

ANÁLISE E DESENVOLVIMENTO DE SOFTWARE

TÓPICOS INTEGRADORES II

Documentação de Projeto de Software

Versão 1.2

Autor(a): Ygor Abreu

GitHub: https://github.com/YgorAbreu

2020

Controle de Versão do Documento

|  |  |
| --- | --- |
| Versão | Descrição |
| 1.1 | Exemplo de descrição: Construção básica estrutura do documento do projeto e Modelo de Visão. |

Sumário

[1. Introdução ao Documento 4](#_Toc36326375)

[1.1. Área de negócio do Sistema 4](#_Toc36326376)

[1.2. Principais funcionalidades 4](#_Toc36326377)

[1.3. Método de trabalho 4](#_Toc36326378)

[2. Modelo de Dados 4](#_Toc36326379)

[2.1. Modelo de Visão 4](#_Toc36326380)

[2.2. Modelo Conceitual 4](#_Toc36326381)

[2.3. Modelo Lógico 4](#_Toc36326382)

[2.4. Dicionário de Dados 4](#_Toc36326383)

[2.5. Modelo Físico 5](#_Toc36326384)

[3. Análise e Design 5](#_Toc36326385)

[3.1. Diagrama de Classes 5](#_Toc36326386)

[4. Arquitetura do Software 5](#_Toc36326387)

[4.1. Padrão de projeto 5](#_Toc36326388)

[4.2. Protótipo 5](#_Toc36326389)

1. Introdução ao Documento

## Área de negócio do Sistema

O projeto busca uma iniciativa em uma academia para a prática de exercícios físicos e esportes em geral, apresentando um sistema onde facilite a administração do mesmo e busque o melhor funcionamento e de forma clara os exercícios passados para os clientes.

## Principais funcionalidades

O sistema funcionará para administrar o dia-a-dia do aluno e sua evolução, levará ao cliente a melhor sequencia de exercícios para o seu estado físico e de acordo com sua melhora física sua sequencia de atividades também mudará.

## Método de trabalho

O principal método de trabalho será a criação de um banco de dados, na linguagem de banco de dados SQL, onde será desenvolvido um sistema que garanta a funcionalidade da academia e a praticidade nas práticas físicas

1. Modelo de Dados

## Modelo de Visão

Este é um banco de dados de uma academia, nesta academia serão vendidos serviços para o aluno (cliente).

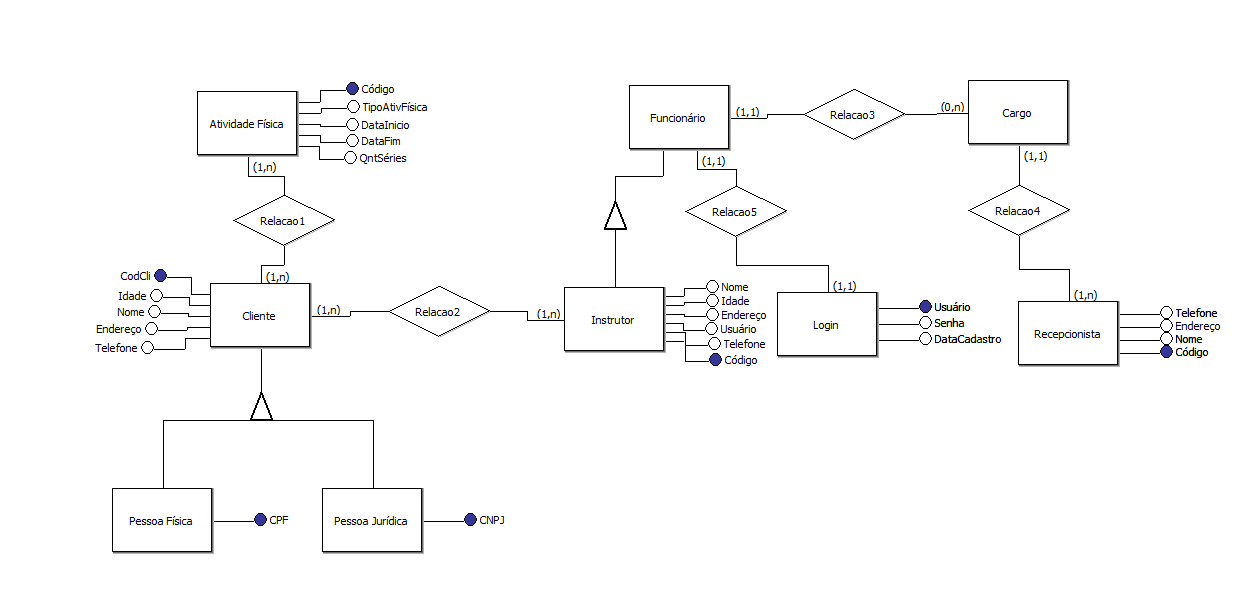
De cada aluno (cliente) é registrado suas informações pessoais como idade, peso, nome, endereço, telefone(um ou mais) cpf e rg. Cada aluno (cliente) pode ser pessoa física ou jurídica, e cada um tem um método de pagamento.

