

Projeto UniCafé

Sistema de Gerenciamento de Computadores de Laboratório de Universidades.

1. INTRODUÇÃO

UniCafé é um sistema que integra alguns softwares e serve para controle de utilizadores de PCs em laboratórios de informática no contexto de uma universidade. Com este sistema será possível bloquear e liberar acesso para visitantes, bem como identificar e manter histórico de utilizadores com Ips usados durante a utilização. Utilizaremos a lista de usuários já cadastrados no sistema da universidade.

Uma das necessidades é possibilitar que seja definida uma cota padrão de horas por intervalo de tempo, aplicada quando o laboratório estiver lotado, permitindo assim que todos tenham acesso ao laboratório. O administrador poderá, por exemplo, definir que cada utilizador na categoria aluno terá uma hora, sendo renovada a cada seis horas.

Neste trabalho descrevo primeiramente os objetivos do programa, onde falarei de cada componente e cada função existentes nos componentes. A saber: UniCaféServer, UniCaféClient, UniCaféAdmin e UniCaféPHPAdmin. Em seguida, a metodologia, onde faço uma discussão sobre as ferramentas utilizadas na criação e estruturação do sistema. Divisão de Equipes e Implementação é uma sessão onde temos uma sugestão de como poderíamos integrar uma equipe e fazer o projeto caminhar. E por fim os resultados esperados, na conclusão.

2. OBJETIVOS

Um dos objetivos é fazer um sistema que possa ser criado por uma equipe e seja de fácil manutenção posterior e posterior adição de novas funcionalidades. Daí a necessidade de se modularizar o projeto.

O Sistema UniCafé se compõe de quatro aplicações: UniCaféServer, UniCaféClient, UniCaféAdmin e UniCaféPHPAdmin. A seguir, uma breve descrição de cada um, com suas funcionalidades e telas:

2.1. UniCaféServer

Este é o Servidor do UniCafé. Ele cria um serviço na rede para um laboratório de informática. Os clientes instalados buscam o serviço nele e é ele quem envia informação que bloqueia ou desbloqueia o cliente. Através dele também que as autenticações são feitas, o cliente busca nele e ele consulta o banco.

De cara, as primeiras funcionalidades disponíveis neste aplicativo são: Iniciar Serviço, Parar Serviço e Reiniciar Serviço. A maioria das suas funções não são acessíveis por meio dele, e sim pelo UniCaféAdmin e o UniCaféPHPAdmin que são as ferramentas dos administradores que acessam o UniCaféServer para solicitar alguma ação ou obter alguma informação.

2.2. UniCaféCliente

O UniCaféCliente é um programa instalado nas máquinas dos laboratórios. Ele será instalado como serviço no sistema operacional e será inicializado juntamente com o sistema. Ele está intimamente ligado à máquina que está instalado, ele verifica o nome da máquina e o seu endereço MAC. Através disso ele busca saber do banco de dados qual o seu servidor e tenta conexão com o mesmo. Sua tela inicial, após essa verificação e conexão é uma tela de login e senha, na qual o usuário não pode fechar, nem sair nem fazer nada além de fazer o login.

Caso o usuário faça seu login, as informações serão enviadas ao UniCaféServer, que fará autenticação no banco e depois enviará resposta para que o cliente seja desbloqueado por tempo determinado.

O cliente recebe apenas a informação de que deverá ser desbloqueado por um tempo e o nome da pessoa que está usando. Ele passa a contar regressivamente esse tempo, informando através de uma pequena janela. Nessa janela poderá também abrir um chat com o usuário que está usando o UniCaféAdmin, que é o software utilizado pelo administrador do laboratório, sendo capaz de responder no chat e atribuir mais tempo aos usuários que estão no UniCaféAdmin essa comunicação é feita por Socket, trataremos mais desse assunto na sessão metodologia.

Ao terminar o tempo de acesso, o UniCaféCliente fechará a janela, mandando também a informação ao UniCaféServer, que não está mais em uso.

