

UNIVERSIDADE FEDERAL DE SERGIPE

ARTHUR HENRIQUE SILVA DE LIMA

DÉBORAH ABREU SALES

GABRIELA ANDRADE MOTA

SANDERSON LACY ALVES DOS SANTOS

VINÍCIUS DIAS VALENÇA

ENGENHARIA DE SOFTWARE 2

ATIVIDADE 2

TJBA - Folha de Pagamento

São Cristóvão

2023

Sumário

1. Conjunto de dados.....	3
2. Repositório.....	3
3. Diagramas UML.....	4
3.1 Diagrama de Sequência.....	4
3.2 Diagrama de Classes.....	5
4. Modelo de dados.....	7
4.1 Modelo Conceitual.....	7
4.2 Modelo Físico.....	8

1. Conjunto de dados

Folha de pagamento TJBA - 2012 a 2023

Os dados que serão utilizados pela aplicação projetada no presente conjunto de documentos estão disponibilizados no portal do Tribunal de Justiça da Bahia. Tendo em vista que a fonte está situada no seguinte endereço web: https://www.tjba.jus.br/transparencia/gestao_de_pessoas/remuneracao/, a equipe utiliza das seções seguintes para caracterizar, definir e modelar o formato dos dados para aplicação em projeto.

Faz se necessário salientar o estado atual dos dados no portal fonte, com pesquisa feita em 28/01/2024, a equipe constata que o portal possui dados desde o ano de 2012 até 2023, sendo assim, ainda não foram postados conteúdos referentes ao ano de 2024. Além disso, os dados estão dispostos no portal nos seguintes formatos: interface web padrão e os tipos de arquivo csv, xml e pdf.

As seguintes seções detalham os diagramas que a equipe de projeto irá utilizar para desenvolver a aplicação, de modo que os diagramas descritos abaixo representarão a primeira versão do banco de dados da aplicação.

2. Repositório

Para o desenvolvimento do projeto foi criado um repositório público no GitHub onde serão inseridos todos os os artefatos elaborados pela equipe. O repositório pode ser acessado através do link a seguir:

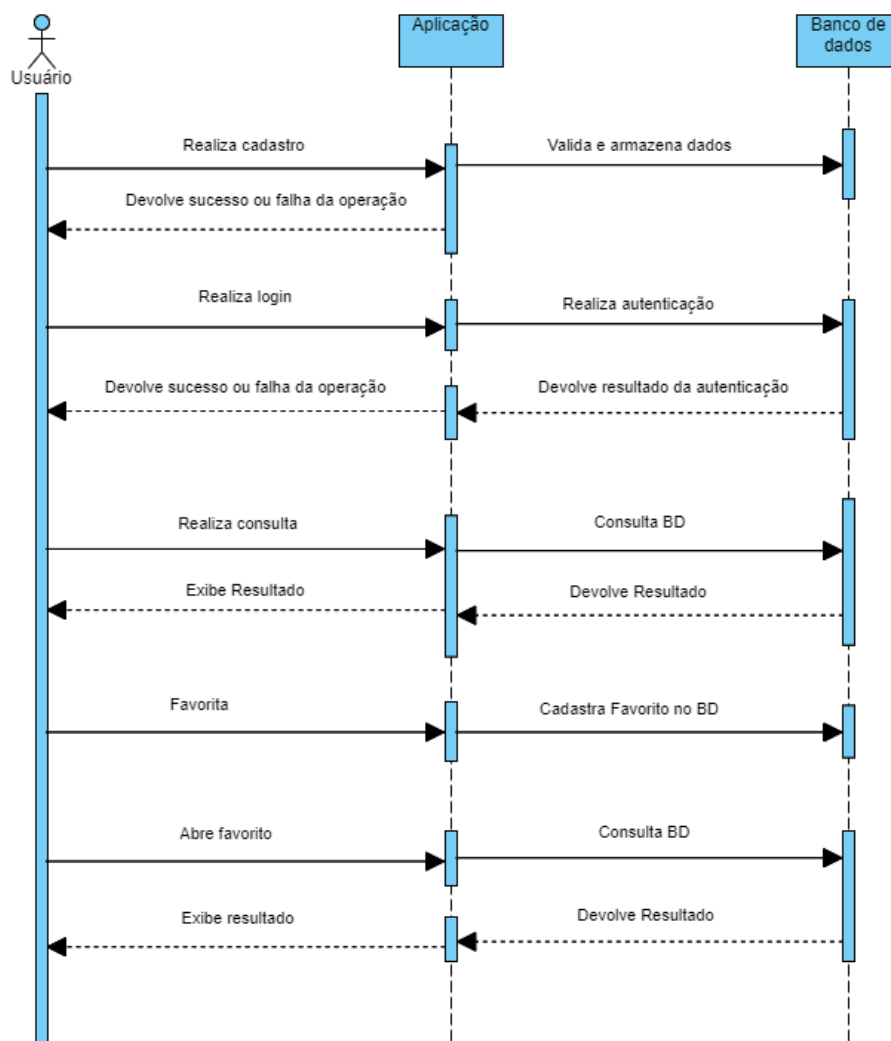
https://github.com/ArthurHenriqueSilva/Eng_SW_2.git

