

Universidade Estadual do Sudoeste da Bahia – UESB
Departamento de Ciências Exatas e Tecnológicas – DCET
Colegiado do Curso de Ciência da Computação – CCComp
Banco de dados I

DOCUMENTAÇÃO DE SISTEMA PARA O RESTAURANTE UNIVERSITÁRIO

Deyvson Luís de Oliveira Nogueira, Edson Araujo de Souza Neto, Pedro Lucca
Silva Martins, Matheus Souza Lopes e Christian Kauan Moreno Silveira.

Histórico de Alteração

Data	Versão	Descrição	Autor
30/04/2024	1.0	Criação do documento de requisitos	Edson Araújo de Souza Neto
28/05/2024	1.1	Criação do modelo conceitual	Christian Kauan Moreno Silveira
28/05/2024	1.3	Criação do diagrama relacional	Matheus Souza Lopes
05/07/2024	1.4	Criação do Modelo Lógico	Edson Araújo de Souza Neto

Conteúdo

1. Introdução

Este documento especifica os requisitos do: “Sistema para funcionamento eficiente do RU”, fornecendo aos projetistas e desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1 Visão Geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo:

- **Seção 2 – Descrição geral do sistema**
- **Seção 3 – Requisitos**
- **Seção 4 – Modelo Conceitual**
- **Seção 5 – Diagrama Relacional**
- **Seção 6 – Modelo Lógico**
- **Seção 7 - Bibliografia**

1.2 Convenções, termos e abreviações

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos seguidos do identificador do requisito, de acordo com a especificação a seguir: [nome da subseção. identificador do requisito]. Por exemplo, o requisito funcional [Incluir Usuário.RF016] deve estar descrito em uma subseção chamada “Incluir Usuário”, em um bloco identificado pelo número [RF016]. Já o requisito não funcional [Confiabilidade.NF008] deve estar descrito na seção de requisitos não funcionais de Confiabilidade, em um bloco identificado por [NF008].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

2. Descrição geral do sistema

Atualmente o Restaurante Universitário nos oferece café da manhã, almoço e janta. Apesar da ótima comida, os estudantes vivem diariamente com várias dúvidas como: “A fila está grande?”, “Será que dá tempo de almoçar sem perder a próxima aula?”.

Há dois tipos de usuários que consomem no RU, estudante com auxílio, que pode ser PRAE (paga R\$ 2,00), e 50% (paga metade do valor final), externo que paga o valor total, que pode ser estudante sem auxílio, ou não ser estudante.

O problema é visível na hora de almoço, entre 11h às 14h, pois se concentram um grande número de clientes. Os alunos precisam para usufruir do RU entrar em uma fila para pegar um ticket que fica no lado externo do RU, passando o tipo de cliente, assinar o nome, e ir para outra fila para enfim colocar o alimento no prato. **Usuários externos não precisam do ticket** e já podem ir para fila do almoço.

Após colocar o alimento no prato, o usuário se dirige ao caixa para colocar o prato na balança e, **caso seja estudante com auxílio**, digita a matrícula, entrega o ticket para o atendente para o mesmo assinar a quantidade de alimento e por fim o usuário realiza o pagamento. Usuário externo e estudante sem auxílio, apenas realiza o pagamento.

Para café da manhã e janta não é necessário pegar ticket no lado externo do RU, o mesmo é entregue e assinado no caixa, pois o fluxo de pessoas é menor.

No café da manhã e janta cada usuário pode pegar no máximo 06 itens, seja usuário externo ou estudante com auxílio.

É importante dizer que usuários com auxílio não podem repetir alimento utilizando o auxílio, almoçar duas vezes por exemplo.

Cliente com auxílio PRAE só tem acesso a duas refeições distintas por dia, caso ele queira fazer a terceira, será pago o valor total.

2.1 Problemas

É fácil burlar, visto que, na hora de pegar o ticket de auxílio, os funcionários pedem o tipo de usuário e o número na lista, na maioria das vezes, eles nem conferem se seu nome é aquela posição mesmo na lista, e entregam o ticket.

A **PROAPA** recentemente visando evitar falsificações de identidade, começou a exigir um documento com foto de cada estudante com auxílio na hora de pegar o ticket, mas essa abordagem não soluciona o problema. .