2.3. UniCaféAdmin

No UniCaféAdmin, nós temos um software para administrador do laboratório, nele também há uma verificação de nome da máquina e endereço MAC, através do banco de dados sabe-se qual o IP do seu servidor, portanto sabe-se quais os clientes que ele tem jurisdição. Ele se conecta ao UniCaféServer e pode fazer com que este envie solicitações de bloqueio e desbloqueio a algum computador que tenha o UniCaféClient instalado e a este servidor seja associado. O desbloqueio pode ser feito com tempo determinado ou indeterminado, neste desbloqueio pode ser definido um usuário temporário ou sessão livre para alguma aula específica, essas informações serão salvas e poderão ser posteriormente recuperadas.

Através do UniCaféAdmin é possível listar usuários e redefinir tempo de acesso de cada um. A informação é guardada no banco de dados, que o UniCaféServer utiliza para definir o tempo de acesso quando o cliente pede o Login.

O UniCaféServer poderá ser iniciado sem login, clientes também, mas o UniCaféAdmin será usado por um usuário de nível especial. Daí necessidade de definição de níveis de usuários.

2.4. UniCaféPHPAdmin

O UniCaféPHPAdmin é um software escrito em PHP em um servidor web. Ele é o nível de cúpula, o institucional. Ele possui o controle do administrador dos laboratórios, mas também possui informações gerais, relatórios que serão acessados pelos alunos. Antes de mais nada, temos que ter uma autenticação, dependendo do tipo de usuário, poderá levar a uma tela de administrador master ou de relatórios de consulta. Ele não atuará conectado diretamente ao UniCaféServer, nem a nenhum dos outros sistemas, ele estará monitorando todos através do banco de dados e também terá informações enviadas automaticamente pelo UniCaféServer, como fotos de captura de tela.

Todas as informações de Endereço MAC, usuário que acessou, hora de acesso, de todos os laboratórios poderão ser acessadas na área de administrador master. Algumas informações serão acessíveis por alunos, como relatórios de como estão os laboratórios em tempo real. Poderão saber o status de cada cliente conectado a um UniCaféServer. Cada UniCaféServer corresponderá a um laboratório.

É através deste componente que serão cadastrados os servidores, clientes e as modificações de nível de usuário. A característica peculiar deste software é que tem uma visão global. Enquanto o UniCaféAdmin só administrava o laboratório, este administra todos os laboratórios.

3. METODOLOGIA

3.1. Conexão com Socket no Java usando protocolo UDP

O UniCaféClient fará uma conexão com o UniCaféServer usando DatagramSocket, uma classe de Java que faz conexões de socket com vários clientes usando protocolo UDP e permite a transferência de dados de qualquer tipo do cliente para o servidor e do servidor para o cliente específico. Da mesma forma também se comunicará o UniCaféAdmin com o UniCaféServer para enviar as solicitações. Muita coisa pode ser passada, via socket. Pretendemos utilizá-lo para passar Strings ou capturas de tela.

3.2. Servidor Apache com PHP

O UniCaféPHPAdmin será feito em PHP, por ser uma ferramenta livre e fácil, além de ter suporte a orientação a objetos e muitas classes e padrões disponíveis para serem utilizados. Através do PHP acesaremos o banco de dados para ver o funcionamento dos servidores e clientes. Acesso de histórico e lista de clientes.

3.3. Padrões de Projeto - MVC, Padrão Observer e DAO

Alguns padrões foram desenvolvidos e resolvem muitos problemas com os quais nos deparamos. Por exemplo, nós precisamos separar o trabalho do programador front-end e do back-end. Usando MVC podemos fazer isso, separamos a aplicação em três camadas, onde em uma temos o Model, que são classes que seguem as regras de negócio do nosso software, a outra é o View que contém as telas e visões, temos também o Control, que é a camada responsável por fazer com que as telas apareçam e sejam trocadas, bem como atribuir informações para essas telas, como informações do banco que serão passadas por outras classes. Este padrão poderá ser aplicado em todos os componentes.

O Padrão Observer é outro padrão que pode ser utilizado. Segundo FREEMAN, (p. 36. 2009) “Padrão Observer define a dependência um-para-muitos entre objetos pra que quando um objeto mude de estado todos os seus dependentes sejam avisados e atualizados automaticamente.” O padrão tem um sujeito e vários observadores que estarão ligados a esse sujeito. No nosso caso de laboratório, podemos definir classes que representarão o servidor e os clientes, sendo estes os observadores e o servidor o sujeito que será observado.

