

Controle de Acesso do Restaurante Universitário

# Manual do Usuário

# Índice

## Informações Iniciais

- 3      Pré-Requisitos
- 4      Downloads

## Servidor

- 5      Inicializando o servidor
- 6      Aplicativo Desktop

## Aplicativo Smartphone

- 7      Login no Smartphone
- 8      Atualizar informações
- 11     Entrar no Restaurante Universitário

## Apêndice

- 12     Guia de Uso

# Pré-Requisitos

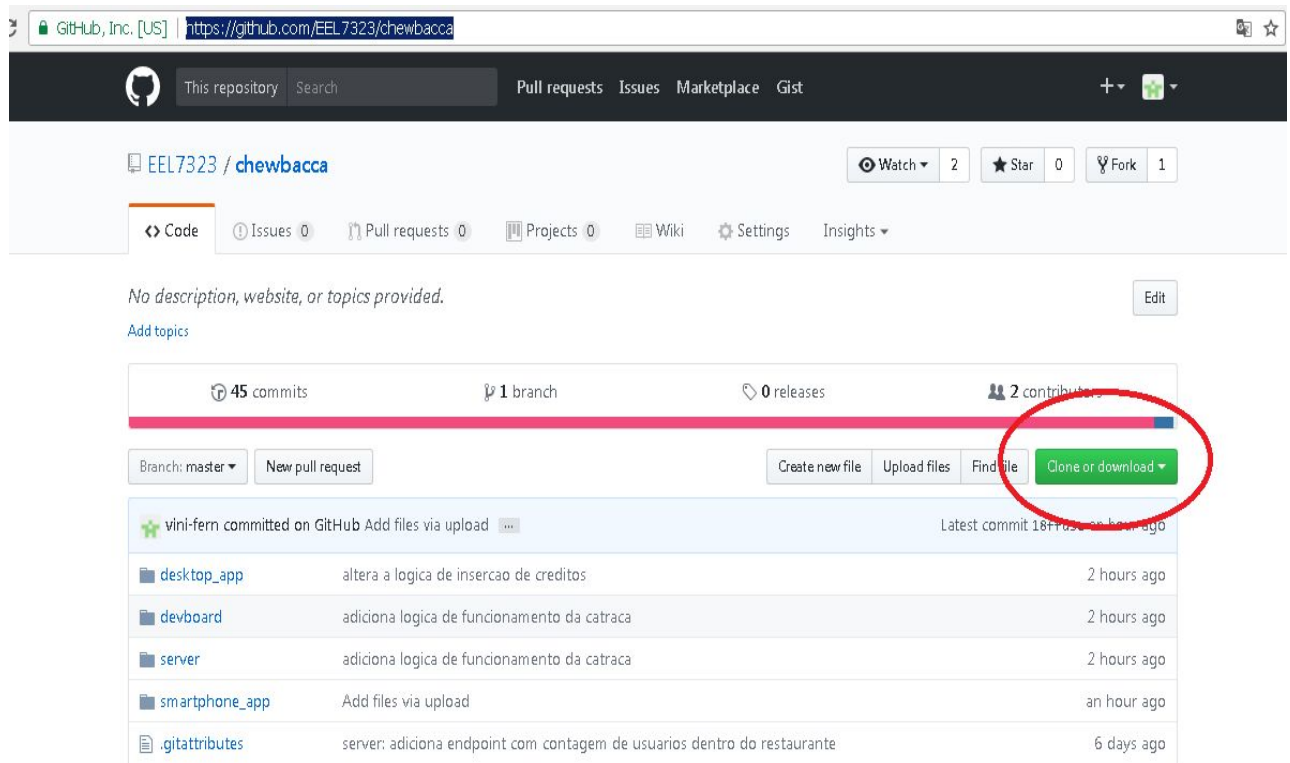
Para utilizar o Controle de Acesso do Restaurante Universitário, é necessário possuir um computador com sistema operacional Linux instalado para a execução do Servidor.

