



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS

TERMO DE APROVAÇÃO

ANDREW CARVALHO BARBOSA FERNANDES

ERICK DOUGLAS SALUSTIANO

TIAGO LUPEPIC

**FREE PASS – Sistema de Controle de Acesso
utilizando cartões de proximidade RFID.**

Trabalho apresentado como requisito
Parcial para obtenção do Título de
Tecnólogo em Análise e
Desenvolvimento de Sistemas da
Universidade Federal do Paraná

Curitiba, 30 de Junho de 2017.

BANCA EXAMINADORA

Professora Andreia de Jesus

Professor Alessandro Brawerman

Professor Rafael Romulado Wandresen

UNIVERSIDADE FEDERAL DO PARANÁ
ANDREW CARVALHO BARBOSA FERNANDES
ERICK DOUGLAS SALUSTIANO
TIAGO LUPEPIC

***FREE PASS - SISTEMA DE CONTROLE DE ACESSO UTILIZANDO CARTÕES DE
PROXIMIDADE RFID***

CURITIBA

2017

ANDREW CARVALHO BARBOSA FERNANDES

ERICK DOUGLAS SALUSTIANO

TIAGO LUPEPIC

***FREE PASS - SISTEMA DE CONTROLE DE ACESSO UTILIZANDO CARTÕES DE
PROXIMIDADE RFID***

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de educação profissional e Tecnológica, da Universidade Federal do Paraná.

Orientadora: Msc. Prof^a Andreia de Jesus.

CURITIBA

2017

AGRADECIMENTOS

Quero agradecer, em primeiro lugar, a Deus, pela força e coragem durante toda esta longa caminhada.

Ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná, na pessoa de seu coordenador Prof. Luiz Antônio Pereira Neves, pelo apoio e oportunidade recebida.

À nossa orientadora, Profa. Andreia de Jesus, pelo auxílio, orientação e paciência.

À Marilza Ribeiro de Carvalho e Tailayne de Souza Demetrio, pela ajuda nos momentos de dificuldade e aflição.

À empresa Go Receba fácil por atender a equipe e participação com o projeto. O agradecimento ao próprio grupo do projeto que, estavam sempre presentes durante todo o processo de elaboração deste trabalho, nos bons e maus momentos.

RESUMO

Com o passar dos anos e desenvolvimento da tecnologia, os sistemas de controle de acesso vem sendo cada vez mais utilizados quando o assunto é segurança. Desses, os sistemas baseados em microcontroladores tem grande uso em locais como: escritórios, bancos, órgãos governamentais, entre outros. Logo, considerando a demanda da sociedade por mais segurança em locais públicos e privados, o presente trabalho tem como objetivo desenvolver um sistema de Acesso Microcontrolado por aproximação de código RFID (Radio-Frequency IDentification) para permitir e negar acessos à ambientes físicos. O principal intuito é propor uma solução para o problema de manter pessoas não autorizadas longe de determinado local. Para tanto, implementou-se uma alternativa simplificada de controle de acesso com um programa funcionando em sistema embarcado, além de uma aplicação web que oferece um recurso de gerenciamento completo das informações para o perfil de administrador. Nossa proposta incide sobre a criação de um sistema denominado FreePass, onde suas validações de controle de acesso são feitas através da aproximação de código RFID, projetado para fazer isso de forma rápida, descomplicada sem deixar de se atentar a segurança, permitindo inclusive que as validações sejam feitas tanto *on* quanto *off-line*.

Palavras-Chave: *RFID; Controle de Acesso; Sistemas Embarcados; Segurança.*

ABSTRACT

Over the years and development of technology, access control systems have been increasingly used when it comes to security. Of these, the systems based on microcontrollers have great use in places such as: offices, banks, government agencies, among others. Therefore, considering the society's demand for more security in public and private places, the present work aims to develop a system of Microcontrolled Access by RFID (Radio Frequency IDentification) code to allow and deny access to physical environments. The main purpose is to propose a solution to the problem of keeping unauthorized persons away from a particular location. For this, a simplified alternative of access control was implemented with a program running on embedded system, besides a web application that offers a complete information management feature for the administrator profile. Our proposal focuses on the creation of a system called FreePass, where its access control validations are done by approaching RFID code, designed to do this in a fast, uncomplicated way without neglecting security, allowing even that the validations Both offline and offline.

Keywords: RFID; Access control; Embedded systems; Safety.

LISTA DE FIGURAS

FIGURA 1: ARQUITETURA BÁSICA DE UM SISTEMA EMBARCADO	19
FIGURA 2: PERIFÉRICOS UTILIZADOS POR SISTEMAS EMBARCADOS PARA A COMUNICAÇÃO COM O AMBIENTE EXTERNO.	21
FIGURA 3: EXEMPLOS DE TAGS E LEITORES DE TAGS	23
FIGURA 4: FUNCIONAMENTO DO MODELO CASCATA.	25
FIGURA 5: SISTEMA PRIME ACESSO SF	26
FIGURA 6: SISTEMA IDACCESS PROX	27
FIGURA 7: SISTEMA EPC/RFID DA MEMOVE	28
FIGURA 8: FLUXOGRAMA DA METODOLOGIA DE TRABALHO DO PROJETO.....	33
FIGURA 9: INTEGRAÇÃO DAS CAMADAS DO SISTEMA.	34
FIGURA 10: MODELO CONCEITUAL DOS DADOS.	41
FIGURA 11: DIAGRAMA DE CASO DE USO	43
FIGURA 12: DIAGRAMA DE CLASSE DE NEGÓCIO	44
FIGURA 13: PLACA DE DESENVOLVIMENTO RASPBERRY PI MODELO V3.....	50
FIGURA 14: MFRC522 COM ESQUEMA DE CONEXÃO COM RASPBERRY PI MODEL V3.....	52
FIGURA 15: DISPLAY TOUCHSCREEN HDMI 5" GPIO FABRICADO PELA 52PI	53
FIGURA 16: DISPLAY TOUCHSCREEN HDMI 5" GPIO ACOPLADA AO RASPBERRY PI.....	54
FIGURA 17: PLACA DE ENSAIO DE 400 PONTOS.....	55
FIGURA 18: JUMPERS DE CONEXÃO.....	56
FIGURA 19: TELA LOGIN	60

FIGURA 20: TELA SOLICITAÇÃO DE ACESSO	61
FIGURA 21: LOGIN ADMINISTRADOR	62
FIGURA 22: TELA USUÁRIOS	63
FIGURA 23: CARTÕES/TAGS	64
FIGURA 24: EDIÇÃO DE CARTÕES/TAG	65
FIGURA 25: SELEÇÃO DE USUÁRIO	66
FIGURA 26: EDIÇÃO DE USUÁRIO/ MEU PERFIL	67
FIGURA 27: TELA HORÁRIOS	69
FIGURA 28: EDIÇÃO DE HORÁRIO	70
FIGURA 29: PERÍODOS	71
FIGURA 30: EDIÇÃO PERÍODO	72
FIGURA 31: TELA FERIADOS	73
FIGURA 32: EDIÇÃO DE FERIADO	74
FIGURA 33: INFO	75
FIGURA 34: MENU DO ADMINISTRADOR	77
FIGURA 35: TELA LOGIN	78
FIGURA 36: CADASTRAR USUÁRIOS	79
FIGURA 37: LISTAR USUÁRIOS	80
FIGURA 38: VISUALIZAR USUÁRIOS	81
FIGURA 39: EDITAR USUÁRIOS USUÁRIOS	82
FIGURA 40: ADICIONAR HORÁRIOS	83
FIGURA 41: LISTAR HORÁRIOS	84
FIGURA 42: VISUALIZAR HORÁRIOS	84
FIGURA 43: EDITAR HORÁRIOS	85

FIGURA 44: ADICIONAR PERÍODO	86
FIGURA 45: LISTAR PERÍODO	87
FIGURA 46: VISUALIZAR PERÍODO	88
FIGURA 47: EDITAR PERÍODO	88
FIGURA 48: ADICIONAR FERIADO	89
FIGURA 49: LISTAR FERIADO	90
FIGURA 50: VISUALIZAR FERIADO	90
FIGURA 51: EDITAR FERIADO	91
FIGURA 52: ADICIONAR CARTÃO / RFID	92
FIGURA 53: LISTAR CARTÃO / RFID	92
FIGURA 54: VISUALIZAR CARTÃO / RFID	93
FIGURA 55: EDITAR CARTÃO / RFID	94
FIGURA 56: EVENTOS	94
FIGURA 57: MENU PRINCIPAL	96
FIGURA 58: ENTREVISTA PARA LEVANTAMENTO DE REQUISITOS FRENTE	100
FIGURA 59: ENTREVISTA PARA LEVANTAMENTO DE REQUISITOS VERSO.....	101
FIGURA 60: AUTORIZAÇÃO PARA ENTREVISTA DE LEVANTAMENTO DE REQUISITOS	102
FIGURA 61: MODELO RELACIONAL DO BANCO DE DADOS	103
FIGURA 62: DIAGRAMA DE CLASSE DA API	121
FIGURA 63: DIAGRAMA DE CLASSE FILTROS	122
FIGURA 64: DIAGRAMA DE CLASSE DECORATOR	123
FIGURA 65: DIAGRAMA DE CLASSE SERVIÇOS	124
FIGURA 66: SOLICITAR ACESSO	125
FIGURA 67: MANTER USUÁRIO	126

FIGURA 68: MANTER CARTÃO	127
FIGURA 69: MANTER PERÍODO	128
FIGURA 70: MANTER HORÁRIO	129
FIGURA 71: MANTER FERIADO	130

LISTA DE TABELAS

TABELA 1: COMPARATIVO ENTRE OS SISTEMAS DISPONÍVEIS NO MERCADO E O SISTEMA PROPOSTO.....	29
TABELA 2: REQUISITOS FUNCIONAIS	36
TABELA 3: REQUISITOS NÃO FUNCIONAIS	37
TABELA 4: ENTREGAS	46

SUMÁRIO

1. INTRODUÇÃO	13
1.1 PROBLEMA.....	15
1.2 OBJETIVO GERAL	15
1.3 OBJETIVOS ESPECÍFICOS	16
1.4 JUSTIFICATIVA	16
2. FUNDAMENTAÇÃO TEÓRICA	18
2.1 CONCEITOS TECNOLÓGICOS	18
2.1.1 SISTEMAS EMBARCADOS	18
2.1.2 RADIO FREQUENCY IDENTIFICATION - RFID.....	23
2.2 METODOLOGIA DE PROJETOS ADOTADA.....	25
3. TRABALHOS CORRELATOS.....	25
3.1. SISTEMA PRIMME SF ACESSO.....	25
3.2. SISTEMA IDACCESS PROX.....	26
3.3. SISTEMA EPC/RFID DA MEMOVE.....	27
4. METODOLOGIA DO TRABALHO	31
4.1. REQUISITOS.....	33
4.1.1. LEVANTAMENTO DOS REQUISITOS.....	33
4.1.2. ESPECIFICAÇÃO DOS REQUISITOS.....	34
4.2. ANÁLISE DOS DADOS.....	40
4.2.1 MODELO DE ENTIDADE E RELACIONAMENTO.....	39
4.3 ANÁLISE DO SISTEMA.....	41
4.3.1 DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO	42
4.3.2 DIAGRAMA DE CLASSES.....	43
4.3.3 DIAGRAMAS DE SEQUÊNCIA	45
4.4. GERÊNCIA DO PROJETO	46
4.5. TECNOLOGIAS E LINGUAGEM UTILIZADAS NO PROJETO.....	48
4.6. PROCESSO DE TESTES DO SISTEMA.....	49
5. APRESENTAÇÃO DO SISTEMA – MANUAL DO USUÁRIO.....	59
5.1 CLIENT.....	59

5.1.1 ACESSO AO SISTEMA.....	59
5.1.2 CADASTROS.....	63
5.1.3 CONFIGURAÇÕES.....	75
5.1.4 MENU.....	76
5.2 TELAS ADMIN.....	78
5.2.1 ACESSO AO SISTEMA.....	78
5.2.2 CADASTROS.....	78
5.2.3 MENU.....	96
6. CONCLUSÃO.....	97
REFERÊNCIAS.....	98
APÊNDICE A – QUESTIONARIO PARA LEVANTAMENTO DE REQUISITOS.....	100
APÊNDICE B – CARTA DE ACEITA DA EMPRESA GO RECEBA FÁCIL.....	102
APÊNDICE C – MODELO RELACIONAL DO BANCO DE DADOS	103
APÊNDICE D – ESPECIFICAÇÃO DOS CASOS DE USO.....	104
APÊNDICE E – DIAGRAMAS DE CLASSE DE IMPLEMENTAÇÃO.....	121
APÊNDICE F – DIAGRAMAS DE SEQUENCIA.....	125

1. INTRODUÇÃO

As formas de se relacionar estão sempre atreladas à algum marco histórico importante. Seja com o advento da agricultura, ou mesmo com a consolidação da família patriarcal como uma instituição. Esses marcos revolucionam a forma de convivência.

A propriedade privada é um desses acontecimentos que foi um dos divisores de água. A sua definição se mantém com poucas modificações desde seu início: "Propriedade privada compreende o direito de usar, gozar e dispor de uma determinada coisa" (PEREIRA, 1992, pág. 72.). As poucas alterações que sofre incidem sobre as garantias que o direito concede ao proprietário, bem como seus deveres.

Com o passar dos anos, esse conceito de posse ultrapassa a tangibilidade e passa a ser aplicada a outras áreas que não apenas a propriedade residencial. Exemplo disso é seu uso para definir a posse de empresas, objetos de valor e, mais recentemente, as patentes e softwares.

A vida atual leva à necessidade de proteção para determinados bens que se tenha a posse. Para que isso ocorra, muitos métodos são utilizados, entre eles o controle de acesso no âmbito físico. Este, por sua vez, é um dos métodos mais efetivos para se atingir a segurança de determinada propriedade ou bem. Essa visão é compartilhada pelos autores Nakamura e Geus (2007. p.37) que, segundo eles, "o mundo da segurança[...] é marcado pela evolução contínua, no qual novos ataques têm como resposta novas formas de proteção". E é por esse motivo que acaba sendo um dos mais utilizados para esse fim. Por isso, seu uso é dado em diversos espaços, tais como: condomínios, bancos, academias, faculdades, escolas, repartições do governo, determinados setores dentro de organizações, empresas privadas, entre muitos outros.

O controle de acesso incide sobre a premissa da segurança por prevenção. De forma resumida, ele se define com o conceito de permitir a entrada e/ou acesso a determinado local ou bem apenas para indivíduos que tenham autorização para tal

(Souza, 2010). Impedindo, assim, a entrada de quem não tem permissão prévia para acessar o local.

Para o professor e pesquisador Rafael R. Obelheiro, o controle de acesso é entendido como uma espécie de mediação entre a requisição de um sujeito a determinado objeto ou espaço:

O controle de acesso é, portanto, a mediação das requisições de acesso a objetos iniciadas pelos sujeitos. Um monitor de referência é um modelo conceitual do subsistema responsável pelo controle de acesso; é a entidade que recebe todas as requisições de acesso dos sujeitos e autoriza ou nega o acesso de acordo com a política de segurança implantada. (OBELHEIRO. 2009. p.1)

É importante destacar que controlar o acesso a um espaço faz parte de um sistema e demais táticas que prezam pela segurança, através de gerenciamento e monitoramento dos indivíduos que desejam acessar o local ou bem.

Sendo assim, a proposta do nosso projeto se mostra totalmente válida. Pois incide sobre a criação de um sistema denominado FreePass, onde suas validações de controle de acesso são feitas de maneira on e off-line, ou seja, mesmo quando não está conectado à internet. Ele foi projetado para fazer isso de forma rápida, descomplicada sem deixar de se atentar a segurança.

As liberações de acesso feitas enquanto on-line são criptografadas. Por outro lado, a base de dados, disponível off-line, é constantemente atualizada enquanto o sistema está conectado a rede, isso faz com que ele possa operar normalmente em momentos que não esteja ligado a internet.

Vale reforçar que um item fundamental para indicar a eficiência de um sistema de controle de acesso é sua base de dados. Conhecer quem pode acessar determinado perímetro faz com que se possa afastar quem não tenha autorização para tal. Com isso, a base de dados do FreePass, que opera inclusive off-line, garante que o acesso ao local protegido pelo sistema seja adentrado somente mediante validação no banco de dados. Tais características garantem ao FreePass a possibilidade de atuar plenamente como um eficiente sistema de controle de acesso que, como tal, visa aumentar a segurança de determinado local ou bem ao possibilitar a entrada somente de usuários previamente cadastrados em banco de dados.

Além disso, o sistema FreePass possibilita ao utilizador cadastrar as pessoas desejadas, associando a elas cartões RFID (do inglês "Radio-Frequency IDentification") e, com isso, gerenciar os períodos em intervalo de dias/horas para permitir seu acesso à portas, portões, catracas e derivados. Tudo isso, de maneira segura e de fácil gerenciamento, com validações em tempo real, além de contar uma base de dados off-line como contingência.

1.1 PROBLEMA

Atualmente, a segurança é algo que pode preocupar muitas pessoas. O medo causado pela violência urbana, bem como a insegurança pertinente a demais meios de convivência, faz com que as pessoas recorram a mais alternativas para se manterem distantes do perigo. Exemplo disso é a elevação nas buscas por itens de segurança em perímetro residencial. Essa situação também se repete, mantendo as devidas proporções e ajustes no âmbito empresarial.

Seja para proteger sua casa, sua empresa ou mesmo um bem como um carro, as pessoas cada vez mais buscam o auxílio em itens tecnológicos para garantir a integridade de seu patrimônio. Sendo assim, é de suma importância que ao recorrerem a esse tipo de recurso, encontrem algo de simples compreensão e que consigam operar. Logicamente, tais itens custam determinado valor, custeado, na maioria das vezes, por quem busca por maior sensação de segurança.

Com isso, os problemas de segurança atuais, muitas vezes, nos levam a seguinte questão: **como manter pessoas não autorizadas longe de determinados locais?**

1.2. OBJETIVO GERAL

Fornecer a um mercado de consumidores em potencial uma alternativa simplificada de controle de acesso, utilizando a tecnologia de reconhecimento por aproximação com RFID a determinado local, visando aumentar a segurança.

1.3. OBJETIVOS ESPECÍFICOS

- Levantar requisitos em uma empresa que necessite controlar o acesso das pessoas em sua dependências;
- Gerar cadastro de usuários previamente em banco de dados;
- Mesclar o uso das plataformas *on* e *off-line* no mesmo sistema;
- Aplicar a tecnologia de RFID no desenvolvimento do sistema de controle de acesso.

1.4. JUSTIFICATIVA

Em tempos em que a busca pelo aumento da segurança é algo extremamente ansiado pelas pessoas, a formulação de um sistema de controle de acesso se justifica. Pois, o detalhamento de informações de quem tem autorização a acessar determinado perímetro é algo extremamente importante para o aumento dos níveis de segurança.

Então, a implementação do sistema *FreePass*, uma vez operante, garantirá o controle e gerenciamento de acesso a determinado ambiente. Isso resulta em uma segurança mais efetiva e plena ao usuário. Ademais, o uso do sistema também se justifica pelo fato de que a insegurança é um problema extremamente presente na sociedade contemporânea. Muitas vezes, questões relacionadas a furto, roubo, violência, entre outros, fogem completamente ao controle dos órgãos públicos responsáveis por eles.

Por isso, manter o acesso a determinado local ou bem somente a quem possui autorização previamente concedida através de cadastro em banco de dados, é uma maneira extremamente viável para que se alcance níveis cada vez mais satisfatórios de segurança ao usuário. Além disso, é importante salientar que a modelagem do sistema *FreePass* foi viabilizada tendo sempre em vista a futura interação com o usuário. Ou seja, a forma simplificada de operação do sistema tende a não gerar insatisfações com os futuros usuários, pois ele poderá ser operado sem requerer muito embasamento técnico. Além disso, disponibilizar uma base de dados *off-line* é um diferencial que pode tornar o sistema ainda mais simples, seguro e eficaz.

Logo, o desenvolvimento de um sistema simplificado de controle de acesso garante a possibilidade de fornecer a um mercado de consumidores em potencial uma alternativa simplificada deste tipo de matriz que visa a segurança.

2. FUNDAMENTAÇÃO TEÓRICA

Neste tópico são apresentadas as duas tecnologias que foram aplicadas no desenvolvimento do sistema, bem como uma breve explanação da metodologia de gerência de projeto utilizada.

2.1 CONCEITOS TECNOLÓGICOS

2.1.1 SISTEMAS EMBARCADOS

Sistemas embarcados, segundo Andrade (2010), são sistemas computacionais de arquitetura simples, porém são completos e independentes, que geralmente trabalham interno a um outro sistema, seja ele elétrico, mecânico, hidráulico, etc.

Normalmente são desenvolvidos para executarem tarefas específicas ou pré-determinadas, desde um simples acender e apagar de luzes, até manter órgãos funcionando corretamente. Por intermédio deles pessoas também se comunicam, aviões traçam suas rotas com segurança e mantêm o funcionamento de praticamente quase tudo, tornando-se indispensáveis atualmente, sem que sequer sejam notados.

O primeiro sistema embarcado desenvolvido foi batizado com o nome de *Apollo Guidance Computer*. Era um sistema utilizado para comandar em tempo real as naves das operações *Apollo* nos Estados Unidos. Este sistema trabalhava com uma frequência de 1,294 MHz e fazia o processamento através de circuitos compostos de unidades discretas com portas NOR ao invés de um processador dedicado. Atualmente, estes sistemas estão bem mais sofisticados, respondem rapidamente a estímulos do ambiente externo e na maioria dos casos operam em tempo real. Porém dependendo da segurança de sua função alguns possuem certas restrições quanto a esta forma de execução.

Segundo Cunha (2007), os sistemas embarcados são de baixo custo, com tamanho discreto e design simplificado. Mas ao contrário do que se pensa, estes tipos de sistemas excedem com grande vantagem a quantidade de computadores, *notebooks*, celulares, entre outros tipos de dispositivos que nos acostumamos a pensar

que são os únicos com capacidade de processamento de informações, tomadas de decisões e armazenamento de dados. Porém, apesar destes sistemas serem simples, eles funcionam de forma independente, e possuem uma arquitetura basicamente semelhante a estes mais robustos citados anteriormente.

Os dados são recebidos do mundo externo através de periféricos, que servem para comunicação, realizando a troca de informações, geralmente através de teclas, conversores ou sensores. A memória, por sua vez, armazena os dados e instruções referentes a realização de cálculos, tomada de decisões, tratamento de eventos, etc. Geralmente, conforme coloca Andrade (2010), em sistemas embarcados os dados são separados das instruções em memórias distintas. Então, essas instruções são interpretadas pelo microprocessador/microcontrolador, que possuem arquitetura muito semelhante a de um processador de *desktop* convencional, e então retornam uma saída para o usuário final executando a função a que foi designado. Todo esse processo pode ser visualizado na Figura 1.

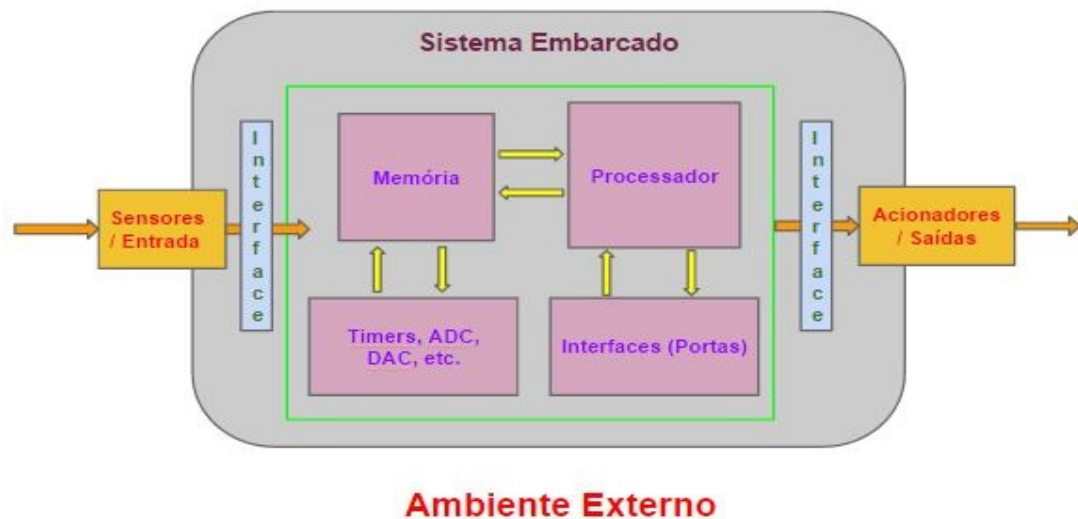


Figura 1: Arquitetura básica de um sistema embarcado.

Fonte: <http://www.bosontreinamentos.com.br/eletronica/eletronica-geral/introducao-aos-sistemas-embarcados/>

Devido a simplicidade, conforme coloca Cunha (2007), estes sistemas eventualmente funcionam com recursos computacionais limitados: sem periféricos, sem tela e com pouca memória. Também como não possuem um disco rígido, o armazenamento é feito como um *firmware*¹ em uma memória ROM ou memória flash.

Em contrapartida, a simplicidade de sua arquitetura acarreta em uma economia de recursos computacionais e custos de *hardware*, pois o sistema utiliza micro controladores, que além de já conter inúmeros periféricos integrados também são muito mais baratos. Mas é importante colocar que existem alguns sistemas mais complexos que utilizam microprocessadores.

Conforme citado em Cunha (2007), a comunicação destes sistemas com o ambiente externo se faz através de alguns periféricos, tais como (Figura 2):

- a) Interfaces Seriais: comunicação com o usuário, sensores e atuadores para recolher dados do mundo real;
- b) Cartões SD, *Compact Flash*, etc utilizados para armazenamento de dados;
- c) Rede Ethernet;
- d) GPIO (*General Purpose Input/Output Pin*) que são pinos de propósito geral de entrada e saída de informações como no *Raspberry Pi*²;
- e) Conversores ADC e DAC responsáveis pela aquisição dos sinais analógicos para o sistema;
- f) Barramentos especializados como *PROFIBUS*;
- g) Interfaces *wireless* como *XBee* e *Bluetooth*.

¹ Também conhecidos pela nomenclatura “software embarcado”, os Firmware são um conjunto de instruções operacionais que são programadas diretamente no hardware de equipamentos eletrônicos.

² *Raspberry Pi* é um computador do tamanho de um cartão de crédito, que se conecta a um monitor de computador ou TV, e usa um teclado e um mouse padrão, desenvolvido no Reino Unido pela Fundação Raspberry Pi. Todo o hardware é integrado numa única placa.

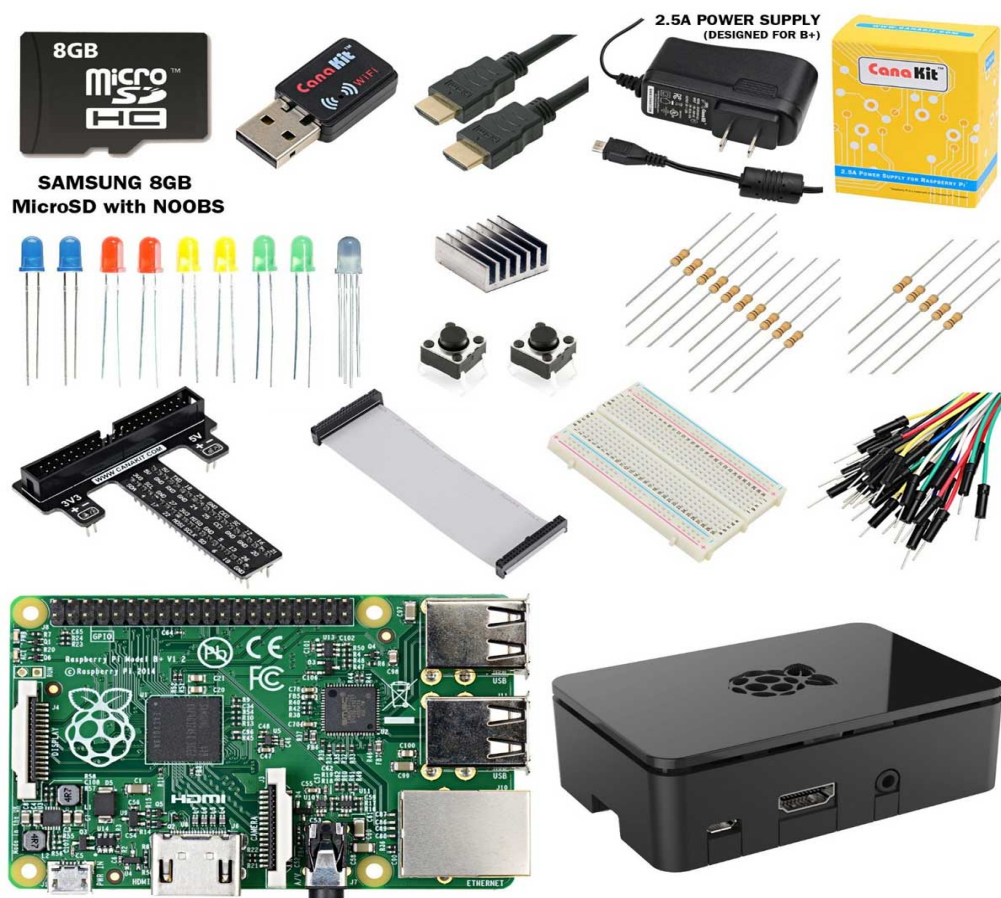


Figura 2: Periféricos utilizados por sistemas embarcados para a comunicação com o ambiente externo.