2.2 Solução

Com objetivo de evitar a fila do ticket, e trazer uma maior velocidade na fila de colocar o alimento, pensamos em um software que busca reconhecer o tipo de usuário com auxílio através da matrícula, que já é digitado atualmente no caixa.

O sistema vai buscar automaticamente o tipo de usuário e através do peso da balança o quanto ele precisa pagar.

Se não for estudante com auxílio, o atendente seleciona na interface “externo” e continua o atendimento.

Porém, se o ticket existe, é porque é necessário para algum setor que administra, logo, vamos continuar gerando comprovações, estes, através de relatórios, contendo a quantidade de usuários, data, o nome de quem atendeu, etc.

3. Requisitos

3.1 Requisitos Funcionais

Nosso sistema terá dois tipos de usuário que vai de fato utiliza-lo: **atendente** e o **administrador**.

- **[RF001]** Como atendente, eu gostaria de escolher a entre café da manhã, janta e almoço.
- **[RF002]** Como atendente, eu gostaria de obter o tipo de usuário através da matrícula digitada.
- **[RF003]** Como atendente, eu gostaria de colocar a forma de pagamento escolhida pelo usuário.
- **[RF004]** Como atendente, eu gostaria de colocar a quantidade do usuário.
- **[RF005]** Como atendente, eu gostaria de obter o troco automaticamente, caso há.
- **[RF005]** Como atendente, eu gostaria de imprimir um relatório para fim de turno.
- **[RF006]** Como atendente, eu gostaria de colocar os itens, caso seja café da manhã ou janta.
- **[RF007]** Como administrador, eu gostaria de registrar os novos usuários do PRAE no banco de dados do sistema.
- **[RF008]** Como administrador, eu gostaria de registrar os novos usuários que obtiveram o auxílio, seja PRAE ou 50%.
- **[RF009]** Como administrador, eu gostaria de remover usuários que tiveram seu auxílio suspendo do banco de dados do sistema.
- **[RF010]** Como administrador, eu gostaria de registrar um atendente no sistema.

- **[RF011]** Como administrador, eu gostaria de remover um atendente no sistema.

3.2 Requisitos não funcionais

- **[NF001]** Velocidade, visto que essa busca através da matrícula deve ser rápida para não atrasar o atendimento.
- **[NF002]** Estabilidade, o servidor não pode cair para o atendimento ser realizado.
- **[NF003]** Usabilidade, para qualquer atendente conseguir utilizar o software.

3.3 Descrição de casos de uso

1. Atender um usuário no almoço

Ator: Atendente

Fluxo normal:

1. Atendente realiza o login, antes do expediente começar.
2. Atendente seleciona no sistema o tipo de refeição do expediente, neste caso almoço.
3. Atendente pede ao usuário do RU digitar a matrícula.
4. Atendente pede ao usuário colocar o prato na balança.
5. Atendente digita a quantidade de comida no sistema.
6. Atendente verifica o valor a ser pago no sistema pelo usuário.
7. Atendente solicita o pagamento ao usuário.
8. Atendente confirma no sistema a operação.

Extensões

- 3a – Se não possuir auxílio, seleciona EXTERNO.
- 3b – Se for estudante com auxílio, o sistema verifica através da matrícula se é usuário com auxílio PRAE, ou auxílio 50%.
- 3c – Se o teclado do usuário não funciona, o atendente digita com seu teclado.
- 7b – Se tiver troco, aparecerá no sistema o valor do troco.

2. Atender um usuário no café da manhã

Ator: Atendente

Fluxo normal:

1. Atendente realiza o login, antes do expediente começar.

2. Atendente seleciona no sistema o tipo de refeição do expediente, neste caso café da manhã.
3. Atendente pede ao usuário do RU digitar a matrícula.
4. Atendente seleciona os alimentos que o usuário pegou para seu café da manhã.
5. Atendente verifica o valor a ser pago no sistema pelo usuário.
6. Atendente solicita o pagamento ao usuário.
7. Atendente confirma no sistema a operação.