Para utilizar o Aplicativo Smartphone, é necessário que o smartphone em questão possua o sistema operacional Android instalado.

Para o controle de acesso local, é necessário fazer a aquisição da placa [modelo].

# Downloads

Para fazer o download, acessar o seguinte endereço:  
<https://github.com/EEL7323/chewbacca> , clicar no botão “Clone or download”



Extraia o arquivo .ZIP no diretório desejado

# Inicializando o Servidor

Para inicializar o servidor basta abrir um terminal na máquina Linux, acessar o diretório "server" e executar os seguintes comandos:

```
./install.sh
```

```
./run.py --host=0.0.0.0
```

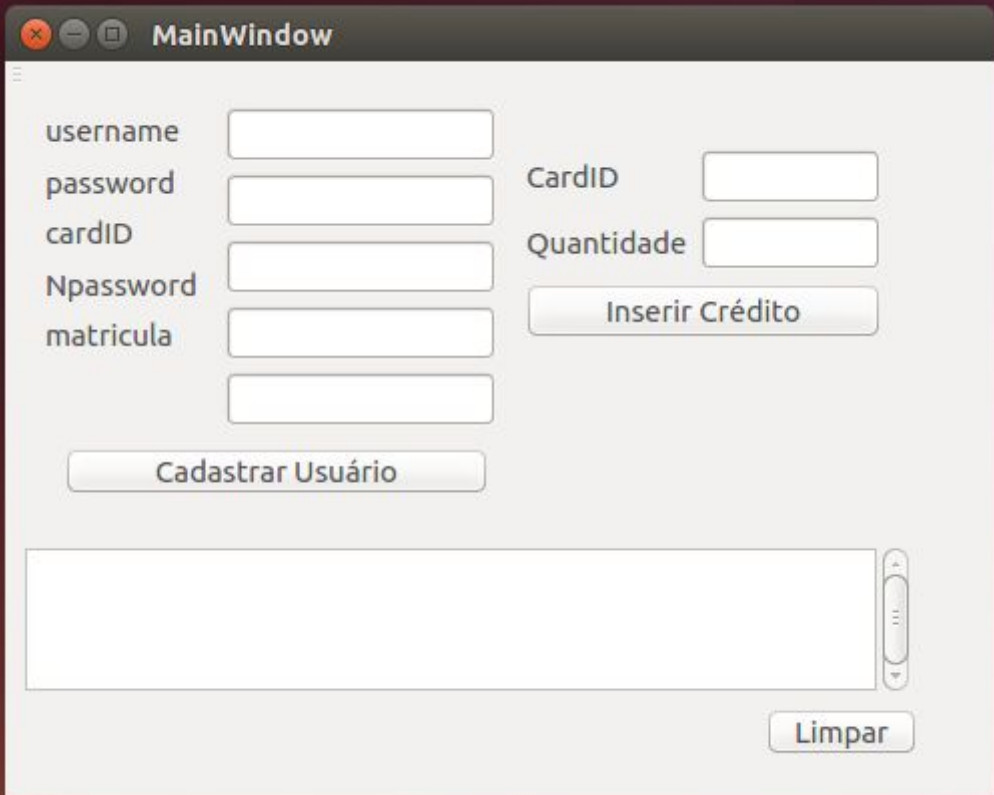
Após a execução dos comandos, o servidor estará inicializado e o terminal exibirá:



```
v@v-VirtualBox:~/Documents/chewbacca-master/server$ ./run.py --host=0.0.0.0
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 212-696-157
```

# Aplicativo Desktop

Acessando o diretório desktop\_app e executando o arquivo desktop\_app.exe será mostrado ao operador a seguinte janela:

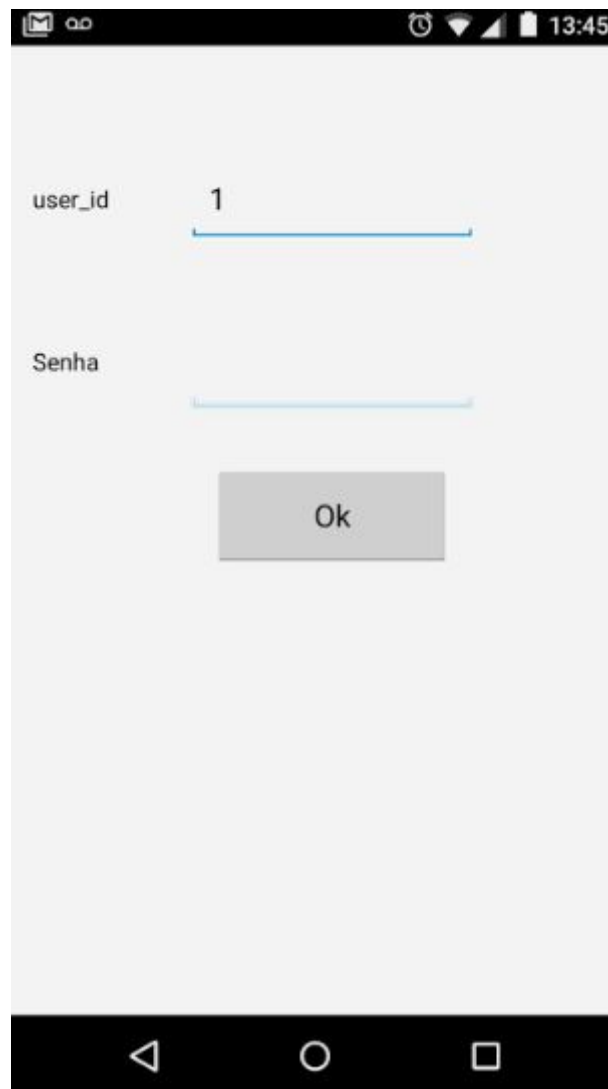


The screenshot shows a desktop application window titled "MainWindow". The interface is divided into two main sections. On the left, there are five text input fields labeled "username", "password", "cardID", "Npassword", and "matricula", with an additional empty field below "matricula". Below these fields is a button labeled "Cadastrar Usuário". On the right, there are two text input fields labeled "CardID" and "Quantidade", followed by a button labeled "Inserir Crédito". At the bottom of the window, there is a large text area with a vertical scrollbar and a button labeled "Limpar" to its right.

Onde os campos a esquerda: “username”, “password”, “cardId”, “Npassword” e “matricula” são onde as informações do usuário devem ser digitadas para cadastro, e os campos a direita: “CardID” e “Quantidade” são onde o operador deve digitar o número do cartão que vai receber os créditos e a quantidade de créditos a ser inserida, respectivamente.

# Login no Smartphone

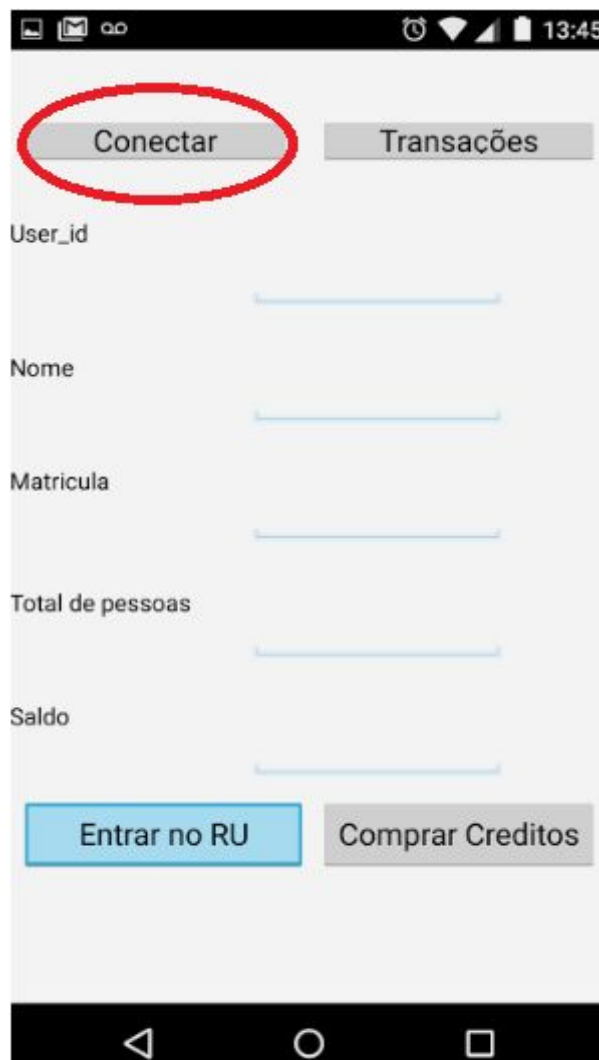
Ao iniciar o aplicativo do Smartphone a seguinte tela aparecerá:

A screenshot of a smartphone screen displaying a login form. The screen has a light gray background. At the top, there is a black status bar with icons for mail, a signal strength indicator, a battery level indicator, and the time 13:45. Below the status bar, the form consists of two text input fields. The first field is labeled 'user\_id' and contains the number '1'. The second field is labeled 'Senha' (Password) and is empty. Below these fields is a gray rectangular button with the text 'Ok' in black. At the bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.

Para realizar o login, o usuario deve digitar o user\_id (numero cadastrado pelo desktop\_app).

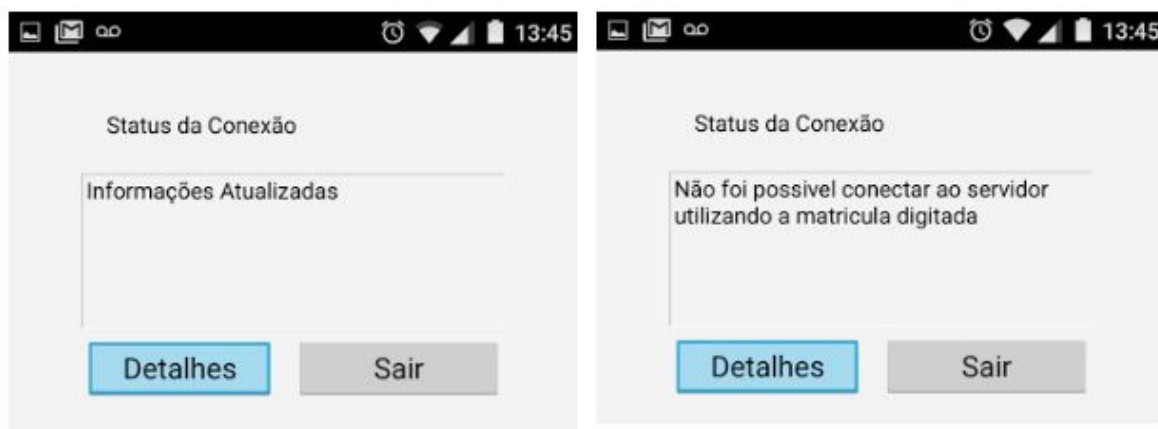
# Atualizar Informações

Após o login, será mostrada a tela principal, onde são mostradas informações como matrícula, nome, saldo, pessoas no RU e user\_id depois de estabelecer a conexão com o servidor. Para conectar o smartphone ao servidor, deve-se apertar o botão “Conectar”