Fonte: <https://br.pinterest.com/pin/293085888222010021/>

Os sistemas embarcados operam com diversos tipos de processadores, dentre eles estão: DSPs (*Digital Signal Processors* - Processadores Digitais de Sinais), micro-controladores, micro-processadores. Diferentemente dos *desktops* onde predominam os processadores de arquitetura x64 da Intel e AMD, os sistemas embarcados dispõe das arquiteturas ARM, Famílias Intel Atom e Intel Quark, PIC, Qualcomm, Hitachi, 8051, Coldfire, TMS320, blackfin, entre outros.

É notável que os sistemas embarcados, além de dominarem o mercado em quantidade segundo Zurita (2013), com a sua importância faz com que se tornem uma das áreas mais promissoras da tecnologia. Pois, além de consumirem menos recursos que os computadores de uso geral, também são muito mais estáveis, o que faz com que sejam aplicados a tarefas em que é essencial o perfeito funcionamento, onde uma simples queda no sistema pode custar até mesmo vidas.

2.1.2 RADIO FREQUENCY IDENTIFICATION - RFID

O *Radio Frequency Identification* (RFID) é um sistema tecnológico que utiliza a radiofrequência³ para a comunicação entre dispositivos Glover e Bhatt (2007).

O sistema se baseia numa premissa simples, uma *tag*⁴ e um leitor de *tags*. Exemplos de tags e leitores de tags encontram-se na Figura 3. As *tags* podem variar dependendo do fabricante, mudando o formato tanto físico quanto digital. A comunicação ocorre através das ondas magnéticas de radiofrequência, possibilitando a leitura dos dados da *tag* em grandes distâncias e mesmo não estando visível ao leitor.

³ A radiofrequência, em suma, é a parte do espectro electromagnético que abarca desde os 3 kilohertz aos 300 gigahertz.

⁴ É um [transponder](#) (dispositivo de comunicação eletrônico), pequeno objeto que pode ser colocado em uma pessoa, animal, equipamento, embalagem ou produto, dentre outros. Contém chips de [silício](#) e [antenas](#) que lhe permite responder aos sinais de rádio enviados por uma base transmissora.



Figura 3: Exemplos de tags e leitores de tags.

Fonte: http://www.jesic-tech.com/RFID_tag.html

Com o desenvolvimento da tecnologia, se tornou viável a utilização do RFID nas empresas, deixando o processo de monitoramento dos itens fabricados com maior controle e eficiência. O RFID também pode ser utilizado para monitorar incêndios florestais; controle de temperatura em locais fechados; garantir o crescimento de áreas florestais; no rastreamento de objetos, de carga, carros, vestuários, pneus e inclusive livros, para identificar e impedir furtos.

O uso do RFID agilizou e melhorou alguns processos rotineiros, tais como: transferências em tempo real de dados entre dispositivos; facilidades para identificação, localização e monitoramento de objetos físicos; acesso quase que instantâneo a bases de conhecimentos utilizadas para solução de um problema; inibi fraudes de maneira mais eficaz, pois objetos possuem RFID único, sendo impossível realizar uma cópia da identificação.

2.2 METODOLOGIA DE GERÊNCIA DE PROJETOS ADOTADA

Para o gerenciamento deste projeto, foi aplicada o modelo cascata onde foram adotadas algumas práticas da metodologia ágil Scrum.

O *Scrum* é uma metodologia ágil utilizada para gerenciar e controlar o desenvolvimento de *software*, através de práticas iterativas e incrementais (Cruz, 2015). Com isso, baseado nas práticas de gerenciamento do *Extreme Programming* e no RUP (Rational Unified Process), o *Scrum* produz os benefícios do desenvolvimento ágil com a vantagem de ser uma implementação simples (YOSHIMA, 2007).

O modelo cascata é um modelo de desenvolvimento moldado para ser utilizado na implementação de diversos tipos de projetos. Sua principal característica é que ele segue um modelo de desenvolvimento linear e sequencial, quando uma etapa é finalizada o desenvolvimento passa para a próxima fase sem que se permita retornar para a fase anterior.

A vantagem deste tipo de desenvolvimento é que ele possui fases bem definidas dando um maior foco no planejamento, já que só se avança para a próxima etapa quando o cliente valida e aceita os produtos finais da tarefa atual, dando uma maior interação e proximidade com o cliente reduzindo as chances de falhas ao final do projeto.

Neste modelo de desenvolvimento começamos então com o levantamento de requisitos, onde realizamos entrevista junto ao cliente para avaliar as necessidades que o sistema deverá atender, depois partimos para a fase de planejamento para então estabelecer estimativas, cronograma e acompanhamento avaliados pelos requisitos que foram levantados na fase anterior, com essas etapas finalizadas iniciamos então a modelagem do sistema para a análise do projeto, definição da estrutura, etc. Após isto partimos para o desenvolvimento do sistema seguidos de codificação e testes para localizar possíveis falhas e carências do sistema, por fim realizamos a implementação na entrega ao cliente e em seguida fornecemos suporte e feedback, para sanar dúvidas e realizar a manutenção do sistema. A Figura 4 ilustra o funcionamento da metodologia Scrum.

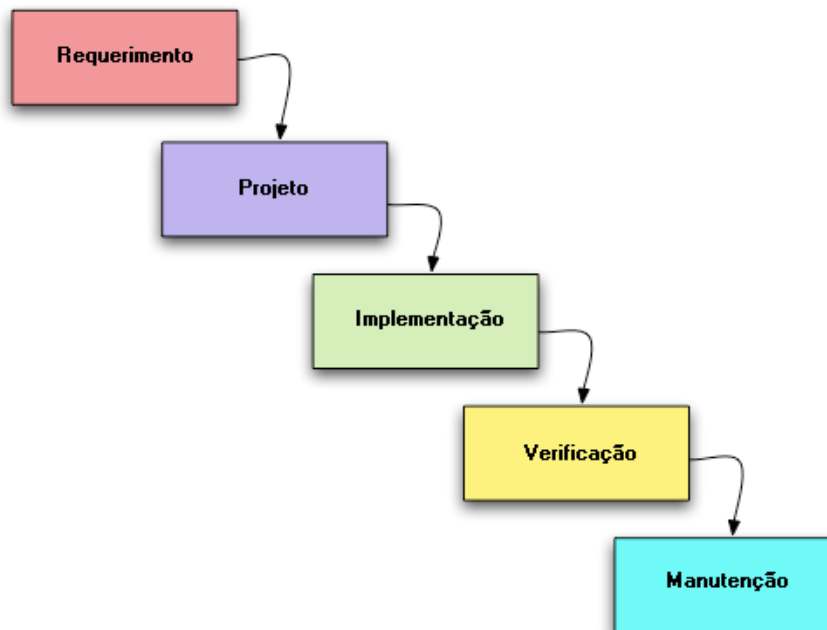


Figura 4: Funcionamento do Modelo Cascata.

3. TRABALHOS CORRELATOS

Veremos a seguir alguns trabalhos correlatos que possuem funcionalidades e estrutura semelhantes ao sistema *Freepass*.

3.1. SISTEMA *PRIMME SF ACESSO*

Equipamento utilizado para controle de acesso de portas, condomínios, estacionamentos e entradas de veículos (Figura 5). Informações detalhadas sobre esse sistema estão disponíveis no site <http://www.henry.com.br/controle-de-acesso/primmesfaccess>.

Os pontos relevantes do *Primme SF Acesso* são:

- Gerencia até 15.000 matrículas de usuários.
- Possibilita a instalação de diversas tecnologias de leitores, como biometria, proximidade, *smart card contactless* e código de barras.
- Armazena os últimos 8 milhões de eventos.

- Sinalização visual e sonora de status de acesso.
- Aplicativo já incluso no equipamento, dispensando software de instalação.
- O equipamento mostra no *display* e emite aviso sonoro para indicar a necessidade de revista no usuário, podendo ser configurada a percentagem de sorteio;



Figura 5: Sistema Prime Acesso SF

Já pontos fracos do *Primme SF Acesso* são:

- Não possui auto-pré-cadastro.
- Tela monocromática, poucas informações
- Limitação na navegação utilizando teclas físicas causando uma experiência de usuário menos interativa
- Poucas informações do usuário no cadastro

3.2. SISTEMA IDACCESS PROX

O sistema *iDAccess Prox* é capaz de controlar com eficiência e rapidez a entrada, saída e frequência de pessoas a ambientes (Figura 6). Informações mais detalhadas sobre esse sistema estão disponíveis no site <https://www.controlid.com.br/blog/produto/controlde-de-acesso/controlde-de-acesso-idaccess/>.

Os pontos relevantes do *iDAccess Prox* são:

- Capacidade para mais de 1.000.000 usuários.
- Capacidade para mais de 1.000.000 de registros.

- Serviço de monitoramento, sincronização e backup na nuvem.
- Detecção de vandalismo; alarme configurável de remoção do equipamento do suporte.
- Central de alarme integrada.



Figura 6: Sistema iDAccess Prox

Já os pontos fracos do *iDAccess Prox* são:

- Não possui auto-pré-cadastro;
- Poucas informações do usuário no cadastro
- A tela pequena e ausência de teclado físico, fazem com que a digitação de textos na tela de tamanho reduzido do equipamento se torne dificultosa.

3.3. Sistema EPC/RFID da MEMOVE

Memove é uma loja varejista que comercializa roupas e acessórios para homens e mulheres. Ela se destina a consumidores modernos com seus 20 e poucos anos, propiciando uma experiência de compra inovadora, pois é a primeira da rede varejista no Brasil a adotar o EPC/RFID de ponta a ponta na cadeia de suprimentos, para realizar mais precisamente o controle de estoque e efetivamente na prevenção de

furtos (Figura 7). Informações mais detalhadas sobre o Sistema EPC/RFID da Memove estão disponíveis no site <http://www.valdacglobalbrands.com.br/site/memove-grupo-valdac-a-primeira-operacao-de-varejo-de-moda-100-rfid-no-brasil/>.

Os pontos fortes desse sistema são:

- Precisão e agilidade na localização de peças em estoque.
- Processo de recebimento 300% mais rápido e 100% de precisão do inventário com o EPC/RFID em nível de item.
- Automatização no processo de compra, pois existem cestas disponíveis com a *tag* RFID. O cliente coloca a sua seleção no cesto, as *tags* são lidas por um leitor embutido e o valor total a pagar é calculado automaticamente. O pagamento é feito em uma máquina de cartões ao lado das cestas evitando filas.
- *Check-out* é em média 60% mais rápida e os vendedores das lojas podem passar mais tempo com os clientes quando eles compram e experimentam as roupas.

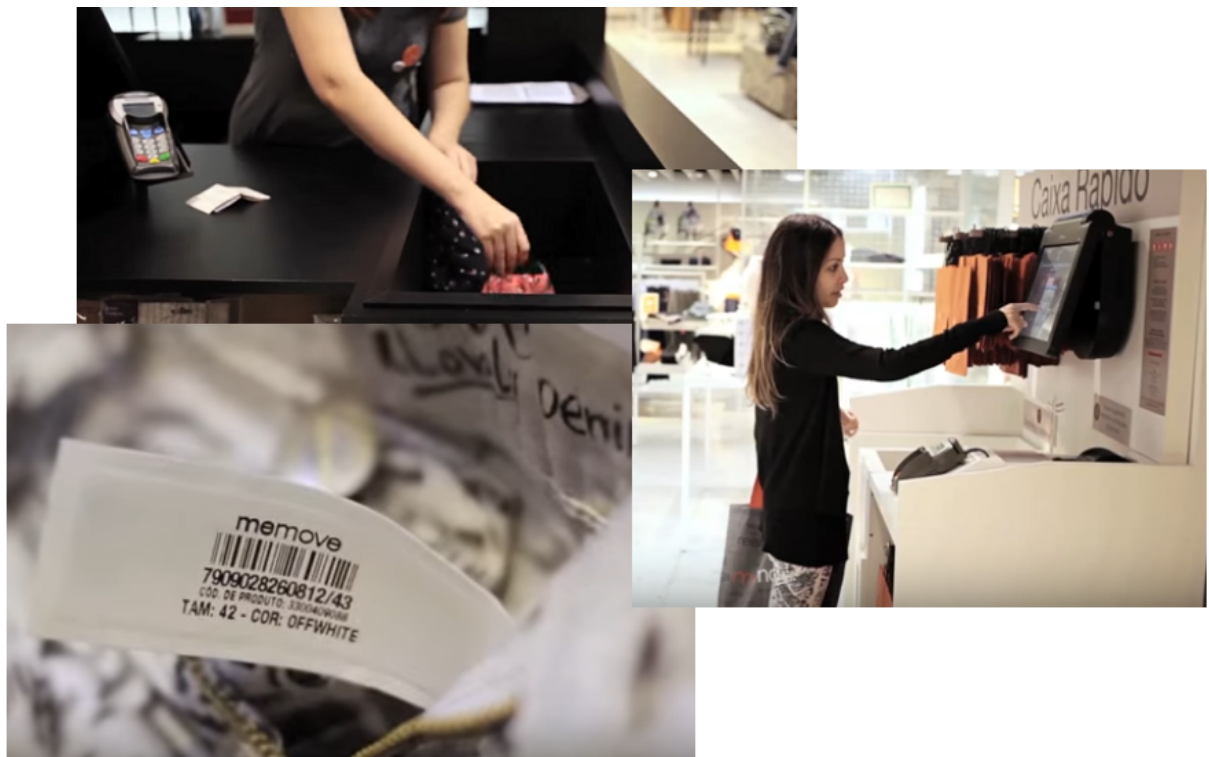


Figura 7: Sistema EPC/RFID da Memove

Pontos negativos:

- Capacidade de Armazenamento.
- Custo da tag RFID, que é bem maior que o convencional código de barras.
- Má interação com metais. (Pode ser contornado através de encapsulamentos).
- Inexistência de padronização das frequências.

A Tabela 1 apresenta um comparativo, com base nos principais requisitos para um sistema de controle de acesso, entre os sistemas disponíveis no mercado, descritos nesse trabalho, e o sistema proposto – *FREE PASS*.

Tabela 1: Comparativo entre sistemas disponíveis no mercado e o sistema proposto.

	Prime SF	iDAccess Prox	EPC/RFID da MEMOVE	Free Pass

Quantidade de Armazenamento de usuários				x
Sistema de gerenciamento local	x	x	x	x
Facilidade de operação do Sistema				x
Quantidade de formas de acesso	x	x		
Segurança do sistema	x	x		
Auto pré-cadastro				x
Visibilidade das telas			x	x

4. METODOLOGIA DO TRABALHO

O processo de desenvolvimento de *software* nada mais é do que etapas compostas de metodologias que auxiliam a definir, desenvolver, testar e manter um sistema (Soares, 2004). Para que um projeto seja estável as etapas de desenvolvimento devem estar bem estruturadas e definidas, com revisões sendo realizadas ao final de cada uma para garantir a consistência do sistema.

Abaixo serão descritas as etapas que foram executadas no desenvolvimento do sistema proposto e seus conceitos relevantes, utilizados para obter um padrão de qualidade no desenvolvimento da aplicação.

Etapa 1: Fase de Contextualização- Iniciação

a) Levantamento de Requisitos:

Antes de iniciar o desenvolvimento foram definidas as funcionalidades do sistema e quais problemas buscava-se solucionar. Ou seja, foram modeladas as principais funções que o sistema deveria ter considerando os problemas e as necessidades do cliente, dando uma maior visão para a implementação do projeto.

Para tanto, foi realizada uma entrevista com o cliente com a aplicação de um questionário, a fim de alcançar um melhor entendimento de cada processo, compreendendo cada requisito, para que as demais etapas ocorressem em conformidade com o processo anterior.

Após este levantamento cada funcionalidade passou por uma análise para tratar e avaliar inconsistências, ambiguidades, riscos e prioridades dos requisitos indicados na identificação.

b) Análise de Requisitos: Pertencente a Fase de Iniciação, está foi a etapa em que uma análise mais precisa foi realizada nos requisitos levantados na etapa anterior. Em reuniões, juntamente com o cliente, foram elaborados modelos de escopos que serviram de base para representação do sistema. Além disso, foram apuradas todas as

necessidades do cliente para um melhor entendimento do que o sistema deveria atender.

Etapa 2: Fase de Design

a) Projeto: nesta etapa foram elaboradas, de forma detalhada, as especificações do projeto, tais como arquitetura do sistema, prototipação das telas, banco de dados, seleção das linguagens utilizadas para o desenvolvimento, definição dos *hardwares* a serem utilizados, sistema operacional, entre outros. Também foram modeladas as relações pertinentes a cada módulo, a fim de segmentar as funcionalidades do sistema e garantir que futuras funcionalidades possam ser atendidas.

Etapa 3: Fase de Desenvolvimento

a) Implementação: nesta etapa as especificações definidas na Fase de Design foram traduzidas em o código fonte executável. Foram definidas as classes de objetos e os níveis de segurança, e elaborados os mecanismos de testes. Então, iniciou-se a implementação a partir da especificação de cada componente, integrando os módulos conforme a arquitetura interna do *software* (Figura 9).

Etapa 4: Fase de Conclusão

a) Testes: Nesta etapa foram aplicados os mecanismos de testes a fim de testar e validar cada requisito do sistema, com o propósito de eliminar qualquer rastro de erros, seja de implementação ou de especificação. Nesta fase redobrou-se a atenção, pois é muito comum que apareçam uma série de *bugs* que precisaram ser reparados para garantir a consistência do sistema.

A Figura 8 apresenta a hierarquia e as relações entre as etapas de desenvolvimento do sistema.

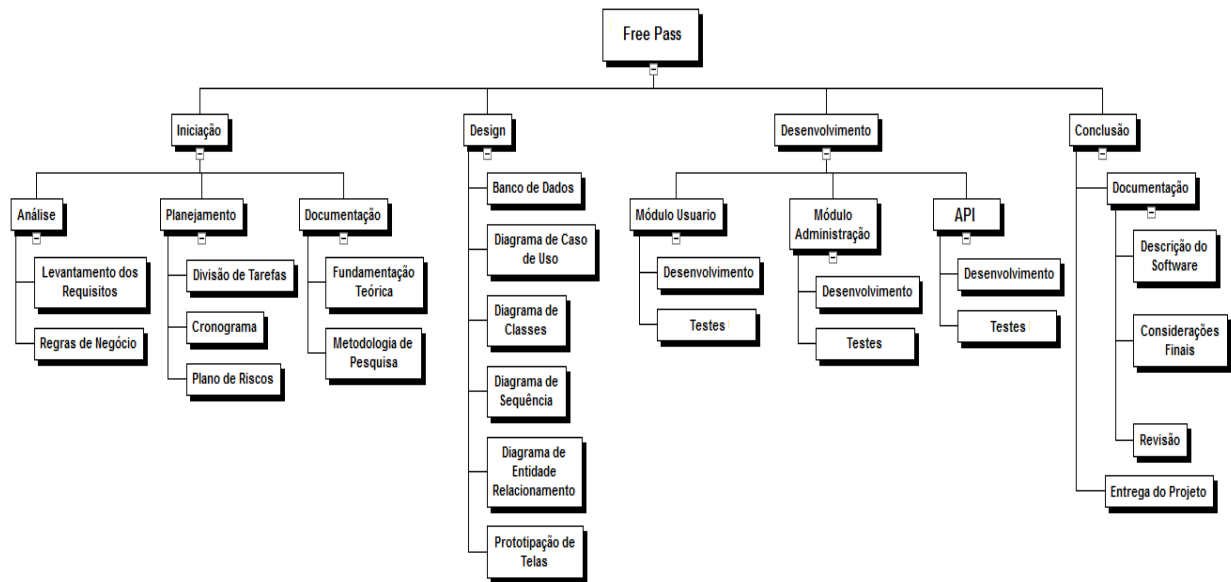


Figura 8: WBS da metodologia de trabalho do projeto.

4.1.REQUISITOS

Na etapa de definição dos requisitos, faz-se o levantamento das necessidades que envolvem o projeto e as especificações dessas necessidades transformadas são transformadas em requisitos funcionais e não funcionais.

4.1.1.LEVANTAMENTO DOS REQUISITOS

A etapa de levantamento de requisitos deve levar em consideração as necessidades reais do cliente(usuário). Como o objetivo desse projeto é fornecer a um mercado de consumidores em potencial uma alternativa simplificada de controle de acesso, não há um cliente específico.

Mas, para garantirmos o levantamento de requisitos reais para o sistema proposto foi realizada uma entrevista, tendo como base um questionário (Apêndice A) de 18 questões, com uma empresa em potencial chamada *Go Receba Fácil* (Apêndice B – carta da empresa concordando em colaborar com o projeto) com o intuito de elaborar um escopo da realidade para o projeto.

A GO Receba Fácil é uma empresa que oferece aos seus clientes, Pessoa Física e Pessoa Jurídica, a possibilidade de receberem suas vendas em débito e a crédito em qualquer lugar do território nacional, através das suas máquinas móveis sem a necessidade de uso de celulares ou qualquer outro tipo de terminal ou aplicativo. Ela se diferencia no mercado por oferecer a seus clientes a possibilidade de trabalharem com a opção sem juros, e poderem receber todas as suas vendas, sejam de crédito ou parceladas, nos cartões VISA e Mastercard, com segurança e tranquilidade em uma única parcela em 02 dias úteis. A empresa está localizada no Bairro Seminário em Curitiba, com 25 funcionários e com 2 anos no mercado.

4.1.2.ESPECIFICAÇÃO DOS REQUISITOS

Com relação ao requisito de arquitetura do sistema, foi definido a arquitetura em camadas. Este modelo de arquitetura foi escolhido para mantermos a responsabilidade em apenas um local. Assim evitamos repetições de regras de negócios no cliente, API e cliente web. Tornando assim apenas a API responsável por carregar as regras de negócios, facilitando a manutenção, correção de bugs e implementação de novas funcionalidades.

A integração entre as camadas é demonstrada na Figura 9.

Na camada cliente (*raspberry pi*), a arquitetura está dividida basicamente em *view*, módulo de integração com a API, uma camada de aplicação e o módulo de integração com o *hardware*.

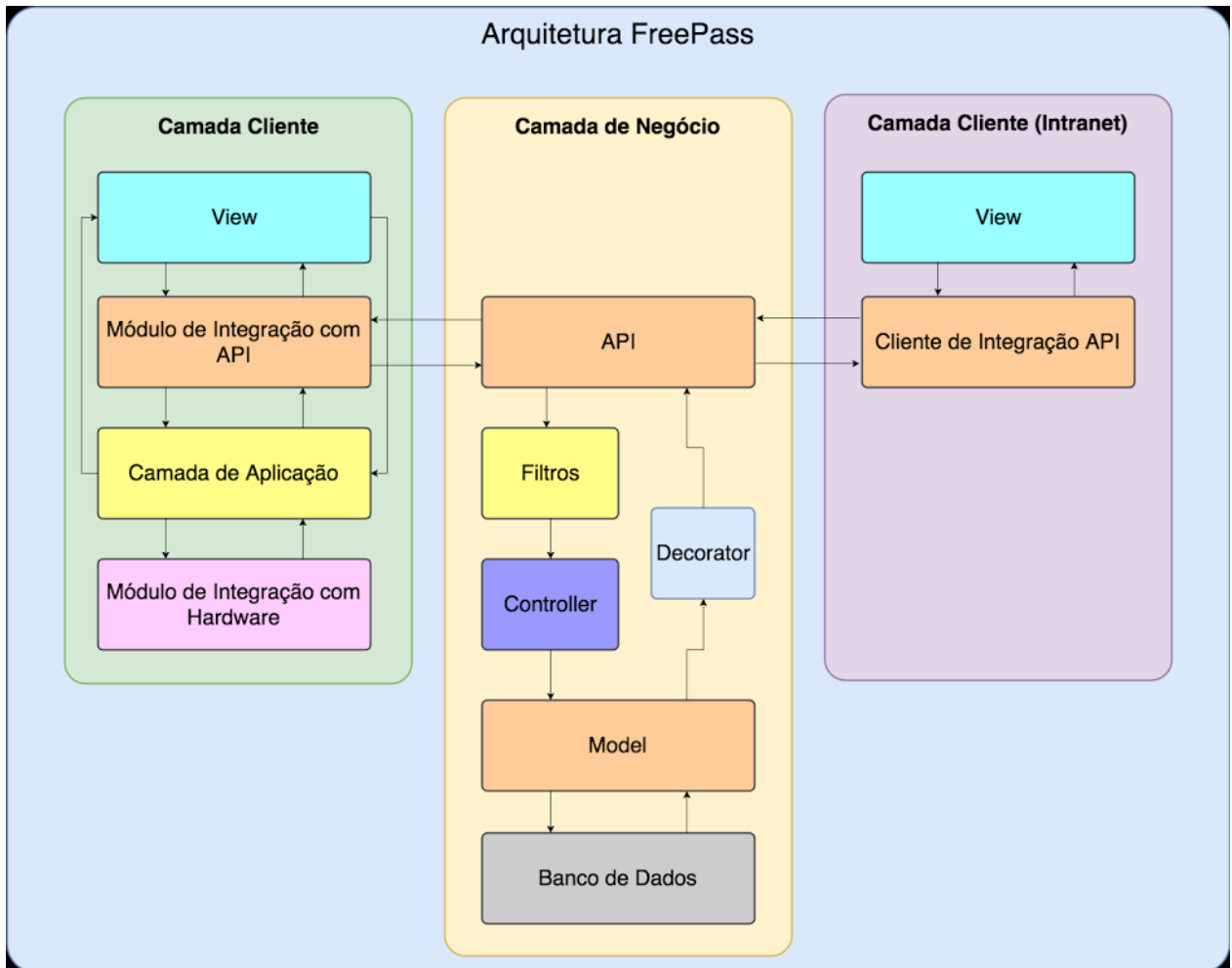


Figura 9: Integração das camadas do sistema.

A arquitetura do Raspberry Pi consiste basicamente de três camadas, são elas: Camada de Negócio e Duas Camadas Cliente, sendo uma para acesso diretamente na tela do dispositivo, e a outra (Intranet) para acesso em rede.

As três camadas estão hospedadas localmente no hardware embarcado, portanto, a divisão entre elas deve-se à tecnologia empregada para desenvolvê-las e hospedá-las e o fato de terem sido desenvolvidas como 3 sub-projetos em paralelo.

A Camada Cliente, funciona como uma interface para a API da camada de negócio, não toma nenhuma decisão de regra de negócio, apenas efetua requisições à camada de negócio e dependendo da resposta efetua ou não uma interação com o hardware.

A outra Camada Cliente (Intranet), assim como a primeira, não toma decisões de regra de negócio e funciona como uma interface gráfica para a camada de negócio.

A Camada de Negócio toma todas as decisões quanto à liberação de acesso, cadastro das entidades, formatação dos dados, aplicação de buscas e filtros juntamente com a persistência dos dados

Estas integrações estão representadas nos diagramas de sequência localizados no APÊNDICE F.

a) Requisitos Funcionais:

Tabela 2 : Requisitos Funcionais

Módulo do Sistema	Requisito	Descrição
Módulo usuário	Manter usuário	O administrador pode cadastrar, alterar, pesquisar ou excluir um usuário.
Módulo horários	Manter horários	O administrador pode cadastrar, alterar, pesquisar ou excluir um horário.
Módulo períodos	Manter períodos	O administrador pode cadastrar, alterar, pesquisar ou excluir um período.
Módulo feriados	Manter feriados	O administrador pode cadastrar, alterar, pesquisar ou excluir um feriado.
Módulo cartão	Manter cartão	O administrador pode cadastrar, alterar, pesquisar ou excluir um cartão.

Módulo usuário	Solicitar cadastro	O usuário pode solicitar a sua aprovação para realizar o cadastro no sistema.
Módulo administrador	Aprovar cadastro	O administrador pode autorizar o acesso de um usuário as áreas restritas.
Módulo usuário	Solicitar acesso	O usuário pode solicitar acesso a área restrita.
Módulo administrador	Visualizar eventos	O administrador pode visualizar os logs de acessos dos usuários.

b) Requisitos Não funcionais

Tabela 3: Requisitos Não Funcionais

Tipo de Requisito	Requisito	Descrição
Requisitos de produto	Comunicação via <i>webserver</i> local.	Integração dos sistemas deve ser feita em <i>webserver</i> local.
Requisitos de produto	Desenvolvimento das telas do sistema utilizando <i>eléctron</i> (Html 5, css 3).	Deve-se utilizar o <i>framework eléctron</i> para confeccionar as telas com

		tecnologias web como JavaScript, HTML e CSS, pois como não utilizamos um navegador normal o mesmo simula um para que a aplicação pareça nativa no cliente.
Requisitos de produto	Desenvolvimento da API utilizando <i>Ruby</i> ..	Deve-se utilizar a linguagem <i>Ruby</i> para o desenvolvimento da API.
Requisitos de produto	Armazenamento de dados na base <i>Postgres</i> SQL.	Sistema gerenciador de banco de dados com tamanho ilimitado com baixo custo.
Requisitos externos	Curto tempo de resposta.	Processador de quatro núcleos para alta velocidade de processamento proporcionando curto tempo de resposta.
Requisitos de produto	Hospedagem in-loco	Hospedagem da API no servidor Linux Embarcado no <i>Raspberry PI</i> .
Requisitos de interface	Presteza	Na tela são exibidas mensagens para auxiliar o

		usuário, informando-o sobre as ações a serem tomadas.
Requisitos de interface	Feedback	Ao executar uma ação no sistema, o usuário recebe como resposta alguma mensagem detalhando caso haja algum erro na operação ou haja problema no sistema de acesso.
Requisitos de interface	Legibilidade	Devido a tela do sistema de acesso ter apenas 5”, foi necessário adaptar uma interface prezando por este requisito, à fim de evitar dificuldades na visualização.
Requisitos de interface	Densidade Informacional	Com uma tela pequena, o sistema também foi desenvolvido de maneira que as funcionalidades pudessem ser bem visualizadas de cada tela, mantendo assim uma densidade de informações menos poluente.
Requisitos de interface	Mensagens de Erro	O sistema retorna mensagens de erro específicas para cada situação durante a

		<p>operação, permitindo que o usuário saiba na maioria dos casos que ação tomar. Quando o erro é sistêmico, um administrador do sistema treinado entenderá o conteúdo das mensagens.</p>
--	--	--

4.2. ANÁLISE DOS DADOS

Baseando-se nas necessidades do projeto levantadas como requisitos, definimos a estrutura de dados na qual o sistema será construído utilizando-se de diagramações UML para representação das entidades e seus relacionamentos.

