

UNIVERSIDADE FEDERAL DO PARANÁ

GABRIEL STROBEL COELHO SANTOS

LUCAS DE FRANÇA DE CAMPOS

LUIZ HENRIQUE NASCIMENTO DOS ANJOS



SISTEMA DE ACESSO ELETRÔNICO

CURITIBA

2016

GABRIEL STROBEL COELHO SANTOS
LUCAS DE FRANÇA DE CAMPOS
LUIZ HENRIQUE NASCIMENTO DOS ANJOS

SISTEMA DE ACESSO ELETRÔNICO

Trabalho apresentado como requisito para
nota parcial na disciplina Implementação de
Aplicação para Computador no curso de
graduação em Tecnologia em Análise e
Desenvolvimento de Sistemas, Setor de
Educação Profissional e Tecnológica da
Universidade Federal do Paraná.

Profº.Luiz Antonio Pereira Neves

CURITIBA

2016

SUMARIO DAS IMAGENS

FIGURA 1 – TELA DE ACESSO	6
FIGURA 2 - TELA DE INÍCIO.....	6
FIGURA 3 - PESQUISA DE USUÁRIOS.....	7
FIGURA 4 - PESQUISA	7
FIGURA 5 - CADASTRO.....	8
FIGURA 6 - CONTROLE DE ACESSO.....	8
FIGURA 7 - DESVINCULAR TAG.....	9
FIGURA 8 - ALTERAÇÕES.....	10
FIGURA 9 - CADASTRO DE SALAS	11
FIGURA 10 - EDIÇÃO DE SALAS	11
FIGURA 11 - INCLUSÃO DE SALAS.....	11
FIGURA 12 - BUSCA DE SALAS.....	12
FIGURA 13 - CADASTRO DE TAG	12
FIGURA 14 – EDIÇÃO DE TAG.....	13
FIGURA 15 - HISTÓRICO DE TAG	13
FIGURA 16 - FLUXO DE TELAS	14
FIGURA 17 - MOVIMENTO EM F.....	15
FIGURA 18 - ACESSO.....	17
FIGURA 19 - CATEGORIA TAG	18
FIGURA 20 - CATEGORIA USUÁRIO	18
FIGURA 21 - PERMISSÕES.....	19
FIGURA 22 - RELACIONAMENTO USUÁRIO TAG	19
FIGURA 23 - SALA	20
FIGURA 24 - TAG.....	20
FIGURA 25 - USUÁRIO	21
FIGURA 26 - MODELAGEM LÓGICA.....	22
FIGURA 27 – DIAGRAMA DE CONTEXTO	33
FIGURA 28 – DFD NÍVEL 0	34
FIGURA 29 – DFD 1 – ACESSO SALA	35
FIGURA 30 – DFD 1 – CONTROLE DE ACESSO.....	35

SUMÁRIO

1.	INTRODUÇÃO.....	4
2	CAPÍTULO 1 – PROTOTIPAÇÃO DE TELAS	5
2.1	PROTÓTIPOS.....	5
2.2	FLUXO DE TELAS	14
2.3	ERGONOMIA.....	15
2.4	CONCLUSÃO	16
3	CAPÍTULO 2 – MODELO LÓGICO DE DADOS.....	17
3.1	ACESSO	17
3.2	CATEGORIAS.....	18
3.3	PERMISSÕES	18
3.4	RELACIONAMENTO USUÁRIO TAG.....	19
3.5	SALA.....	20
3.6	TAG.....	20
3.7	USUÁRIO.....	21
3.8	CONCLUSÃO	23
4	CAPÍTULO 3 – MODELO FÍSICO DE DADOS.....	24
4.1	SCRIPT DE BANCO DE DADOS.....	24
4.2	CONCLUSÃO	31
5	CAPITULO 5 – DIAGRAMA DE FLUXO DE DADOS	32
5.1	DIAGRAMA DE CONTEXTO	33
5.2	DFD – NÍVEL 0	33
5.3	DFD NÍVEL 1	34
5.4	CONCLUSÃO	36
6	FONTES	37
7	CONCLUSÃO	39
8	REFERÊNCIAS	40

1. INTRODUÇÃO

A disciplina Implementação de Aplicação para Computador propõe aos alunos uma experiência prática que engloba diversos conteúdos ministrados ao longo do curso de Análise e Desenvolvimento de Sistemas. O objetivo desta disciplina é implementar um sistema, fazendo a modelagem estruturada e sua programação. Para isso, os alunos responsáveis por este trabalho realizaram todas as etapas do desenvolvimento de um software: análise dos requisitos, prototipação de telas, modelagem de dados, desenvolvimento do script e diagrama de fluxo de dados, até que, enfim, pudessem implementar o sistema.

O Sistema de Acesso Eletrônico é um método de gerência de acesso baseado na tecnologia RFID, isto é, identificação por radiofrequência. Esta tecnologia, muito usada no setor de logística e varejo, se mostra extremamente eficiente no objetivo deste trabalho: controlar o acesso de usuários em determinados ambientes.

O objetivo dos autores, após concluir o trabalho, é aprender a trabalhar com as diversas áreas técnicas e administrativas que o curso propõe. Além disso, espera-se apresentar um sistema que atenda aos requisitos dados pelo professor e, ainda, novas soluções que otimizem a experiência do usuário.

2 CAPÍTULO 1 – PROTOTIPAÇÃO DE TELAS

A prototipação é um processo que tem como finalidade facilitar a compreensão dos requisitos, apresentar conceitos e funcionalidades do software. Assim, aumentamos a sua percepção de valor, já que podemos oferecer uma solução mais adequada as necessidades do cliente. Um protótipo também é a forma mais rápida de definir e experimentar um projeto.

Os protótipos podem ser desenvolvidos em diferentes níveis de fidelidade. Quanto maior a fidelidade maior o tempo de produção ou modificação. A escolha do nível de fidelidade é feito de acordo com o nível de complexidade do projeto e das exigências do cliente.

A prototipação das telas deste projeto foi feita através do software Balsamiq Mockups, atingindo um nível baixo de fidelidade. Apresentamos todas as principais telas, que servem tanto para administrador quanto usuário. Um usuário comum, no entanto, possui algumas opções desabilitadas.

2.1 PROTÓTIPOS

Para iniciar, apresentamos a tela de acesso (FIGURA 1). Nesta tela, há os campos para inserção da *tag* ou e-mail, e a senha do usuário, além do link para solicitação de cadastro de novos usuários.

Após o login, é apresentada uma tela inicial (FIGURA 2) que mostra o histórico com os acessos mais recentes e o menu ao lado esquerdo. Vale ressaltar que o usuário comum tem acesso somente ao histórico mais recente. O menu completo aparece apenas para um usuário de *tag* master. As solicitações de inclusões ou retiradas de *tags* sempre são feitas por um administrador.

Nesta tela inicial vemos ao lado esquerdo o menu principal que dá acesso ao cadastro e pesquisa de usuários, salas, *tag* e histórico de *tag*. No lado superior direito há uma identificação de boas vindas e um link para finalizar a sessão.