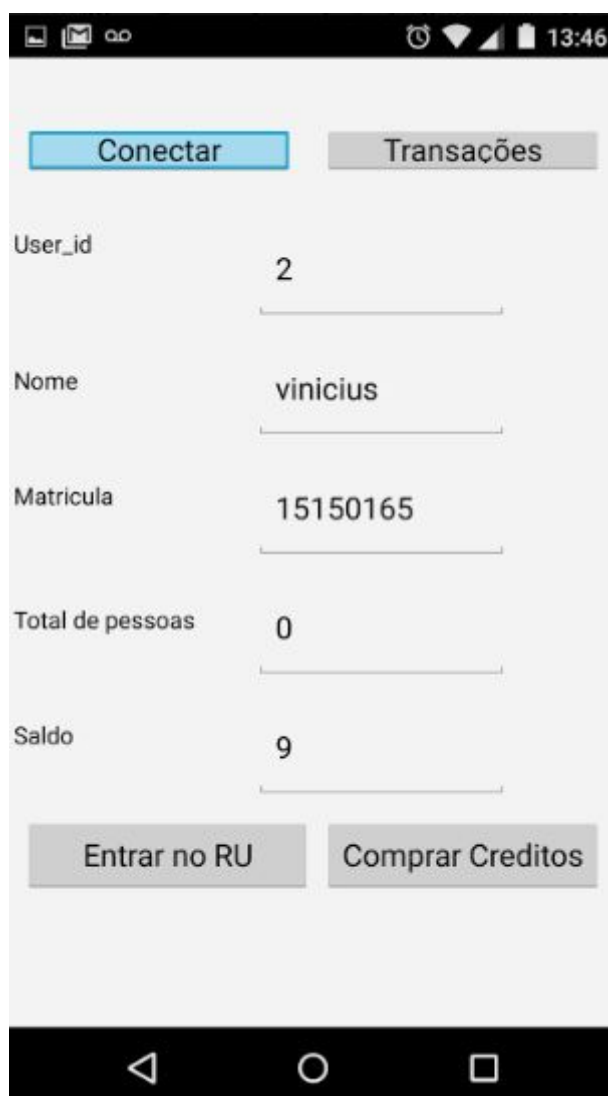


Após pressionar o botão “Conectar” será exibido ao usuário o status da conexão, informando se a conexão foi estabelecida corretamente.





Conectado ao servidor, as informações do usuário são atualizadas na tela principal.



Pressionando o botão “Transações”, será exibido na tela o histórico de todas as transações feitas pelo usuário (utilização e compra de créditos com suas respectivas data e hora).



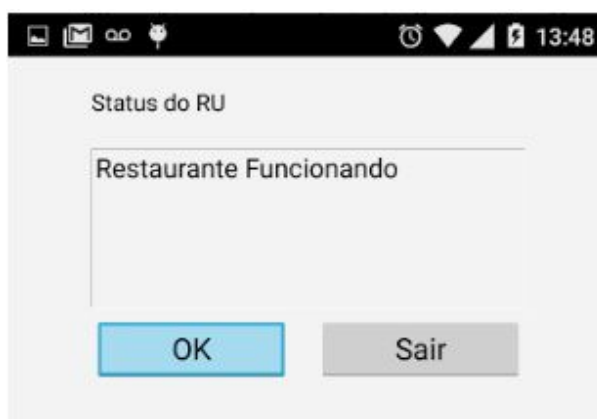
# Entrar no Restaurante Universitário

Para ter o acesso liberado, basta ter um saldo positivo e pressionar o botão “Entrar no RU” dentro do horário de funcionamento do estabelecimento.

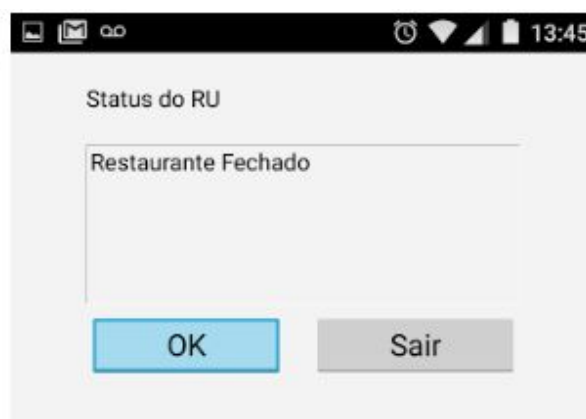


A screenshot of a mobile application interface. At the top, there are two buttons: "Conectar" (highlighted in blue) and "Transações" (grey). Below these are five input fields with labels on the left and values on the right: "User\_id" with value "2", "Nome" with value "vinicius", "Matricula" with value "15150165", "Total de pessoas" with value "0", and "Saldo" with value "9". At the bottom, there are two buttons: "Entrar no RU" (highlighted with a red circle) and "Comprar Creditos" (grey). The Android navigation bar is visible at the very bottom.