4.2.1 Modelo de Entidade e Relacionamento

A construção do banco de dados foi baseada em uma modelagem relacional, para permitir uma facilidade na manipulação dos dados e maior integridade dos mesmos.

As principais tabelas são: Usuário, Cartão, Período, Horário, Feriado e Eventos.

A tabela usuário é responsável por armazenar todos os novos registros e dados dos usuários do sistema. Já os cartões pertinentes aos usuários que possuem a *tag RFID* tem seus dados armazenados na tabela Cartão. Para conceder ou não o acesso ao usuário, o sistema verifica nas tabelas de Período, Horário e Feriado, as respectivas permissões. A tabela de Eventos por sua vez consiste em armazenar os eventos (entradas e saídas), que podem ser consultadas por todos os usuários do sistema.

A Figura 10 apresenta o modelo conceitual de dados do sistema. Já no Apêndice 3 encontra-se o modelo relacional do banco de dados.

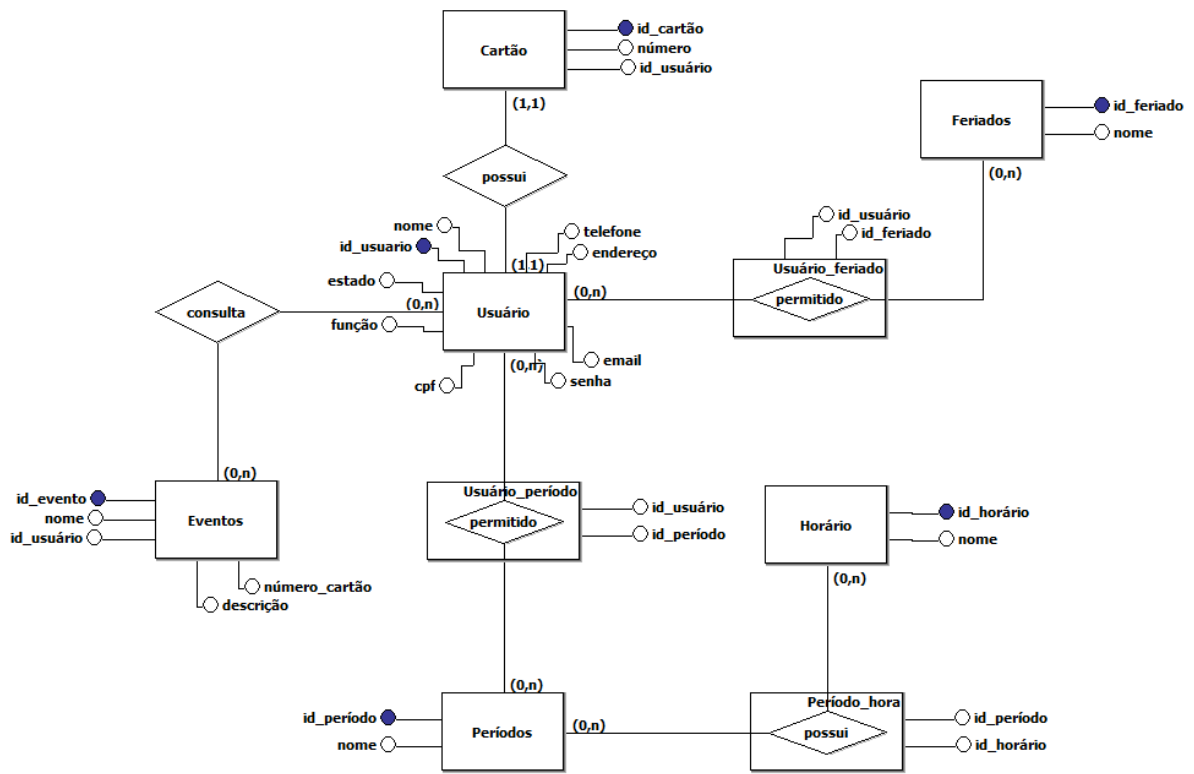


Figura 10: Modelo conceitual dos dados.

4.3 ANÁLISE DO SISTEMA

A UML (*Unified Modeling Language*) é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos (FOWLER, 2005). Sintetiza os principais métodos existentes, sendo considerada uma das linguagens mais expressivas para modelagem de sistemas orientados a objetos.

Como a proposta deste projeto é modelar e implementar um sistema utilizando o paradigma orientado a objetos, optou-se por utilizar a UML para a especificação e modelagem do sistema.

Dos 13 tipos diferentes de diagramas disponíveis na UML, neste projeto foram desenvolvidos três, que são: Diagrama de Casos de Uso, Diagrama de Classe e Diagrama de Sequência.

4.3.1 DIAGRAMA E ESPECIFICAÇÃO DE CASOS DE USO

Guedes (2011) descreve que o Diagrama de Casos de Uso (DCU) tem por objetivo possibilitar a compreensão do comportamento externo do sistema por meio de uma linguagem simples e abstrata.

DCUs são compostos por dois componentes principais: atores e casos de uso. Um Ator é uma entidade que interage com o sistema, podendo ser um humano ou um outro sistema. O símbolo utilizado para a representação do ator é um boneco palito e dos processos é uma elipse (MELO, 2010).

No Diagrama de casos de uso (Figura 11) podemos visualizar todas as funcionalidades do sistema, divididas em 3 atores. O primeiro é o perfil de usuário, que são os funcionários que necessitam realizar os acessos as áreas restritas, o segundo perfil é o de Administrador que herda todas as funcionalidades pertinentes ao ator usuário além das suas próprias funcionalidades de gerenciamento, e por fim, o terceiro ator é a aplicação, responsável por gerar solicitações de acessos.

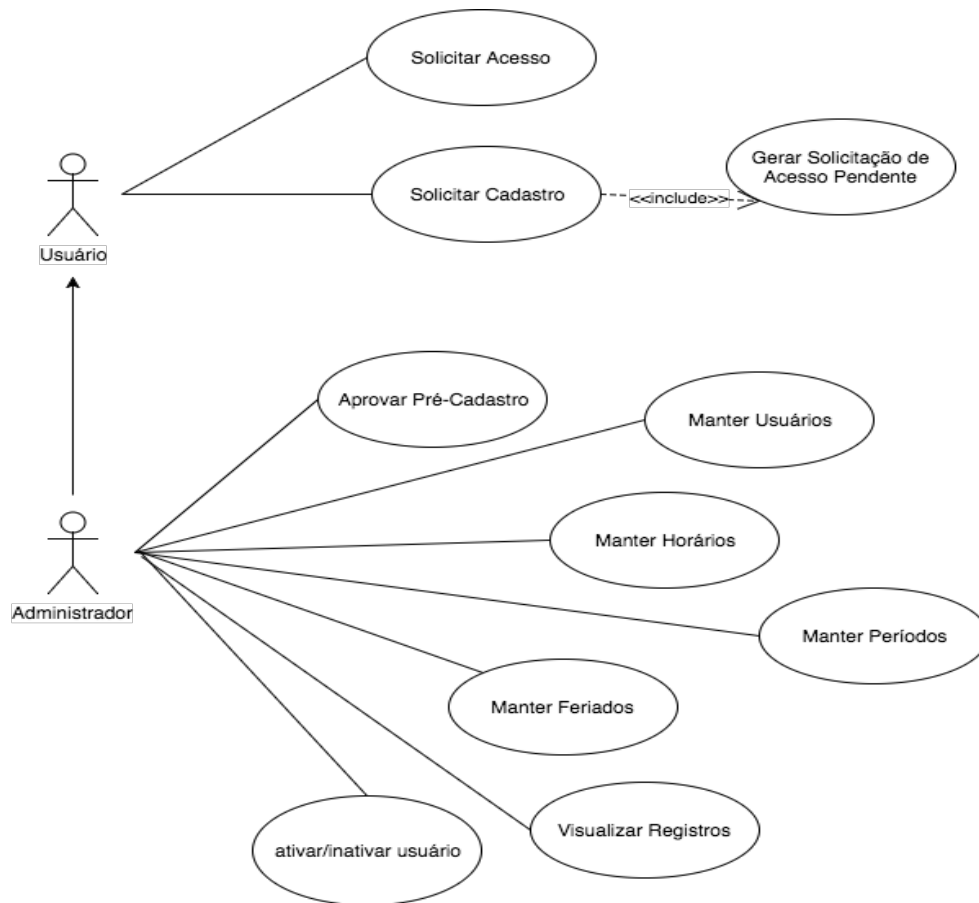


Figura 11: Diagrama de Caso de Uso.

As especificações dos casos de uso encontram-se no Apêndice 4.

4.3.2 Diagrama de Classes

Um diagrama de classes é um modelo fundamental de uma especificação orientada a objetos. Produz a descrição mais próxima da estrutura do código de um programa, ou seja, mostra o conjunto de classes com seus atributos e métodos e os relacionamentos entre classes. As classes e relacionamentos constituem os elementos sintáticos básicos do diagrama de classes (SILVA, 2007).

Podemos visualizar as principais classes do sistema *FreePass* na Figura 12.

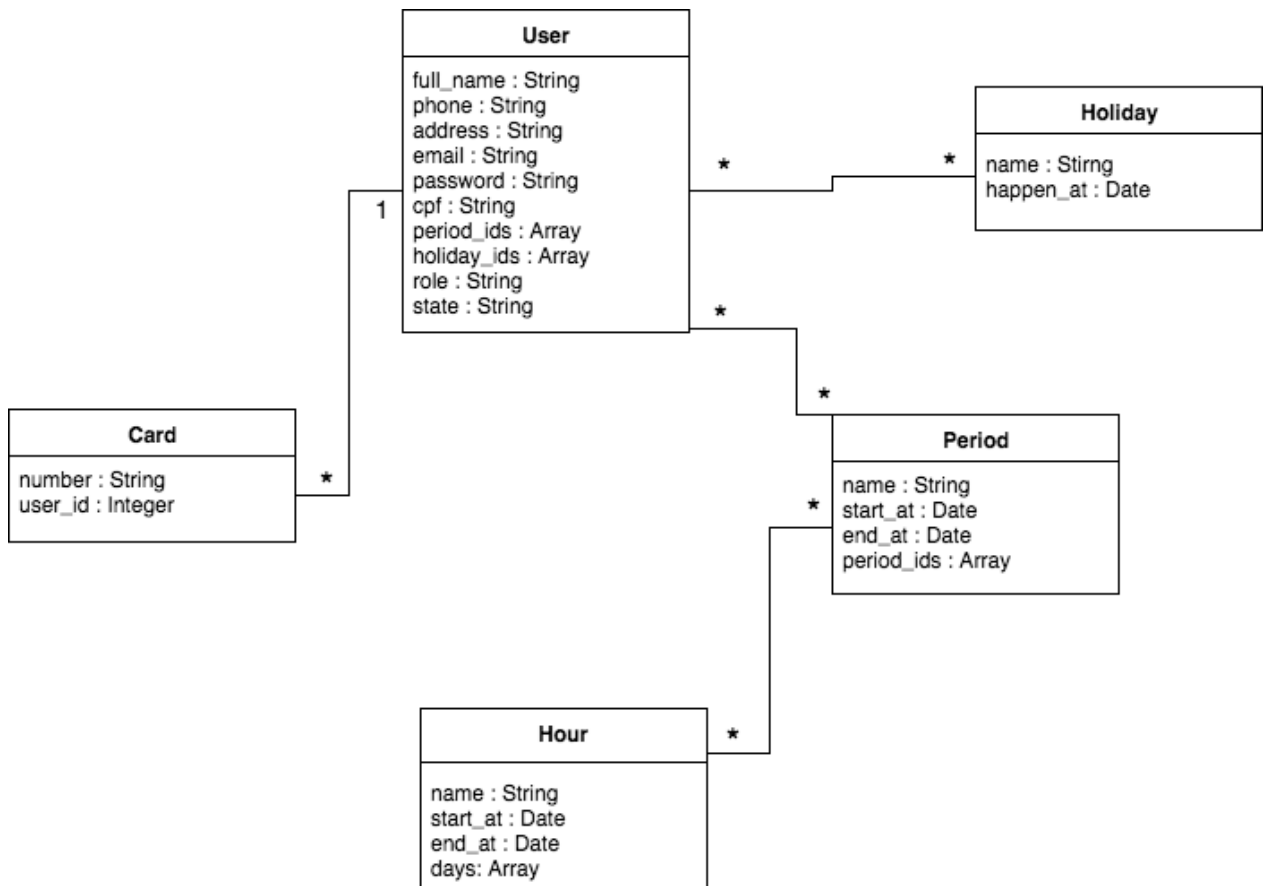


Figura 12: Diagrama De Classe De Negócio

Com relação aos diagramas de Classe de Implementação, estes encontram-se no Apêndice 5:

- Diagrama de Classes *Filters*: são responsáveis por receber qualquer tipo de parâmetros e através de uma *white list* filtrar apenas os necessários e válidos para o *Model*.
- Diagrama de Classes *Decorators*: possuem a responsabilidade de tratar o dado de uma forma amigável, por exemplo, converter o uma data e hora para o fuso horário solicitado pelo cliente.
- Diagrama de Classes *Services*: são classes que possuem regras de negócios para cada *Model*, por exemplo, buscar informações de feriados no banco de dados e validar se a data de acesso é um feriado válido.
- Diagrama de Classes API's: herdam a responsabilidade da classe *Roda*, onde possuem regras para entender requisições RESTFUI HTTP.

4.3.4 DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência mostra a relação entre os atores e as classes do sistema, onde os objetos são retratados por linhas verticais tracejadas, contendo um nome como atributo de identificação. Há troca de mensagens entre as classes/objetos que são representadas por setas, através da linha do tempo com os parâmetros correspondentes a mensagem (Booch, 2005).

Os diagramas apresentados no apêndice 6 correspondem as estruturas: ao solicitar um novo acesso utilizando o cartão que é somente utilizada no client. E os diagramas de manter usuário, cartão, período, horário e feriado que são correspondentes a camada de negócios da API.

4.4.GERÊNCIA DO PROJETO

Para a realização deste projeto foi adotado um modelo de gerenciamento que fosse possível de aplicar em um time reduzido. Desta forma optou-se por aplicar o método Cascata, a fim de tornar a gerência do projeto mais eficiente e adaptada a realidade da equipe.

Na metodologia de desenvolvimento de projetos de software utilizamos práticas baseadas na metodologia ágil *Scrum*, onde tem-se um modelo mais adequado para a elaboração de projetos de software orientados a objetos. Trata-se de um método flexível e adaptativo que foca seu desenvolvimento na colaboração entre pequenos grupos de profissionais, sujeitos a circunstâncias de frequentes mudanças, como é próprio num desenvolvimento de software (SOARES, 2004).

As reuniões diárias eram realizadas por *Skype* onde dividimos as funções de *Product Owner* (dono do produto) e *Scrum Master* entre os membros de forma equivalente.

As Entregas foram divididas conforme a tabela abaixo.

Tabela 4: Entregas

	Objetivos da sprint	Data inicial	Data entrega	Time box
Entrega 1	<ul style="list-style-type: none">• Definição do cronograma do projeto;• Divisão do backlog do produto nas sprints;• Criação de repositório de código-fonte para versionamento do projeto;• Levantamento e análise de requisitos;• Desenho de Telas.	12/08/2016	19/08/2016	1 semana
Entrega 2	<ul style="list-style-type: none">• Prototipação das telas;• Desenvolvimento de	19/08/2016	26/08/2016	1 semana

	<p>documentação UML (Diagramas);</p> <ul style="list-style-type: none"> • Definição da arquitetura do Sistema; • Configuração do Servidor. 			
Entrega 3	<ul style="list-style-type: none"> • Finalização das telas; • Início com a documentação; • Criação dos Scripts de Banco de Dados; • Implementação da API. 	26/08/2016	03/09/2016	1 semana
Entrega 4	<ul style="list-style-type: none"> • Criação do CRUD de pessoa; • Criação do CRUD de cartão; • Criação do CRUD de feriado; • Criação do CRUD de horário; • Criação do CRUD de período. 	03/09/2016	16/09/2016	1 semana
Entrega 5	<ul style="list-style-type: none"> • Montagem do sistema embarcado Raspberry pie. 	16/09/2016	23/09/2015	1 semana
Entrega 6	<ul style="list-style-type: none"> • Implementação do módulo cliente; • Implementação de possível backlog. 	23/09/2016	30/09/2016	1 semana
Entrega 7	<ul style="list-style-type: none"> • Integração com API; • Ajustes na documentação. 	30/09/2016	07/10/2016	1 semana
Entrega 8	<ul style="list-style-type: none"> • Realização de Testes no Sistema; • Implementação de possível backlog. 	07/10/2016	14/10/2016	1 semana

4.5. TECNOLOGIAS E LINGUAGEM UTILIZADAS NO PROJETO

a) Na modelagem do sistema

- Br Modelo: ferramenta voltada para modelagem de banco de dados relacional, com interface simples mas extremamente funcional. Permite alterações estruturais no modelo diante de novas decisões durante o processo de modelagem; promove a simplificação do esboço escondendo ou mostrando atributos de acordo com a necessidade do tipo de diagrama.
- *Navicat for PostgreSQL*: Ferramenta de modelagem de banco de dados com suporte ao banco de dados PostgreSQL.
- Draw.io: é uma aplicação de diagramação de modelagem de dados totalmente gratuita do *Google Drive*, livre de propagandas, registro e possui tradução para a língua portuguesa. Ferramenta usual, de fácil manuseio e intuitiva, possuindo grande vantagem de preferência em comparação aos *softwares* pagos.

b) No desenvolvimento do sistema

- Angular JS: possibilita o desenvolvimento rápido de *controllers*, com gerenciamento eficiente de rotas. Além disso, permite manipular a *view* em tempo real a partir das *controllers* de maneira muito prática, facilitando o desenvolvimento do projeto no geral.
- Node JS: possui uma plataforma ágil de desenvolvimento de cliente/servidor em *javascript*, permitindo a rápida instalação de módulos para integração com linguagens. Além disso, permite desenvolver interfaces utilizando html/css3. Executa localmente e tem compatibilidade com multiplataforma, além de possuir o *Electron Framework* que transforma uma aplicação *node.js* em uma espécie de executável local para exibição do projeto *node* como um aplicativo nativo em qualquer sistema operacional.
- *Ruby*: Linguagem *free* e *open source*, com foco na simplicidade e produtividade. Permite grande velocidade no desenvolvimento; código mais legível e de fácil

manutenção. A versão utilizada no projeto foi a 2.3.1.

- *Python*: permite o desenvolvimento rápido de módulos em *script* no *raspberry pi* devido a grande quantidade de bibliotecas prontas.
- *Python-shell*: módulo para *node js* que permite a integração rápida entre chamadas de *scripts python* e seu tratamento dentro do *node.js*.
- *Gems*: São bibliotecas empacotadas em um único arquivo que contém código Ruby desenvolvido para alguma finalidade. São muito comuns em projetos que utilizam a linguagem, pois muito deste código empacotado é reaproveitado em outros projetos diminuindo muito o retrabalho. Gems utilizadas:
 - *roda*: Responsável por criar uma interface REST simples e eficaz no servidor da Api onde recebe as requisições.
 - *puma*: Através do protocolo tcp/udp cria um servidor que pode ser acessado pelo navegador, foi utilizado no projeto pois consumir baixa memória em aplicações pequenas.
 - *pg*: Biblioteca que contém as orientações de conexão com o banco de dados PostgreSQL
 - *activerecord*: Cria uma camada de conexão com o banco de dados, com métodos específicos para criar, alterar, excluir e listar informações do banco de dados. Também é responsável por transformar o código Ruby em linguagem SQL.
 - *rake*: Gerenciador de tarefas que podem ser definidas pelo desenvolvedor, como por exemplo: *script rake* para criar o banco de dados.
 - *will_paginate*: Biblioteca para realizar paginação de listas no banco de dados.
 - *rack-cors*: Habilita a Api para receber requisições de sites confiáveis.
 - *state_machines-activerecord*: Responsável por gerenciar a máquina de estado. Oferece um controle de estado de cada objeto, podendo criar regras de transição de um estado e outro.

- *pg_search*: Habilita o banco de dados PostgreSQL com suporte a Full-text search.
- *dotenv*: Carrega instruções do arquivo .env para o ambiente de desenvolvimento.
- *hanami*: Framework REST Ruby modular. Por ser modular foi mais adequado para ser usado no admin.
- *spyke*: Permite a interação com serviços REST na forma do ActiveRecord.
- *request_store*: Armazenamento de variáveis. Permite guardar informações na sessão de uma forma mais segura e confiável.
- *multi_json*: Responsável por receber strings no formato JSON e converter para objetos JSON.

c) Na gerência e documentação do projeto

- Editor de Texto *Word*: oferece um painel de navegação fácil permitindo ver representações visuais de muitas das funções que você pode precisar, com ferramenta de detecção automática de erros, melhor configuração de numeração e marcação de páginas.
- *Google Drive*: ferramenta simples que permite o armazenamento em nuvem, fazendo com que dessa forma o projeto fique muito mais seguro e acessível. Além disso, permite edições em tempo real, podendo ser utilizada sem muitas dificuldades já que é uma plataforma extremamente intuitiva.

d) No hardware do Sistema

- Raspberry Pi



Figura 13:Placa de desenvolvimento Raspberry Pi Modelo v3

A placa de desenvolvimento Raspberry Pi 3 é um componente de hardware com diversos dispositivos integrados de alto desempenho com o tamanho próximo de um cartão de crédito. Com uma construção robusta poderoso suporte da comunidade de desenvolvedores, torna-se uma alternativa muito interessante para o desenvolvimento de protótipos de projetos embarcados, com ele é possível acessar portas GPIO, integrar displays, conectar dispositivos USB, alto-falantes entre outros dispositivos de maneira muito ágil. Esses módulos, que em sua grande maioria já possuem bibliotecas de integrações de código aberto, possibilitam então uma prototipação utilizando-se de diversas tecnologias integradas sem exigir um trabalho pesado de integração de hardware.

- Especificações:

ChipSet: Broadcom BCM2837

Processador: 4× ARM Cortex-A53, 1.2GHz

GPU: Broadcom VideoCore IV

Memória RAM: 1GB LPDDR2 (900 MHz)

Rede: 10/100 Ethernet, 2.4GHz 802.11n wireless

Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy

Armazenamento: microSD

GPIO: 40-pinos

Portas: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Fonte: <https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>

- MFRC522

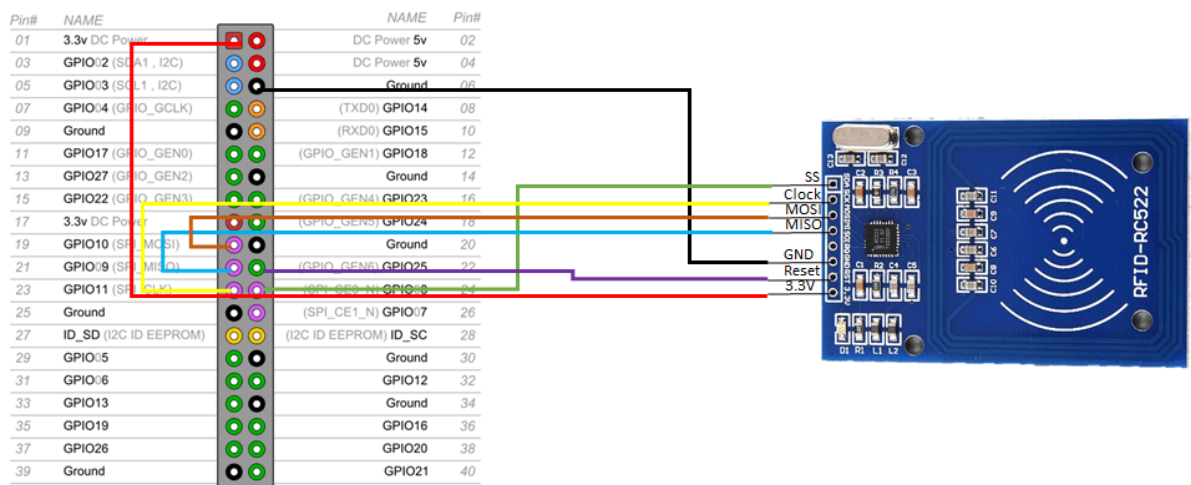


Figura 14: MFRC522 com Esquema de Conexão com Raspberry Pi Model v3

O MFRC522 é um componente leitor/gravador de informações de proximidade a 13.56MHz. Suporta os protocolos ISO/IEC 14443 A/MIFARE e NTAG. O transmissor interno do MFRC522 é capaz de transmitir e receber informações de cartões e transponders de proximidade sem nenhum circuito adicional. O receptor possui uma implementação robusta e eficiente para desmodular e decodificar os sinais de rádio-frequência compatíveis.

Fonte: <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>

- Display TouchScreen HDMI 5" GPIO - 52pi



Figura 15: Display TouchScreen HDMI 5" GPIO Fabricado Pela 52pi

Este modelo de display fabricado pela 52Pi, foi especialmente desenhada para o Raspberry Pi, a 52Pi disponibiliza com a documentação do dispositivo uma imagem de sistema para o RaspberryPi pré-configurado para operar com as portas GPIO, o que agiliza o processo de integração. A conexão com o raspberry Pi é extremamente fácil, bastando acoplar o módulo na placa e conectando o adaptador HDMI como na imagem:



Figura 16: Display TouchScreen HDMI 5" GPIO Acoplada ao Raspberry Pi

- Especificações:
Energia: 5V Power via GPIO from Raspberry pi

Tipo do display: 5" TFT LCD

Resolução: 800x480

Touchscreen: SPI Via Raspberry pi

Dimensões: 120mm x 74mm x 7mm

Fonte: <http://wiki.52pi.com/index.php/5-Inch-800x480-HDMI-GPIO-Touch-Screen> SKU:EP-0072

- ProtoBoard

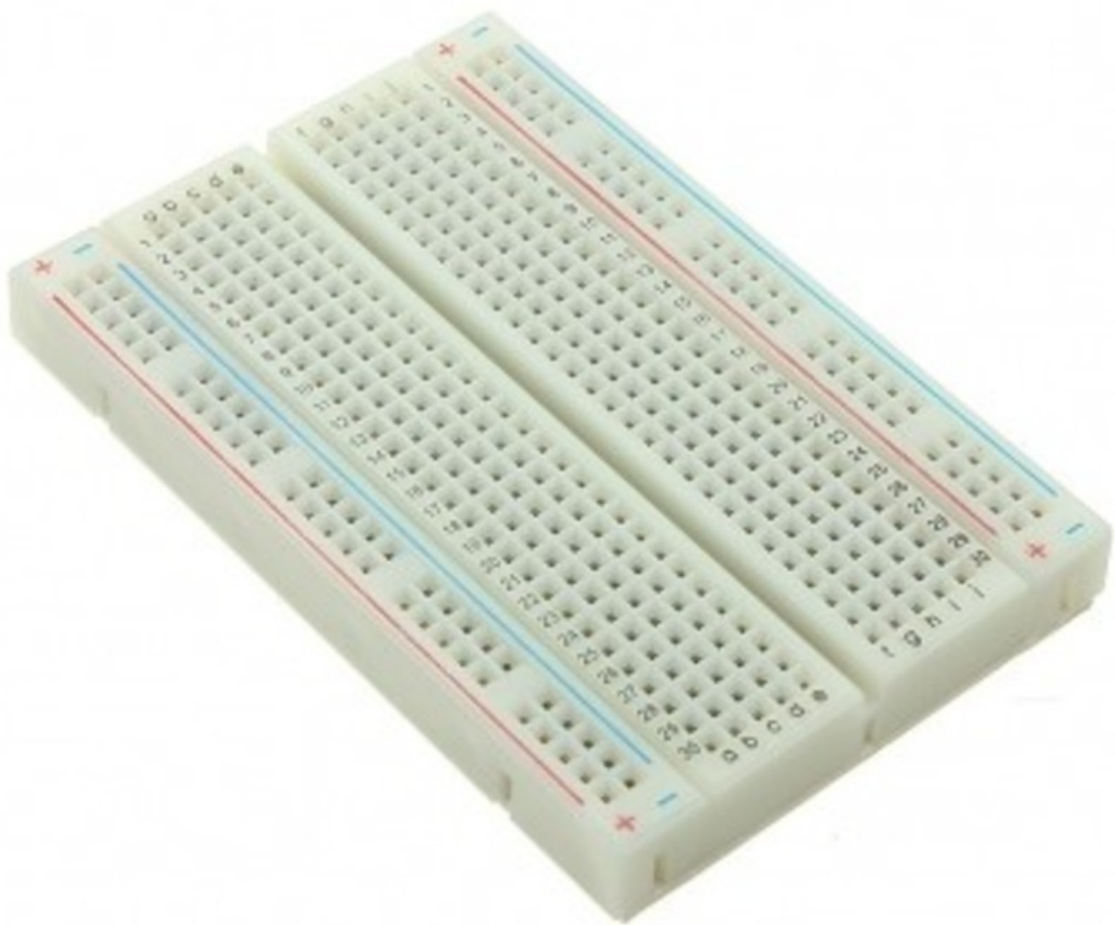


Figura 17: Placa de Ensaio de 400 Pontos

Também chamada de placa de ensaio ou matriz de contato, possibilita a construção de protótipos de circuitos eletrônicos sem a necessidade de efetuar soldas ou emendas, utilizando-se de “Jumpers” (Figura abaixo) Para comutar conexões entre pinos GPIO, periféricos, circuitos externos, LEDS, Resistores entre outros componentes eletrônicos.

