    function __destruct() {
        $this->conn->close();
    }

    // Returns all user information in associative array
    public function getUserInfoFromSessionCookie($cookie) {
        $email = extractEmailFromCookie($cookie);

        $query = "SELECT name, email, type, grr, cardId, balance FROM
Users WHERE email = '$email'";
        $result = $this->conn->query($query);
        if (!$result) {
            error_log("db_operations: Error when fetching user info");
            return false;
        }
        else {
            $row = $result->fetch_assoc();
            $result->close();
            return $row;
        }
    }

    public function getLastFiveTransactions($cookie) {
        $cardId = $this->getUserInfoFromSessionCookie($cookie) ["cardId"];

        $query = "SELECT Transactions.tranId, Transactions.tranTime,
Transactions.value, Transactions.type, Restaurants.restName FROM
Transactions LEFT JOIN Restaurants ON
Transactions.restId=Restaurants.restId WHERE
Transactions.cardId='$cardId' ORDER BY Transactions.tranTime DESC
LIMIT 5;";
    }
}
```

```

        return $this->conn->query($query);
    }

    public function isUserInDb($email, $cardId = NULL) {
        $email = $this->escape_string($email);
        if (!is_null($cardId)) {
            $cardId = $this->escape_string($cardId);
        }

        $checkQuery = is_null($cardId) ? "SELECT * FROM Users WHERE
email='$email';" :
                                "SELECT * FROM Users WHERE
email='$email' OR cardId='$cardId';";

        $checkCursor = $this->conn->query($checkQuery);
        if (!$checkCursor) {
            error_log("db_operations: Failed to check if user exists
in db");
            die();
        }

        $hasMatch = ($checkCursor->num_rows > 0);
        $checkCursor->close();
        return $hasMatch;
    }

    // Return: confirmation key used in confirmation email
    public function insertUserInTemporaryTable($cardId, $name, $email,
$passwd, $grr, $role) {

        // Escape strings
        $cardId = $this->escape_string($cardId);
        $name = $this->escape_string($name);
        $email = $this->escape_string($email);
        $passwd = $this->escape_string($passwd);
        $grr = $this->escape_string($grr);
        $role = $this->escape_string($role);

        // Converts our role string to a correspondent number before
inserting into the db
        $roleToNumber = array (
            "Estudiante" => 0,
            "Professor" => 1,
            "Servidor" => 2
        );
        $roleNumber = $roleToNumber[$role];

        // Hashes password using BCrypt method
        $passwdHashed = password_hash($passwd, PASSWORD_BCRYPT);

        // Creates random confirmation key
        $confirmationKey = $name . $email . date('mY');
        $confirmationKey = md5($confirmationKey);

        // Get registration date with current time
        $date = date_create();
        $regdate = date_format($date, "Y-m-d");
    }

```

```

        $query          =          "INSERT          INTO          Tempusers
(cardId,name,email,password,grr,type,regdate,confirmkey)
        VALUES (

'$cardId','$name','$email','$passwdHashed','$grr','$roleNumber','$regdate',
'$confirmationKey');"
        $retval = $this->conn->query($query);

        if (!$retval) {
            error_log("db_operations: Failed to insert user in
temporary table.");
            return false;
        }
        else {
            return $confirmationKey;
        }
    }

    public function insertUserInPermanentTable($confirmationKey) {
        $confirmationKey = $this->escape_string($confirmationKey);

        $query = "SELECT * FROM Tempusers WHERE confirmkey =
'$confirmationKey'";
        $result = $this->conn->query($query);

        if (!$result) {
            error_log("db_operations: Error when querying db.");
            return false;
        }

        if ($result->num_rows == 0) {
            return false;
        }
        else if ($result->num_rows > 1) {
            $delQuery = "DELETE FROM Tempusers WHERE confirmkey =
'$confirmationKey'";
            $this->conn->query($delQuery);
            $this->conn->commit();
            return false;
        }

        $row = $result->fetch_assoc();
        $result->close();