Dependendo do horário, será exibido ao usuário:



A screenshot of a dialog box titled "Status do RU". It contains a text field with the text "Restaurante Funcionando". At the bottom, there are two buttons: "OK" (highlighted in blue) and "Sair" (grey). The status bar at the top shows the time as 13:48.



A screenshot of a dialog box titled "Status do RU". It contains a text field with the text "Restaurante Fechado". At the bottom, there are two buttons: "OK" (highlighted in blue) and "Sair" (grey). The status bar at the top shows the time as 13:45.

# Guia de Uso

servidor:

```
cd server/  
install.sh
```

Este arquivo instalará os pacotes básicos e criará um ambiente virtual do python de onde rodará o servidor. No ambiente virtual os pacotes necessários do projeto serão instalados e a inicialização e criação do servidor serão feitas.

A inicialização colocará algumas informações no banco de dados para demonstrações.

```
./run.py --host=host --port=port
```

Iniciará o servidor.

host=localhost (default) para rodar na mesma máquina  
host=0.0.0.0 para responder através do IP da máquina  
port=5000 (default)

catraca - placa:

```
libcurl
```

Dependência externa necessária em /usr/lib/x86\_64-linux-gnu normalmente disponível via libcurl4-openssl-dev.

```
cd devboard/  
make
```

Compila os códigos fonte do projeto

```
./placa
```

Com o servidor acima rodando, inicializará o funcionamento da catraca.

Funcionamento

Inserir os dados como endereço do servidor e porta.

Espera um comando do tipo

(e|s)(c|s)(dados)

onde:

primeiro campo: e - entrada

s - saída

segunda campo : c - cartão

s - senha

terceiro campo: dados referentes as opções anteriores

cartão formato: abcd0000

senha formato: matricula+senha -> 151501620000

Exemplo:

ecabcd0000 - entrada com cartão, e identificador do cartao "abcd0000"

ss151501620000 - saída com a senha, onde a matricula é 15150162 e a senha é 0000

Desktop App

Fazer o build do projeto no qtCreator (não houve tempo de criar o executável final)

No lado esquerdo é possível cadastrar usuários. Importante memorizar ou anotar os dados inseridos para usá-los na placa em seguida. **IMPORTANTE:** cadastrar o cartão no formato ABCD01234 com as letras em maiúsculo para consistência do projeto, pois não foi possível corrigir todas as situações.

No lado direito insere-se créditos através da informação do cartão e quantidade de crédito (passe), e não por valor.

O app pode funcionar ao mesmo tempo que a placa e deve estar conectado ao servidor (servidor deve estar rodando na mesma máquina que o app). Pessoas recém cadastradas ou com crédito zerado que inseriram créditos após a inicialização da placa, devem inserir seus dados para acesso ao restaurante duas vezes. A primeira os busca no servidor e a segunda efetua a entrada conforme mensagens aparecem na tela.

TODO

Fazer com que a placa atualize o servidor quando a conexão cai  
tornar o desktop-app um executável e deixa-lo mais configurável

Tratar mais casos de inconsistência na entrada dos dados  
Alterar a forma de entrada de dados para acesso  
Tornar o servidor e o banco de dados mais amigável, onde seja possível ver as informações armazenadas principalmente

Projetos externos utilizados

Json parser - <https://github.com/nlohmann/json>

C++ RestClient - <https://github.com/mrtazz/restclient-cpp>

RFC1321 (MD5) - <http://www.bzflag.org>