- Jumpers



Figura 18: Jumpers de conexão

Os jumpers, são cabos pré-montados com ponteiros em três combinações: macho-macho, macho-fêmea e fêmea-fêmea, são utilizados para prototipar circuitos eletrônicos evitando soldagem, na maioria das vezes, em conjunto com uma ProtoBoard.

4.6. PROCESSO DE TESTES DO SISTEMA

No projeto foram aplicados os Testes Unitários e os Testes de Integração. Estes testes de um modo geral abrangem simulações de entradas e saídas de dados no caso de apenas uma classe para testes unitários e quando ocorre integração de mais classes são utilizados testes de integração. Os testes unitários e de integração são definidos pelo desenvolvedor sendo aplicados em uma ambiente de testes isolado do

ambiente de desenvolvimento e produção. Para ambas as situações de testes foi utilizado a ferramenta Rspec.

Abaixo seguem dois exemplos de testes aplicados no projeto.

Exemplo de teste de integração entre a classe “Api::Cards” e a classe “Card”: neste exemplo testamos a integração em todas as camadas da Api.

São testados os comportamentos para quando são recebidos parâmetros válidos ao criar um cartão é esperado que seja incluído no banco de dados o cartão e que o código de retorno seja 200. E recebendo parâmetros inválidos, esperamos que a Api retorne os erros de uma forma amigável e o código http 422.

```
require 'spec_helper'
```

```
RSpec.describe Api::Cards do
  let(:request_headers) { headers(token, params) }

  let(:token) { nil }
  let(:params) { { number: '123456', user_id: user.id } }
  let(:user) { create :user }

  describe 'POST create' do
    it 'should create card' do
      post '/cards', {}, request_headers
      expect(response.status).to eq 201
      expect(Card.count).to eq 1
    end

    context 'with invalid params' do
      let(:params) { { number: nil } }

      it 'should return error' do
        post '/cards', {}, request_headers
        expect(response.status).to eq 422
      end

      it 'should not create card' do
        post '/cards', {}, request_headers
        expect(Card.count).to eq 0
      end

      it 'should return errors' do
        post '/cards', {}, request_headers
        expect(json_response['errors']).to be_present
      end
    end
  end
end
```

Exemplo de teste unitário da classe “Card”: nestes exemplos, verificamos a validação das informações necessárias para persistir os dados de um cartão no banco de dados, se existe o relacionamento entre a classe Card e User e ainda se a classe PG:Search está sendo incluída na classe Card.

```

require 'spec_helper'

RSpec.describe Card, type: :model do
  it { expect(described_class.ancestors).to include(PgSearch) }

  it { should validate_presence_of(:number) }
  it { should validate_presence_of(:user) }
  it { should belong_to :user }

  describe '.search' do
    it 'should have method' do
      expect(Card).to respond_to(:search)
    end
  end
end

```

5. APRESENTAÇÃO DO SISTEMA – MANUAL DO USUÁRIO

Este capítulo tem o objetivo de explicar o funcionamento o sistema *Free Pass*.

Serão abordadas suas características e funcionalidades, telas do sistema, processos de utilização e descrições dos processos de utilização que dão uma melhor visão dos aspectos e objetivos das interfaces.

5.1. CLIENT

5.1.1. Acesso ao Sistema

○ LOGIN

A tela principal, permanece visível no dispositivo desde a inicialização do sistema e enquanto não houver interação com o aparelho, seja através da aproximação de um cartão ou interação através de toque na tela.

Através desta tela, é possível aproximar um cartão ou preencher através do teclado virtual o CPF e Senha do usuário para efetuar o acesso à dependência onde o equipamento encontra-se instalado.

Também é possível ir para a tela de Solicitar Acesso utilizando o botão "Solicite seu Acesso" que redirecionará o usuário para a tela de Solicitação de Acesso.

No canto inferior direito da tela, podemos observar um botão de "Menu", o qual leva à tela de Login para acesso ao Menu de Administrador do equipamento.



Figura 19 - Tela login

- SOLICITAÇÃO DE ACESSO

Na tela de Solicitação de Acesso, o usuário pode solicitar o seu cadastramento no sistema, informando os campos necessários: "CPF", "Nome", "E-Mail" e "Senha".

Ao solicitar um acesso, um administrador poderá ativar o cadastro, permitindo assim o acesso do usuário solicitante em uma próxima visita.

19:55:04
03/12/2016

← Solicitação de acesso

Preencha com seus dados para solicitar acesso ao administrador

CPF

Nome

E-Mail

Senha

Solicitar Acesso

Menu

Figura 20 - Tela solicitação de acesso

- LOGIN ADMINISTRADOR

Nesta tela, o Administrador, munido de sua senha, poderá efetuar o login para acessar o menu de administração do dispositivo. A tela de login é acessada através do botão "Menu" localizada no canto inferior direito na tela Principal.

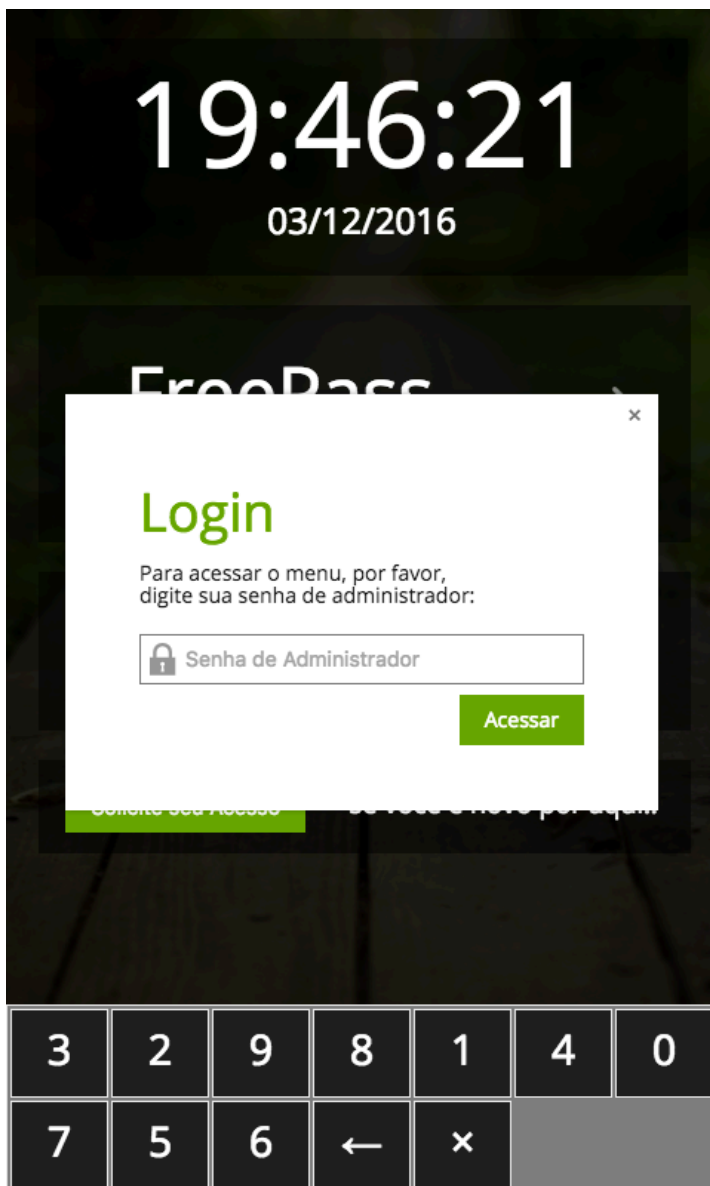


Figura 21 - Login do Administrador

5.1.2. CADASTROS

○ TELA USUÁRIO

A página usuários traz uma lista de usuários do sistema, paginada em resultados de 5, permitindo a busca de um usuário pelo nome ou documento.

Ao acionar o botão "Novo", é carregada a página de Edição de Usuário (PAG 05) limpa para criação de um novo usuário.

Se o administrador tocar sob um usuário da lista, será apresentada a tela de "Edição de Usuário" citada acima com as informações deste usuário carregadas para visualização/edição.

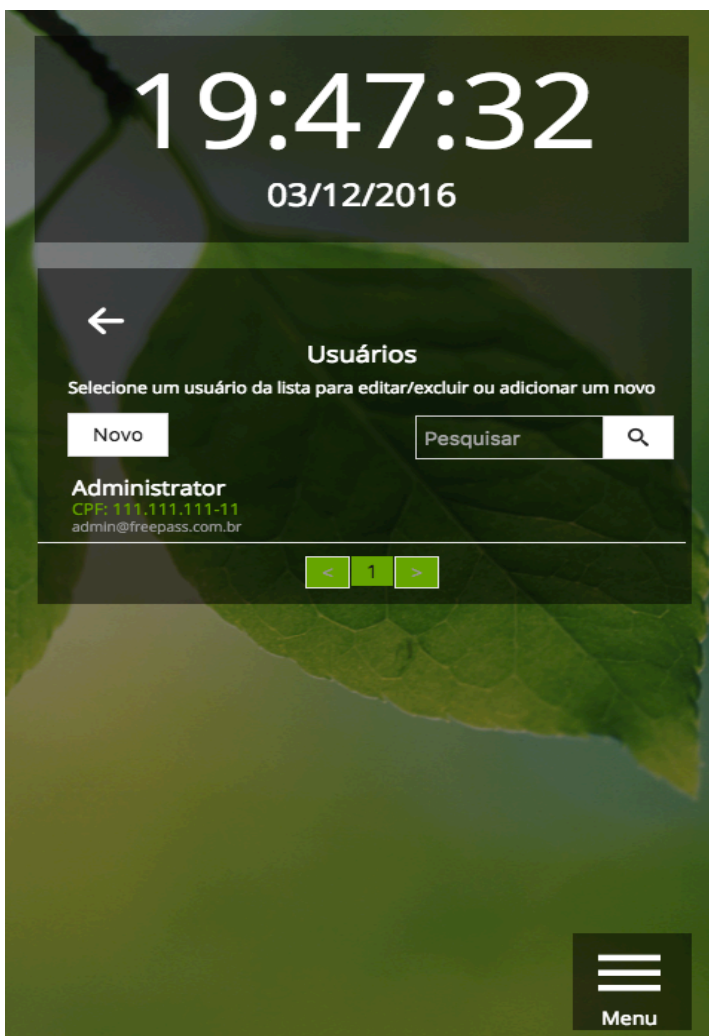


Figura 22 - Tela Usuários

- TELA CARTÕES/TAGS

A tela de tags, traz apenas um botão "Novo" para permitir o cadastramento de uma nova tag e possibilita a busca para visualização/edição de uma tag quando o usuário aproxima o cartão no sensor RFID.

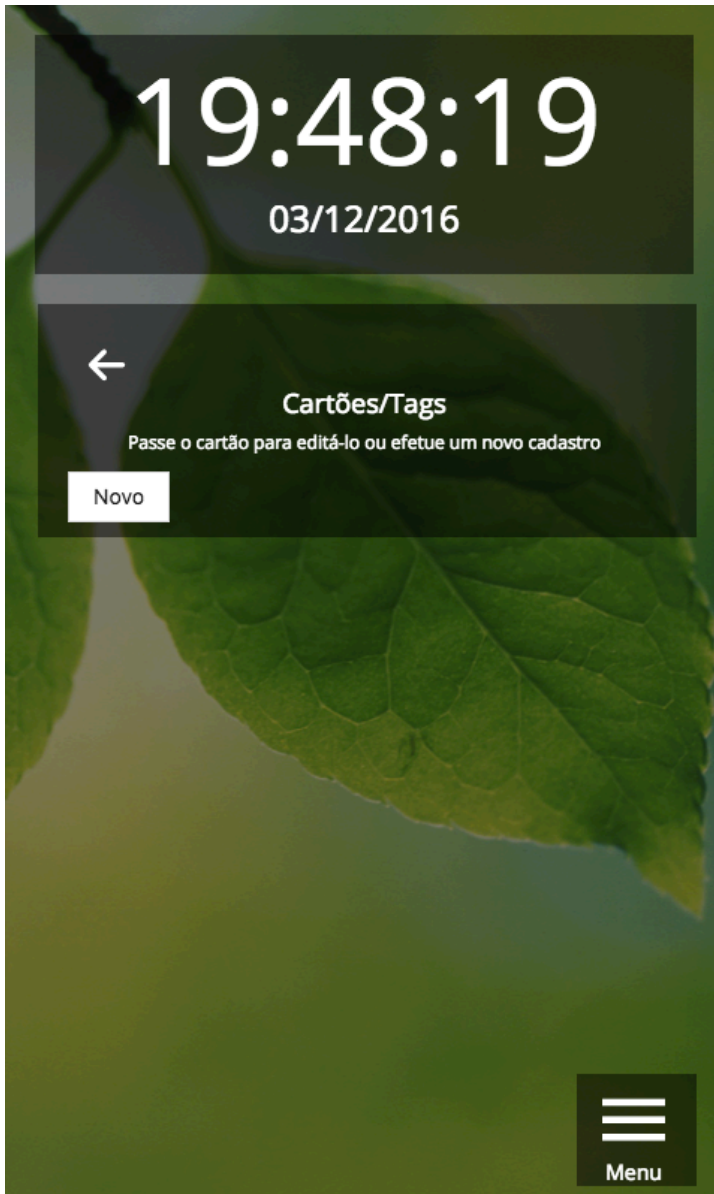


Figura 23 - Cartões/Tags

- EDITAR CARTÕES

A tela de edição de cartão permite que o administrador aproxime uma tag no sensor RFID para efetuar seu cadastramento no sistema, associando ao um usuário que deve ser escolhido através do campo de busca "Usuário".

Ao preencher o campo e acionar a pesquisa, o administrador visualizará uma lista de usuários que atender ao parâmetro da busca e que estão disponíveis para atribuição à tag/cartão em questão.

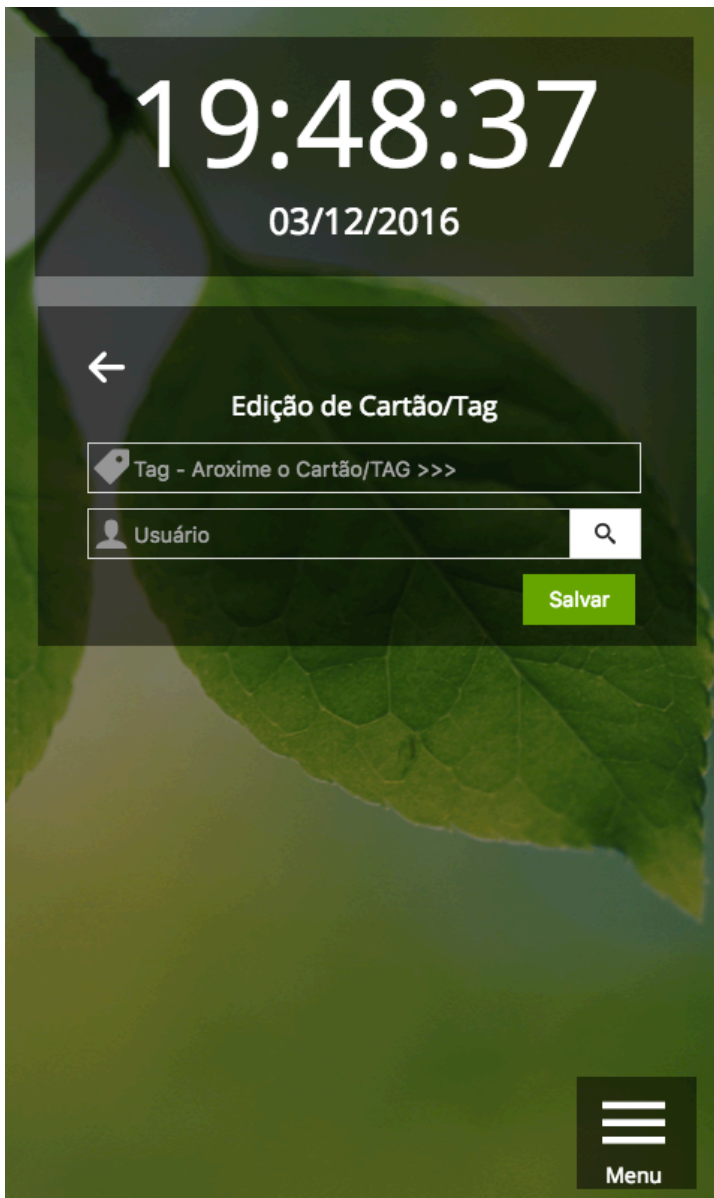


Figura 24 - Edição de Cartão/Tag

- SELECIONAR USUÁRIO

A tela de seleção de usuário é acionada quando um administrador efetua uma pesquisa de um usuário para atribuir à uma tag/cartão através da tela de "Edição de Cartão/Tag".

Essa tela exibe uma lista de usuários como resultado da busca e ao tocar sob um usuário da lista, o mesmo é carregado no campo "Usuário" da tela referenciada acima.

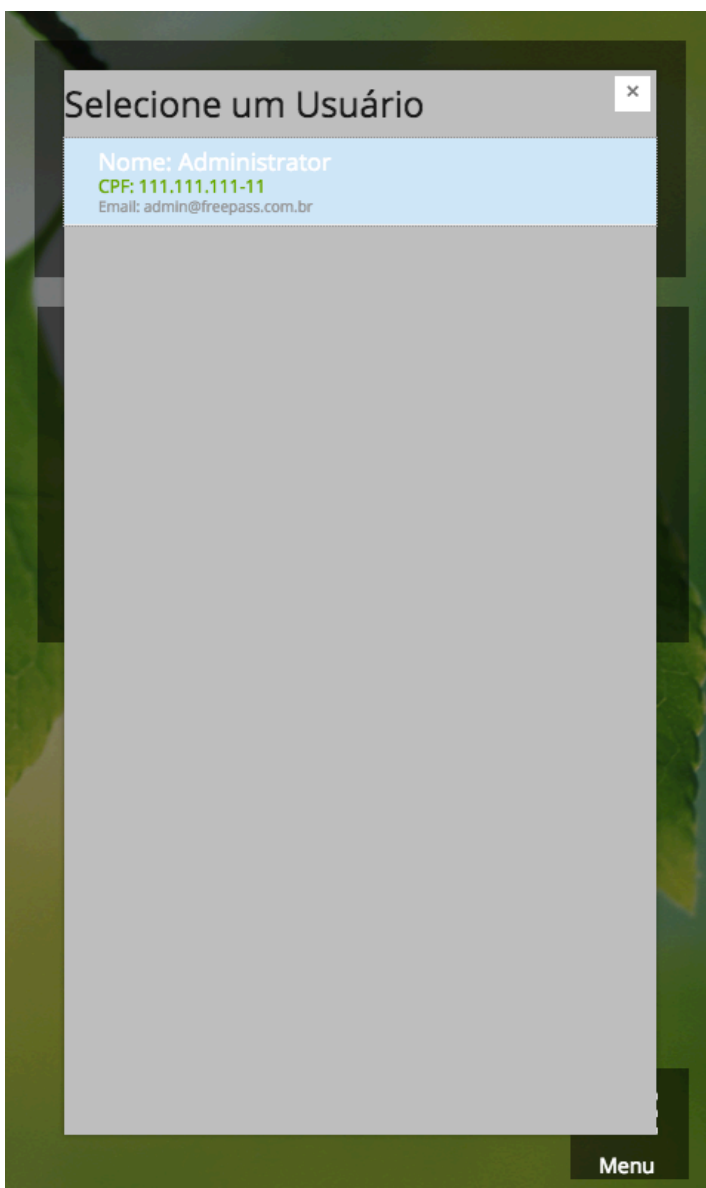


Figura 25 - Seleção de Usuário

- EDITAR USUÁRIO

Na tela de edição de usuário, o administrador poderá visualizar/editar as informações do usuário selecionado, ou suas próprias informações, caso tenha acessado a página através do botão "Meu Perfil" na página de "Menu do Administrador".



Figura 26 - Edição de usuário/Meu Perfil

- TELA HORÁRIOS

A página usuários traz uma lista de horários do sistema, paginada em resultados de 5, permitindo a busca de um horário pelo nome.

Ao acionar o botão "Novo", é carregada a página de Edição de Horário (DV 11) limpa para criação de um novo horário.

Se o administrador tocar sob um horário da lista, será apresentada a tela de "Edição de Horário" citada acima com as informações deste horário carregadas para visualização/edição da vigência deste horário.



Figura 27 –Tela Horários

- EDITAR HORÁRIOS

A tela de edição de horários permite que o administrador visualize, edite ou crie um horário no sistema informando o nome, horário de início e de horário de termino.

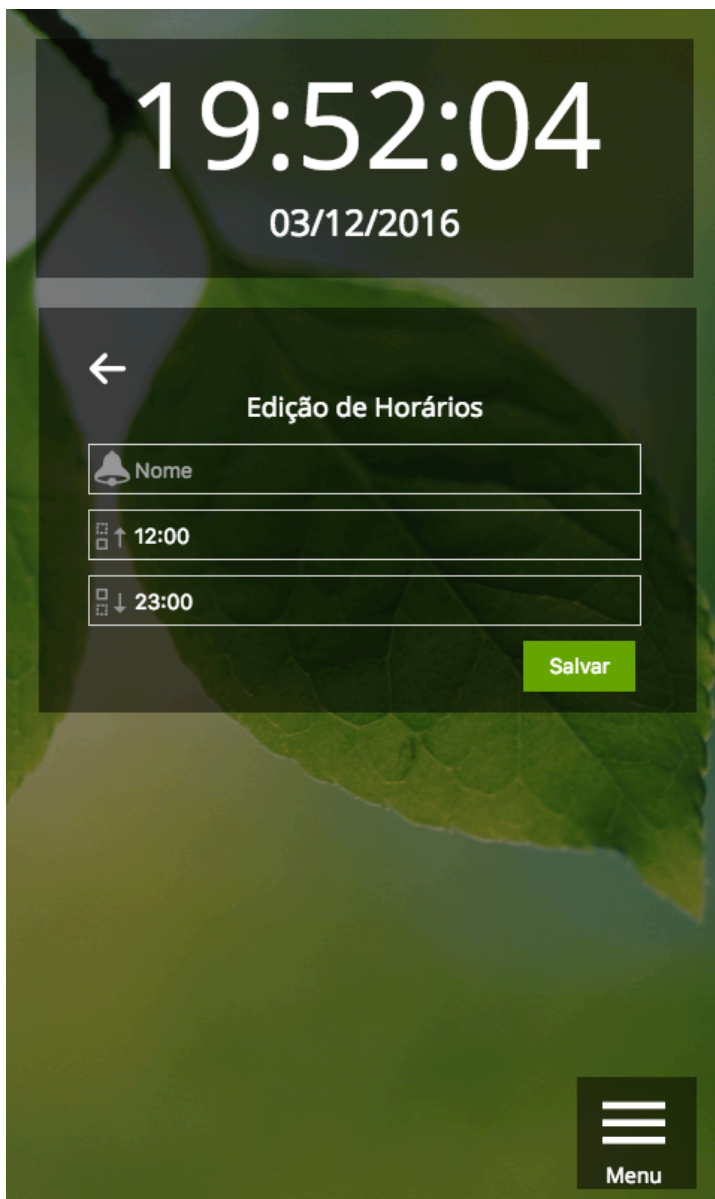


Figura 28 - Edição de horário

- TELA PERÍODOS

A página usuários traz uma lista de períodos do sistema, paginada em resultados de 5, permitindo a busca de um período pelo nome.

Ao acionar o botão "Novo", é carregada a página de Edição de Período limpa para criação de um novo período.

Se o administrador tocar sob um período da lista, será apresentada a tela de "Edição de Período" citada acima com as informações deste período carregadas para visualização/edição.

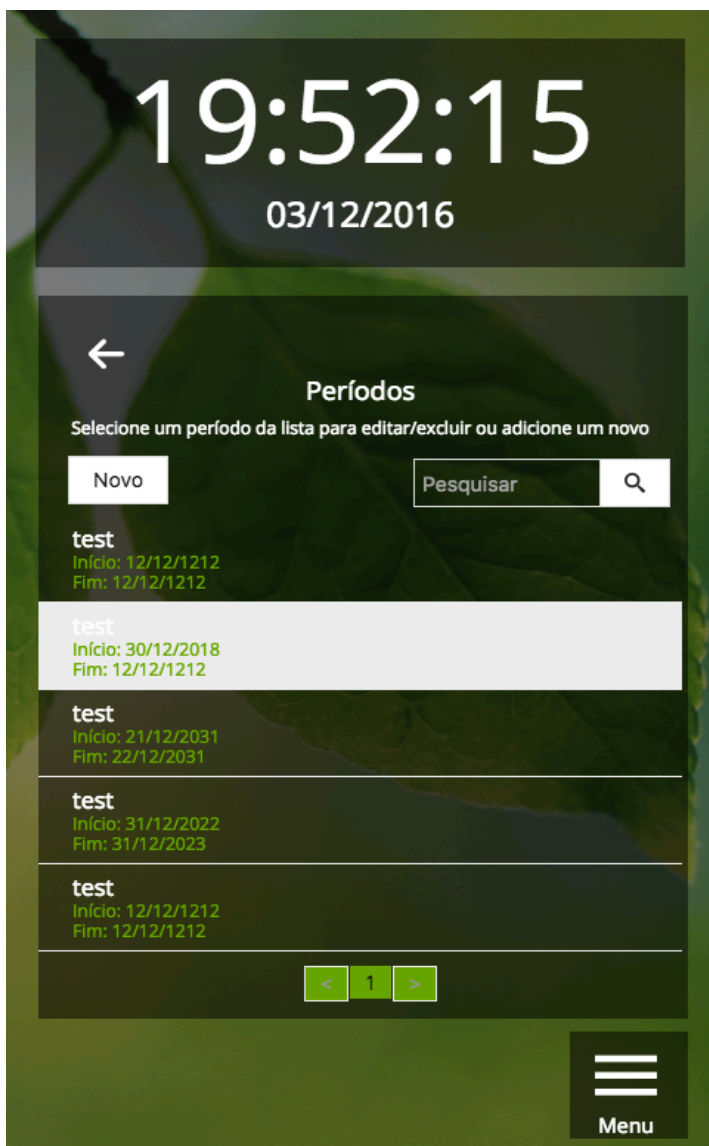


Figura 29 - Tela Períodos

- EDITAR PERÍODO

A tela de edição de períodos permite que o administrador visualize, edite ou crie um horário no sistema informando o nome, data de início e de data de termino da vigência deste período.



Figura 30 - Edição período

- TELA FERIADOS

A página usuários traz uma lista de feriados do sistema, paginada em resultados de 5, permitindo a busca de um feriado pelo nome.

Ao acionar o botão "Novo", é carregada a página de "Edição de Feriado" limpa para criação de um novo feriado.

Se o administrador tocar sob um feriado da lista, será apresentada a tela de "Edição de Feriado" citada acima com as informações deste feriado carregadas para visualização/edição.



Figura 31 - Tela Feriados

○ EDITAR FERIADO

A tela de edição de feriados permite que o administrador visualize, edite ou crie um feriado no sistema informando o nome e data do feriado em questão.



Figura 32 - Edição de feriado

5.1.3. CONFIGURAÇÕES

- TELA INFO

A tela Info, apresenta apenas o endereço IP em uso pelo dispositivo para possibilitar o acesso externo ao equipamento através de um navegador desktop/mobile para gerenciamento do mesmo.

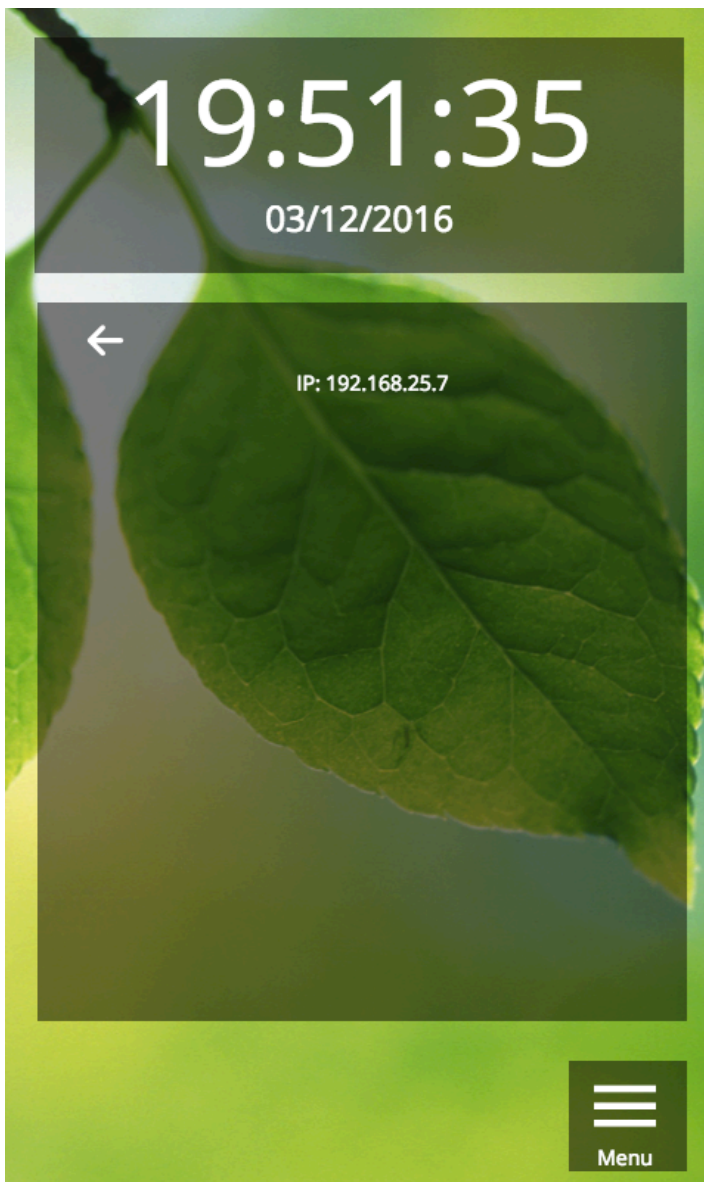


Figura 33 - Info

5.1.4. MENU

- MENU ADMINISTRADOR

Na tela de menu, o administrador verá os links para os cadastros no sistema.

