

UNIVERSIDADE ESTADUAL PAULISTA “JULIO DE MESQUITA FILHO”

CAMPUS BAURU

DOCUMENTAÇÃO REPMANAGER

RELATÓRIO FINAL DE TRABALHO

PAULO EDUARDO MANZONE MAIA

RELATÓRIO FINAL DE ATIVIDADES

Relatório elaborado pelo aluno **Paulo Eduardo Manzone Maia** RA: **161022261**, matriculado no curso de Bacharelado em Ciência da Computação, do Campus de Bauru, orientado pelo professor Aparecido Nilceu Marana e apresentado como requisito formal para aprovação na disciplina de Banco de Dados I.

BAURU

2018

RESUMO

Repúblicas estudantis são, geralmente, locais de baixo nível de organização. No entanto, tal organização é fundamental para a boa convivência entre seus moradores, que em sua maioria estão vivendo a primeira experiência de morar fora da casa de seus pais. Para auxiliar neste quesito, foi desenvolvido um software de gestão de repúblicas, o RepManager, que visa disponibilizar ferramentas organizacionais para essas repúblicas.

Palavras-chave: República; Gerenciamento; Organização.

SUMÁRIO

| | |
|------------------------------------|----|
| RELATÓRIO FINAL DE ATIVIDADES..... | ii |
| 1. INTRODUÇÃO..... | 4 |
| 2. MÉTODOS E PROCEDIMENTOS..... | 5 |
| 3. RESULTADOS E DISCUSSÕES..... | 6 |
| 4. CONCLUSÕES..... | 7 |

1. INTRODUÇÃO

1.1. OBJETIVO

O objetivo geral deste relatório é descrever as atividades realizadas a partir do trabalho proposto, que envolve atividades como modelagem, normatização e implementação de banco de dados e desenvolvimento de software

1.2. ORGANIZAÇÃO DO TRABALHO

Este relatório foi organizado em 3 capítulos.

No primeiro, faz-se uma introdução a este trabalho, enquanto o capítulo 2 trata dos métodos e procedimentos empregados para a realização do projeto.

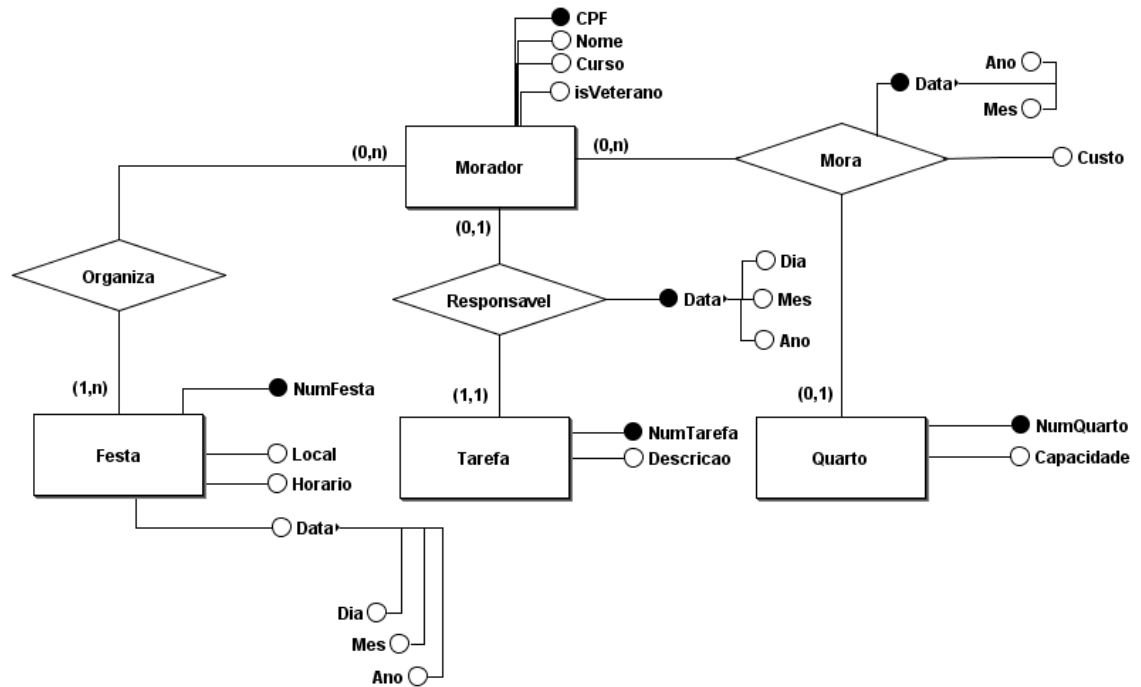
O capítulo 3 trata das conclusões obtidas com o desenvolvimento do projeto e também faz uso do subcapítulo de recomendações para trabalhos futuros para indicar condutas que facilitariam a realização do mesmo.