        // Get info from temporary record
        $cardId = $row['cardId'];
        $name = $row['name'];
        $email = $row['email'];
        $password = $row['password'];
        $grr = $row['grr'];
        $type = $row['type'];

        //Get current date
        $date = date_create();
        $regdate = date_format($date,"Y-m-d");
        $expdate =
date_add($date,date_interval_create_from_date_string("10 years"));
        $expdate = date_format($expdate,"Y-m-d");

        $addQuery = "INSERT INTO Users (

```

```

        cardId,name,email,password,grr,type,regdate,expdate)
VALUES (

'$cardId','$name','$email','$password','$grr','$type','$regdate','$exp
date');"
    if (!$this->conn->query($addQuery)) {
        error_log("db_operations: Couldn't insert user in
permanent table.");
        $this->conn->rollback();
        return false;
    }

    // Delete user from temporary table
    $delQuery = "DELETE FROM Tempusers WHERE confirmkey =
'$confirmationKey'";
    $this->conn->query($delQuery);

    // Commit transaction changes
    $this->conn->commit();
    return true;
}

public function isPasswordValid($email, $passwd) {
    $email = $this->escape_string($email);
    $passwd = $this->escape_string($passwd);

    $checkQuery = "SELECT password FROM Users WHERE
email='$email'";
    $resultCursor = $this->conn->query($checkQuery);

    if (!$resultCursor) {
        error_log("db_operations: Error querying the password in
db.");
        return false;
    }

    $passInDb = $resultCursor->fetch_assoc()["password"];
    $resultCursor->close();

    // Checks if typed password matches with hashed one in db
    return password_verify($passwd, $passInDb);
}

public function escape_string($string) {
    return $this->conn->real_escape_string($string);
}

}

?>

```

7.15. UTILITIES.PHP

```

<?php

// Validates a cookie received from a client.
// Returns true if cookie is valid, false otherwise
function validateCookie($cookie) {

```

```

$secretKey = Keys::getCookieSecretKey() ;

$splitCookie = explode("|", $cookie);
$numElements = count($splitCookie);
// Parse number of elements in cookie
if ($numElements != 2) {
    return false;
}

require_once("db_operations.php");
$db = new DBOperator();

// Check if user is in db
if (!$db->isUserInDb($splitCookie[0])) {
    return false;
}

// Check if cookie hash is correct
if (hash_hmac("sha256", $splitCookie[0], $secretKey) ===
$splitCookie[1]) {
    return true;
}
else {
    return false;
}
}

// Creates a secure cookie using hmac with a secret key
function createSecureCookie($email) {
    $secretKey = Keys::getCookieSecretKey();
    return $email . "|" . hash_hmac("sha256", $email, $secretKey);
}

// Deletes session cookie
function deleteCookie() {
    setcookie("session", "", time()-36000, "/", "arion.ddns.net");
    return;
}

function extractEmailFromCookie($cookie) {
    $splitCookie = explode("|", $cookie);
    return $splitCookie[0];
}

// Class containing the functions to retrieve passwords stored in file
class Keys {

    public static function getDbUser() {
        return Keys::getFileArray()[0];
    }
    public static function getDbPasswd() {
        return Keys::getFileArray()[1];
    }
    public static function getCaptchaKey() {
        return Keys::getFileArray()[2];
    }
    public static function getEmailPasswd() {
        return Keys::getFileArray()[3];
    }
    public static function getCookieSecretKey() {

```

```

        return Keys::getFileArray()[4];
    }

    static function getFileArray() {
        $mKeys = file_get_contents("../../../keys");
        return explode("\n", $mKeys);
    }
}

?>

```

7.16. WELCOME.PHP

```

<?php

session_start();

require_once("utilities.php");
require_once("db_operations.php");

if (!isset($_COOKIE["session"])) {
    // Redirect to login page in case there are no session cookies
    header("Location: login.php");
}
if (!validateCookie($_COOKIE["session"])) {
    // Redirect to login page in case the session cookie is invalid
    header("Location: login.php?logout=true");
}