Através desta tela, o administrador poderá:

- Visualizar/editar seu próprio perfil acessando a tela "Meu Perfil" através do botão "Meu Perfil";
- Acessar a tela de busca de Tag, para localizar, visualizar e editar Tags/Cartões
- Acessar a tela "Info" para visualizar o IP atual do dispositivo, o qual é utilizado para acesso da página de administração externamente na mesma rede.
- Acessar a página de listagem, visualização e alteração de horários, períodos e feriados.



Figura 34 - Menu do Administrador

5.2. TELAS ADMIN

5.2.1. ACESSO AO SISTEMA

○ LOGIN ADMINISTRADOR

Nesta tela, o Administrador, munido de sua senha, poderá efetuar o login para acessar o sistema de administração do dispositivo. A tela é apresentada sempre que o sistema verifica que o Administrador ainda não está logado.

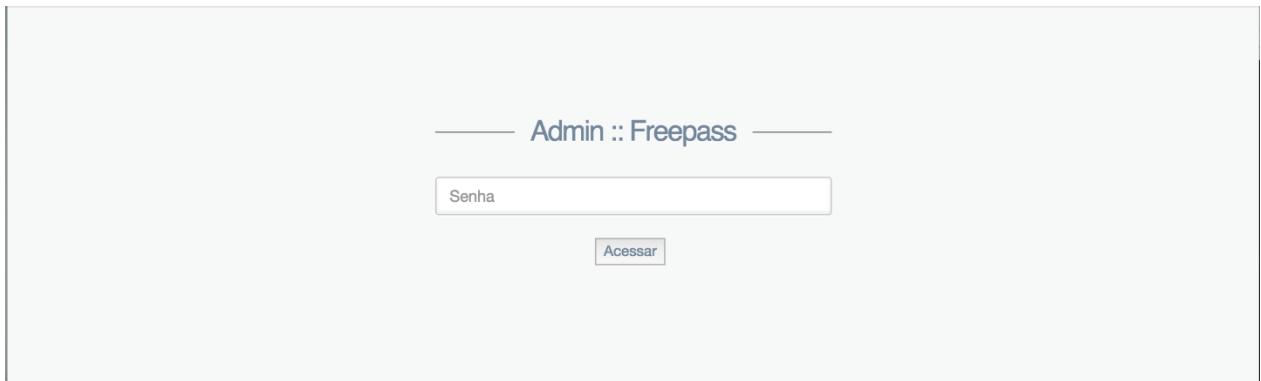
A imagem mostra a interface de login do administrador. No topo, há o texto "Admin :: Freepass" centralizado entre duas linhas horizontais. Abaixo disso, há um campo de entrada de texto com o rótulo "Senha". Logo abaixo do campo, há um botão com o texto "Acessar".

Figura 35 – Tela Login

5.2.2. CADASTROS

○ CADASTRAR USUÁRIO

Nesta tela o administrador pode realizar o cadastro de um novo usuário onde os dados obrigatórios são: CPF, Nome Completo, E-Mail, Senha e Nível, como opcional o Administrador poderá utilizar os campos de Períodos, Feriados e vincular algumas estratégias de acesso já definidas.

The image shows a web application interface for 'Freepass'. On the left is a dark blue sidebar with the 'Freepass' logo and a user profile for 'Administrador'. The main content area is titled 'Usuários' and contains a form titled 'Adicionar Usuário'. The form includes the following fields: CPF, Nome Completo, E-Mail, Senha, Telefone, Endereço, Períodos, and Feriados. A 'Nível' dropdown menu is set to 'Administrador'. At the bottom of the form are two buttons: 'Salvar' (green) and 'Cancelar' (blue).

Figura 36 – Cadastrar usuário

- LISTAR USUÁRIOS

São listados os usuários em grupos de 5, o Administrador tem uma opção de filtrar os usuários pelo primeiro nome. Nesta tela ainda é possível excluir um

usuário, visualizar suas informações, ativar ou inativar e excluir um usuário.

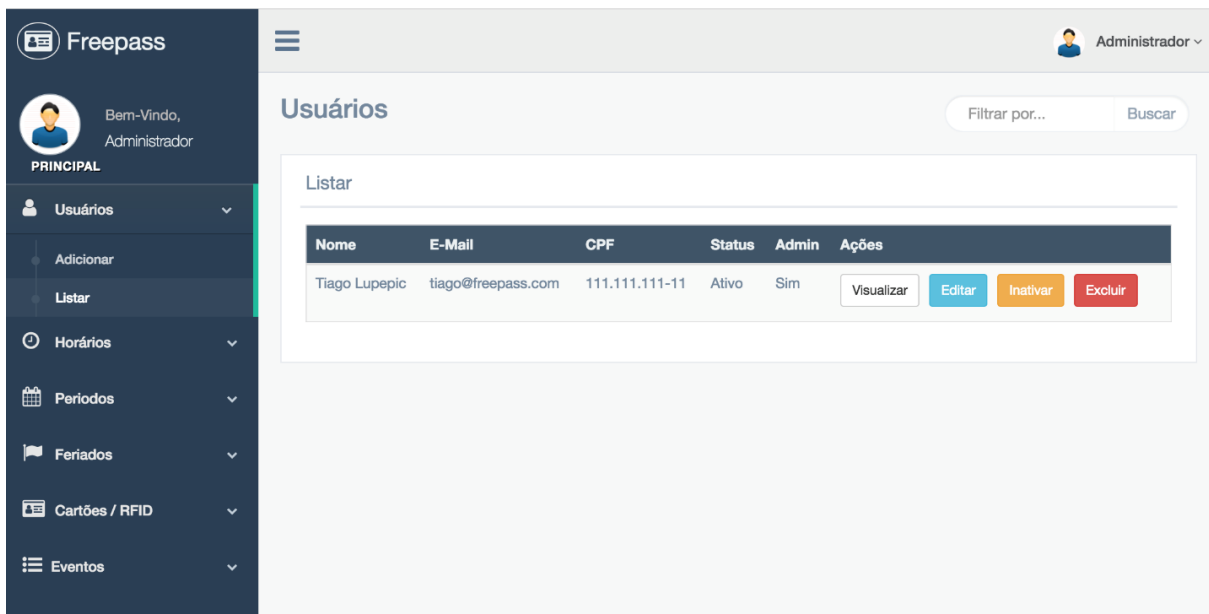


Figura 37 – Listar usuários

- VISUALIZAR USUÁRIO

São apresentadas as informações do usuário, a partir desta tela o Administrador do sistema tem as opções de voltar para a lista de usuários, editar as informações do usuário apresentado, inativar ou ativar e ainda excluir o usuário.

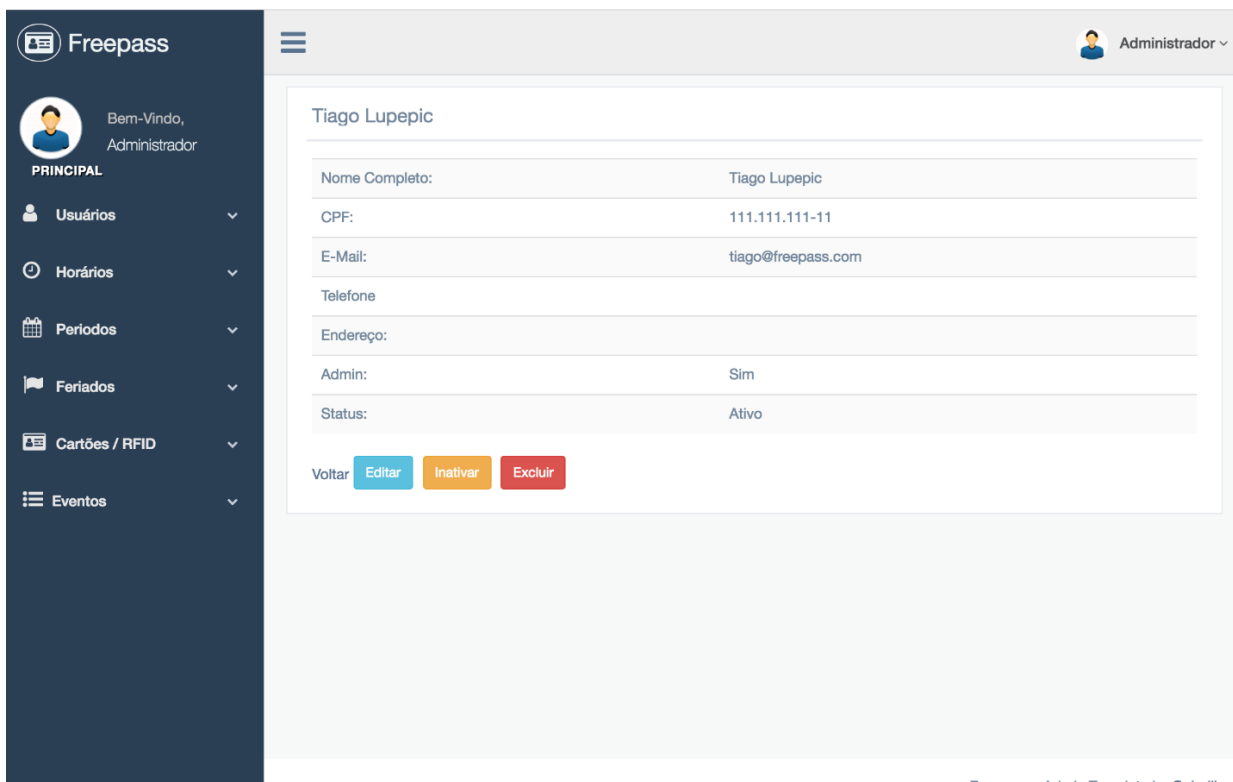


Figura 38 – Visualizar usuário

- EDITAR USUÁRIO

Ao acessar esta tela o Administrador terá opções de alterar informações do usuário como, CPF, Nome Completo, E-Mail, Endereço, Períodos, Feriados e alterar o nível de acesso. É possível controlar novos Períodos de acesso ou Incluir exceções de acesso para Feriados.

The screenshot displays the 'Freepass' user management interface. On the left is a dark blue sidebar with the 'Freepass' logo and a navigation menu including 'Usuários', 'Horários', 'Períodos', 'Feriados', 'Cartões / RFID', and 'Eventos'. The main content area is titled 'Usuários' and contains a form titled 'Alterar Usuário'. The form fields are: CPF (111.111.111-11), Nome Completo (Tiago), E-Mail (tiago@freepass.com), Telefone (empty), Endereço (empty), Períodos (empty), Feriados (empty), and Nível (Administrador). At the bottom of the form are 'Salvar' and 'Cancelar' buttons. The top right corner shows the user 'Administrador' with a dropdown arrow.

Figura 39 – Editar usuário

- ADICIONAR HORÁRIO

O Administrador poderá incluir novos horários de acesso, utilizando o menu principal é possível incluir um novo horário onde os campos obrigatórios são: Nome, Início e Fim, o formato apropriado para o horário é HH:MM, ainda opcionalmente é possível selecionar os dias da semana disponíveis.

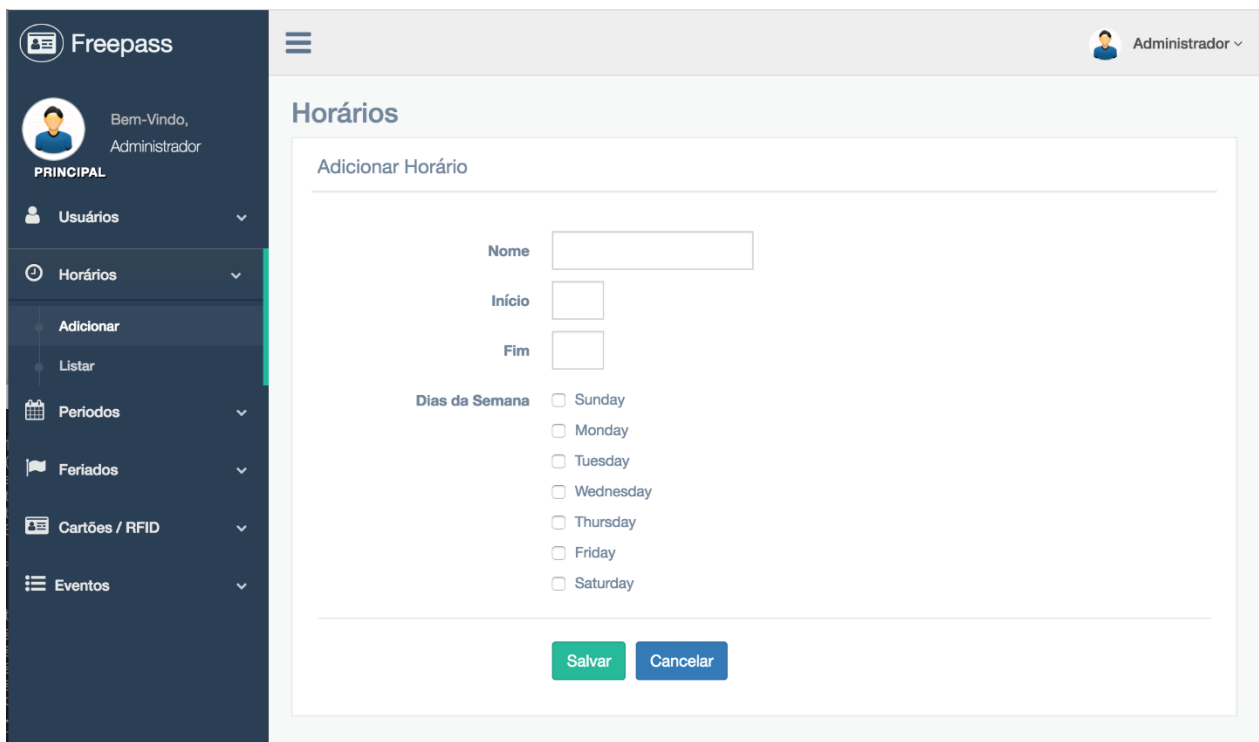


Figura 40 – Adicionar horário

- LISTAR HORÁRIOS

Nesta tela o Administrador poderá buscar os horários cadastrados, visualizar as informações, editar ou excluir.

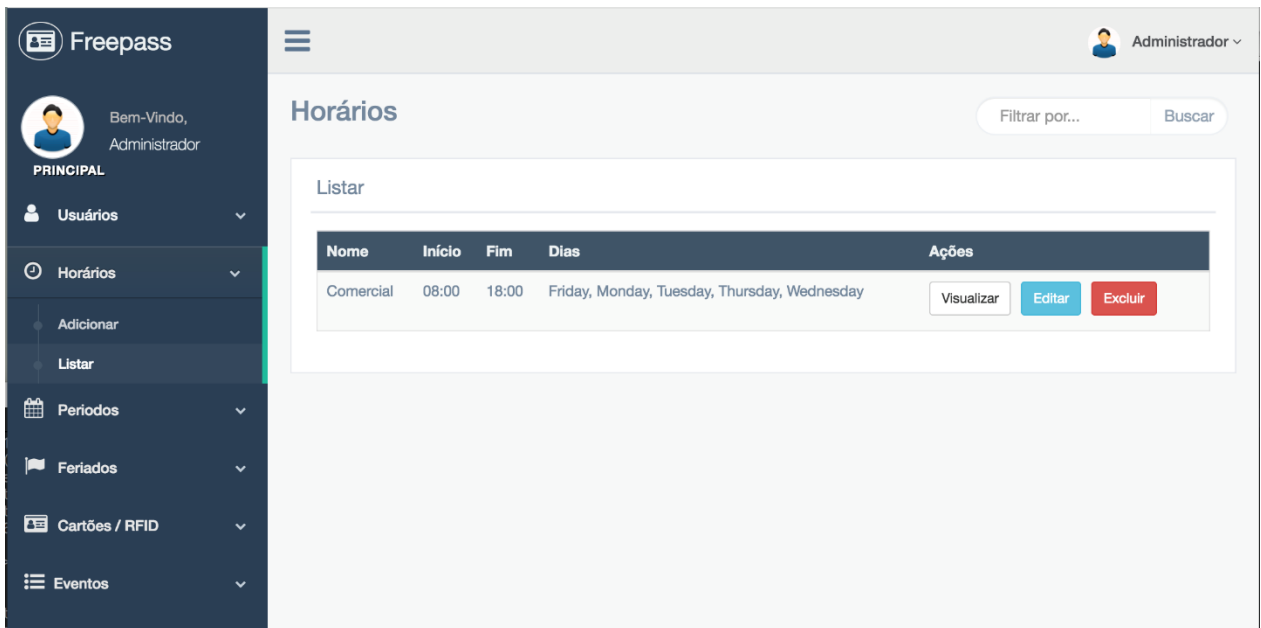


Figura 41 – Listar horários

○ VISUALIZAR HORÁRIO

Nesta tela é apresentada as informações do horário como Nome, Início, Fim e os Dias de Semana selecionados no cadastro, é possível através desta tela, voltar a listagem de horários, editar informações de horário e excluir.

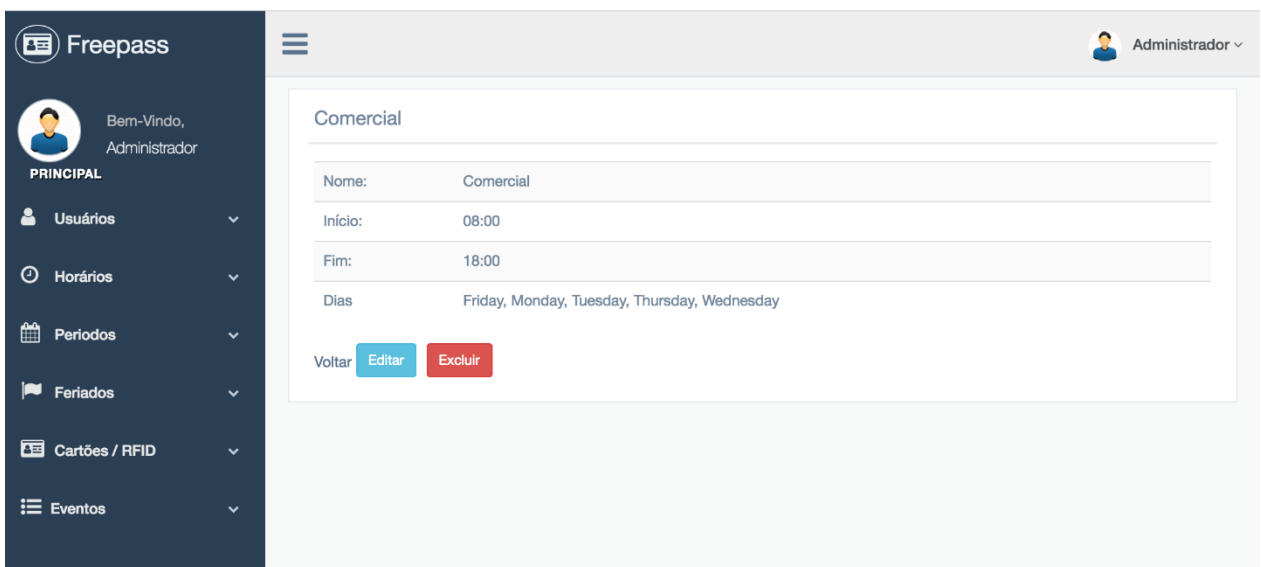


Figura 42 – Visualizar horários

- EDITAR HORÁRIO

Nesta tela é possível ser acessada através da busca de um horário ou na visualização do horário, nela é possível alterar regrad dos horários de início e fim ou ainda incluir ou excluir dias da semana.

The screenshot displays the 'Freepass' application interface. On the left is a dark blue sidebar with the 'PRINCIPAL' menu containing options like 'Usuários', 'Horários', 'Períodos', 'Feriados', 'Cartões / RFID', and 'Eventos'. The main content area is titled 'Horários' and contains a form titled 'Alterar Horário'. The form has three input fields: 'Nome' with the value 'Comercial', 'Início' with '09:00', and 'Fim' with '18:00'. Below these is a section for 'Dias da Semana' with checkboxes for Sunday through Saturday. Monday, Tuesday, Wednesday, Thursday, and Friday are checked. At the bottom of the form are two buttons: 'Salvar' (green) and 'Cancelar' (blue). The top right of the interface shows the user 'Administrador'.

Figura 43 – Editar horários

- ADICIONAR PERÍODO

Ao acessar esta tela através do menu principal o Administrador deverá preencher os campos Nome, Data Início, Data Fim e caso necessário selecionar os horários.

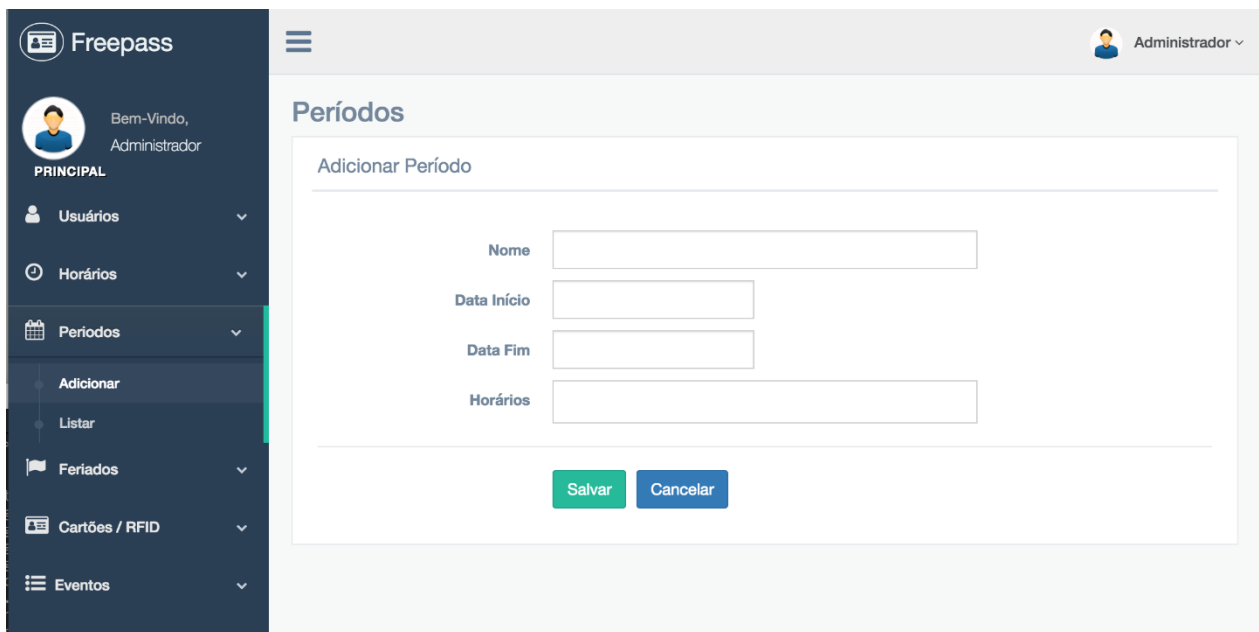


Figura 44 – Adicionar período

- LISTAR PERÍODOS

O acesso desta tela é feito através do menu principal, nesta tela são apresentados os períodos em grupos de 5, o Administrador poderá visualizar as informações completas do período, editar ou excluir.

The screenshot shows the 'Períodos' page in the Freepass system. The sidebar on the left contains the following menu items: PRINCIPAL, Usuários, Horários, Períodos (highlighted), Adicionar, Listar, Feriados, Cartões / RFID, and Eventos. The main content area has a header with 'Períodos', a search bar with 'Filtrar por...' and 'Buscar' buttons, and a table titled 'Listar'. The table has the following data:

Nome	Início	Fim	Horário(s)	Ações
1º Semestre	01/01/2016	31/07/2016	Comercial	Visualizar Editar Excluir

Figura 45 – Listar período

- VISUALIZAR PERÍODO

Nesta tela é apresentada todas as informações sobre o período acesso, é possível ao Administrador: voltar a listagem de períodos, editar o período ou ainda excluir o período.

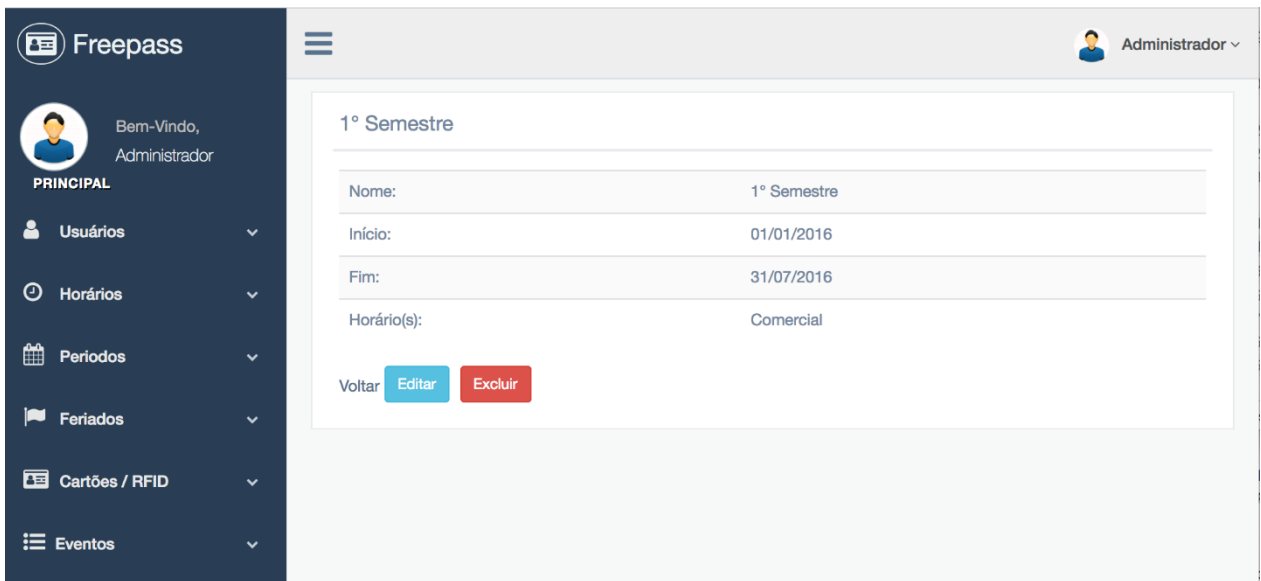


Figura 46 – Visualizar período

- EDITAR PERÍODO

A tela de editar poderá ser acessada buscando um período ou ainda na visualização, nela o Administrador poderá editar as informações do horário, vinculando novas permissões de horários, alterando nome ou data de início e data de fim.

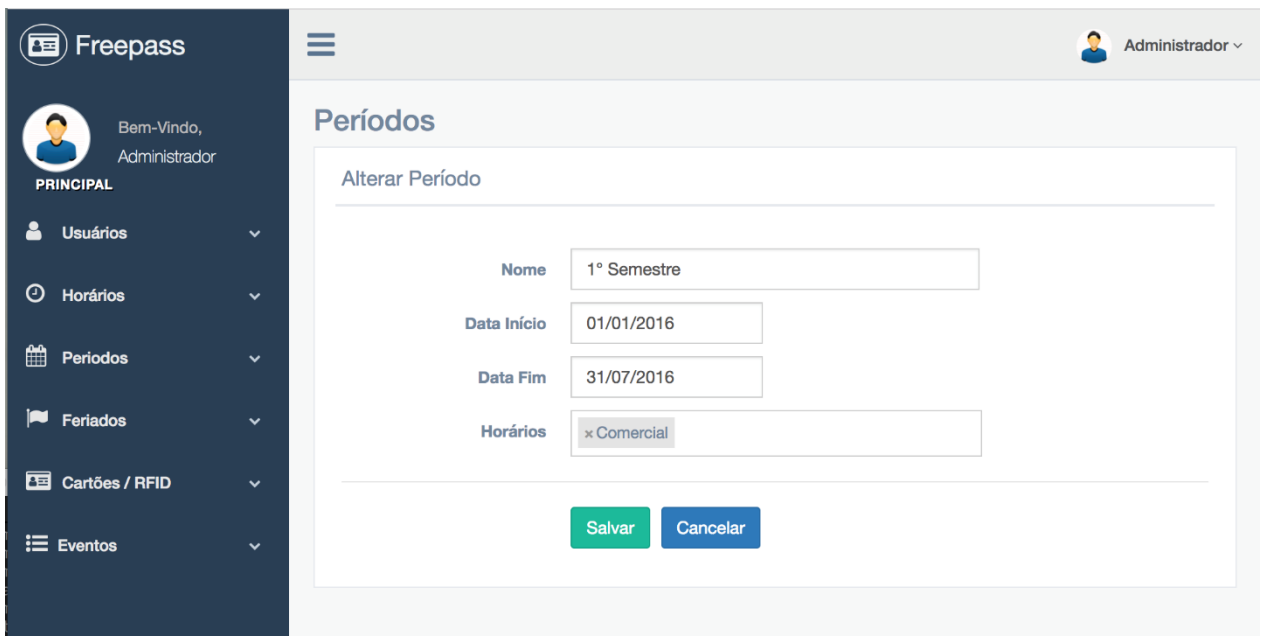


Figura 47 – Editar período

- ADICIONAR FERIADO

A tela de adicionar feriado é acessada através do menu principal, onde o Administrador poderá inserir um novo feriado com as informações de Nome e Data.