FIGURA 1 – TELA DE ACESSO



Sistema de Controle de Acesso

Login

E-mail/TAG:

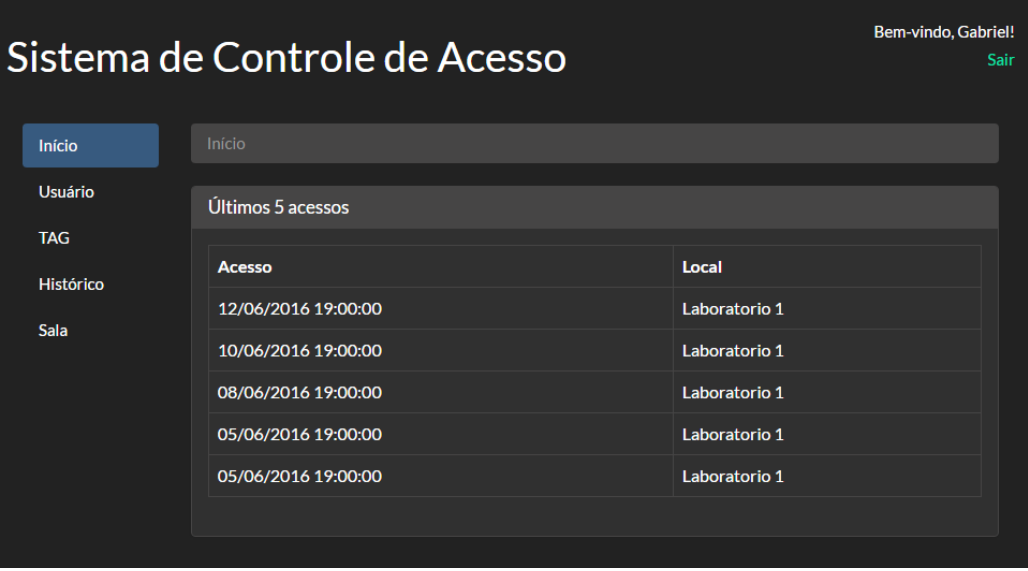
Senha:

Login

Novo usuário? Solicite seu cadastro aqui.

FONTE: Os autores (2016).

FIGURA 2 - TELA DE INÍCIO



Sistema de Controle de Acesso

Bem-vindo, Gabriel! [Sair](#)

Início

Usuário

TAG

Histórico

Sala

Últimos 5 acessos

Acesso	Local
12/06/2016 19:00:00	Laboratorio 1
10/06/2016 19:00:00	Laboratorio 1
08/06/2016 19:00:00	Laboratorio 1
05/06/2016 19:00:00	Laboratorio 1
05/06/2016 19:00:00	Laboratorio 1

FONTE: Os autores (2016).

Em seguida, o primeiro item do menu nos apresenta o cadastro de usuários (FIGURA 3). Essa tela fornece uma maneira de um administrador efetuar uma busca de um usuário por nome ou matrícula.

FIGURA 3 - PESQUISA DE USUÁRIOS

Sistema de Controle de Acesso

Bem-vindo, Gabriel!
Sair

Usuários

Nome: Matrícula:

Pesquisar

Usuários

Nenhum registro para exibir.

Incluir

FONTE: Os autores (2016).

FIGURA 4 - PESQUISA

Sistema de Controle de Acesso

Bem-vindo, Gabriel!
Sair

Usuários

Nome: Matrícula:

Luiz

Pesquisar

Usuários

	Nome	Matrícula	Tipo de Usuário
■	Luiz Antonio Passos	8978	Cliente
■	Luiz Henrique	12344	Cliente

Incluir Editar Excluir

FONTE: Os autores (2016).

Na sequência são exibidos os resultados e uma opção para que o usuário selecione um dos resultados caso queira verificar mais detalhes (FIGURA 4).

A tela de cadastro contém os campos para inserção dos dados dos usuários, além de um arquivo de imagem (FIGURA 5). O controle de acesso é feito na tela

que possui opções para inserção do local a que se concede acesso e do período de permissão (FIGURA 6).

FIGURA 5 - CADASTRO

Sistema de Controle de Acesso Bem-vindo, Gabriel! [Sair](#)

Usuários / Incluir

Nome: Matricula:

CEP: Rua:

Número: Bairro: Cidade:

Estado: E-mail: Telefone:

Departamento: CPF:

Tag: Tipo:

Senha:

Foto: Nenhum arquivo selecionado

FONTE: Os autores (2016).

FIGURA 6 - CONTROLE DE ACESSO

Sistema de Controle de Acesso Bem-vindo, Gabriel! [Sair](#)

Usuários / Editar / Controle de Acesso

Local	Dia	Início	Fim
Laboratório 1	segunda	08:00	09:00
Laboratório 1	terça	11:00	12:00
Laboratório 2	terça	13:00	15:00

Local: Início: Fim:

FONTE: Os autores (2016).

FIGURA 7 - DESVINCULAR TAG

A imagem mostra a interface de um sistema de controle de acesso. No topo, há o título 'Sistema de Controle de Acesso' e uma saudação 'Bem-vindo, Gabriel!' com um link 'Sair'. À esquerda, há um menu lateral com opções: 'Início', 'Usuário' (destacado), 'TAG', 'Histórico' e 'Sala'. No topo da área principal, há uma barra de navegação com 'Usuários / Editar / Desvincular Tag'. Abaixo, há o texto 'Selecione o motivo:' seguido de uma lista suspensa com a opção 'Perda/Roubo'. Na base, há dois botões: 'Salvar' e 'Cancelar'.

FONTE: Os autores (2016).

É possível desvincular a *tag* do usuário. Para isso, é necessário informar o motivo (FIGURA 7).

Ao editar o cadastro de um usuário, selecionado após a pesquisa, o administrador pode alterar suas permissões de acesso e desvincular uma *tag* (FIGURA 8). É possível, por exemplo, alterar somente os horários ou locais em que o usuário tem acesso, sem ter que criar ou excluir um cadastro novo a cada operação.

FIGURA 8 - ALTERAÇÕES

Sistema de Controle de Acesso

Bem-vindo, Gabriel! [Sair](#)

Usuários / Editar

Alterar/Incluir Foto: Nenhum...ionado

Nome: Matrícula:

Rua: Número:

Bairro: CEP: Cidade:

Estado: E-mail:

Telefone: Departamento:

Tag: Tipo:

Permissões de acesso			
Local	Dia	Início	Fim
Laboratório 1	segunda	08:00	09:00
Laboratório 1	terça	11:00	12:00
Laboratório 2	terça	13:00	15:00

FONTE: Os autores (2016).

O menu seguinte é o de cadastro de salas (FIGURA 9). Ele é bem simples, já que serve para incluir, excluir ou editar as salas que os usuários terão acesso (FIGURAS 10 e 11). Há também uma tela que permite buscar uma sala (FIGURA 12).

FIGURA 9 - CADASTRO DE SALAS

Sistema de Controle de Acesso

Bem-vindo, Gabriel!
Sair

Início: Sala / Incluir

Usuário: Nome: Número: Descrição:

TAG: [Input fields]

Histórico: [Input fields]

[Salvar] [Cancelar]

Sala

FONTE: Os autores (2016).

FIGURA 10 - EDIÇÃO DE SALAS

Sistema de Controle de Acesso

Bem-vindo, Gabriel!
Sair

Início: Sala / Editar

Usuário: Nome: Número: Descrição:

TAG: Laboratório 1 101 laboratório de informática para pesquisa:

Histórico: [Input fields]

[Salvar] [Cancelar]

Sala

FONTE: Os autores (2016).

FIGURA 11 - INCLUSÃO DE SALAS

Sistema de Controle de Acesso

Bem-vindo, Gabriel!
Sair

Início: Sala / Incluir

Usuário: Nome: Número: Descrição:

TAG: [Input fields]

Histórico: [Input fields]

[Salvar] [Cancelar]

Sala

FONTE: Os autores (2016).

FIGURA 12 - BUSCA DE SALAS

Sistema de Controle de Acesso Bem-vindo, Gabriel! [Sair](#)

Início
 Usuário
 TAG
 Histórico
Sala

Sala

Nome: Número: Descrição:

101

Q Pesquisar

Nome	Número	Descrição
Laboratório 1	101	laboratorio de informatica para pesquisas

+ Incluir ✎ Editar ✖ Excluir

FONTE: Os autores (2016).