Extensões

- 3a – Se não possuir auxílio, seleciona EXTERNO.
- 3b – Se for estudante com auxílio, o sistema verifica através da matrícula se é usuário com auxílio PRAE, ou auxílio 50%.
- 3c – Se o teclado do usuário não funciona, o atendente digita com seu teclado.
- 7b – Se tiver troco, aparecerá no sistema o valor do troco.

3. Acrescentar novo usuário com auxílio

Ator: Administrador

Fluxo normal:

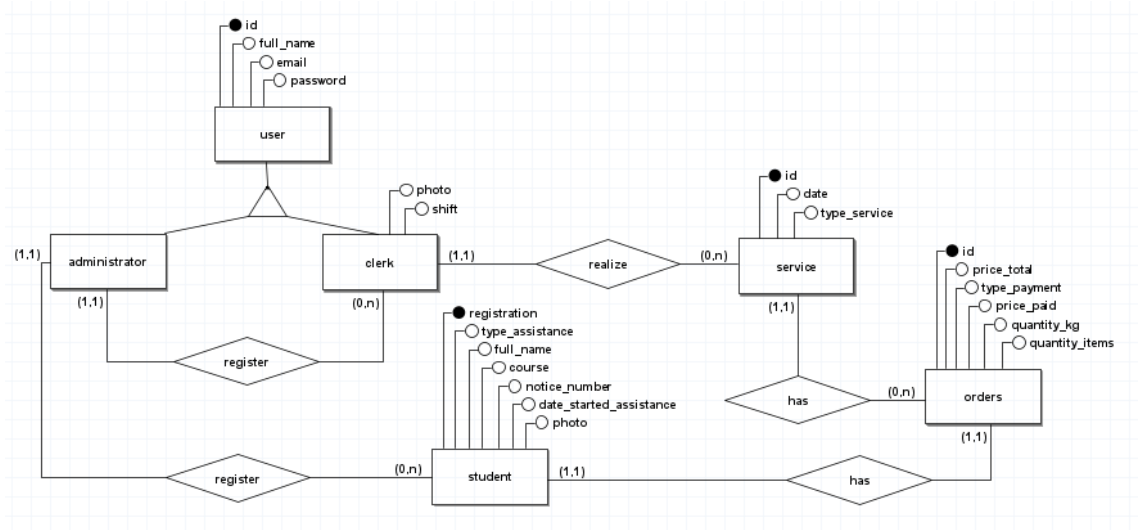
1. Administrador realiza o login.
2. Administrador verifica os aprovados pelo último edital.
3. Administrador cadastra a matrícula, nome e o tipo de auxílio.

Extensões

- 3a – Se for estudante com auxílio 50% que foi aprovado no auxílio PRAE, remove o auxílio 50%.

4. Modelo conceitual

Um modelo conceitual é uma representação abstrata e de alto nível de um sistema em desenvolvimento. Ele descreve os principais conceitos, entidades e suas relações, sem detalhes específicos de implementação. Esse modelo ajuda a definir requisitos, guiar o design e a arquitetura do software, e é essencial para que desenvolvedores e stakeholders compreendam o sistema de forma clara e concisa. Ele pode ser representado por diagramas como ER (entidade-relacionamento) ou de classe, dependendo do foco em estrutura de dados, comportamento ou fluxo de informações.



5. Diagrama Relacional

Um diagrama relacional é uma representação gráfica das relações entre tabelas em um banco de dados relacional. Ele mostra como as tabelas estão interligadas por chaves primárias e estrangeiras, ajudando a visualizar a estrutura e as conexões dos dados de forma clara. Esses diagramas são essenciais para o design e a manutenção de bancos de dados, permitindo aos desenvolvedores e administradores entender e comunicar eficientemente como os dados estão organizados e inter-relacionados no sistema.