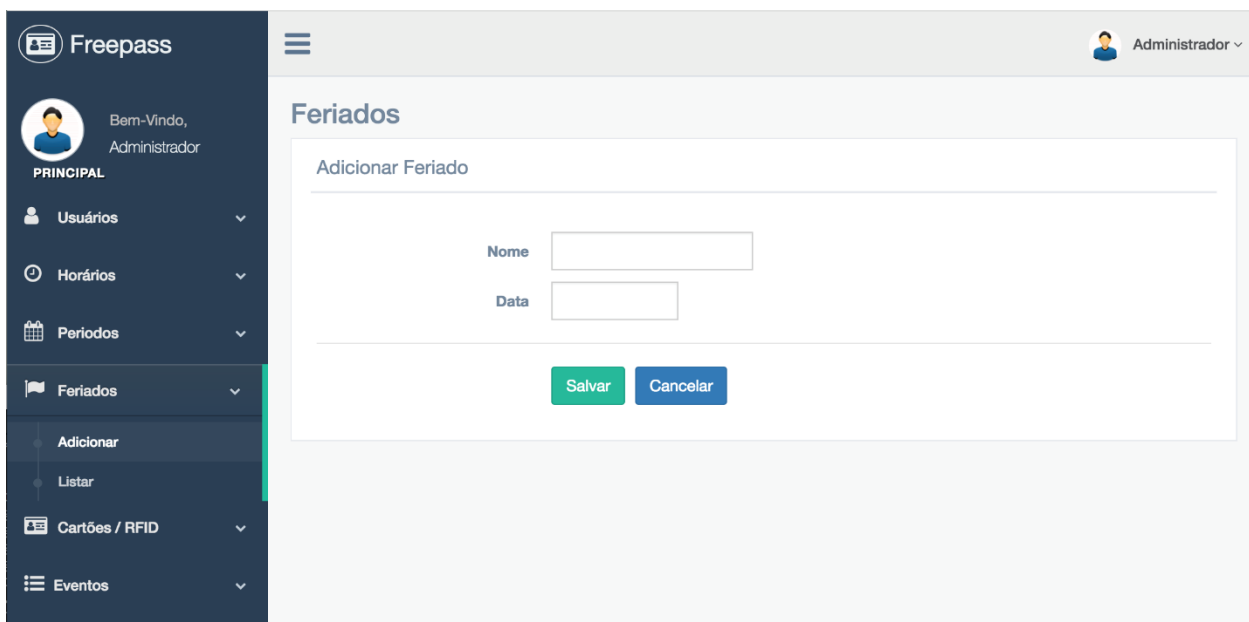


Figura 48 – Editar feriado

- LISTAR FERIADOS

Nesta tela o Administrador poderá visualizar as informações de cada feriado e ainda editar e excluir um feriado. Ainda é possível realizar uma busca de feriados através do Nome ou Data.

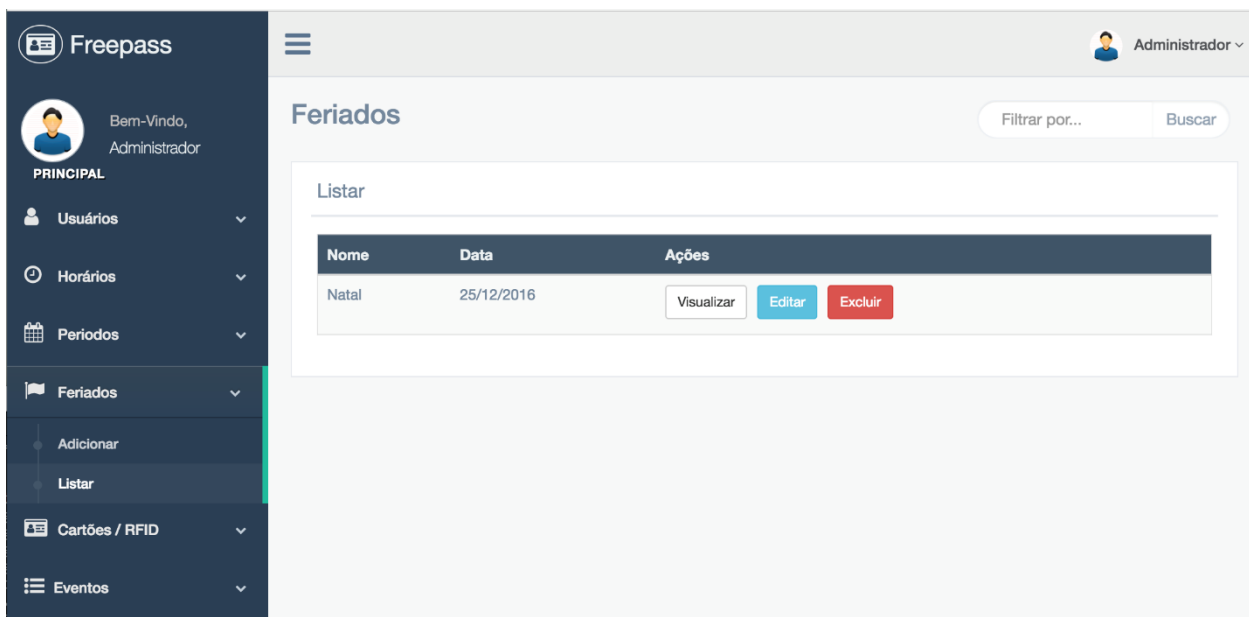


Figura 49 – Listar feriado

- VISUALIZAR FERIADO

Nesta tela o Administrador tem uma visão geral do Feriado onde é possível voltar para a lista de feriados, editar as informações do feriado ou excluir.

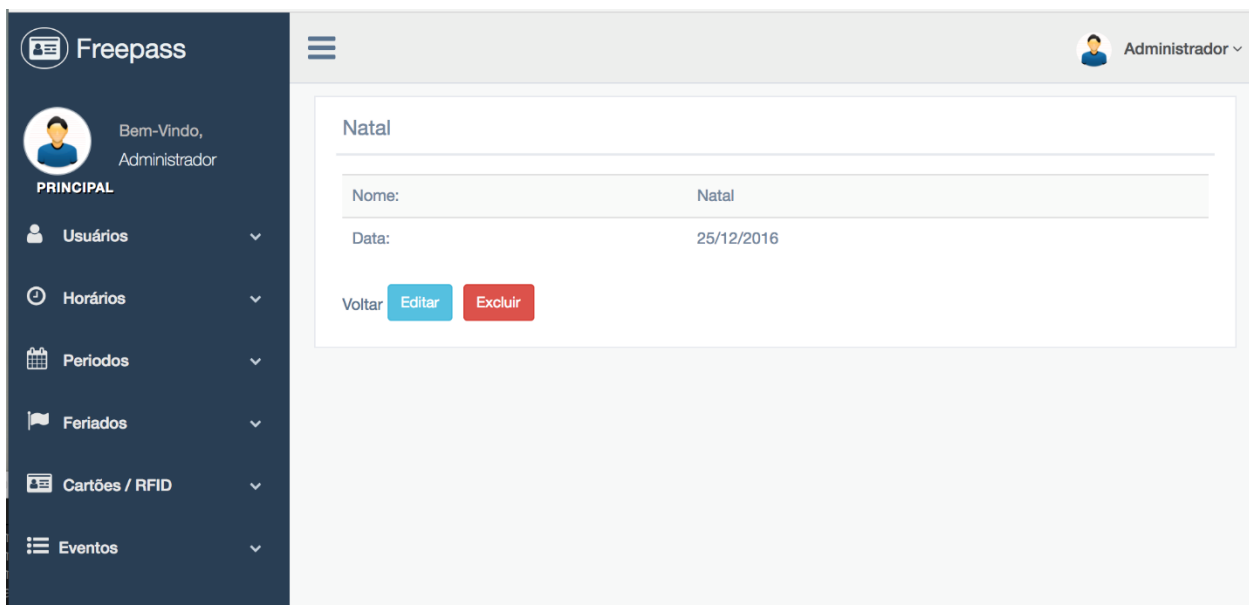


Figura 50 – Visualizar feriado

- EDITAR FERIADO

Para acessar a tela de alteração de feriados o Administrador deve acessar o Menu Principal, entrar no sub-menu “Listar” e buscar pelo feriado, nesta tela é possível de alterar as informações de Nome e Data.

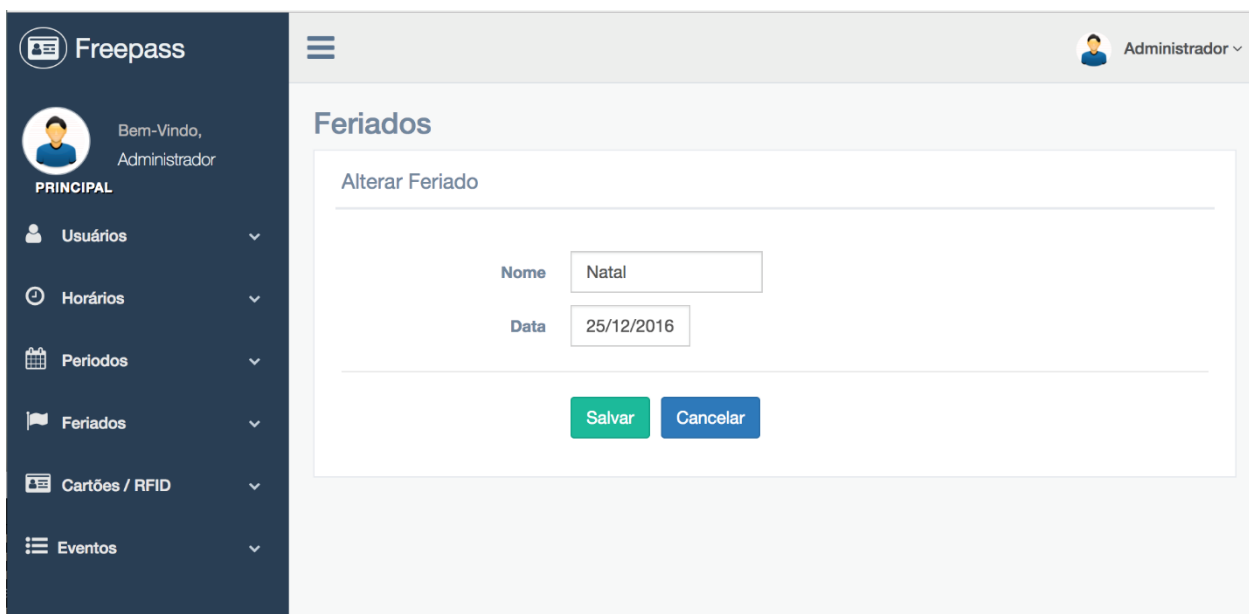


Figura 51 – Editar feriado

- ADICIONAR CARTÃO / RFID

Para adicionar um novo cartão o Administrador deve acessar o menu principal e logo após o sub-menu “Adicionar” nesta tela o Administrador deverá inserir o número referente ao Cartão/RFID e vincular a um usuário cadastrado do sistema.

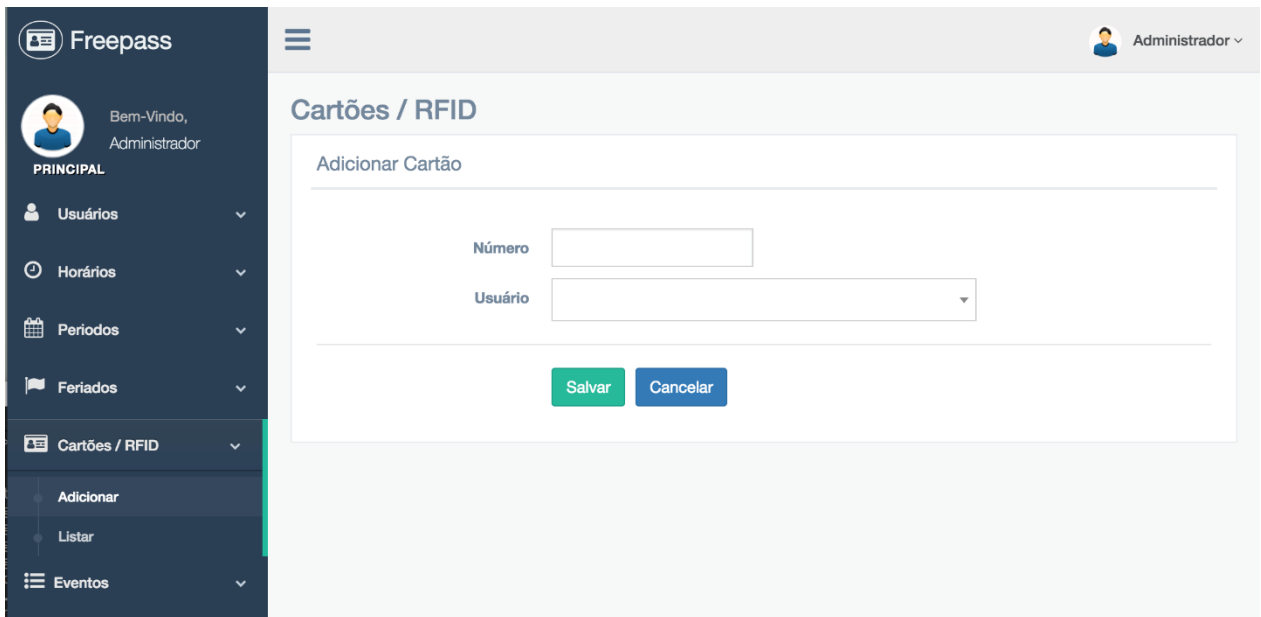


Figura 52 – Adicionar cartão / RFID

- LISTAR CARTÕES / RFID

Nesta tela o Administrador tem uma lista de cartões em grupo de 5, na tela é possível buscar cartões através do número, visualizar as informações do cartão, editar ou excluir o cartão.

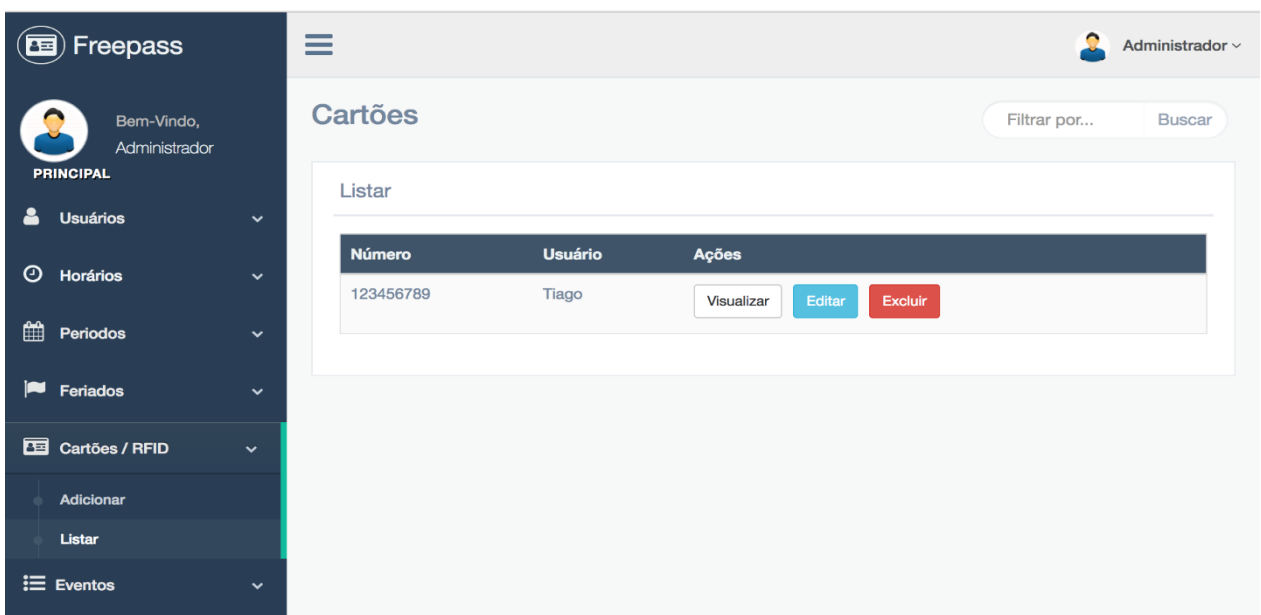


Figura 53 – Listar cartão / RFID

- VISUALIZAR CARTÃO / RFID

Para visualizar as informações de um cartão o Administrador deve acessar através do menu principal e logo após o sub-menu “Listar” e então encontrar o cartão a ser visualizado, nesta tela o usuário pode voltar a listagem de cartões, editar ou excluir.

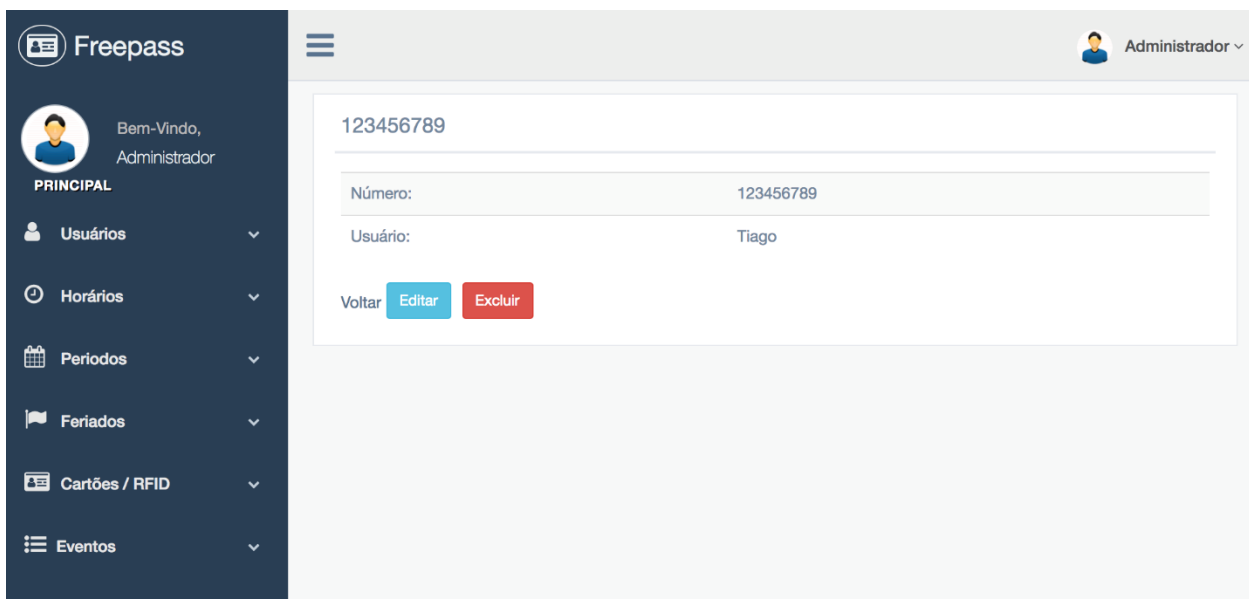


Figura 54 – Visualizar cartão / RFID

- EDITAR CARTÃO / RFID

Nesta tela o usuário poderá mudar o vínculo do usuário ao cartão, ou corrigir o número.

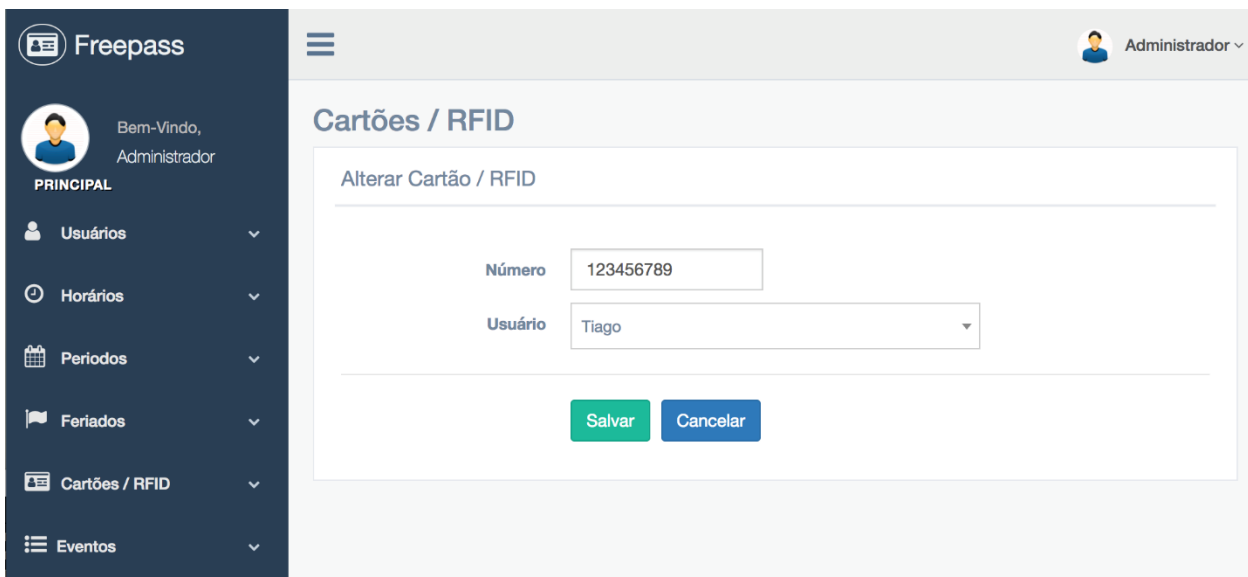


Figura 55 – Editar cartão / RFID

- LOG EVENTOS

Esta tela deverá ser acessada através do Menu Principal, na tela são apresentados os históricos de acesso de administradores e usuários, a listagem é ordenada de acordo com a data, os registros mais recentes são apresentados por primeiro.

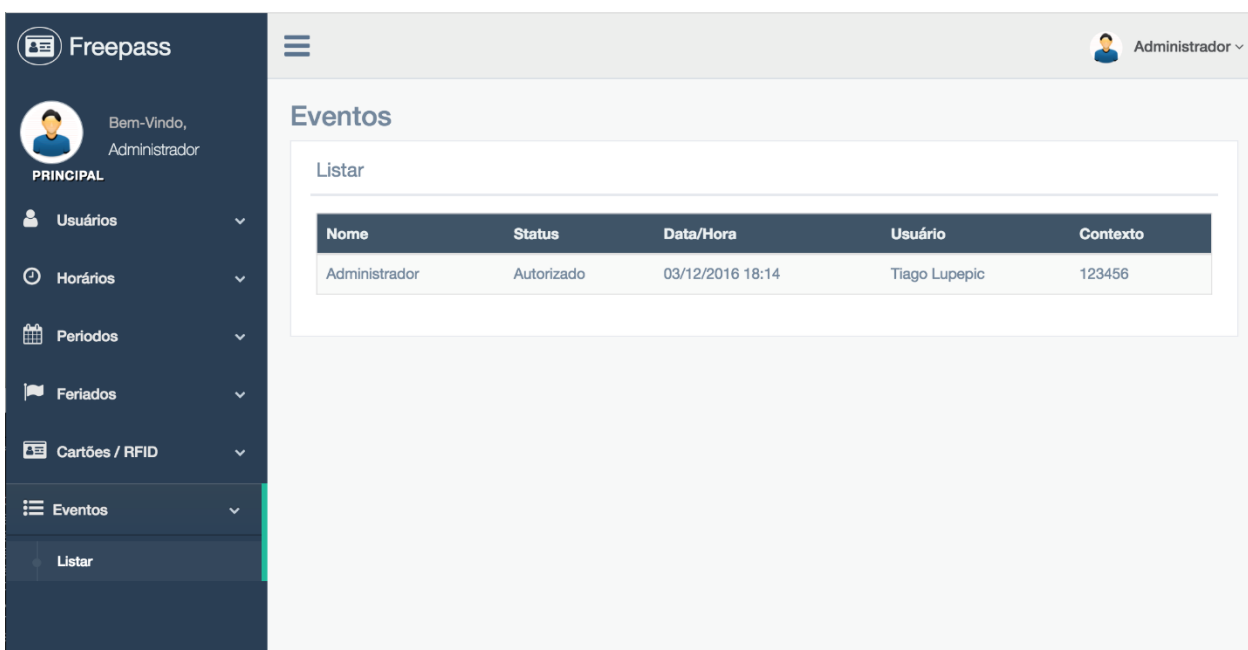


Figura 56 – Eventos

5.2.3. MENU

- MENU PRINCIPAL

O menu principal sempre fica visível ao lado esquerdo após o login de sucesso do Administrador. Nele o Administrador encontra as seguintes opções:

Usuários, Horários, Períodos, Feriados, Cartões / RFID, Eventos.



Figura 57 – Menu principal

6. CONCLUSÃO

Tendo em vista a segurança como um aspecto contemporâneo cada vez mais necessário, o presente trabalho buscou evidenciar o como a elaboração de um sistema simplificado de controle de acesso pode se mostrar uma alternativa eficaz e assertiva quanto ao problema de manter pessoas não autorizadas fora de determinados locais. Para isso, os autores desse estudo tiveram a oportunidade de colocar todo o conhecimento adquirido durante todo o período da graduação.

Sendo assim, a proposta primordial deste trabalho se mostra totalmente válida. Ela incide sobre a criação de um sistema denominado *FreePass*, onde suas validações de controle de acesso, feitas através da aproximação de código RFID, são feitas de maneira *on* e *off-line*. O sistema foi projetado para fazer isso de forma rápida, descomplicada sem deixar de se atentar a segurança.

Entre as maiores dificuldades que surgiram durante o desenvolvimento deste projeto, a que se destaca foi a configuração do sistema operacional para dar suporte as bibliotecas do RC522 (leitor do cartão). Como o sistema todo é embarcado e possuiu uma distribuição Linux não tão recente, tivemos que, fazer a compilação manual do Ruby para a mesma versão que foi utilizada em desenvolvimento, neste caso foi o passo mais fácil do que voltar a versão do Ruby que foi utilizado em desenvolvimento pois seria mais trabalhoso e demorado revisão todo o código da API e Administrador. Porém após uma série de pesquisas e testes foram resolvidas tornando nosso projeto funcional e pronto pra operação, atendendo todos os objetivos que estipulamos no início do mesmo, e as exigências da empresa entrevistada.

Será aplicado ainda mais melhorias neste projeto as quais serão denominadas como trabalhos futuros, que não irão alterar a estrutura final do projeto. Entre eles, está o controle do acesso separada por departamentos e controle de acesso por impressão digital incluso. Tais melhorias tem o intuito de aumentar a segurança do usuário desse sistema de forma simplificada tornando-o cada vez mais aprimorado e adaptável às necessidades de quem o utiliza.

REFERÊNCIAS

ANDRADE , Fernando de S. **Sistemas Embarcados**, 2a ed., São Paulo: Editora Érica, 2010.

BOOCH, Grady, RUMBAUGH, James, JACOBSON, Ivar. **UML: Guia do Usuário**. 2ª Ed., Rio de Janeiro: Editora Campus, 2005.

CRUZ, Fábio. **Scrum e Agile em Projetos Guia Completo**, São Paulo: Editora Brasport, 2015.

CUNHA, Alessandro. **Sistemas Embarcados**, Revista Saber Eletrônica, p. 414, São Paulo: Editora Saber, 2007.

PEREIRA, Caio Mário da Silva. **Instituições de direito civil**, volume IV, 2ª edição, p. 72., Rio de Janeiro: Forense, 1992.

GLOVER, B.; BHATT, H. **Fundamentos de RFID**, Rio de Janeiro: Alta Books, 2007.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**, São Paulo: Novatec, 2011.

MARÇAL, Ana Sofia. et al. **Entendendo Scrum para gerenciar projetos de forma ágil**. Recife, 2007. Disponível em <http://www.cesar.org.br/files/file/SCRUM_MundoPM-Abril-Maio-2007.pdf>. Acesso em: 03/04/2017.

MELO, Ana Cristina. **Desenvolvendo aplicações com UML 2.2 do conceito à implementação**. 3º edição. Rio de Janeiro: Brasport, 2010.

MILANI, Fabiano . **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre: Bookman Editora, 2014.

NAKAMURA, Emilio T. GEUS, Paulo Lício de. **Segurança de Redes em Ambientes Corporativos**. São Paulo: Editora Novatec. 2007.

WAZLAWICK, R. **Engenharia de software: conceitos e práticas**, Rio de Janeiro: Editora Elsevier, 2013.

OBELHEIRO, Rafael. **Controle de Acesso**. Disponível em: <<http://www2.joinville.udesc.br/~dcc2rro/seg-bcc/2009.1/resumo-controle-acesso.pdf>>

REIS, Claiton . **Sistemas Operacionais para Sistemas Embarcados**, Tutorial, Bahia : Editora EDUFBA, 2004.

SILVA, Bruno. At all. **Minicursos do XIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais — SBSeg 2013**. Disponível em: <<http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=cseseg:2013-sbseg-mc3.pdf>> acesso em 15/05/2017.

YOSHIMA, R. **Gerenciamento de Projetos com Scrum**. 2007. Disponível em <<http://www.aspercom.com.br/ead/mod/resource/view.php?id=245>> Acesso em 15/02/2017

ZURITA, Marcos E. P. V. **Projeto de Sistemas Embarcados**. 2014. Disponível em <https://www.researchgate.net/publication/267298521_Projeto_de_Sistemas_Embarcados> Acesso em 08/08/2016.