$db = new DBOperator();

    if (!isset($_SESSION['name']) || !isset($_SESSION['email']) ||
!isset($_SESSION['role']) || !isset($_SESSION['grr']) ||
!isset($_SESSION['cardId']) || !isset($_SESSION['balance'])) {
        $info = $db->getUserInfoFromSessionCookie($_COOKIE["session"]);

        if (!$info) {
            die();
        }
        else {
            // Puts info in session variables
            $_SESSION["name"] = $info["name"];
            $_SESSION["email"] = $info["email"];
            $_SESSION["role"] = $info["type"];
            $_SESSION["grr"] = $info["grr"];
            $_SESSION["cardId"] = $info["cardId"];
            $_SESSION["balance"] = $info["balance"];
        }
    }

    if (!isset($_SESSION["transactions"])) {
        $transactions = $db->getLastFiveTransactions($_COOKIE["session"]);
        if (!$transactions) {
            die();
        }
    }
}

```

```

        else {
            $_SESSION["transactions"] = $transactions;
        }
    }
    ?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet">
    <link rel="stylesheet" href="css/welcome.css">
    <title>Acesso RU UFPR - Inicio</title>
    <meta name="description" content="O sistema de acesso oficial da
UFPR">
</head>
<body>
    <div class="container">
        <!-- HEADER SECTION -->
        <?php include("header.php"); ?>
        <!-- MENU SECTION -->
        <nav class="navbar navbar-default">
            <div class="container-fluid">
                <div class="navbar-header">
                    <a class="navbar-brand" href="#">Arion</a>
                </div>
                <ul class="nav navbar-nav">
                    <li class="active"><a href="#">Saldo</a></li>
                    <li><a href="/recharge.php">Recarga</a></li>
                    <li><a href="#">Transações</a></li>
                    <li><a href="/personalinfo.php">Informações
pessoais</a></li>
                </ul>
                <ul class="nav navbar-nav navbar-right">
                    <li>
                        <a href="login.php?logout=true" class="navbar-
left">Sair</a>
                    </li>
                </ul>
            </div>
        </nav>
        <!-- BALANCE SECTION -->
        <div class="row text-center">
            <div class="col-xs-12">
                <h3>Bem-vindo <?php echo $_SESSION["name"] ?>!</h3>
                <h3 class="section-margin">Seu saldo é de: <?php echo
'R$' . $_SESSION["balance"]; ?></h3>
                <a href="/recharge.php">Faça uma recarga aqui</a>
                <h3 class="section-margin">Suas últimas 5 transações
foram:</h3>
                <table class="table table-hover text-center">
                    <thead>
                        <tr>

```

```

        <th>ID</th>
        <th>Data e Hora</th>
        <th>Valor</th>
        <th>Local</th>
    </tr>
</thead>
<tbody>
    <?php
    // Fill up table dynamically
    while ($row = $_SESSION["transactions"]-
>fetch_array()) {
        echo "<tr>";
        echo "<td>" . $row["tranId"] . "</td>";
        echo "<td>" . $row["tranTime"] . "</td>";
        if ($row["type"]) { echo "<td>" .
$row["value"] . "</td>"; }
        else { echo "<td>-" . $row["value"] .
"</td>"; }
        echo "<td>" . $row["restName"] . "</td>";
        echo "</tr>";
    }
    ?>
</tbody>
</table>
<a href="#">Veja seu histórico completo aqui</a>
</div>
</div>
</div>
</body>
</html>

```

7.17. RECHARGE.PHP

```

<?php

session_start();

require_once("utilities.php");

if (!isset($_COOKIE["session"])) {
    // Redirect to login page in case there are no session cookies
    header("Location: login.php");
}
if (!validateCookie($_COOKIE["session"])) {
    // Redirect to login page in case the session cookie is invalid
    header("Location: login.php?logout=1");
}

if (!isset($_SESSION["cardId"])) {
    require_once("db_operations.php");
    $db = new DBOperator();