No cadastro de *tag*, além das opções mais básicas de inserção e exclusão (FIGURA 13) ele possibilita também selecionar a chave para editar seus dados (FIGURA 14).

FIGURA 13 - CADASTRO DE TAG

Sistema de Controle de Acesso Bem-vindo, Gabriel! [Sair](#)

Início
 Usuário
TAG
 Histórico
 Sala

TAGs

Chave: Tipo:

1234567890 Comum

+ Incluir Cancelar

Chave:

selecione

✎ Editar ✖ Excluir

FONTE: Os autores (2016).

FIGURA 14 – EDIÇÃO DE TAG

FONTE: Os autores (2016).

Por fim, há uma tela para consultar o histórico de uma determinada *tag* (FIGURA 15). Nesta tela o usuário deve selecionar uma *tag*, e abaixo é exibido o nome do usuário vinculado a *tag*. É possível conferir data e horário dos últimos acessos registrados com a *tag*, o local atribuído ao acesso.

FIGURA 15 - HISTÓRICO DE TAG

Acesso	Local
12/06/2016 19:00:00	Laboratorio 1
10/06/2016 19:00:00	Laboratorio 1
08/06/2016 19:00:00	Laboratorio 1
05/06/2016 19:00:00	Laboratorio 1
05/06/2016 19:00:00	Laboratorio 1

FONTE: Os autores (2016).

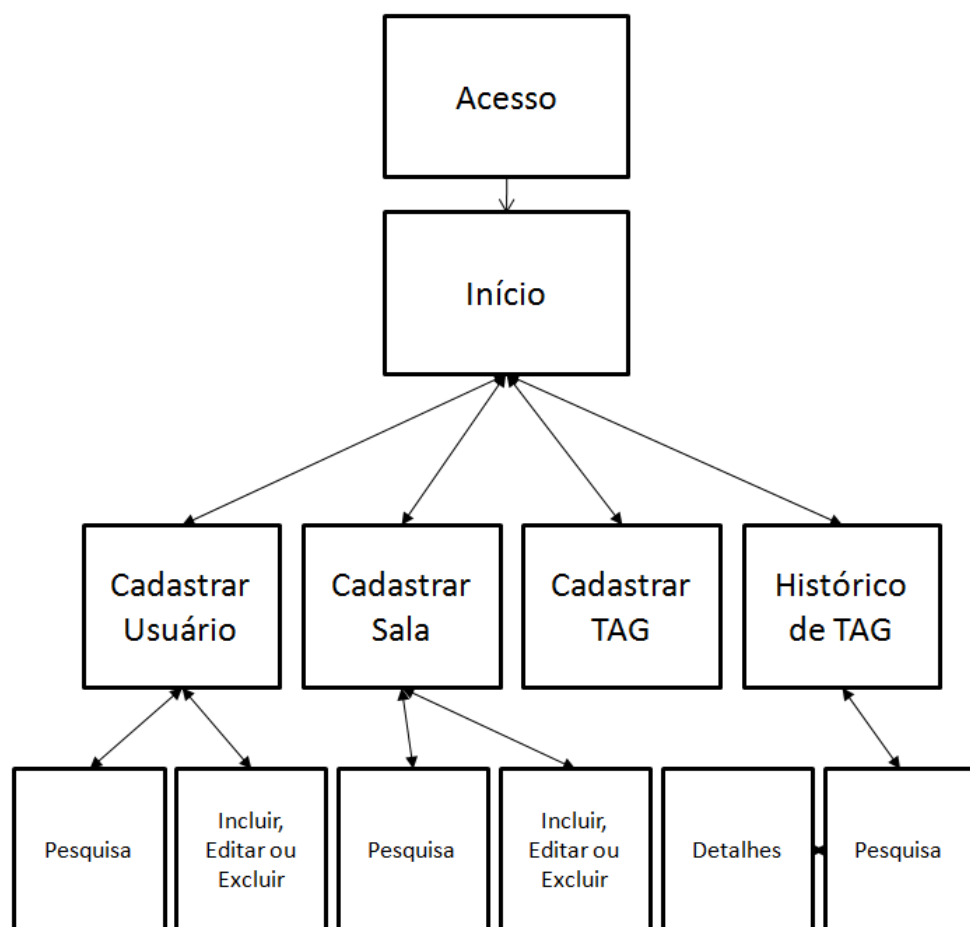
2.2 FLUXO DE TELAS

Para facilitar o entendimento da prototipação, apresentamos o fluxo das telas do sistema (FIGURA 17).

Seguindo os preceitos da Ergonomia, com conhecimento obtido na fase de prototipação, a estrutura do sistema permite que em dois cliques, a partir da tela de início, o usuário consiga acessar a tela de edição ou pesquisa desejada. Isso torna o sistema mais eficiente e amigável.

Ao longo do desenvolvimento desse projeto, essa prototipação sofre alterações, visando melhorar o funcionamento do sistema, e isso implica em alterações no fluxo de telas.

FIGURA 16 - FLUXO DE TELAS



FONTE: Os autores (2016).

2.3 ERGONOMIA

A Ergonomia trata-se de uma ciência que estuda as relações do homem com o trabalho e o meio ambiente. Esses conceitos ganharam força após a Revolução Industrial sobretudo durante a Segunda Guerra Mundial.

Há uma tríade básica da Ergonomia: Conforto, Segurança e Eficiência. Aplicando esses conceitos a sites ou sistemas, chegamos em três pontos principais: Design, Usabilidade e Utilidade.

Com esses conceitos em mente, o sistema foi criado visando contemplar o máximo possível de cada um.

O sistema é bastante limpo visualmente, o que evita o usuário de cansar a visão. Há também o conceito do “movimento F”. Há estudos que comprovam que a maioria das pessoas tendem a observar a página e o caminho percorrido pela visão forma algo parecido com a letra F (FIGURA 18). Considerando isso, o sistema possui o menu principal localizado todo no lado esquerdo da tela. E os submenus, ficam posicionados na parte superior do *frame* central, paralelamente a logo do site.

FIGURA 17 - MOVIMENTO EM F



FONTE: Blog Wix, 2016. Disponível em: <http://goo.gl/y60sDM>

Outro ponto importante é que somente em dois cliques o usuário consegue chegar na tela de consultar ou edição mais relevantes. Não há também outras coisas que distraiam o usuário, além do que ele realmente busca.

As cores utilizadas em uma página também são levadas em conta na questão da ergonomia. Embora o protótipo esteja nas cores padrão do Balsamiq, as cores pré-selecionadas para serem utilizadas vão ser azul, branco ou cinza. Essas cores são associadas à confiança, integridade e seriedade. Essas são características que o sistema busca demonstrar ao usuário, já que o foco é gerenciamento e segurança. São as cores mais apropriadas para o público alvo.

2.4 CONCLUSÃO

A aplicação web desenvolvida tem a utilidade de gerenciar os dados de um sistema de controle de acesso utilizando *tags* RFID, voltado para o ambiente acadêmico. Por meio da administração das entidades envolvidas (usuários, salas, *tags*), é possível definir as regras do controle de acesso às salas de aula e aos laboratórios. A documentação produzida nessa parte do projeto permite visualizar como essas tarefas podem ser realizadas e define que o conceito de ergonomia é priorizado, com o objetivo de otimizar a utilização do sistema.

3 CAPÍTULO 2 – MODELO LÓGICO DE DADOS




Um modelo de dados lógico é uma representação lógica das informações do sistema. Essa abstração permite uma comunicação simplificada entre programador e usuário final, já que é de fácil compreensão. Ele é organizado de forma independente da tecnologia, já que a mesma se altera com o passar do tempo. O modelo lógico também se altera com o tempo, mas essa mudança é muito mais lenta. Com uma modelagem lógica bem estruturada uma mudança de tecnologia é muito mais fácil de ser feita.