3. Diagramas UML

3.1 Diagrama de Sequência

Pensando em descrever a sequência de tarefas que precisam ser realizadas para o funcionamento da aplicação, foi desenvolvido um diagrama UML, que detalha a interação e ordem das tarefas executadas por cada ator e objeto.

Foram identificados como ator o usuário e como objetos a aplicação e o banco de dados. As tarefas que serão performadas pelo usuário foram identificadas como: realiza cadastro, realiza login, realiza consulta, favorita e abre favorito. Para cada uma dessas ações o usuário envia uma requisição para o sistema, que deve validar, armazenar, autenticar, cadastrar ou consultar o banco de dados. A aplicação por sua vez envia a requisição para o banco de dados que devolve o resultado da operação, como identificado no diagrama abaixo. Por fim, a aplicação devolve o sucesso ou falha das operações e exibe os resultados para o usuário.



3.2 Diagrama de Classes

Analisando o diagrama UML, é possível entender as diferentes relações entre as principais entidades dos sistemas.

As entidades são:

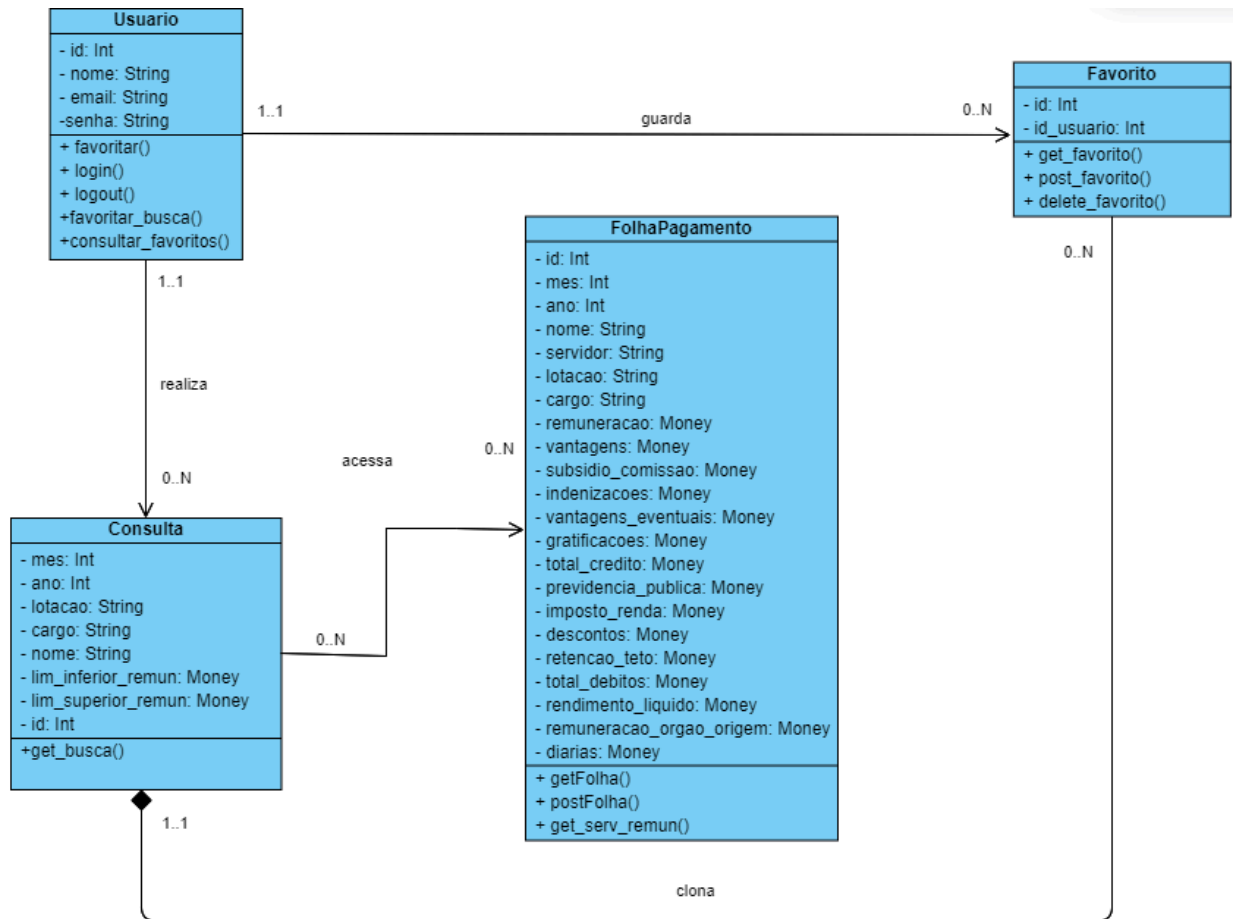
- Usuário: representa a unidade do público que utiliza o sistema. Possui os atributos id, nome e email.
- Favorito: permite que o usuário salve consultas frequentes para acesso rápido. Possui os atributos id e id_usuario.
- FolhaPagamento: possui todas as informações extraídas da base de dados original, além dos métodos essenciais.
- Consulta: representa uma consulta realizada pelo usuário ao sistema. Possui os atributos mês, ano, servidor e cargo, nome, limite inferior de remuneração e limite superior de remuneração.

As relações entre as entidades são:

- guarda: um usuário pode ter vários favoritos.
- realiza: um usuário pode realizar várias consultas.
- acessa: uma consulta acessa a folha de pagamento.
- clona: uma consulta pode ser clonada para gerar uma nova consulta com os mesmos parâmetros.

O sistema oferece as seguintes funcionalidades:

- Login: o usuário pode logar no sistema com seu email e senha.
- Favoritar: o usuário pode salvar uma consulta como favorita para acesso rápido.
- Consultar: o usuário pode consultar sua folha de pagamento informando o mês e ano desejados.
- Logout: o usuário pode sair do sistema.