APÊNDICE A

QUESTIONÁRIO PARA LEVANTAMENTO DE REQUISITOS

QUESTIONÁRIO PARA LEVANTAMENTO DE REQUISITOS

- Você considera a falta de controle e gerenciamento de acesso um problema se não implementado um sistema de segurança?
R- SIM
- Qual seria o papel de um sistema de controle e gerenciamento de acesso na sua empresa?
R- GARANTIR A SEGURANÇA, MONITORAMENTO DO FLUXO, HORÁRIOS ENTRADA SAÍDA
- Você acha importante obter relatórios de entrada/saída dos funcionários por data e horário?
R- SIM
- É interessante gerenciar o tempo que o funcionário permanece na empresa?
R- SIM
- Nos feriados, há a necessidade de manter acesso livre dos funcionários ao ambiente da empresa?
R- SIM
- Quem irá utilizar o sistema?
R- FUNCIONÁRIOS, VISITANTES, CLIENTES, PRESTADORES SERV.
- Como deve ser feita a solicitação do acesso?
R- VIA SISTEMA, VIA EMAIL, CONTATO DIRETO ADM.
- Se implementado um sistema de controle de acesso, como deveria ser o processo de cadastramento de um novo usuário e quais informações seriam necessárias?
R- EMAIL, CPF, NOME, OBR - TELF. EMPRESA, DT NASC. OPC.
- Seria interessante utilizar cartões para realizar o controle de acesso?
R- SIM
- Quais os dados necessários para o cadastro de cartões?
R-
- Como deverá ser o processo de habilitação/deshabilitação de um usuário no sistema?
R- ERRA X VEZES A SEQUÊNCIA LONGO PERÍODO SEM ACESSAR
POR HOR. TEMPO DETERMINADO

Figura 58 – Entrevista Levantamento de requisitos Frente

- Ao seu ver, logs de entrada e saída de visitantes são de importância vital?
R-SIM
- Buscar um usuário num sistema de controle de acesso para editar os dados pessoais, datas e horários de acesso seriam requisito para o sistema?
R-SIM
- Seria interessante fazer também buscas por períodos (horas, dias, meses e feriados)?
R-SIM
- É interessante armazenar um histórico de cadastros?
R-SIM
- Quais devem ser as validações que o sistema deve possuir?
R-SENHA, CARTÃO, PERÍODO, HORÁRIO, ATIVO
- Em que ambiente o sistema será operado?
R- INTERNO E EXTERNO (SKLAS E ACESSO PRINCIPAL)
- O sistema deve possuir níveis de permissão de acesso? Quais?
R-SIM
- ADMIN
PERMANENTES ← SETOR
PER. HOR
TEMPORARIOS ← SETOR
PER. HOR

Figura 59 – Entrevista Levantamento de requisitos Verso

APÊNDICE B

CARTA DE ACEITA DA EMPRESA GO RECEBA FÁCIL



UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Prezados Gestores,

Venho por meio desta declaração informar que **Andrew Carvalho Barbosa Fernandes, Erick Douglas Salustiano, Giampaolo Pierozan e Tiago Lupepic** são alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná.

Neste segundo semestre de 2016 os acadêmicos estão desenvolvendo o Trabalho de Conclusão de Curso, cujo objetivo é o desenvolvimento de um **SISTEMA DE CONTROLE DE ACESSO UTILIZANDO CARTÕES DE PROXIMIDADE RFID – Free Pass**

Para tanto, é necessário uma entrevista com gestores e profissionais de empresas que utilizam algum sistema de controle (manual ou informatizado) para identificar os requisitos essenciais para o desenvolvimento do sistema, a fim de garantir um projeto de software que atenda as necessidades do mercado.

Então, solicitamos a colaboração de você(s), gestor(es) e/ou profissional (is) para responder as perguntas dos acadêmicos e, assim, ajudar no desenvolvimento desse projeto.

Curitiba, 27 de outubro de 2016.

Atenciosamente,

Handwritten signature of Prof. Me. Andreia de Jesus in blue ink.

Prof. Me. Andreia de Jesus
Professora Orientadora do Projeto

Prof.ª Andreia de Jesus
UFPR
SIAPE 1024538

Aceitamos contribuir com o projeto e autorizamos mencionar o nosso(s) nome(s) e o da empresa na documentação do sistema.

Nome da Empresa: GO RECEBA FÁCIL

Nome completo do responsável pela entrevista: MARCELO RAZELA

Assinatura do Responsável pela entrevista:

Handwritten signature of Marcelo Razela in blue ink.
Data da Entrevista: _____

Figura 60 – Entrevista Levantamento de requisitos

APÊNDICE C

MODELO RELACIONAL DO BANCO DE DADOS

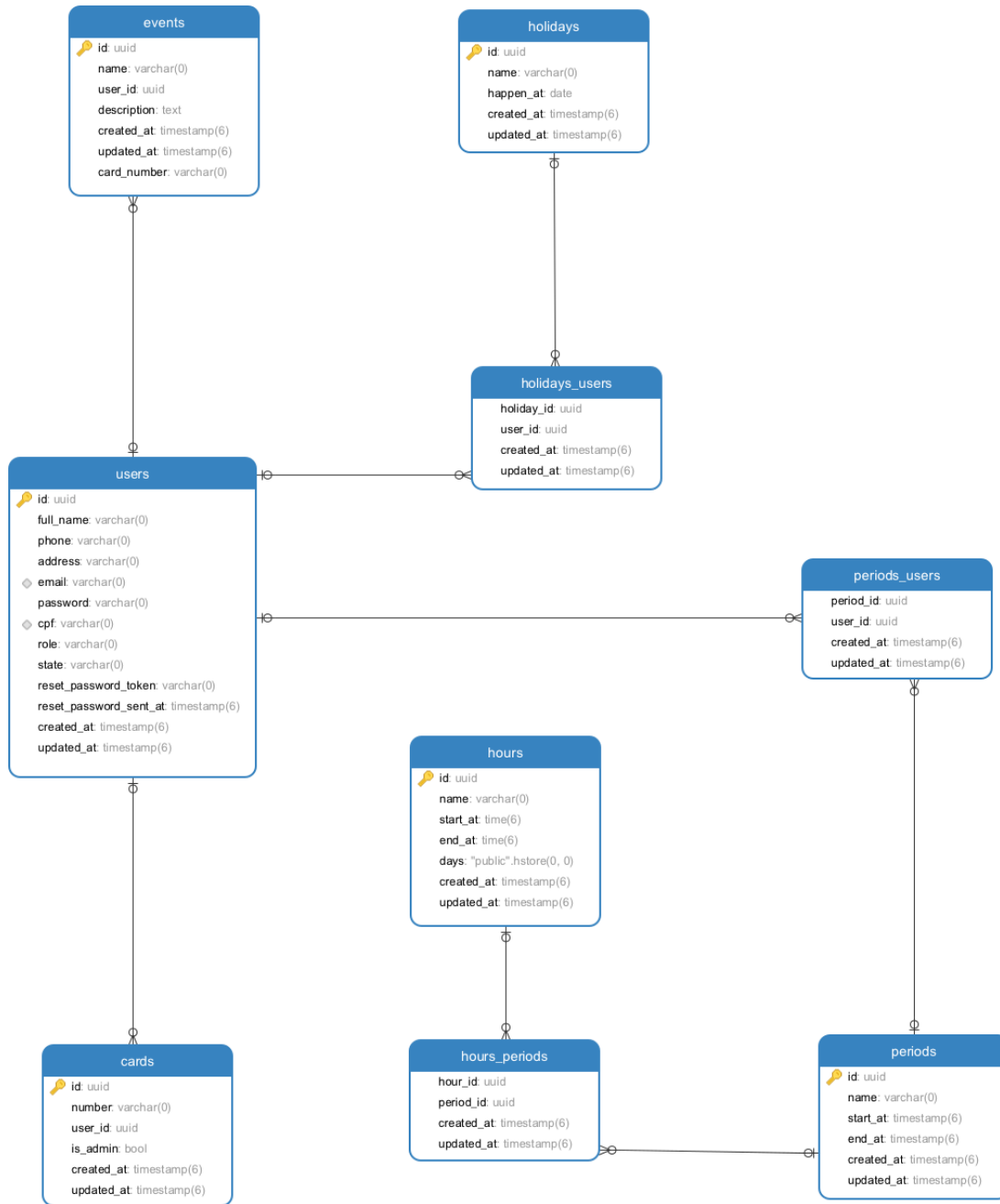


Figura 61 – Modelo relacional Banco de Dados

APÊNDICE D
ESPECIFICAÇÃO DE CASOS DE USO

UC01 – Solicitar Cadastro

Descrição: Este caso de uso serve para que um usuário solicite a sua aprovação para realizar o cadastro no sistema.

Data View:

DV01 – Tela Cadastro.png.

Pré-condições

Não há pré-condição.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Permitir ao usuário realizar o cadastro no sistema.

Ator Principal

Usuário (funcionário).

Fluxo de Eventos Principal

1. O sistema apresenta a tela. (DV01)
2. O usuário insere seu cpf, nome, email, senha nos respectivos campos. Endereço e telefone serão campos opcionais.
3. O usuário clica no botão Entrar.
4. O sistema gera uma solicitação de cadastro ao administrador. (R01) (E01)
5. O sistema informa que o pré-cadastro foi realizado e que deve aguardar uma aprovação por email (R02)

Fluxos Alternativos

Nenhum fluxo alternativo.

Fluxos de Exceção

E01. Dados inválidos:

1. O sistema recarrega a página para que o usuário digite suas credenciais novamente.
2. O Caso de Uso é reiniciado.

Regras de Negócio

R01. Para autenticar o usuário, o sistema faz uma busca pelo cpf no banco de dados, caso o cpf não exista o usuário é cadastrado, senão o sistema informa “usuário já cadastrado”.

UC02 – Solicitar Acesso

Descrição: Este caso de uso serve para que um usuário solicite acesso a área restrita.

Data View:

DV02 – Tela Solicitar acesso.png.

Pré-condições

O usuário precisa ter sido aprovado pelo administrador do sistema após a solicitação de cadastro.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

2. Permitir ao usuário ter acesso as áreas restritas.

Ator Principal

Usuário (funcionário)

Fluxo de Eventos Principal

6. O sistema apresenta a tela. (DV02)
7. O usuário passa o cartão no leitor.
8. O sistema autentica o cartão do usuário. (R01) (R02) (E01)
9. O sistema permite/nega o acesso ao usuário. (R01) (R02) (R03)

Fluxos Alternativos

1. O usuário insere o login no campo usuário.
2. O usuário insere a senha no campo Senha.
3. O usuário clica no botão Entrar.
4. O sistema autentica o login do usuário. (R01) (R02) (E01)
5. O sistema permite/nega o acesso do usuário a área restrita. (R01) (R02) (R03)

Fluxos de Exceção

E01. Cartão inválido ou não reconhecido:

3. O sistema recarrega a página para que o usuário passe o cartão novamente.
4. O Caso de Uso é reiniciado.

Regras de Negócio

R01. Para autenticar o login, o sistema faz uma busca pelo ID do cartão no banco de dados, localiza e faz a validação.

R02. Para autenticar o login via login e senha, o sistema faz uma busca pelo login no banco de dados, descriptografa a respectiva senha do banco de dados e a compara com a senha digitada.

R03 - Impedir acesso ao sistema a usuários bloqueados

UC03 – Aprovar cadastro

Descrição: Este caso de uso serve para que um administrador autorize o acesso de um usuário as áreas restritas.

Data View:

DV03 – Tela Aprovar cadastro.png.

Pré-condições

O usuário precisa ter realizado a solicitação de cadastro.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

3. Conceder ao usuário permissão de acesso as áreas restritas.

Ator Principal

Administrador.

Fluxo de Eventos Principal

10. O administrador acessa o painel administrativo.
11. O administrador verificar na lista de usuários os usuários pendentes.(DV03)
12. O administrador localiza o usuário a ser aprovado.
13. O administrador clica em ativar.
14. O sistema autentica o usuário. (R01)(R02) (E01)
15. O sistema envia um email para o usuário informando da sua aprovação.

Fluxos Alternativos

Não se aplica

Fluxos de Exceção

Não se aplica

Regras de Negócio

R01. Para autenticar o login, o sistema faz uma busca pelo ID do cartão no banco de dados, localiza e faz a ativação do cartão.

R02. Para autenticar o login via login e senha, o sistema faz uma busca pelo login no banco de dados, localiza e faz a ativação do cartão.

UC04 – Manter usuário

Descrição: Este caso de uso serve para que um administrador cadastre, altere, pesquise ou exclua um usuário.

Data View:

DV04 – Tela perfil usuário.png.

DV05 – Tela cadastrar usuário.png.

Pré-condições

1. O administrador precisa estar autenticado no sistema.
2. O administrador precisa estar logado no painel administrativo dentro da lista de usuários cadastrados.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

4. Cadastrar ou atualizar as informações do usuário no banco de dados.

Ator Principal

Administrador

Fluxo de Eventos Principal

16. O administrador acessa o painel administrativo. (DV01)

17. O administrador aciona o botão “Cadastrar” na opção usuário no menu lateral.

18. O administrador preenche os dados do usuário. (DV05)

19. O administrador aciona o botão “Salvar”. (E01)

20. O sistema cadastra os dados do novo usuário no banco de dados

21. O sistema retorna uma mensagem de usuário cadastrado com sucesso. (R02)

22. O sistema envia um email para o usuário confirmando o cadastro.

Fluxos Alternativos

A01: O usuário já é cadastrado no sistema e admin aciona o botão “Alterar”

6. O sistema busca os dados do usuário.

7. O sistema exibe os dados do usuário possibilitando edição.(DV04)

8. O administrador atualiza as informações.

9. O administrador aciona o botão “Atualizar”. (E01) (E02)

10. O sistema autentica o login do usuário. (R02) (E01)

11. O sistema altera os dados do usuário no banco de dados.

12. O caso de uso é encerrado.

A02: O administrador deseja remover um usuário cadastrado.

1. O sistema busca a lista de usuários. (DV03)

2. O administrador localiza o usuário a ser excluído.

3. O administrador aciona o botão “Excluir”.

4. O sistema inativa o perfil em questão.
5. O sistema retorna uma mensagem que o usuário foi excluído com sucesso.
6. O sistema redireciona o administrador para a tela de usuários do painel administrativo.

A03: O usuário já é cadastrado no sistema e aciona o botão de pesquisa.

1. O sistema busca a lista de usuários. (DV03)
2. O administrador digita no campo de pesquisa o usuário a ser localizado
3. O administrador aciona o botão “Pesquisar”
4. O sistema exibe o usuário desejado.

Fluxos de Exceção

E01. Dados inválidos:

1. O usuário deixa de preencher um campo obrigatório ou o preenche com dados inválidos;
2. O sistema exibe uma mensagem em vermelho junto ao campo explicando qual o problema.
3. O caso de uso é reiniciado.

Regras de Negócio

R01. O usuário deve cadastrar um email válido para receber a confirmação de cadastro.

UC05 – Manter horários

Descrição: Este caso de uso serve para que um administrador cadastre, altere, pesquise ou exclua um horário.

Data View:

DV06 – Tela cadastrar horários.png

DV07 – Tela horários.png

Pré-condições

3. O administrador precisa estar autenticado no sistema.
4. O administrador precisa estar logado no painel administrativo dentro da lista de horários.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

5. Cadastrar ou atualizar horários de acesso no banco de dados.

Ator Principal

Administrador

Fluxo de Eventos Principal

24. O administrador acessa o painel administrativo.
25. O administrador clica em “Horários” e depois em “Adicionar” (DV06)
26. O administrador preenche os dados de intervalo do horário.
27. O administrador aciona o botão “Salvar”. (E01)
28. O sistema cadastra os dados do novo horário no banco de dados
29. O sistema retorna uma mensagem de horário cadastrado com sucesso.

Fluxos Alternativos

A01: O horário já está cadastrado no sistema e deseja-se editá-lo.

13. O sistema exibe a lista de horários. (DV07)

14. O administrador aciona o botão “Editar”.
15. O administrador atualiza as informações.
16. O administrador aciona o botão “Atualizar”. (E01)
17. O sistema autentica a alteração.
18. O sistema altera os dados de horário no banco de dados.
19. O caso de uso é encerrado.

A02: O administrador deseja remover um horário cadastrado.

7. O sistema exibe a lista de horários.
8. O administrador aciona o botão “Excluir”.
9. O sistema remove o horário do banco de dados. (R01)
10. O sistema retorna uma mensagem que o horário foi excluído com sucesso.
11. O sistema redireciona o administrador para a tela de horários do painel administrativo.

A03: O usuário já é cadastrado no sistema e aciona o botão de pesquisa.

5. O sistema busca a lista de horários. (DV0X)
6. O administrador digita no campo de pesquisa o horário a ser localizado
7. O administrador aciona o botão “Pesquisar”
8. O sistema exibe o horário desejado.

Fluxos de Exceção

E01. Dados inválidos:

4. O administrador deixa de preencher um campo obrigatório ou o preenche com dados inválidos;

5. O sistema exibe uma mensagem em vermelho junto ao campo explicando qual o problema.
6. O caso de uso é reiniciado.

Regras de Negócio

R01. Quando um horário é excluído automaticamente todas as referências daquele horário são excluídos dos usuários cadastrados no intervalo de horário excluído.

UC06 – Manter períodos

Descrição: Este caso de uso serve para que um administrador cadastre, altere, pesquise ou exclua um período de acesso.

Data View:

DV08 – Tela períodos.png.

DV09 – Tela períodos.png.

Pré-condições

5. O administrador precisa estar autenticado no sistema.
6. O administrador precisa estar logado no painel administrativo dentro da lista de períodos

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

6. Cadastrar ou atualizar períodos de acesso no banco de dados.

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O administrador acessa o painel administrativo.
2. O administrador clica em “Períodos” e depois em “Adicionar” (DV08)
3. O administrador preenche os dados do período.
4. O administrador aciona o botão “Salvar”. (E01)
5. O sistema cadastra os dados do novo período no banco de dados
6. O sistema retorna uma mensagem de horário cadastrado com sucesso.

Fluxos Alternativos

A01: O período já está cadastrado no sistema e deseja-se editá-lo.

1. O sistema exibe a lista de períodos. (DV09)
2. O administrador aciona o botão “Editar”.
3. O administrador atualiza as informações.
4. O administrador aciona o botão “Atualizar”. (E01)
5. O sistema autentica a alteração.
6. O sistema altera os dados do período no banco de dados.
7. O caso de uso é encerrado.

A02: O administrador deseja remover um período cadastrado.

12. O sistema exibe a lista de períodos.
13. O administrador aciona o botão “Excluir”.
14. O sistema remove o período do banco de dados

15. O sistema retorna uma mensagem que o período foi excluído com sucesso.

16. O sistema redireciona o administrador para a tela de horários do painel administrativo.

A03: O usuário já é cadastrado no sistema e aciona o botão de pesquisa.

9. O sistema busca a lista de períodos. (DV03)

10. O administrador digita no campo de pesquisa o período a ser localizado

11. O administrador aciona o botão “Pesquisar”

12. O sistema exibe o período desejado.

Fluxos de Exceção

E01. Dados inválidos:

7. O administrador deixa de preencher um campo obrigatório ou o preenche com dados inválidos;

8. O sistema exibe uma mensagem em vermelho junto ao campo explicando qual o problema.

9. O caso de uso é reiniciado.

Regras de Negócio

R01. Quando um período é excluído automaticamente todas as referências daquele horário são excluídas dos usuários cadastrados no intervalo de horário excluído.

UC07 – Manter feriados

Descrição: Este caso de uso serve para que um administrador cadastre, altere, pesquise ou exclua um feriado.

Data View:

DV10 – Tela cadastrar feriados.png.

DV11 – Tela feriados.png.

Pré-condições

7. O administrador precisa estar autenticado no sistema.
8. O administrador precisa estar logado no painel administrativo dentro da lista de feriados.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

7. Cadastrar ou atualizar feriados de acesso no banco de dados.

Ator Principal

Administrador

Fluxo de Eventos Principal

1. O administrador acessa o painel administrativo.
2. O administrador clica em “Feriados” e depois em “Adicionar” (DV10)
3. O administrador preenche os dados do feriado.
4. O administrador aciona o botão “Salvar”. (E01)
5. O sistema cadastra os dados do feriado no banco de dados
6. O sistema retorna uma mensagem de horário cadastrado com sucesso.

Fluxos Alternativos

A01: O feriado já está cadastrado no sistema e deseja-se editá-lo.

1. O sistema exibe a lista de períodos. (DV11)
2. O administrador aciona o botão “Editar”.

3. O administrador atualiza as informações.
4. O administrador aciona o botão “Atualizar”. (E01)
5. O sistema autentica a alteração.
6. O sistema altera os dados do feriado no banco de dados.
7. O caso de uso é encerrado.

A02: O administrador deseja remover um feriado cadastrado.

17. O sistema exibe a lista de feriados.
18. O administrador aciona o botão “Excluir”.
19. O sistema remove o feriado do banco de dados
20. O sistema retorna uma mensagem que o feriado foi excluído com sucesso.
21. O sistema redireciona o administrador para a tela de horários do painel administrativo.

A03: O usuário já é cadastrado no sistema e aciona o botão de pesquisa.

13. O sistema busca a lista de feriados. (DV0X)
14. O administrador digita no campo de pesquisa o feriado a ser localizado
15. O administrador aciona o botão “Pesquisar”
16. O sistema exibe o feriado desejado.

Fluxos de Exceção

E01. Dados inválidos:

10. O administrador deixa de preencher um campo obrigatório ou o preenche com dados inválidos;
11. O sistema exibe uma mensagem em vermelho junto ao campo explicando qual o problema.

12. O caso de uso é reiniciado.

Regras de Negócio

R01. Quando um feriado é excluído automaticamente todas as referências daquele horário são excluídos dos usuários cadastrados no intervalo de horário excluído.

UC08 – Visualizar eventos

Descrição: Este caso de uso serve para que o administrador visualize os logs de acessos dos usuários.

Data View:

DV12 – Tela eventos.png.

Pré-condições

9. O administrador precisa estar autenticado no sistema.
10. O administrador precisa estar logado no painel administrativo dentro dos logs de acesso.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

8. Redirecionar o administrador à home do painel.

Ator Principal

Administrador

Fluxo de Eventos Principal

7. O administrador acessa os logs de eventos. (DV12)
8. O sistema carrega os dados do log de acessos.
9. O sistema exibe os logs para o administrador (E01)
10. O caso de uso é encerrado.

Fluxos de Exceção

E01. Não há dados para serem exibidos:

13. O sistema exibe a lista sem conter nenhum dado, com a mensagem “Não há dados de logs de acesso”.
14. O caso de uso é encerrado.

Regras de Negócio

Não se aplica

UC09 – Gerar solicitação de acesso pendente

Descrição: Este caso de uso serve para que o administrador receba a solicitação de aprovação de um novo usuário.

Data View:

DV01 – Tela.png.

Pré-condições

11. O usuário precisa ter realizado a solicitação de cadastro

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

9. Gerar uma solicitação para aprovação de um administrador.

Ator Principal

Administrador

Fluxo de Eventos Principal

11. O usuário realiza a solicitação de cadastro (DV01)
12. O sistema valida os dados
13. O sistema gera a solicitação de acesso para o administrador (R01) (E01)
14. O caso de uso é encerrado.