• **Administrator** (*id:int , full_name:varchar, email: varchar, password: varchar*)

• **Clerk** (*id:int , full_name:varchar, email:varchar, password:varchar, shift: varchar, photo:blob, id_administrator:int*)

id_administrator referência Administrator

• **Student** (*registration: char, type_assistance:varchar, full_name: varchar, course:varchar, notice_number:varchar, date_started_assistance:date, photo:blob, id_administrator:int*)

id_administrator referência Administrator

• **Service** (*id:int, date_service:timestamp, type_service:varchar, id_clerk:int*)

id_clerk referência Clerk

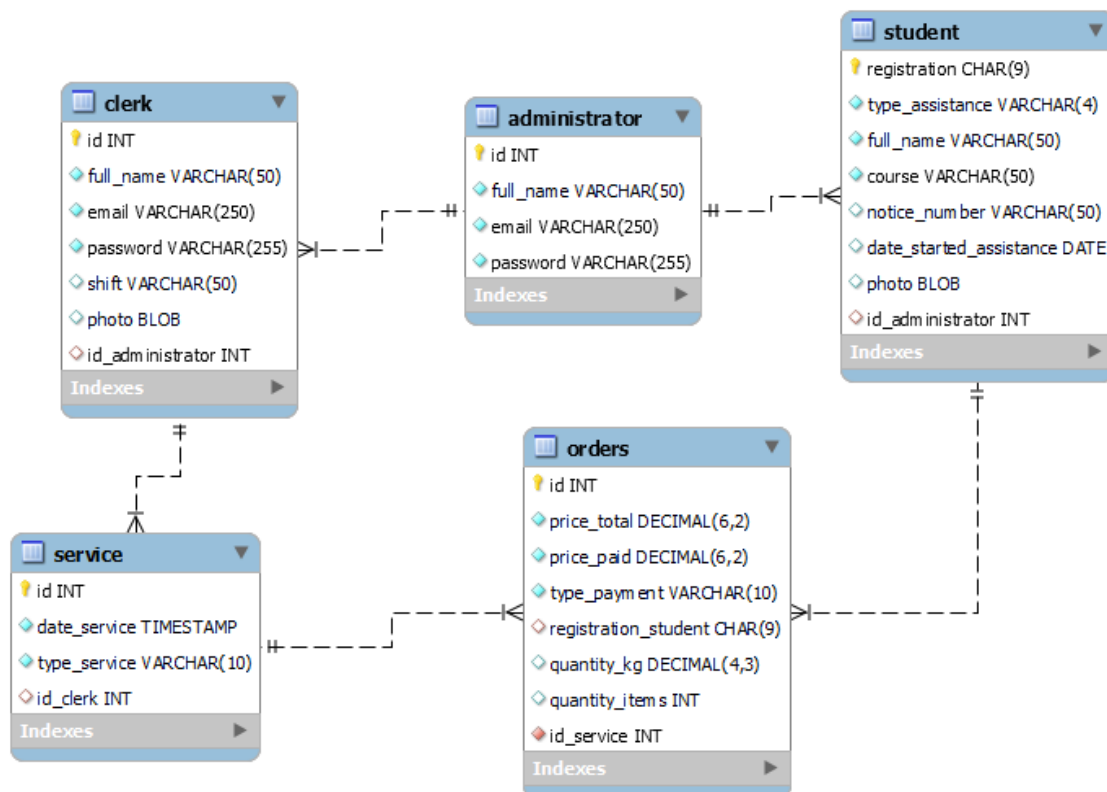
• **Orders** (*id:int, price_total:decimal, price_paid:decimal, type_payment:varchar, registration_student:char, quantity_kg:decimal, quantity_items:int, id_service:int*)

id_service referência Service

registration_student referência Student

6. Modelo lógico

Um modelo lógico é uma representação detalhada dos dados e suas relações após a fase conceitual, descrevendo a estrutura das tabelas, colunas, tipos de dados e as conexões entre elas de forma precisa. Ele transforma o modelo conceitual em um esquema de banco de dados relacional, definindo chaves primárias, estrangeiras e outras restrições para garantir a integridade e eficiência dos dados. Esse modelo é essencial para guiar o desenvolvimento e a manutenção de sistemas, assegurando que a implementação do banco de dados esteja alinhada com os requisitos funcionais e técnicos do sistema.



7. Bibliografia

DEITEL, H. M.; DEITEL, P. J. Java Como Programar: 4 ed.

ELMASRI, Ramez e NAVATHE, Shamkant B. Sistemas de Banco de Dados. Pearson Addison Wesley. 6 a Edição, 2011.