Nessa etapa são definidas as informações que devem fazer parte do banco de dados, as funções e todas as suas regras. É o passo imediatamente anterior ao modelo físico, que já representa o script de banco de dados. Portanto é fundamental um bom capricho nessa etapa para que sejam evitados erros na etapa seguinte e que acabem gerando retrabalho. A ferramenta utilizada neste capítulo foi o brModelo.

3.1 ACESSO

A tabela de Acesso (FIGURA 19) refere-se ao acesso do local ao qual a *tag* se refere. Ela irá conter a identificação da *tag* e da sala. Com esses itens armazenados nessa tabela um histórico será criado, para consultas posteriores. Ele possui duas chaves estrangeiras referenciando as tabelas de *tags* e de salas.

FIGURA 18 - ACESSO


acesso	
	id: INTEGER
	id_tag: INTEGER
	id_sala: INTEGER
	data_hora: DATETIME

Fonte: Os autores (2016).

3.2 CATEGORIAS


As categorias de *tag* e de Usuário (FIGURAS 20 e 21) são simples porém fundamentais. Essas tabelas guardam as informações que categorizam o usuário e a *tag*. Um usuário pode ser comum ou administrador. E uma *tag* pode ser normal ou master. A tabela de usuários faz referência com a tabela *categoria_usuario* para acessar essa informação. Da mesma forma acontece com a tabela de *tags*, que faz referência a *categoria_tag*.

FIGURA 19 - CATEGORIA TAG

categoria_tag	
	id: INTEGER(4)
	tipo: VARCHAR(20)

FONTE: Os Autores (2016).

FIGURA 20 - CATEGORIA USUÁRIO




categoria_usuario	
	id: INTEGER
	tipo: VARCHAR(20)

FONTE: Os autores (2016).

3.3 PERMISSÕES

Durante o cadastro de um novo usuário comum é necessário cadastrar as permissões. Ou seja, é informado a sala, dias e os horários que o usuário está autorizado a acessar. Com isso, ele terá o acesso negado em outras salas ou horários a não ser que seja um administrador. A tabela (FIGURA 22) contém chaves estrangeiras referenciando as tabelas Salas e *tags*. Além disso, há os campos para inserção dos horários de início de fim. Os dias da semana são booleanos.

FIGURA 21 - PERMISSÕES



permissoes	
	id: INTEGER
	id_sala: INTEGER
	id_tag: INTEGER
	hora_inicio: TIME
	hora_fim: TIME
	data_inicio_permissao: DATE
	data_fim_permissao: DATE
	segunda: INT(1)
	terca: INT(1)
	quarta: INT(1)
	quinta: INT(1)
	sexta: INT(1)
	sabado: INT(1)
	domingo: INT(1)

FONTE: Os autores (2016).

3.4 RELACIONAMENTO USUÁRIO TAG

A tabela seguinte (FIGURA 23) apresenta um relacionamento de Usuário com *tag*. Este relacionamento gerou uma tabela extra pois há particularidades nesse relacionamento. Um usuário pode não estar associado a uma *tag*. Uma *tag* pode não estar associada a um usuário. Há ainda data e hora de início desse relacionamento, que são informações que somente irão existir no momento que um usuário estiver vinculado com a sua *tag*.

FIGURA 22 - RELACIONAMENTO USUÁRIO TAG


rel_usuario_tag	
	id: INTEGER
	id_usuario: INTEGER
	id_tag: INTEGER
	data_inicio: DATETIME

FONTE: Os autores (2016).

3.5 SALA

Na próxima tabela (FIGURA 24) há um simples cadastro de sala. Ele contém a numeração da sala (chave primária), um nome e a descrição dela. Esta tabela será referenciada pelas tabelas de acesso e de permissão.

FIGURA 23 - SALA



sala	
	id: INTEGER
	numero: INTEGER
	nome: VARCHAR(20)
	descricao: VARCHAR(100)

FONTE: Os Autores (2016).

3.6 TAG

A tabela referente a *tag* (FIGURA 25) vem a seguir, sendo uma das mais requisitadas ao longo da modelagem. São ao todo três tabelas que fazem referência a ela. Após o vínculo de uma *tag* com o seu usuário, torna-se desnecessário referenciá-lo, já que a *tag* já está vinculada a ele e através de consultas SQL pode-se descobrir qual o usuário relacionado. Esta *tag* tem o seu código, que é único, e também uma categoria, que a separa em comum e master.

FIGURA 24 - TAG




tag	
	id: INTEGER
	categoria: INTEGER(4)
	codigo: INTEGER
	senha: VARBINARY

FONTE: Os Autores (2016).

3.7 USUÁRIO

Por fim temos a tabela (FIGURA 26) que representa o usuário no sistema. Nela está contida a matrícula (chave primária), identificação e as chaves estrangeiras que fazem referência a categoria e a *tag*. Pelo fato deste usuário estar vinculado a uma *tag*, faz-se desnecessário referenciá-lo nas demais tabelas, já que a *tag* já o identifica.