Cada atividade física é feita para um tipo de aluno (cliente), para atividade física armazenar: tipo de atividade física (treino funcional, corrida, spinning, musculação, danças, etc.), data início, data fim e quantidade de séries.

Existem vários tipos de funcionários na academia, para recepcionista armazenar código, nome, idade, telefone(um ou mais), para instrutores na academia, cada instrutor pode ter vários alunos (clientes), para o instrutor armazenar, código, nome, telefone (um ou mais), entre outros.

A academia terá um sistema para controle dos alunos (clientes) e professores, o sistema contará com um processo de login e deverá armazenar: Usuário, senha e data de cadastro.

## Modelo Conceitual



## 2.3.Modelo Lógico

Cliente (CódigoCli, CódigoAtivFísica, CódigoInst, CPF, Idade, Nome, Endereço, RG, DataNasc)

CódigoCli (Primary key)

Atividade Física (CódigoAtivFísica, CódigoCli, TipoAtivFísica, DataInício, DataFim, QntdSéries)

CódigoAtivFísica (Primary Key)

CódigoInst referencia instrutor

Instrutor (CódigoInst, CPF, Nome, Endereço, Telefone, RG, DataNasc)

CódigoInst (Primary Key)

CódigoCli referencia Cliente

Usuário referencia Login

Recepcionista (CódigoRecep, CPF, Nome, Endereço, Telefone, RG, DataNasc)

CódigoRecep (Primary Key)

Usuário referencia Login

Login (Usuário, Senha, DataCadastro)

Usuário (Primary Key)

Senha

## Dicionário de Dados

Tabela Cliente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Tipo (tamanho) | Obrigatório | Chave | Descrição |
| CódigoCli | integer | sim | PK | Chave primaria da tabela |
| CódigoInst | integer | sim | FK | Código do instrutor |
| CPF | varchar |  |  | CPF do cliente |
| Idade | integer |  |  | Idade do cliente |
| Nome | Varchar |  |  | Nome do cliente |
| Endereço | Varchar |  |  | Endereço do cliente |
| RG | Integer |  |  | RG do cliente |
| DataNasc | Date |  |  | Data de nascimento do cliente |

Atividade Física

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Tipo (tamanho) | Obrigatório | Chave | Descrição |
| CódigoAtivFísica | integer | sim | PK | Chave primaria da tabela |
| TipoAtivFísica | integer | sim | FK | Código da atividade física |
| CódigoCli | integer | sim | FK | Código do cliente |
| DataInício | Date |  |  | Data de início da atividade fisica |
| DataFim | Date |  |  | Data do fim da atividade fisica |
| QtdSéries | integer |  |  | Quantidade de séries que o aluno irá fazer |

Tabela Instrutor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Tipo (tamanho) | Obrigatório | Chave | Descrição |
| CódigoInst | integer | sim | PK | Chave primaria da tabela |
| CódigoCli | integer | sim | FK | Chave primaria da tabela |
| CPF | integer | sim | FK | CPF do Instrutor |
| Idade | integer |  |  | Idade do Instrutor |
| Nome | Varchar |  |  | Nome do Instrutor |
| Endereço | Varchar |  |  | Endereço do Instrutor |
| RG | Integer |  |  | RG do Instrutor |
| DataNasc | Date |  |  | Data de nascimento do Instrutor |

Tabela Recepcionista

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Tipo (tamanho) | Obrigatório | Chave | Descrição |
| CódigoRecep | integer | sim | PK | Chave primaria da tabela |
| CPF | integer | sim | FK | CPF do Recepcionista |
| Idade | integer |  |  | Idade do Recepcionista |
| Nome | Varchar |  |  | Nome do Recepcionista |
| Endereço | Varchar |  |  | Endereço do Recepcionista |
| RG | Integer |  |  | RG do Recepcionista |
| DataNasc | Date |  |  | Data de nascimento do Recepcionista |