    $info = $db->getUserInfoFromSessionCookie($_COOKIE["session"]);

    if (!$info) {
        die();
    }
    else {

```



```

        // Puts info in session variables
        $_SESSION["name"] = $info["name"];
        $_SESSION["email"] = $info["email"];
        $_SESSION["role"] = $info["type"];
        $_SESSION["grr"] = $info["grr"];
        $_SESSION["cardId"] = $info["cardId"];
        $_SESSION["balance"] = $info["balance"];
    }
}

?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <script type="text/javascript" src="js/jquery-
3.1.0.min.js"></script>
    <script type="text/javascript" src="js/bootstrap.min.js"></script>
    <script type="text/javascript" src="js/validator.min.js"></script>
    <link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/font-
awesome/4.3.0/css/font-awesome.min.css">
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.13.1/jquery.validate.min.js"></script>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery.payment/1.2.3/jquer
y.payment.min.js"></script>
    <script type="text/javascript" src="js/card.js"></script>
    <link rel="stylesheet" href="css/recharge.css">
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet">
    <link rel="stylesheet" href="css/welcome.css">
    <title>Acesso RU UFPR - Recarga</title>
    <meta name="description" content="O sistema de acesso oficial da
UFPR">
</head>
<body>
    <div class="container">
        <!-- HEADER SECTION -->
        <?php include("header.php"); ?>
        <!-- MENU SECTION -->
        <nav class="navbar navbar-default">
            <div class="container-fluid">
                <div class="navbar-header">
                    <a class="navbar-brand" href="#">Arion</a>
                </div>
                <ul class="nav navbar-nav">
                    <li><a href="/welcome.php">Saldo</a></li>
                    <li class="active"><a href="#">Recarga</a></li>
                    <li><a href="#">Transações</a></li>
                    <li><a href="/personalinfo.php">Informações
pessoais</a></li>
                </ul>
                <ul class="nav navbar-nav navbar-right">
                    <li>

```

```

        <a href="login.php?logout=true" class="navbar-
left">Sair</a>
        </li>
    </ul>
</div>
</nav>
<div class="row">
    <div class="col-sm-4 col-sm-offset-4">
        <p>Para realizar uma recarga, preencha os dados do seu
cartão de crédito abaixo. Todas as informações enviadas são
seguras.</p>
        <div class="panel panel-default credit-card-box">
            <div class="panel-heading display-table" >
                <div class="row display-tr" >
                    <h3 class="panel-title display-td"
>Detalhes de pagamento</h3>
                    <div class="display-td" >
                        
                    </div>
                </div>
            </div>
            <div class="panel-body">
                <form role="form" id="payment-form"
method="POST" action="javascript:void(0);">
                    <div class="row">
                        <div class="col-xs-12">
                            <div class="form-group">
                                <label for="cardId">Cartão a ser
recarregado</label>
                                <input type="text" name=cardId
class="form-control" id=cardId value="<?php echo
$_SESSION['cardId']?>" readonly>
                                </div>
                                <div class="form-group">
                                    <label>Valor da
Recarga</label>
                                    <select class="form-control"
name="value">
                                        <option>R$ 10,00</option>
                                        <option>R$ 20,00</option>
                                        <option>R$ 30,00</option>
                                        <option>R$ 50,00</option>
                                        <option>R$ 100,00</option>
                                    </select>
                                </div>
                            </div>
                        <div class="row">
                            <div class="col-xs-12">
                                <div class="form-group">
                                    <label for="cardNumber">Número
do cartão de crédito</label>
                                    <div class="input-group">
                                        <input
type="tel"
class="form-control"
name="cardNumber"
placeholder="Número do
cartão"

```