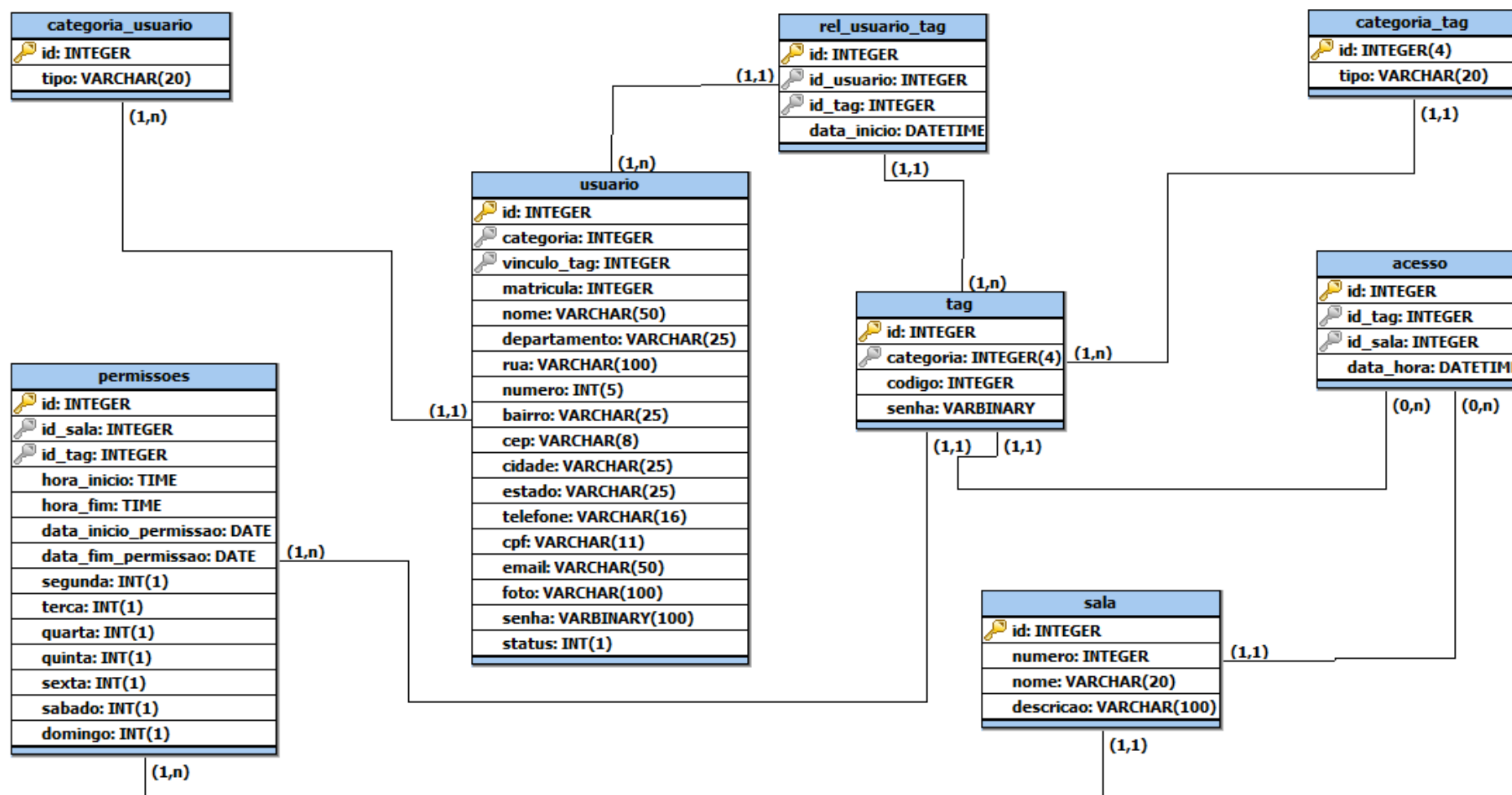
FIGURA 25 - USUÁRIO

usuario	
	id: INTEGER
	categoria: INTEGER
	vinculo_tag: INTEGER
	matricula: INTEGER
	nome: VARCHAR(50)
	departamento: VARCHAR(25)
	rua: VARCHAR(100)
	numero: INT(5)
	bairro: VARCHAR(25)
	cep: VARCHAR(8)
	cidade: VARCHAR(25)
	estado: VARCHAR(25)
	telefone: VARCHAR(16)
	cpf: VARCHAR(11)
	email: VARCHAR(50)
	foto: VARCHAR(100)
	senha: VARBINARY(100)
	status: INT(1)

FONTE: Os Autores (2016).

Com a apresentação da tabela de usuário fecha-se o esquema lógico do sistema. Na imagem (FIGURA 27) pode-se ver todo o relacionamento entre as tabelas bem como as cardinalidades, tudo devidamente normalizado.

FIGURA 26 - MODELAGEM LÓGICA



FONTE: Os Autores (2016).

3.8 CONCLUSÃO

Como pôde ser visto, uma boa modelagem lógica pode trazer um entendimento melhor do funcionamento do sistema antes da sua implantação. Pode-se observar com mais clareza o relacionamento entre as tabelas e com isso a modelagem física torna-se mais precisa, reduzindo as chances de problemas. As tabelas precisam ser normalizadas, ou seja, sem excesso e redundância de dados, fazendo com que cada tabela individual seja fácil de ser compreendida, pois guarda somente as informações necessárias. Com tudo isso pronto, o próximo passo já pode ser dado, que é a modelagem física.

4 CAPÍTULO 3 – MODELO FÍSICO DE DADOS

A modelagem física representa o nível mais baixo de abstração do modelo de dados. Nele são descritos como os dados são armazenados, acessados e manipulados. Essa etapa requer que a tecnologia já esteja definida, pois ela pode ter particularidades em relação às outras.

Nessa etapa o foco são os métodos de armazenamento e de acesso. E não há uma independência de hardware ou software como há nos níveis mais acima na modelagem. Tudo isso é executado através de comandos SQL. Estes comandos desse capítulo serão criados com base no MySQL. Ele foi escolhido pelo fato de ser aberto, robusto e amplamente distribuído no mercado.

4.1 SCRIPT DE BANCO DE DADOS

O modelo de dados da aplicação desenvolvida é composto das tabelas e relacionamentos a seguir.

```
CREATE TABLE `sala` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `numero` int(11) DEFAULT NULL,  
  `nome` varchar(20) DEFAULT NULL,  
  `descricao` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `categoria_tag` (
```

```

`id` int(4) NOT NULL AUTO_INCREMENT,

`tipo` varchar(20) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `tag` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`id_categoria` int(4) DEFAULT NULL,

`codigo` varchar(11) DEFAULT NULL,

`senha` varbinary(100) NOT NULL,

PRIMARY KEY (`id`),

KEY `fk_tag_categoria_tag` (`id_categoria`),

CONSTRAINT `fk_tag_categoria_tag` FOREIGN KEY (`id_categoria`)
REFERENCES `categoria_tag` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

ALTER TABLE `tag` ADD UNIQUE(`codigo`);

```

```

CREATE TABLE `acesso` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`id_tag` int(11) DEFAULT NULL,

`id_sala` int(11) DEFAULT NULL,

`data_hora` datetime DEFAULT NULL,

```

```

PRIMARY KEY (`id`),

KEY `fk_acesso_tag` (`id_tag`),

KEY `fk_acesso_sala` (`id_sala`),

CONSTRAINT `fk_acesso_sala` FOREIGN KEY (`id_sala`) REFERENCES `sala`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk_acesso_tag` FOREIGN KEY (`id_tag`) REFERENCES `tag` (`id`)
ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `categoria_usuario` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`tipo` varchar(20) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `usuario` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`id_categoria` int(11) DEFAULT '2',

`matricula` int(11) DEFAULT NULL,

`nome` varchar(50) DEFAULT NULL,

`departamento` varchar(25) DEFAULT NULL,

`rua` varchar(100) DEFAULT NULL,

`numero` int(5) DEFAULT NULL,

`bairro` varchar(25) DEFAULT NULL,

```

```

`cep` varchar(9) DEFAULT NULL,

`cidade` varchar(25) DEFAULT NULL,

`estado` varchar(25) DEFAULT NULL,

`telefone` varchar(16) DEFAULT NULL,

`cpf` varchar(11) DEFAULT NULL,

`email` varchar(50) DEFAULT NULL,

`foto` varchar(100) DEFAULT NULL,

`senha` varbinary(100) DEFAULT NULL,

`status` int(1) DEFAULT 1,

PRIMARY KEY (`id`),

KEY `fk_usuario_categoria_usuario` (`id_categoria`),

CONSTRAINT `fk_usuario_categoria_usuario` FOREIGN KEY (`id_categoria`)
REFERENCES `categoria_usuario` (`id`) ON DELETE NO ACTION ON UPDATE NO
ACTION

) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=utf8;

SELECT * FROM gandalf.categoria_usuario;

ALTER TABLE `usuario` ADD UNIQUE(`cpf`);

ALTER TABLE `usuario` ADD UNIQUE(`email`);

CREATE TABLE `permissao` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`id_sala` int(11) DEFAULT NULL,

```

```

`id_tag` int(11) DEFAULT NULL,

`hora_inicio` time DEFAULT NULL,

`hora_fim` time DEFAULT NULL,

`data_inicio_permissao` date DEFAULT NULL,

`data_fim_permissao` date DEFAULT NULL,

`segunda` int(1) DEFAULT NULL,

`terca` int(1) DEFAULT NULL,

`quarta` int(1) DEFAULT NULL,

`quinta` int(1) DEFAULT NULL,

`sexta` int(1) DEFAULT NULL,

`sabado` int(1) DEFAULT NULL,

`domingo` int(1) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk_permissao_tag` (`id_tag`),

KEY `fk_permissao_sala` (`id_sala`),

CONSTRAINT `fk_permissao_sala` FOREIGN KEY (`id_sala`) REFERENCES
`sala` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk_permissao_tag` FOREIGN KEY (`id_tag`) REFERENCES `tag`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `rel_usuario_tag` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`id_usuario` int(11) DEFAULT NULL,