Através do DAO, podemos ter um pacote que se preocupará exclusivamente com transações com o banco e retornará listas de objetos ou fará cadastros. Criamos uma classe para retornar conexão com banco e classes específicas de envio e acesso ao banco relacionado com entidades específicas, por exemplo. UsuarioDAO é a classe que acessa a lista de usuários do banco de dados, retornando lista de objetos ou faz cadastro de usuários.

3.4. Banco de dados LDAP e PostGreSql

O sistema terá acesso à lista de usuários no LDAP, mas também terá seu banco de dados específicos para suas operações em um banco PostGreSql, como por exemplo acesso à lista de servidores, cadastro de fotos dos usuários, etc. A seguir detalharemos mais a estrutura do sistema e do banco de dados.

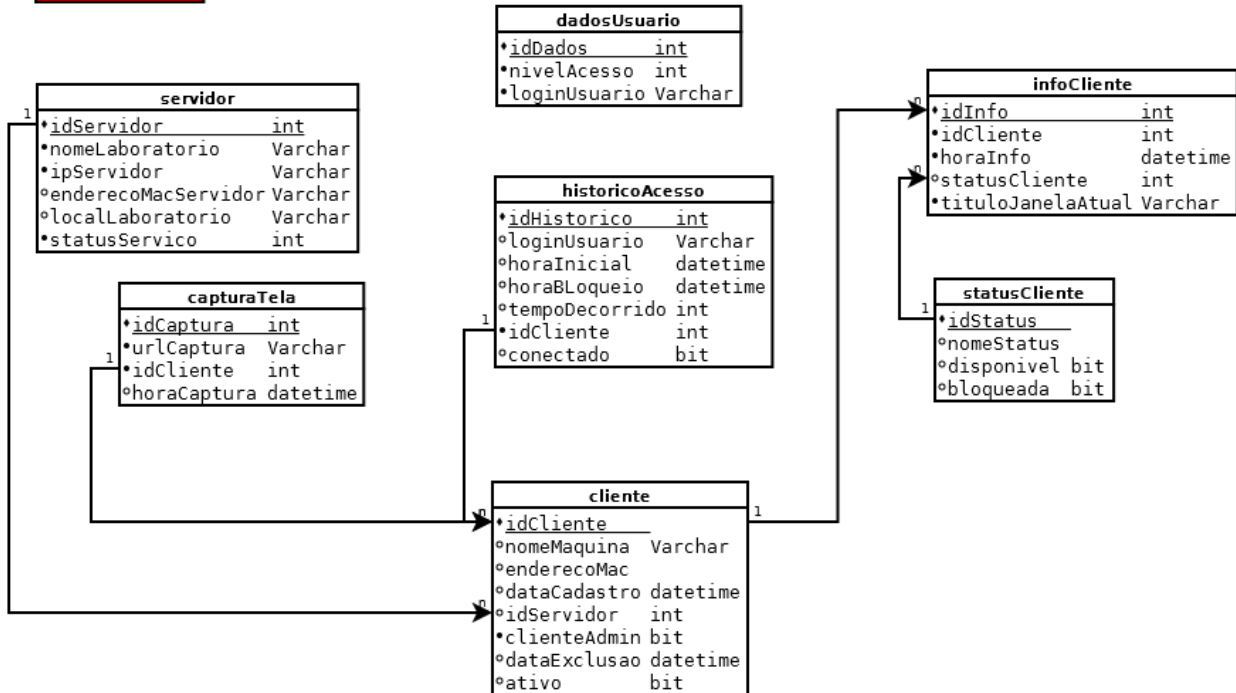
4. ESTRUTURA DO SISTEMA

4.1. Modelagem de banco de dados

Tabela existente no LDAP

usuario	
•id	int
•login	Varchar
•senha	Varchar

Banco específico no PostgreSQL



4.2. Modelagem de classes

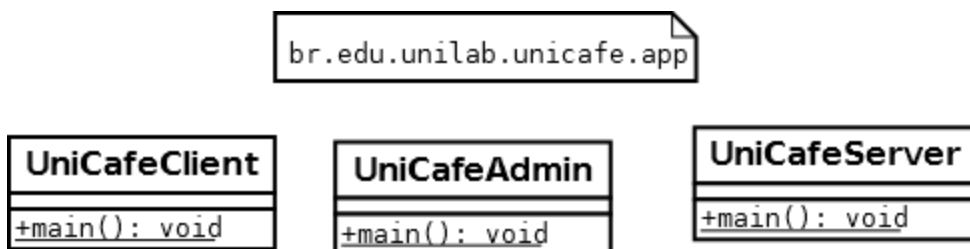
Devido ao fato de que uma das aplicações é em Java e a outra é em PHP, temos bibliotecas padrões diferentes, logo foi necessário fazer uma modelagem de classes para as aplicações UniCaféServer, UniCaféClient e UniCaféAdmin. Para a aplicação UniCaféPHPAdmin é outra modelagem. A seguir cada uma das modelagens com explicação geral das funcionalidades e forma de interação entre objetos.