Fluxos de Exceção

Não se aplica

Regras de Negócio

Não se aplica

APÊNDICE E

DIAGRAMAS DE CLASSE DE IMPLEMENTAÇÃO

Diagrama de classe da API

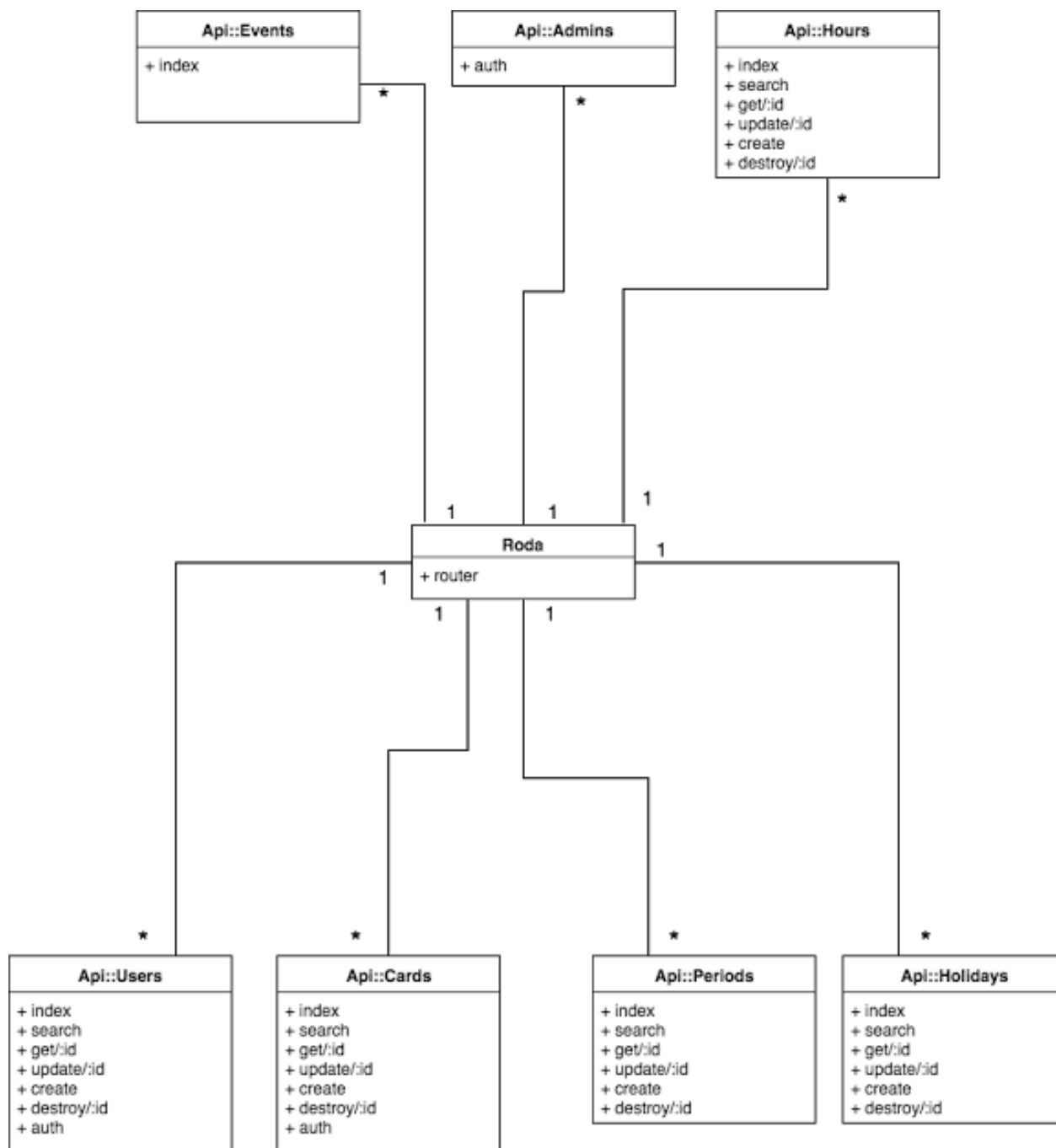


Figura 62 – Diagrama de classe da API

Diagrama de classe Filtros

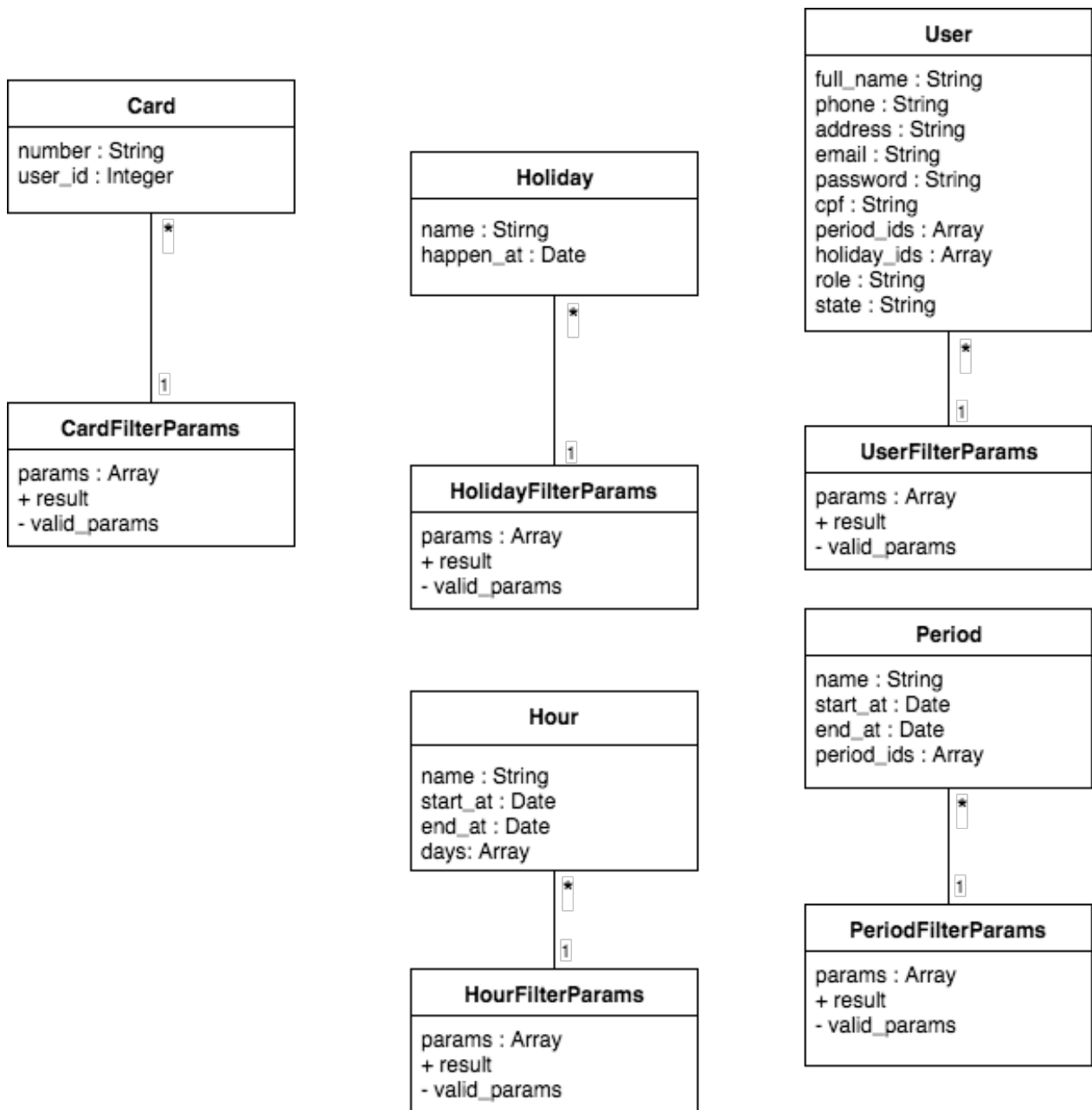


Figura 63 – Diagrama de classe Filtros

Diagrama de classe Decorator

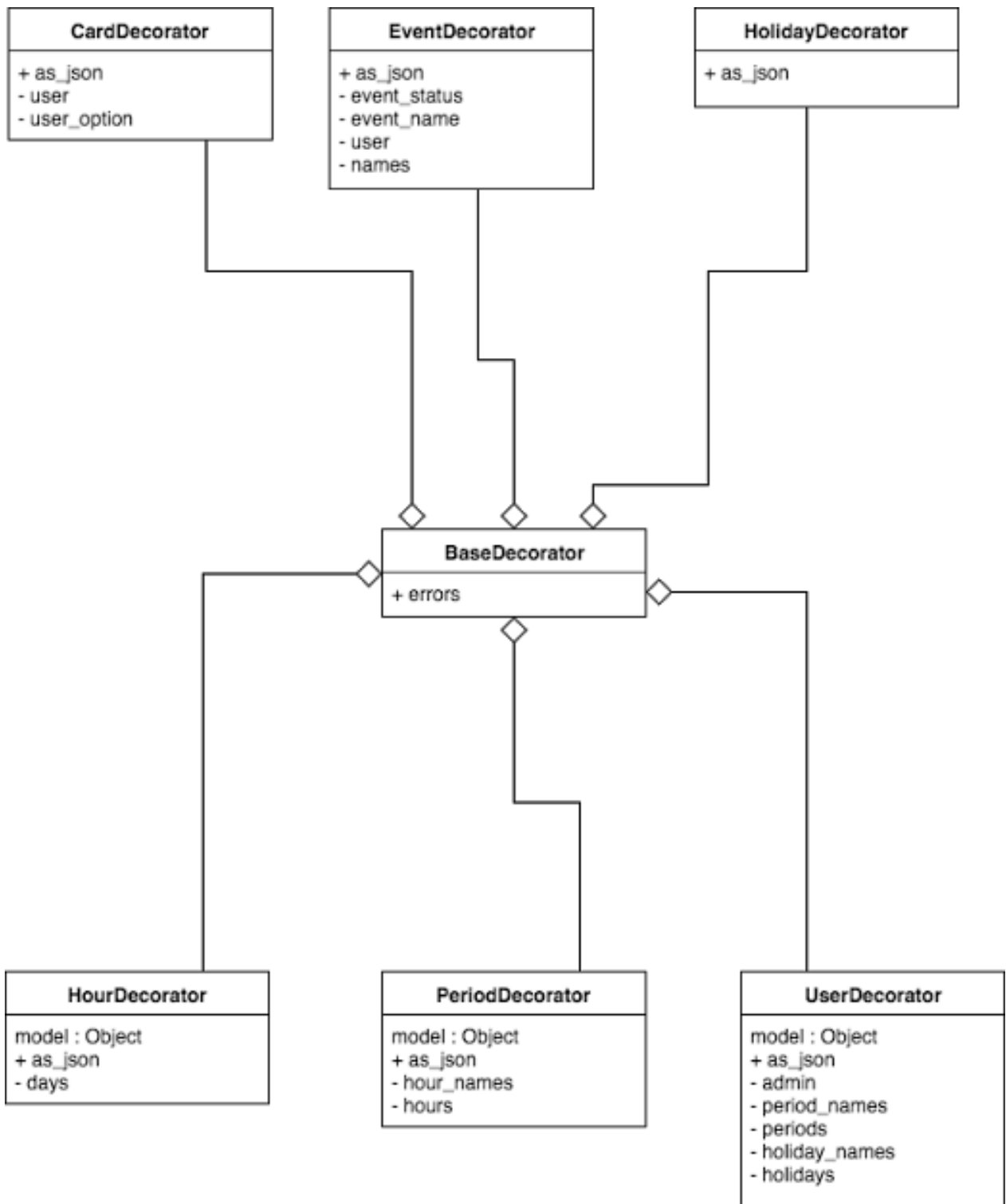


Figura 64 – Diagrama de classe Decorator

Diagrama de classe serviços

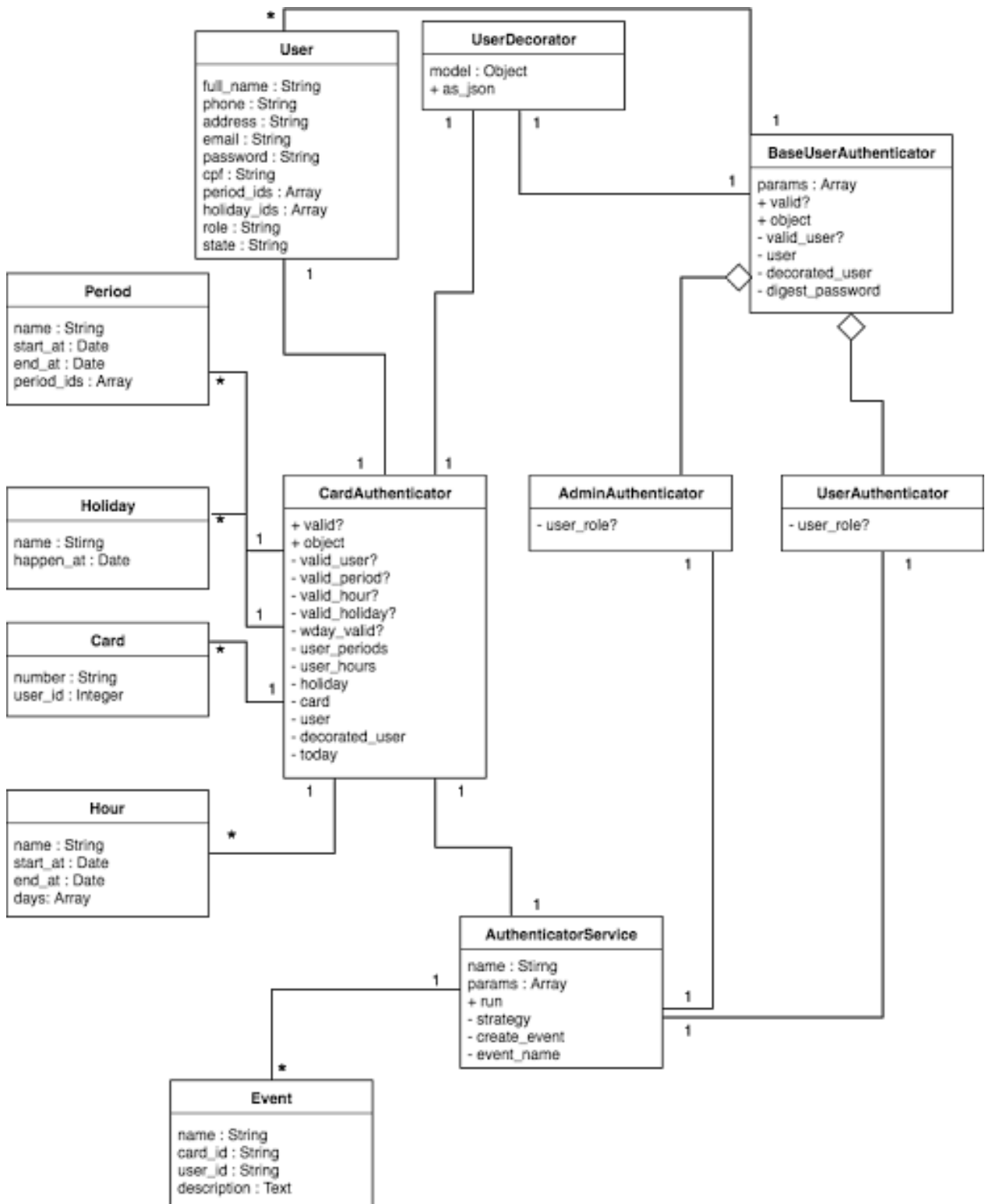


Figura 65 – Diagrama de classe serviços

APÊNDICE F

DIAGRAMAS DE SEQUÊNCIA

Solicitar acesso

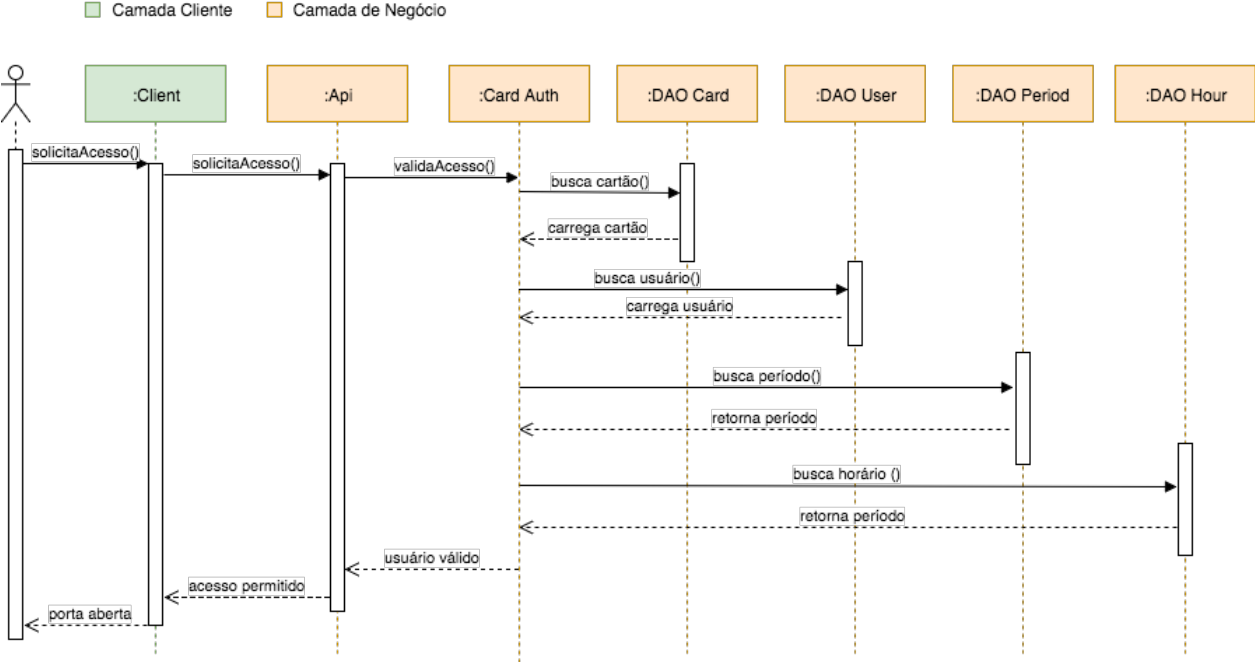


Figura 66 – Solicitar acesso

Manter usuário

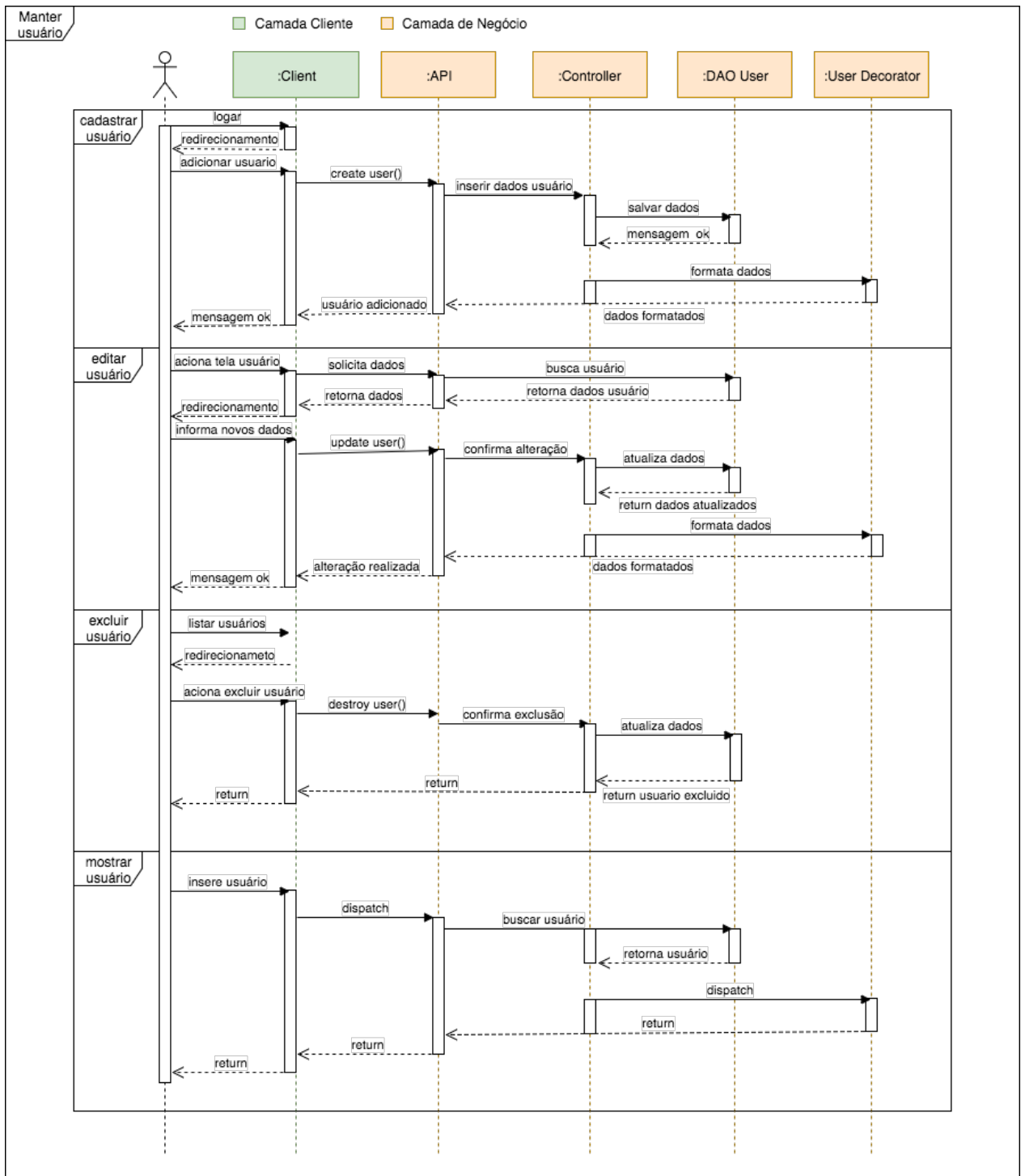


Figura 67 – Manter usuário

Manter cartão

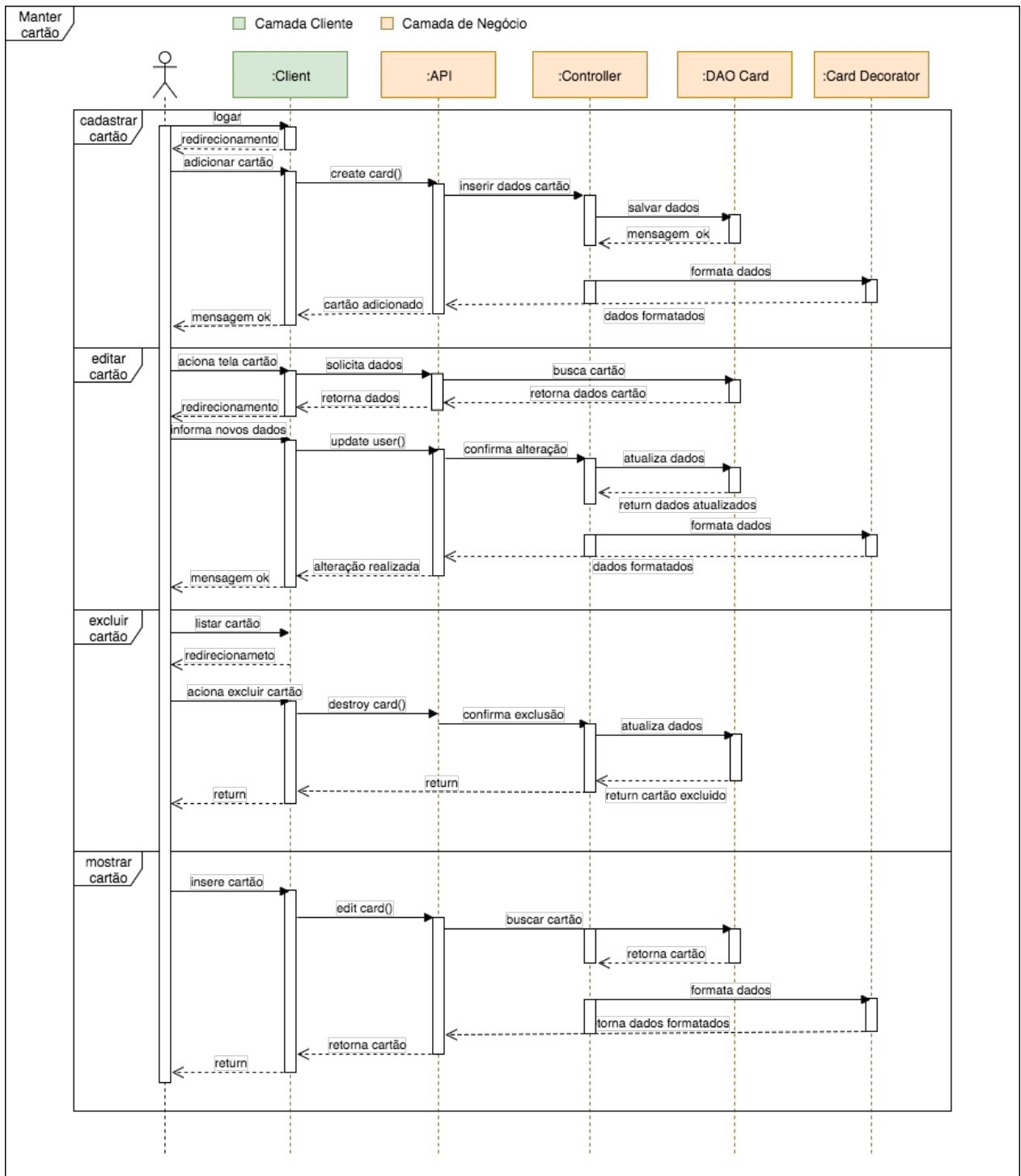


Figura 68 – Manter cartão

Manter período

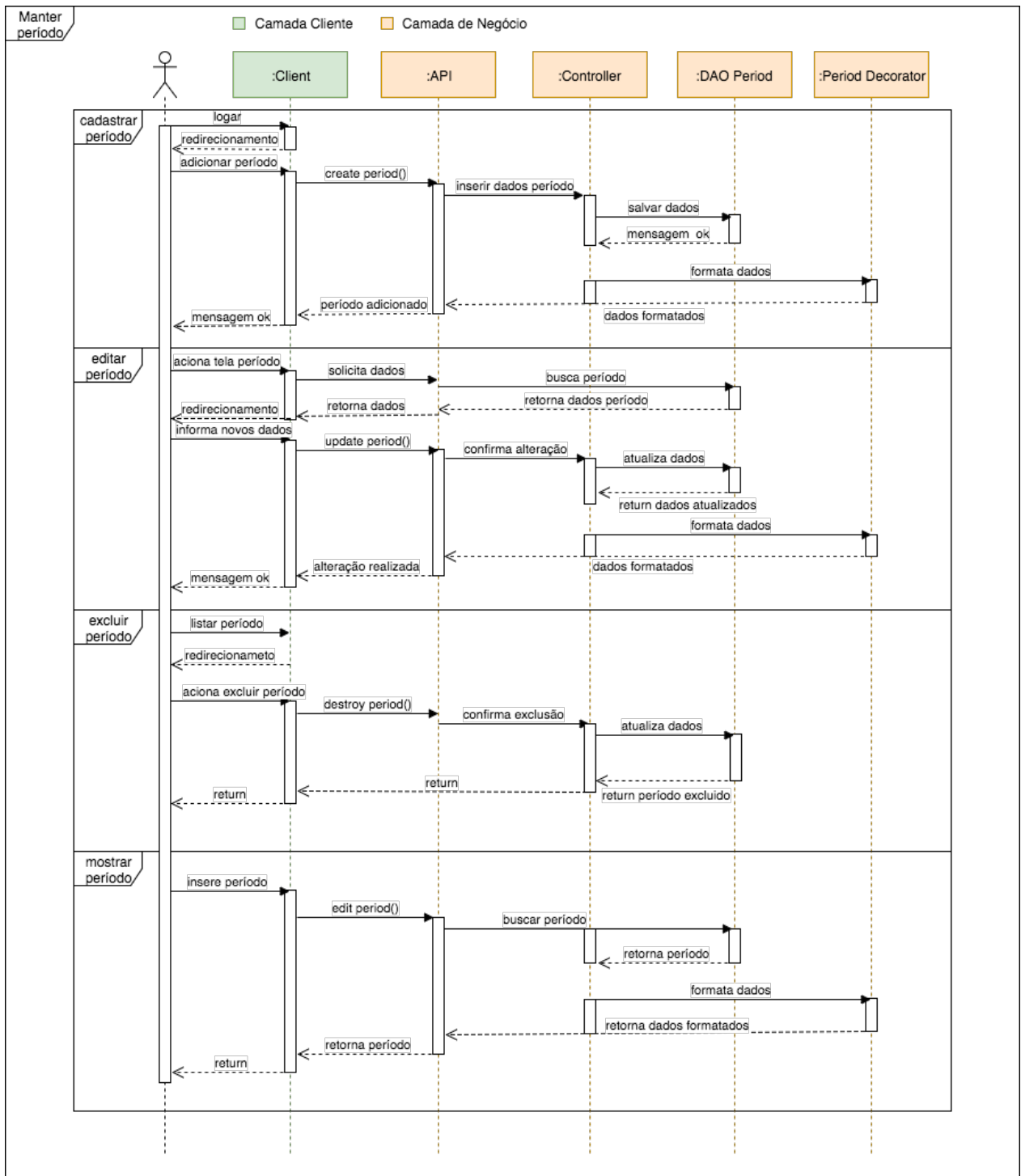


Figura 69 – Manter período

Manter horário

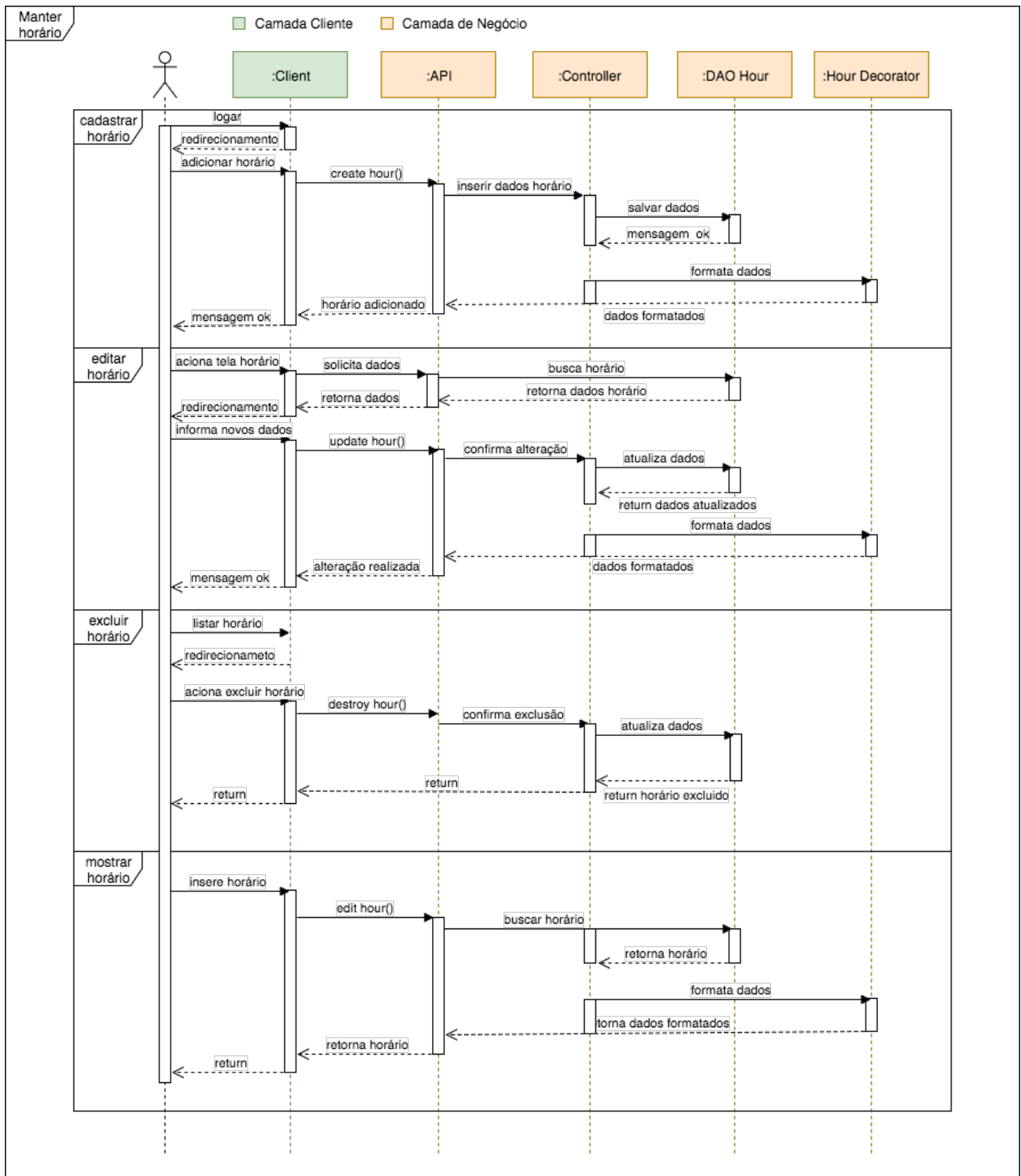


Figura 70 – Manter horário

Manter feriado

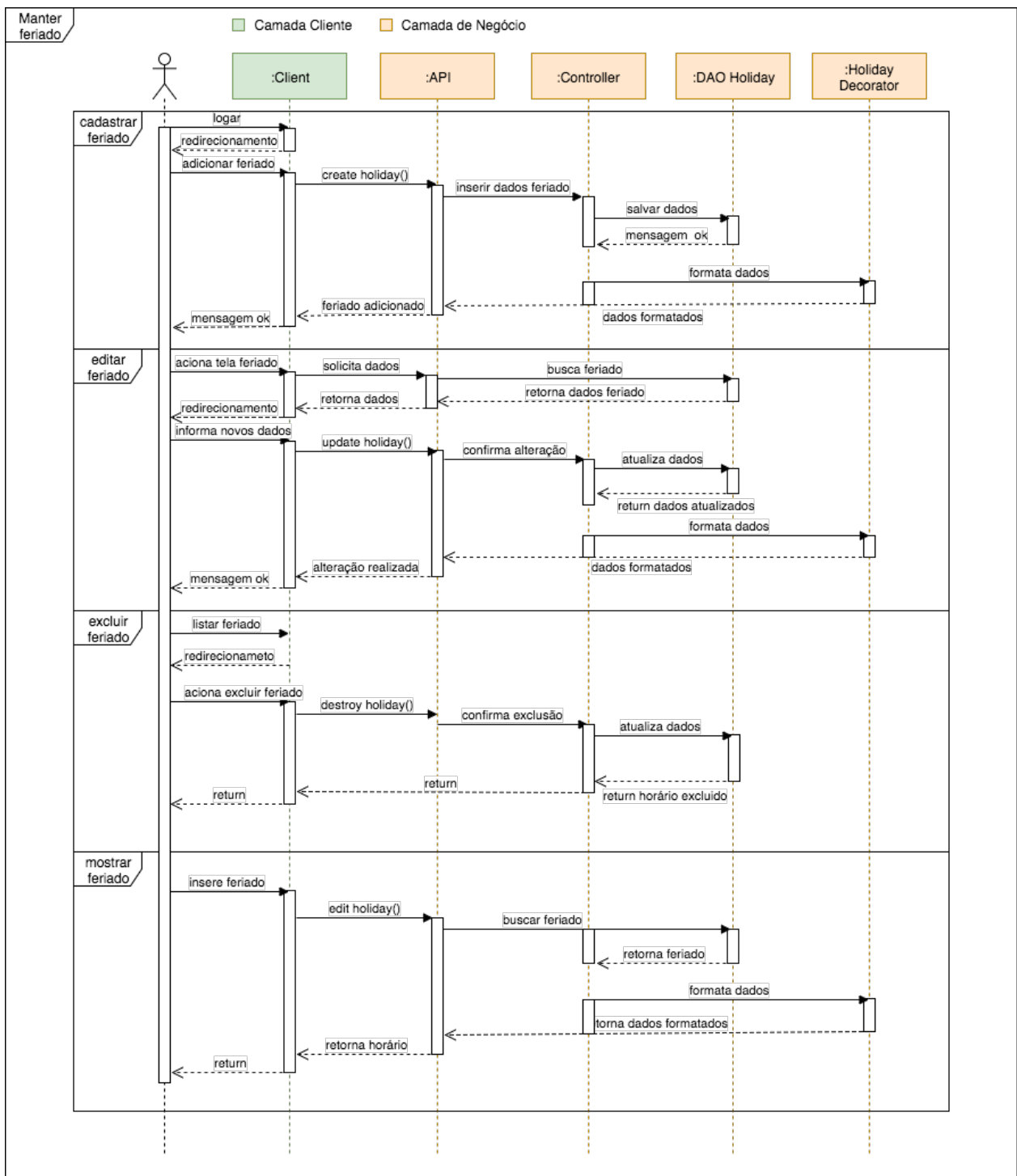


Figura 71 – Manter feriado