```

```

`id_tag` int(11) DEFAULT NULL,

`data_inicio` datetime DEFAULT CURRENT_TIMESTAMP,

`data_fim` datetime DEFAULT CURRENT_TIMESTAMP,

PRIMARY KEY (`id`),

KEY `fk_rel_usuario_tag_tag` (`id_tag`),

KEY `fk_rel_usuario_tag_usuario` (`id_usuario`),

CONSTRAINT `fk_rel_usuario_tag_tag` FOREIGN KEY (`id_tag`) REFERENCES
`tag` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk_rel_usuario_tag_usuario` FOREIGN KEY (`id_usuario`)
REFERENCES `usuario` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE IF NOT EXISTS `pais` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`nome` varchar(60) DEFAULT NULL,

`sigla` varchar(10) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=2 ;

```

```

CREATE TABLE IF NOT EXISTS `estado` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`nome` varchar(75) DEFAULT NULL,

`uf` varchar(5) DEFAULT NULL,

`pais` int(7) DEFAULT NULL,

```

```

PRIMARY KEY (`id`),

KEY `fk_Estado_pais` (`pais`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=28 ;

CREATE TABLE IF NOT EXISTS `cidade` (

`id` int(11) NOT NULL AUTO_INCREMENT,

`nome` varchar(120) DEFAULT NULL,

`estado` int(5) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk_Cidade_estado` (`estado`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=5565 ;

```

De acordo com o script acima, a estrutura do banco de dados é composta por usuários, *tags*, salas, permissões e acessos. Existem outras tabelas, que servem para classificar os dados principais, como categorias de usuários e categorias de *tags*, relacionar usuários com *tags*, de maneira que vários usuários possam ter várias *tags* ao longo do tempo, e tabelas auxiliares ao cadastro de usuário.

Um usuário é identificado pela matrícula, tem um nome, uma categoria e vínculo com várias *tags*. A regra de que um usuário comum tenha apenas uma *tag* é implementada na aplicação.

A *tag* é identificada pelo código e tem uma categoria. A sala é identificada pelo número, tem nome e descrição. Permissões relacionam salas e *tags*, definindo o dia da semana, a hora de início e a hora de fim da permissão de acesso da *tag*. Além disso, os campos *data_inicio_permissao* e *data_fim_permissao* possibilitam que uma *tag* seja associada temporariamente a uma sala e que o histórico de acessos seja mantido após a expiração do período da permissão. A tabela acesso

registra todas as entradas de uma *tag* em uma sala, a data e a hora nas quais ocorreu o acesso.

4.2 CONCLUSÃO

A etapa da modelagem física dos dados de um sistema serve para definir a estrutura do banco de dados. Com isso, é possível iniciar o planejamento de como ocorre o fluxo de dados no sistema.

Vimos que o sistema baseia-se principalmente no relacionamento entre a *tag* e as demais entidades, refletindo o fato de que o sistema é baseado nas *tags*. A modelagem demonstra como os dados são armazenados e como se relacionam.

5 CAPÍTULO 5 – DIAGRAMA DE FLUXO DE DADOS

O Diagrama de Fluxo de Dados (DFD) é uma ferramenta que nos permite visualizar um sistema como uma rede de processos funcionais, depósitos e setas que representam o fluxo de dados entre eles.

Ele é uma das principais formas de modelagem de sistemas. Uma boa modelagem irá colocar no papel todos os processos do sistema e os seus relacionamentos com os outros processos e depósitos de dados. É uma representação simplificada do sistema. Com isso o sistema se torna mais fácil de visualizar e até mesmo um leigo pode entender o funcionamento do mesmo.

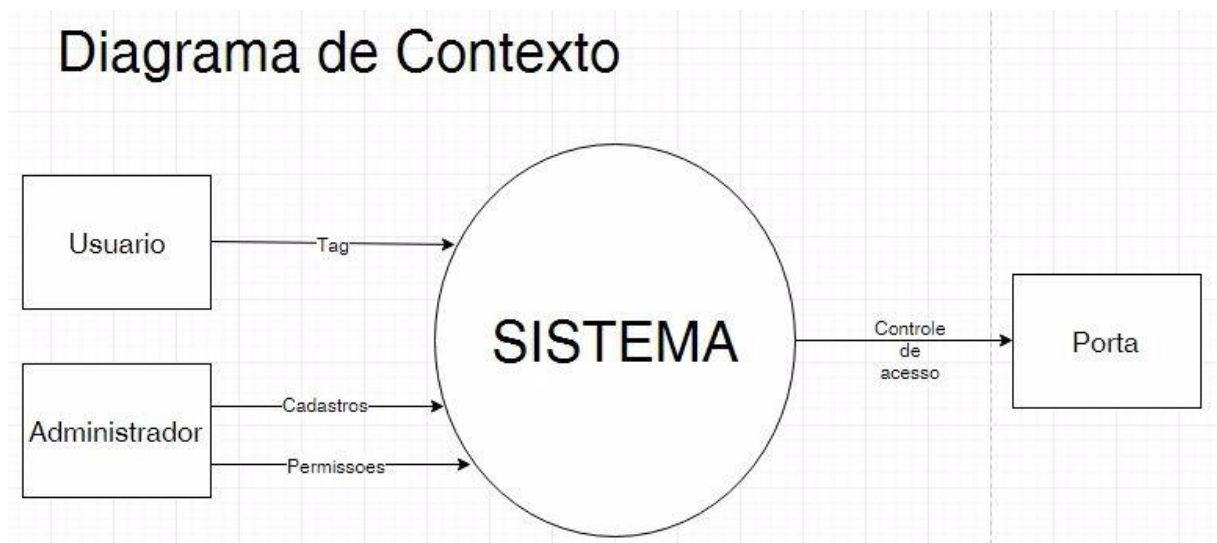
Softwares de sucesso geralmente tem por base uma boa modelagem. Essa modelagem através de diagramas pode evitar erros e facilitar o desenvolvimento do programa. Quanto maior a complexidade do sistema mais necessário se faz o uso de algum tipo de modelagem documentada pois as futuras manutenções podem acabar gerando uma mão-de-obra maior ao administrador e com maiores chances de erros.

A modelagem adotada neste sistema será com a utilização de um Diagrama de Contexto e os DFD's níveis 0 e 1. O Diagrama de Contexto apresenta o sistema como um único processo e exibe o seu relacionamento com as entidades externas. O DFD nível 0 demonstra o sistema todo em funcionamento, enquanto o DFD nível 1 seleciona um determinado processo e exibe-o mais detalhadamente.

5.1 DIAGRAMA DE CONTEXTO

O sistema é simples e por conta disso há poucas entidades externas envolvidas. Basicamente o usuário e o administrador (FIGURA 28). E o relacionamento deles gera uma saída que possibilita o acesso a porta através da TAG. O administrador cadastra e determina permissões ao usuário.

FIGURA 27 – DIAGRAMA DE CONTEXTO



FONTE: Os autores (2016).