4.2.1. Modelagem UniCaféServer, UniCaféClient e UniCaféAdmin

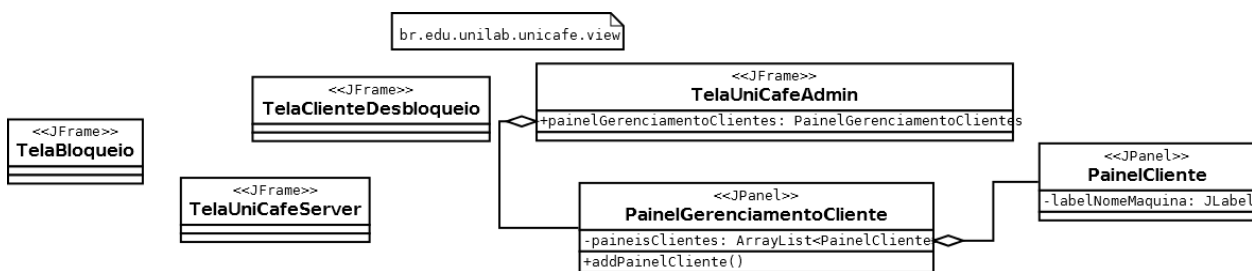
4.2.1.1. Aplicação

A primeira camada, é a camada app. Ela possui uma classe para cada aplicação, e cada uma organiza os objetos a serem inicializados na aplicação, no caso do

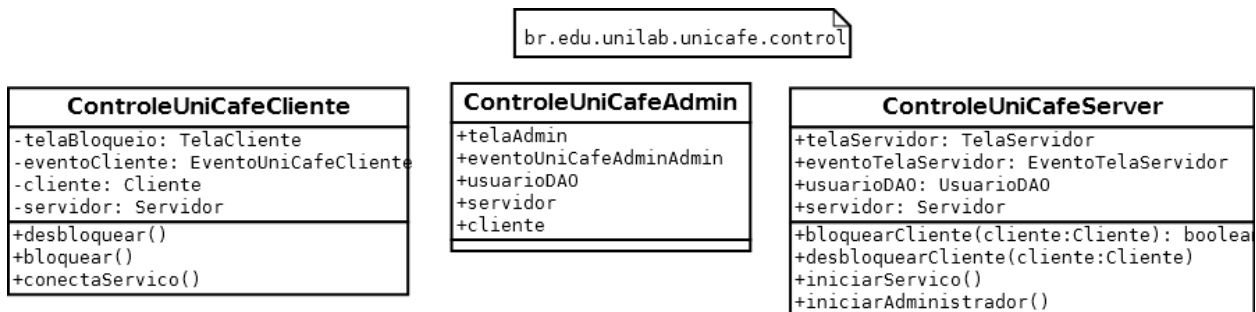
UniCaféServer, ele inicia o ControleUniCaféServer, que por sua vez vai chamar as do pacote view. Isso será explicado mais adiante.