```

                                autocomplete="cc-
number"
                                required autofocus
                                />
                                <span class="input-group-
addon"><i class="fa fa-credit-card"></i></span>
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                <div class="col-xs-12">
                                <div class="form-group">
                                <label for="cardName">Nome no
cartão</label>
                                <input
                                type="text"
                                class="form-control caps-
lock"
                                name="cardName"
                                placeholder="Nome no
cartão"
                                autocomplete="cc-name"
                                required autofocus
                                />
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                <div class="col-xs-7 col-md-7">
                                <div class="form-group">
                                <label for="cardExpiry">Data
de validade</label>
                                <input
                                type="tel"
                                class="form-control"
                                name="cardExpiry"
                                placeholder="MM / AA"
                                autocomplete="cc-exp"
                                required
                                />
                                </div>
                                </div>
                                <div class="col-xs-5 col-md-5 pull-
right">
                                <div class="form-group">
                                <label for="cardCVC">Cód.
Verificação</label>
                                <input
                                type="tel"
                                class="form-control"
                                name="cardCVC"
                                placeholder="CVC"
                                autocomplete="cc-csc"
                                required
                                />
                                </div>
                                </div>
                                </div>
                                <div class="row">

```



```

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link href="https://fonts.googleapis.com/css?family=Lato"
rel="stylesheet">
    <link rel="stylesheet" href="css/welcome.css">
    <title>Acesso RU UFPR - Inicio</title>
    <meta name="description" content="O sistema de acesso oficial da
UFPR">
</head>
<body>
    <div class="container">
        <!-- HEADER SECTION -->
        <?php include("header.php"); ?>
        <!-- MENU SECTION -->
        <nav class="navbar navbar-default">
            <div class="container-fluid">
                <div class="navbar-header">
                    <a class="navbar-brand" href="#">Arion</a>
                </div>
                <ul class="nav navbar-nav">
                    <li><a href="/welcome.php">Saldo</a></li>
                    <li><a href="/recharge.php">Recarga</a></li>
                    <li><a href="#">Transações</a></li>
                    <li class="active"><a href="#">Informações
pessoais</a></li>
                </ul>
                <ul class="nav navbar-nav navbar-right">
                    <li>
                        <a href="login.php?logout=true" class="navbar-
left">Sair</a>
                    </li>
                </ul>
            </div>
        </nav>
        <!-- BALANCE SECTION -->
        <div class="row text-center">
            <div class="col-xs-12">
                <h3>Dados cadastrais de <?php echo
$_SESSION['name']?></h3>
            </div>
        </div>
        <!-- REGISTRATION FIELDS -->
        <div class="row section-margin">
            <div class="col-sm-4 col-sm-offset-4">
                <form name=PersonalInfo role="form">
                    <!-- NOME -->
                    <div class="form-group">
                        <label for="name">Nome</label>
                        <input type="text" name=name class="form-
control" id=name value="<?php echo $_SESSION['name']?>" readonly>
                    </div>
                    <!-- EMAIL -->
                    <div class="form-group">

```

```

        <label for="email">Endereço de Email</label>
        <input type="text" name=email class="form-
control" id=email value="<?php echo $_SESSION['email']?>" readonly>
    </div>
    <!-- VINCULO -->
    <div class="form-group">
        <label for="role">Vínculo com a UFPR</label>
        <input type="text" name=role class="form-
control" id=role value="<?php
            if      ($_SESSION['role'] == 0)      echo
'Estudante';
            else if ($_SESSION['role'] == 1)      echo
'Professor';
            else echo 'Servidor';
        ?>" readonly>
    </div>
    <!-- GRR -->
    <div class="form-group">
        <label for="grr">Número de matrícula
(GRR)</label>
        <input type="text" name=GRR class="form-
control" id=grr value="<?php echo $_SESSION['grr']?>" readonly>
    </div>
    <!-- NUM. CARTEIRINHA -->
    <div class="form-group">
        <label for="cardId">Número da
carteirinha</label>
        <input type="text" name=cardId class="form-
control" id=cardId value="<?php echo $_SESSION['cardId']?>" readonly>
    </div>
</form>
</div>
</div>
</div>
</body>
</html>

```