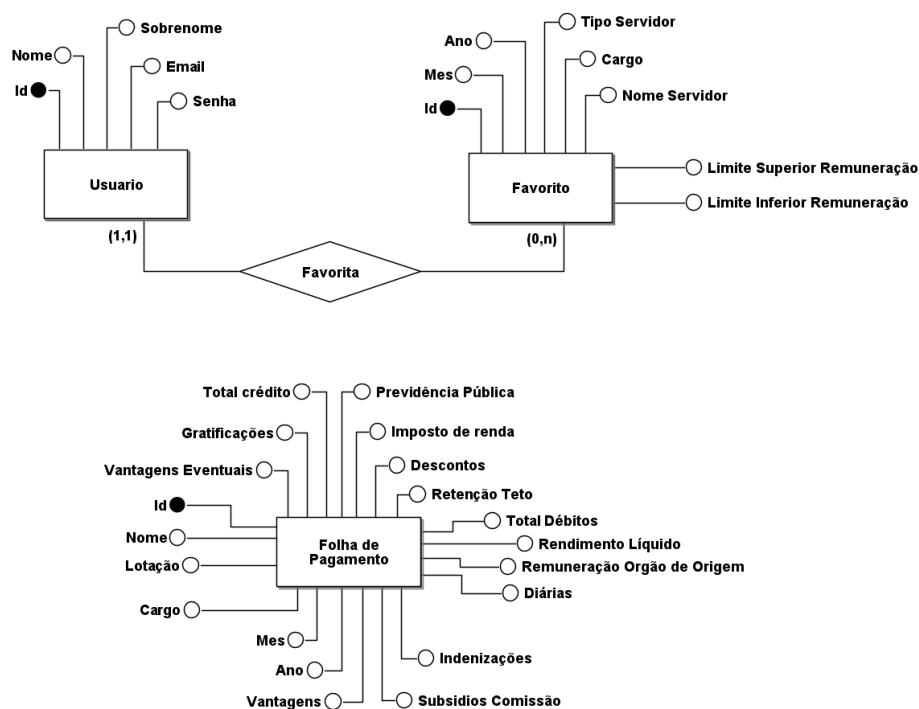
4. Modelo de dados

4.1 Modelo Conceitual

Partindo dos modelos de dados, começamos com o modelo conceitual. Esse modelo servirá como base para o modelo físico. As entidades são:

- **Usuário:** Representa os usuários em si e as informações atreladas a eles. Essas informações são: Id, Nome, Sobrenome, Email, Senha.
- **Favorito:** Representa as consultas favoritadas pelo usuário. Cada consulta favoritada possui essas informações: id, Mês, Ano, Tipo Servidor, Cargo, Nome Servidor, Limite Superior Remuneração, Limite Inferior Remuneração.
- **Folha de Pagamento:** Representa todos os dados originais adquiridos da base de dados referente a folha de pagamento do TJBA. Essas informações são: id, mês, ano, nome, servidor, lotação, cargo, remuneração, vantagens, subsídio comissão, indenizações, vantagens eventuais, gratificações, total crédito, previdência pública, imposto de renda, descontos, retenção teto, total débitos, rendimento líquido, remuneração órgão de origem, diárias.

Além disso, temos o relacionamento da entidade Usuário e a entidade Favorito. Esse relacionamento representa a ação de um usuário favoritar uma consulta. A cardinalidade dessa relação é (1,1) ----- (0,n), o que representa que um usuário poderá ter nenhuma ou várias consultas favoritadas.



4.2 Modelo Físico

O modelo físico explora os relacionamentos definidos no modelo conceitual voltando-se para um nível mais baixo na modelagem de dados. Foram definidas três tabelas pensando nas principais entidades definidas anteriormente:

- **usuario:** Seus atributos são: nome, sobrenome, email e senha, onde todos eles possuem dados do tipo varchar não nulo. Menção ao email que além disso possui restrição de unicidade associada. As restrições not null impostas são necessárias, pois todos atributos são cruciais para identificação da entidade e bom funcionamento do sistema.
- **favorito:** Seus atributos possuem dados do tipo integer, varchar e double, sem restrições associadas, dado que as opções de filtragem são opcionais, e podem se repetir para diferentes usuários. É responsabilidade da aplicação gerenciar quais campos serão preenchidos. Possui relacionamento de 1 para N com a tabela usuario; graças a isso, contém chave estrangeira de usuario com restrição not null, que garante que um favorito sempre estará associado a um e somente um usuário, e um usuário poderá se relacionar com 0 ou N favoritos.
- **pagamento:** Armazena todos os campos contidos no arquivo bruto original. Não possui relacionamento com outras classes, sendo voltada para consulta. Seus atributos são not null, visto que os campos das base importada nunca são vazios.