4.2.1.2. View



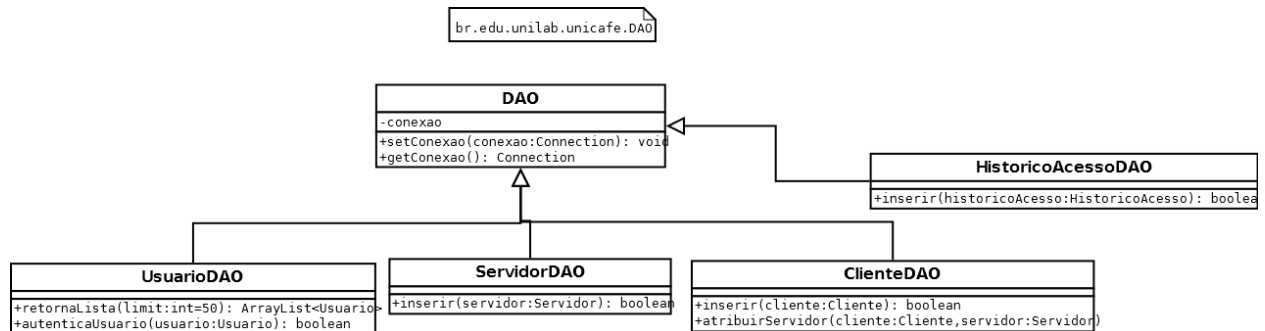
4.2.1.3. Control



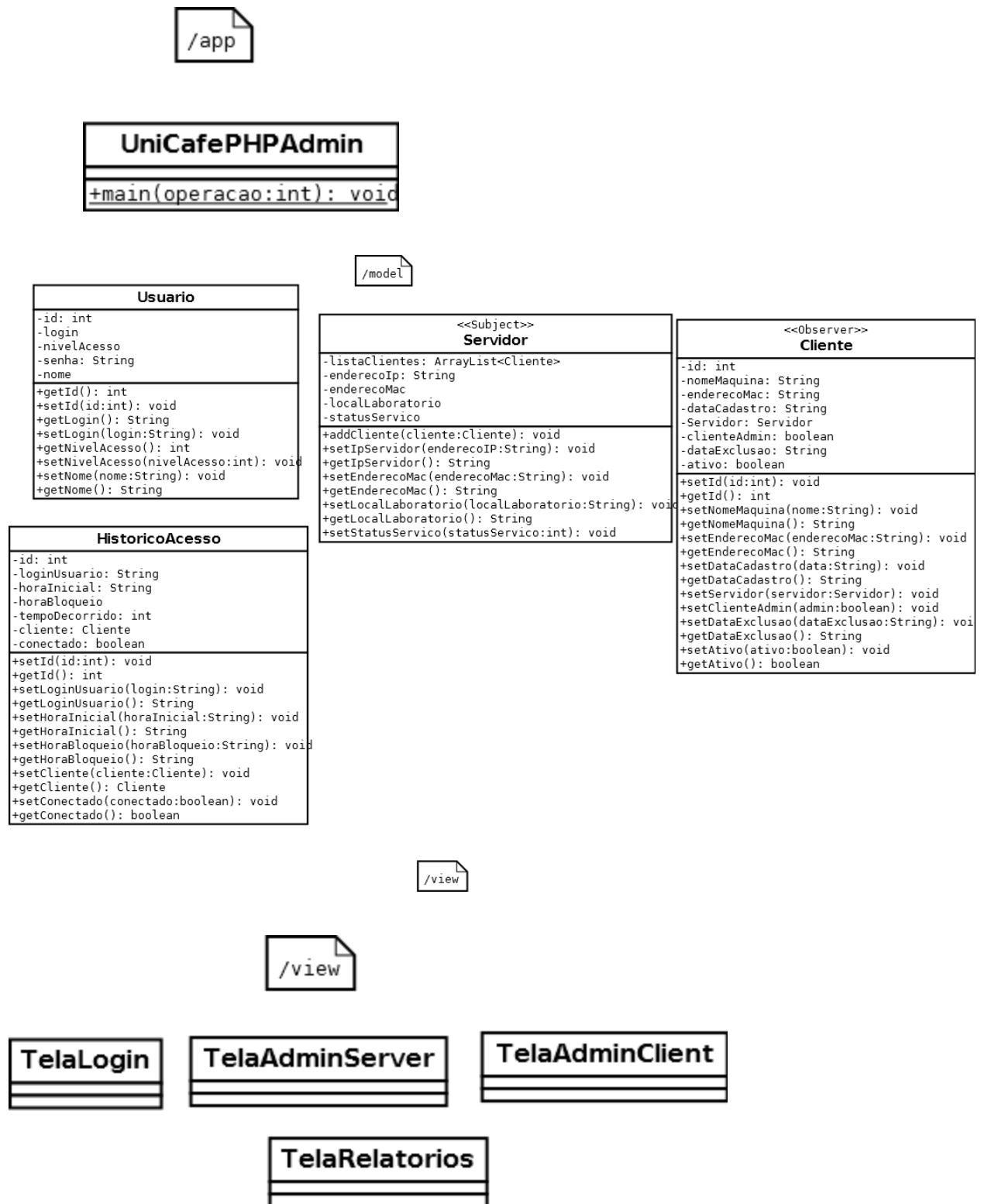
4.2.1.4. Model

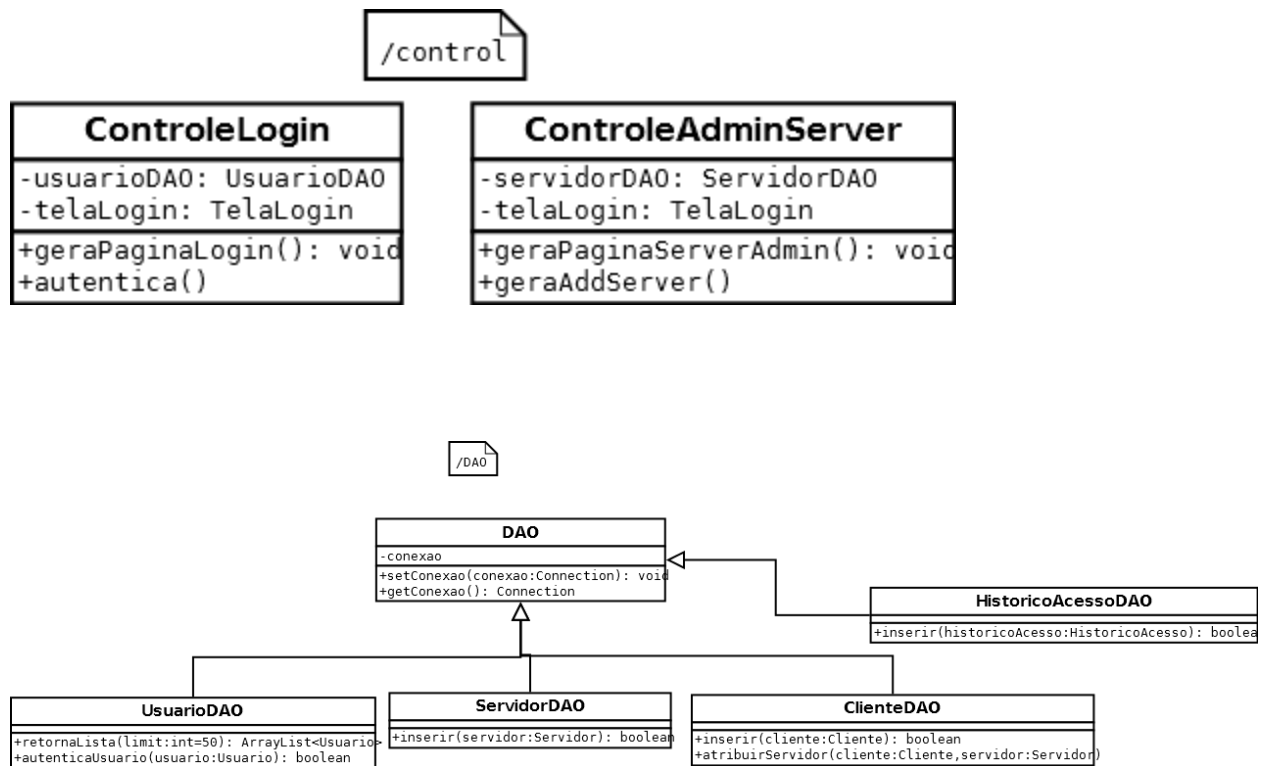


4.2.1.5. DAO



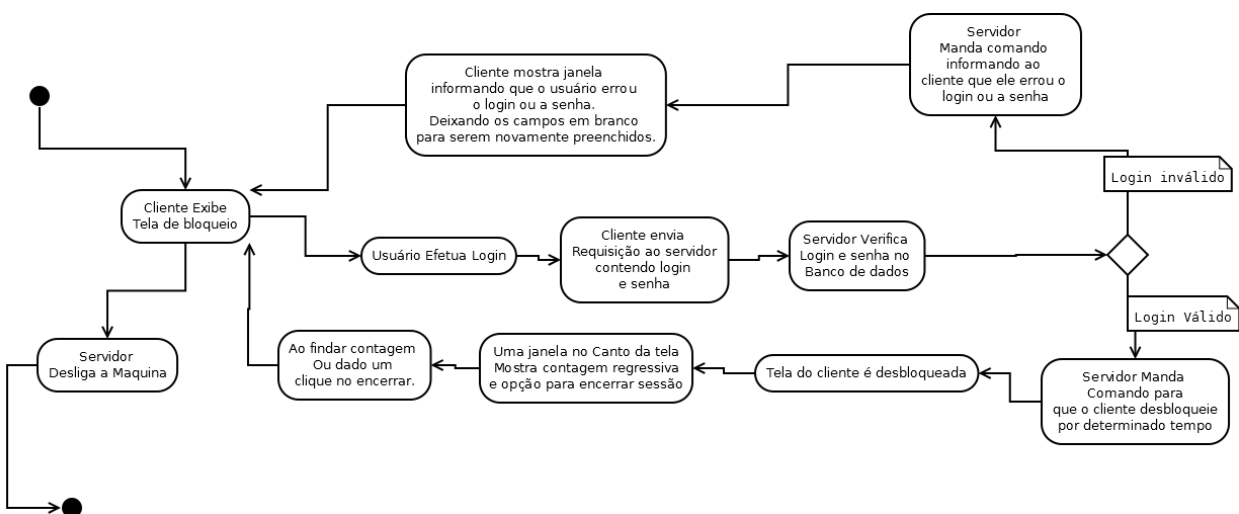
4.2.2. Modelagem UniCafePHPAdmin





4.3. Diagramas de Caso de Uso

4.3.1. Perspectiva do usuário comum, no dia-a-dia do laboratório.



5. DIVISÃO DE EQUIPES E IMPLEMENTAÇÃO

As modelagens ainda estão imperfeitas, é necessário discutir com a equipe sobre modificações nas mesmas. Sobre divisão de equipe, podemos dividir o trabalho em tarefas, tipo: fazer validação de usuário no LDAP, criar as classes do modelo, desenhar telas ou programar as telas.

Podemos admitir uma pessoa que fique administrando e juntando os componentes feitos, e disponibilize os arquivos usando o servidor SVN. Outra forma bem útil de agilizar os processos e ter transparência em relação ao que cada um está fazendo ou fez é utilizar o processo de desenvolvimento SCRUM.

6. CONCLUSÃO

Com este projeto é possível iniciar o desenvolvimento com um rumo. Ele está incompleto, pois carece de mais discussão em relação às suas funcionalidades e sua modelagem de classes.