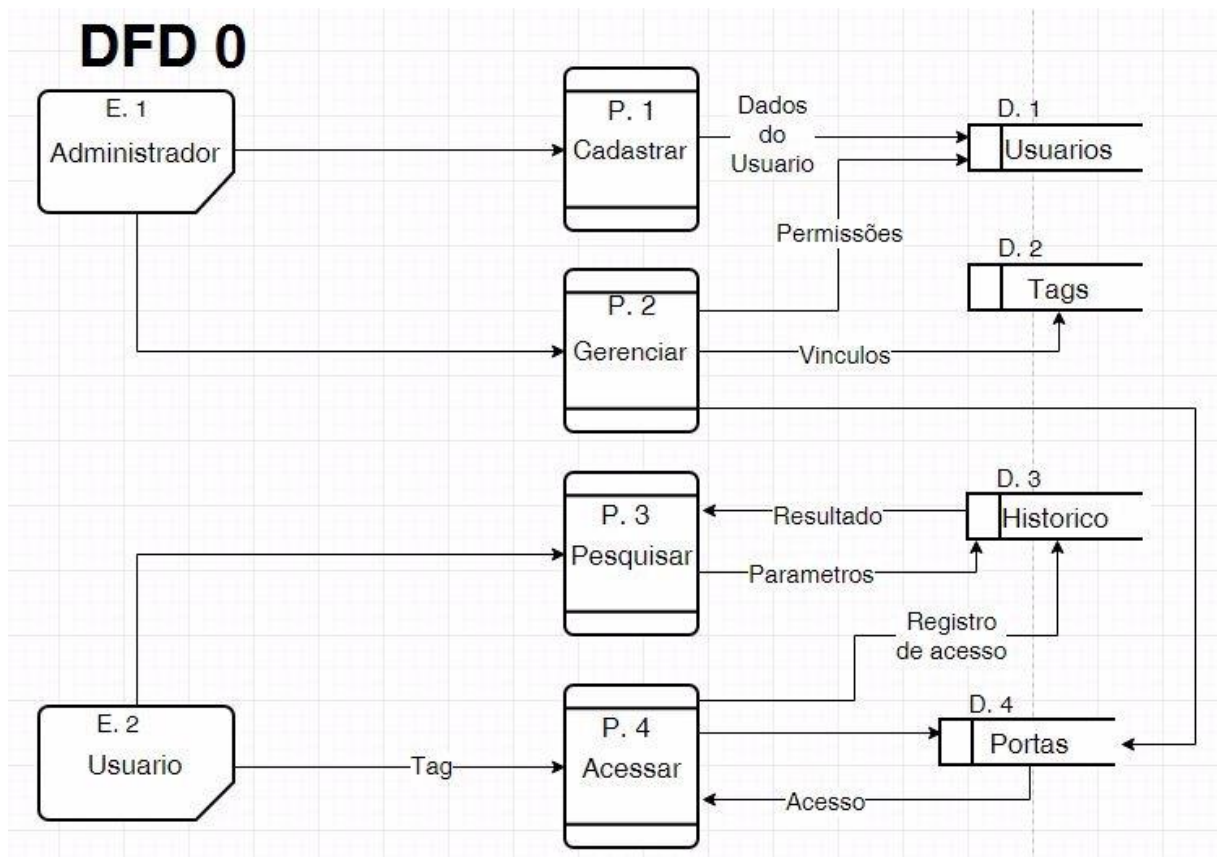
5.2 DFD – NÍVEL 0

O DFD Nível 0 é a modelagem de fluxo de dados mais global entre os DFD's pois exibe, de forma simplificada, todo o sistema e suas interações. É um passo a frente do Diagrama de Contexto. Nele já é possível verificar as interações das duas entidades externas com o sistema de modo mais detalhado, embora ainda simplificado. Há quatro grandes processos que são o cadastro, gerenciamento, pesquisa e acesso.

O processo Cadastrar (FIGURA 29) recebe os dados do usuário e irá trabalhar em conjunto com o processo Gerenciar para emitir as permissões deste usuário, bem como o seu vínculo com a *tag*. Após a conclusão deste cadastro essa *tag* será adicionada ao depósito de *tags* e o usuário ao depósito de usuários. Esses depósitos irão guardar todas as informações para possibilitar validações e consultas. Há também o depósito de portas que permite cadastrar novas portas ou exibir as que já estão cadastradas para vincular ao novo usuário.

A seguir é apresentado o processo referente ao Usuário. Ele com a sua *tag* previamente cadastrada irá realizar o Acesso, aonde é feito uma validação, e em seguida liberado o acesso. Também é armazenado num depósito próprio o registro desse acesso, para posterior consulta. Há também o processo relacionado a busca, aonde baseado em parâmetros informados pelo usuário o sistema efetua a busca no histórico.

FIGURA 28 – DFD NÍVEL 0



FONTE: Os autores (2016).

5.3 DFD NÍVEL 1

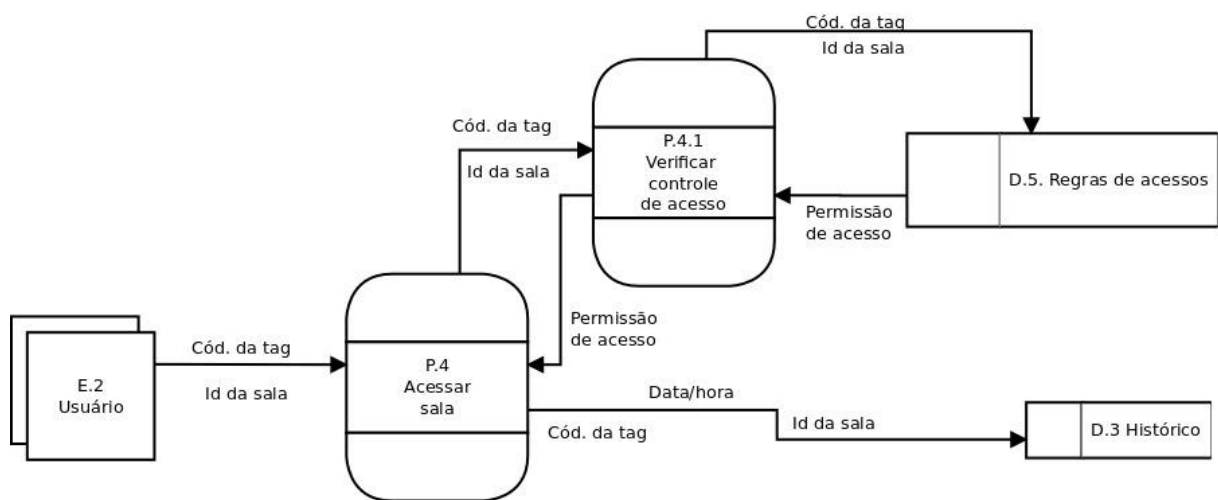
Neste nível o diagrama se aprofunda nos processos do sistema. Há processos que são complexos, mas há alguns que se aprofundados nos apresentam alguns novos processos menores, que num nível acima do DFD não é importante aparecer. Neste sistema detectou-se dois processos mais complexos do que os demais: Acessar Sala e Controle de Acesso.

No processo que controla o acesso da sala (FIGURA 30) observa-se um processo auxiliar para verificar o controle de acesso e que realiza uma consulta num

depósito com as regras dos acessos. Este processo é que irá conceder a permissão para o usuário acessar a sala. Observa-se também diversos fluxos menores nesse diagrama como códigos e ids, que são fundamentais nessas transações de informações.

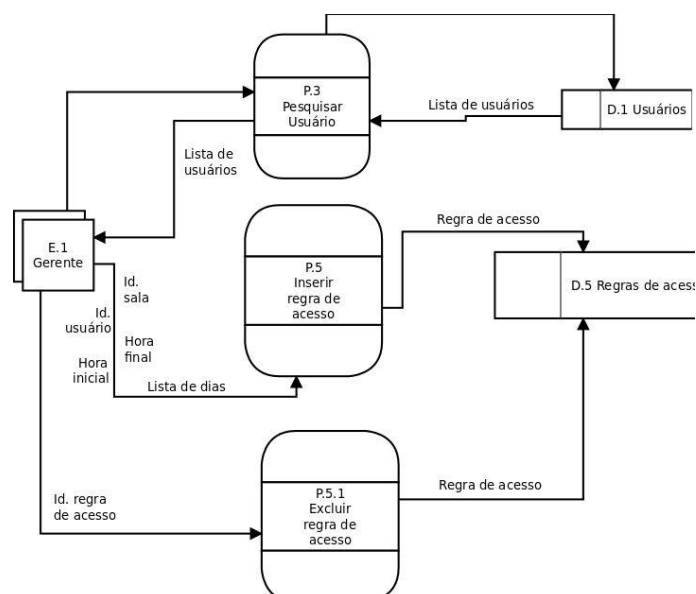
Já no processo que controla os acessos é aonde são criadas as regras que irão permitir ou não um usuário de acessar a sala que está requisitando o acesso. Há também a possibilidade da exclusão de um critério definido anteriormente. Essas informações após processadas irão compor o depósito das regras de acesso.