1.3. MINIMUNDO

Uma república bauruense possui diversos problemas relacionados a organização. Seus moradores moram em quartos que podem ser ocupados por mais de um morador. Cada tarefa é responsabilidade de um morador em um dia específico, sendo possível alocar a mesma tarefa para duas pessoas diferentes em datas distintas. É necessário para fins de gestão financeira, registros sobre a estadia dos moradores nos quartos contendo o mês e o ano da estadia, e também de seu custo (que pode variar dependendo do mês, por motivos de reajuste, quantidade de pessoas dividindo o mesmo quarto ou algum outro motivo). Nessa república são realizadas festas visando arrecadar dinheiro para os moradores, que são organizadas exclusivamente por seus moradores veteranos. Para sua realização, é

definida uma data, um horário e um grupo de moradores veteranos que será responsável por aquela festa.

1.4. MODELO ENTIDADE RELACIONAMENTO



Modelo Entidade Relacionamento – Repúblicas Bauruenses

2. MÉTODOS E PROCEDIMENTOS

2.1. MÉTODOS

Para a modelagem do banco de dados, houve primeiramente uma análise do problema. Após determinadas suas entidades e relacionamentos houve a necessidade de verificar sua forma normal e normalizar as tabelas, utilizando os métodos de normalização ensinados em aula.

Após a normalização, foi desenvolvido o software utilizando Java, na IDE Eclipse, utilizando JDBC e DAO para interação com o banco de dados, que foi construído com MySql rodando em um servidor local pelo Wamp.

O projeto de software foi baseado no padrão de arquitetura MVC.

2.2. PROCEDIMENTOS

Os procedimentos seguidos foram:

1. Modelagem do Banco de Dados
2. Normalização do Banco de Dados
3. Criação do Banco de Dados em MySQL
4. Definição das classes necessárias
5. Desenvolvimento dos JavaBeans
6. Desenvolvimento das Interfaces com usuário
7. Desenvolvimento das Interfaces de banco, tais como:
 1. ConnectionFactory
 2. Data Access Objects
8. Integração da “lógica de negócio”

3. RESULTADOS E DISCUSSÕES

3.1. RESULTADOS

Os resultados do presente trabalho foram quase satisfatórios, visto que pela falta de otimização nas queries e pela utilização de alguns artifícios via software não foram empregadas todas as técnicas e boas práticas ensinadas.

3.2. DISCUSSÕES

Ao se analisar a modelagem e o problema proposto, percebe-se certos requisitos funcionais do programa, como a capacidade dos quartos de hospedar moradores, a restrição de que somente veteranos organizem festas. Ambos podem ser satisfeitos tanto via banco de dados, quanto via aplicação.

A solução implementada foi a utilização de um atributo adicional veterano (o que elimina a necessidade de duas tabelas o que causaria a necessidade de inserções e deleções ao se alterar o status de veterano). Apesar de não ser otimizado, é o programa verifica o status de veterano conforme cpf via query SQL, retornando o morador como um todo e verificando no bean o seu atributo booleano 'veterano'. Uma saída mais otimizada seria a query retornar apenas o status e o mesmo ser verificado diretamente da query.

Para a restrição de capacidade de quartos, foi feita uma query count, que contava quantas pessoas já moravam nesse quarto e era comparado via software com a capacidade do quarto em si, o que poderia ser resolvido através de triggers, o que otimizaria o processo.

Embora as soluções pudessem ser otimizadas, o fator tempo de execução levou ao desenvolvimento do projeto como está sendo apresentado, já que as soluções utilizadas foram de rápida implementação devido a conhecimento na linguagem.

4. CONCLUSÕES

4.1. CONCLUSÕES

Conforme comentado ao longo das seções, foram empregadas várias tecnologias já existentes. Apesar disso houve muito retrabalho, principalmente na interação com o banco de dados, já que existem diversas tecnologias que interfaceiam esta conexão, como Hibernate e JPA, que não foram utilizadas.

Percebe-se que tanto funções Java, quanto queries e triggers SQL podem resolver o mesmo tipo de problema. Equivocadamente tende-se a utilizar solução por software de aplicação, no entanto em muitos dos casos as soluções em SQL são mais otimizadas.

4.2. RECOMENDAÇÕES PARA OUTROS TRABALHOS

A atividade permitiu observar que pode-se desenvolver aplicações com interação com banco de dados de forma mais modular, visando o reuso de código, já que grande parte do código foi simplesmente retrabalho e adaptações para cada caso, o que poderia ser diminuído com a modularidade.