Utilizando uma abordagem orientada a objetos e com padrão de projetos, podemos ter um projeto grande que pode ser trabalhado em equipe e de fácil manutenção posterior. Também visamos um sistema modularizado e reutilizável.

7. BIBLIOGRAFIA

FREEMAN, Eric; FREEMAN, Elisabeth. Use a Cabeça! Padrões de Projetos(Design Patterns). O'REILLY

<http://www.vaniomeurer.com.br/2009/02/19/como-usar-a-classe-robot/>

Anexo 1 - Tabela com componentes da aplicação e lista de funcionalidades com descrição e método de realização da função.

UniCaféClient

Funcionalidade	Descrição	Método de realização
Bloqueio de Tela	<p>É a primeira coisa que o programa faz. Ele cria uma tela cheia no PC e impede que o usuário feche.</p> <p>Esta tela será padrão para quando o usuário não estiver logado.</p>	Utilizando registro do Windows iremos bloquear utilização de teclas especiais do teclado.
Autenticação	<p>É o procedimento de validação do usuário através de formulário de login e senha. Este é o procedimento que envia informações ao servidor de quem está acessando.</p>	Através da conexão via socket já estabelecida, o cliente envia a informação de login para autenticação no servidor UniCaféServer, este compara com os dados no Banco de dados, decide quanto tempo o UniCaféClient ficará desbloqueado e envia informação do quanto tempo o cliente estará desbloqueado.
Desbloqueio de tela	<p>É a funcionalidade que o programa tem, capaz de desbloquear a tela e permitir que o usuário utilize alguns recursos do sistema, necessários ao laboratório, como navegador, editor de texto, planilhas, etc.</p> <p>Neste momento de desbloqueio é mostrada uma janela pequena com contagem de tempo e algumas funcionalidades que poderão ser criadas depois, chat por exemplo.</p>	Ele será feito através do recebimento de informação do UniCaféServer, que envia comando de desbloqueio pela conexão de socket, juntamente com o comando envia o tempo de duração desse desbloqueio.
Envio de informações ao UniCaféServer	O UniCaféServer poderá ter acesso a informações relativas ao acesso do computador, para isso o UniCaféClient envia informações do que está sendo feito, informações acerca de	O programa UniCaféClient, rodando em background pegará informações da máquina e da utilização através de comandos ao sistema operacional e enviará essas informações via