FIGURA 29 – DFD 1 – ACESSO SALA



FONTE: Os autores (2016).

FIGURA 30 – DFD 1 – CONTROLE DE ACESSO



FONTE: Os autores (2016).

5.4 CONCLUSÃO

A modelagem de fluxo de dados através de diagramas mostra-se muito importante embora alguns profissionais resistam em utilizar esta ferramenta. Enquanto um sistema simples dispense esse tipo de ferramenta os mais complexos precisam de algo assim para poder controlar melhor o sistema no futuro em manutenções. Porém, mesmo que o sistema seja simples, um cliente leigo pode ter dificuldades em compreender o sistema que está adquirindo. Nesse caso um diagrama simples pode resolver. Essas ferramentas são de grande utilidade na compreensão de qualquer tipo de sistema.

6 FONTES

O desenvolvimento do projeto foi possível graças a utilização de variadas ferramentas que facilitam a programação em alto nível. O objetivo aqui é explicar os motivos pelos quais cada uma destas ferramentas foram escolhidas.

Para começar, utilizamos o software editor de texto, o Word. O mesmo foi utilizado para a confecção deste texto. A escolha pelo programa se deu pela grande familiaridade que todos os integrantes do grupo possuem com o mesmo.

Para a prototipação de telas o “Balsamiq Mockups” foi o escolhido. Esta ferramenta possibilitou a fácil visualização das telas, possibilitando assim correções e ajustes que seriam bastante difíceis de se visualizar em abstrações.

Outro software essencial, o BrModelo, foi utilizado para a criação do Modelo Relacional. Ele é simples e intuitivo, tornando bem simplificado a criação e manipulação das tabelas e relacionamentos. A facilidade em gerar o Esquema Físico de dados foi outro diferencial, poupando tempo. O script já estava adequado ao MySQL, que foi escolhido devido à larga utilização no mercado e também pelo fato do mesmo estar integrado com o ambiente do XAMPP. Este nos ofereceu um pacote com Apache, PhpMyAdmin e PHP, necessários para o desenvolvimento do projeto e sem grande complexidade.

Para a criação do DFD, foram utilizados o Dia e o Draw.io. O Dia fornece uma ampla gama de comandos, que combrem as necessidades de uma boa modelagem. O Draw.io é um sistema hospedado diretamente na web, o que faz com que o trabalho seja acessível em qualquer local com acesso à internet.

Partindo então para a implementação do projeto, a plataforma de controle de versões GitHub foi escolhida como ambiente de desenvolvimento, utilizada de maneira contínua durante o desenvolvimento.

A interface visual foi desenvolvida em HTML 5 e CSS 3, utilizando-se largamente o framework Bootstrap. A interatividade do sistema foi possível através da utilização da popular biblioteca Java Script, JQuery. Com o auxílio desta biblioteca foi possível realizar algumas validações do lado do cliente e criar as máscaras utilizadas nos formulários.

O Webservice “República Virtual” foi selecionado para que os formulários com endereços pudessem ser capturados e preenchidos automaticamente através do preenchimento do CEP. Com isso aumentamos a confiabilidade das informações depositadas no banco de dados.

Para o desenvolvimento do backend as ferramentas IDE Sublime Text e Notepad++, pela versatilidade, foram utilizados para construir todo o código html,

php e javascript. Essa versatilidade tornou o trabalho mais fácil, já que com a mesma ferramenta pode abrir documentos de extensões variadas.

Conclui-se que com o uso dessas ferramentas o trabalho pôde ser desenvolvido com mais facilidade e confiabilidade. A versatilidade e integração que as ferramentas oferecem facilitou a interação entre os membros da equipe de desenvolvimento. Por outro lado, o uso de várias ferramentas enriquece o aprendizado, já que muitas delas são amplamente utilizadas no mercado.

7 CONCLUSÃO

O desenvolvimento deste projeto envolveu muita pesquisa e testes por parte da equipe. A experiência da busca pela melhor solução enriqueceu o aprendizado dos membros, que cumpriram os requisitos passados pelo professor da disciplina. Os objetivos traçados pela equipe foram alcançados satisfatoriamente.

O sistema gerado é prático e ergonômico, de modo que o usuário pode operá-lo com facilidade em poucos cliques. Procuramos fornecer ao usuário uma experiência agradável, e para isso nos colocamos em seu lugar. Além disso, construímos estruturas funcionais, de fácil manutenção. Foi tomada a iniciativa, ainda, de propor soluções diferentes as exigidas pelo orientado do projeto.

Como proposto no início deste trabalho, foi possível colocar em prática diversos conhecimentos adquiridos em diferentes disciplinas, como a implementação do bando de dados e a construção de algoritmos em diferentes linguagens. Sendo assim, concluímos que este projeto obteve sucesso e contribuiu para a formação dos envolvidos.

8 REFERÊNCIAS

PROTOTIPAÇÃO e sua Importância no Desenvolvimento de Software. Disponível em: <<http://goo.gl/e0Ns2l>>. Acesso em 14 abr. 2016.

A IMPORTÂNCIA dos Protótipos no Desenvolvimento de Sistemas. Disponível em: <<http://goo.gl/k4ZnpY>>. Acesso em 14 abr. 2016.

7 MANEIRAS de Melhorar a Ergonomia do seu Site. Disponível em: <<http://goo.gl/y60sDM>>. Acesso em 14 abr. 2016.

5 USABLY Lessons from Website Eye Tracking Studies. Disponível em: <<http://goo.gl/rH7dG>>. Acesso em 15 abr. 2016.

DEFINIÇÕES de Ergonomia e Usabilidade. Disponível em: <<http://goo.gl/l6Bk9Z>>. Acesso em 14 abr. 2016.

ERGONOMIA Ciência do Conforto. Disponível em: <<http://goo.gl/rRXMSs>>. Acesso em 14 abr. 2016.

USANDO as Cores Certas para o Seu Site. Disponível em: <<http://goo.gl/kLoHND>>. Acesso em 15abr. 2016.

COMO Escolher as Cores do seu Site. Disponível em: <<http://goo.gl/ZTJwLP>>. Acesso em 15 abr. 2016.

MODELAGEM Conceitual, Lógica e Física de Dados. Disponível em: <<http://goo.gl/w9g7ro>>. Acesso em 28 abr. 2016.

POR QUÊ Construir um Modelo de Dados Lógico – Parte I. Disponível em: <<http://goo.gl/9Oe8B6>>. Acesso em 28 abr. 2016.

MODELAGEM de Dados. Disponível em: <<https://goo.gl/F0btjR>>. Acesso em 28 abr. 2016.

POSTGRESQL x MySQL. Qual escolher? Disponível em: <<http://goo.gl/1GPU7B>>. Acesso em 13 mai. 2016.

A IMPORTÂNCIA da Modelagem de Objetos no Desenvolvimento de Sistemas. Disponível em: <<http://goo.gl/uhBshk>>. Acesso em 13 mai. 2016.