	programas em execução e títulos na barra de título do programa atual.	socket para o UniCaféServer, que ficará sempre atualizado com esses dados. Essas ações podem ser feitas pela classe Robot, que é padrão do java e pode fazer capturas de telas e muitas outras coisas.
--	---	--

UniCaféServer

Iniciar serviço	É ação de iniciar o servidor, que vai esperar a conexão dos clientes.	A inicialização do serviço abrirá uma conexão com Socket usando DatagramSocket do java e um laço infinito para que fique sempre executando a ação de esperar um pacote para processar.
Parar serviço	Faz com que o serviço seja encerrado, fechando todas as conexões se houverem.	
Reinicia serviço	É o comando de parar o serviço, seguido de iniciar serviço.	

UniCaféAdmin

Autenticação	Antes de qualquer coisa, para usar o UniCaféAdmin é necessário autenticar. Essa autenticação.	Autenticação consulta o banco de dados.
Visualização de status de clientes	Essa função é a visualização do status de cada um dos clientes conectados ao servidor.	A informação vem do cliente até o servidor, este as envia ao UniCaféAdmin. Toda essa comunicação é via conexão da classe DatagramSocket.
Liberação de cliente	Essa função permite a liberação de qualquer dos	Solicitação feita via socket ao UniCaféServer e este

	clientes conectados.	envia a solicitação de bloqueio ao cliente. Tal solicitação pode ter como parâmetro o tempo indeterminado ou algum tempo para ser contado pelo próprio cliente.
Bloqueio do cliente.	Essa manda o comando para que o cliente seja bloqueado.	Via socket o comando será enviado para o servidor, este envia para o cliente.
Captura de tela de cliente	O UniCaféAdmin poderá ter a captura de tela do cliente no momento desejado.	UniCaféAdmin faz a solicitação ao servidor, indicando o cliente que quer tirar o print, servidor envia a solicitação ao cliente, este envia uma captura de tela usando a classe Robot, obviamente que localmente executada e envia os bytes da imagem por socket para o servidor, este envia para o UniCaféAdmin, que renderizará esta imagem em um arquivo, usando ImageIO.

UniCaféPHPAdmin

Autenticação	Essencial, nela é feita a identificação do usuário que está acessando.	Feita via formulário HTML e consulta ao banco de dados LDAP. Dependendo do nível do usuário, a pessoa poderá ser enviada para uma tela de administrador ou para uma tela de visualização serviços para alunos.
Listagem e edição de servidores e clientes.	Será possível saber quais os servidores cadastrados, quais os clientes cadastrados nos servidores. Podendo adicionar	Página em HTML com acesso a banco de dados.

	novos servidores e clientes. (Admin)	
Visualização de informações de clientes e servidores em tempo real.	Nesta área poderá ser visualizado o comportamento do laboratório, status de cada computador, se em funcionamento ou não, se ocupado ou desocupado. Os clientes estarão agrupados por laboratório. E também será possível ver informações do servidor.	Essas informações serão encontradas no banco de dados e serão atualizadas pelo UnifCaféServer, o UniCaféPHPAdmin pesquisa no banco e terá essas informações.
Visualização de captura de tela	No servidor Web será possível pegar uma captura de tela de qualquer um dos clientes. (Admin)	O UniCaféServer estará eventualmente enviando capturas de telas dos clietnes via HTTP para o servidor Web. Esta foto será enviada com uma marca com data, hora e informações de acesso.
Relatórios	Página com relatórios, com pico de acessos, quantidade de usuários atendidos por semana, mês e ano. Outros relatórios poderão ser criados e implementados.	Informações guardadas no banco de dados. Para criar os gráficos do lado do servidor pode-se usar JPGraph e do lado do cliente usando biblioteca de javaScript, Higcharts.