Tabela Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificador | Tipo (tamanho) | Obrigatório | Chave | Descrição |
| Usuário | Varchar | sim | PK | Chave primaria da tabela |
| Senha | integer | sim |  | Senha do usuário |
| DataCadastro | Date |  |  | Data de login |

## Modelo Físico

**Create** **Table** Cliente(

codCli **integer** **not** **null** **primary** **key**,

nome **varchar**(40) **not** **null**,

cpf\_cnpj **char**(14) **not** **null**,

rg **integer** **not** **null**,

rua **varchar**(50),

bairro **varchar**(45),

cidade **varchar**(45),

uf **char**(2),

fone **varchar**(15) ,

dataNasc **DATE**);

**Create** **Table** Instrutor(

codInst **integer** **not** **null** **primary** **key**,

nome **varchar**(40) **not** **null**,

cpf\_cnpj **char**(14) **not** **null**,

rua **varchar**(50),

bairro **varchar**(45),

cidade **varchar**(45),

uf **char**(2),

fone **varchar**(15) ,

dataNasc **DATE,**

codCli **integer** **not** **NULL**,

**foreign** **key** (codCli) **references** Cliente (codCli));

**Create** **Table** Recepcionista(

codRecep **integer** **not** **null** **primary** **key**,

nome **varchar**(40) **not** **null**,

cpf\_cnpj **char**(14) **not** **null**,

rua **varchar**(50),

bairro **varchar**(45),

cidade **varchar**(45),

uf **char**(2),

fone **varchar**(15),

dataNasc **DATE,**

usuario **varchar(40**) not null);

**Create** **Table** Login(

usuario **varchar**(40) **not** **null** **primary** **key**,

senha **varchar**(40) **not** **null**,

dataCadastro **char**(14) **not** **NULL**);

**Create** **Table** Atividade(

codativfis **integer** **not** **null** **primary** **key**,

tipoativfis **varchar**(40) **not** **NULL**,

datainicio **date**,

datafim **date**,

qtdseries **INTEGER** **not** **NULL**,

codCli **integer** **not** **NULL**,

codInst **integer** **not** **NULL**,

**foreign** **KEY** (codCli) **REFERENCES** cliente(codCli),

**foreign** **key** (codInst) **REFERENCES** instrutor(codInst));

INSERTS

**Insert** **into** cliente **values** (1,'José Manoel','12345678904','1027152','Rua da Vitoria','Casa Amarela','Recife','PE','34441589','1965-08-28');

**Insert** **into** cliente **values** (2,'Maria Santos','78945612303','8737852','Rua da Mata','Torre','Recife','PE','34444256','1977-03-25');

**Insert** **into** cliente **values** (3,'Rafael Bezerra','45612378908','8769022','Alameda da Conquista','Derbi','Recife','PE','34439061','1990-10-05');

**Insert** **into** cliente **VALUES** (4,'Júlio Santos','45620378908','8027152','Rua da moeda','Derbi','Recife','PE','34440919','1992-04-23');

**Insert** **into** cliente **VALUES** (5,'Felipe Gabriel','45612848908','8767152','Rua da Conquista','Agua Fria','Recife','PE','34410967','1995-06-24');

**commit**;

**Insert** **into** instrutor **values** (1,'Felipe Barros','12336278904','Rua do Arruda','Casa Amarela','Recife','PE','34127589','1965-04-27',2);

**Insert** **into** instrutor **values** (6,'Felipe Barros','12336278904','Rua do Arruda','Casa Amarela','Recife','PE','34127589','1965-04-27',3);

**Insert** **into** instrutor **values** (2,'Ronaldo Lima','01245612303','Rua da misericordia','Torre','Recife','PE','34983256','1977-02-20',1);

**Insert** **into** instrutor **values** (3,'Renan Oliveira','45612378294','Rua da Conquista','Derbi','Recife','PE','34410661','1990-11-09',1);

**Insert** **into** instrutor **VALUES** (4,'Fábio Manoel','45620378029','Rua da Historia','Derbi','Recife','PE','34440293','1992-12-30',3);

**Insert** **into** instrutor **VALUES** (5,'José Felipe','45283848908','Rua do mundo','Agua Fria','Recife','PE','34410967','1995-06-07',4);

**commit**;

**Insert** **into** recepcionista **values** (1,'Ana Vitória','12294278904','Rua do Arruda','Arruda','Recife','PE','34271589','1992-04-27','Ana@academy.com');

**Insert** **into** recepcionista **values** (2,'Fernanda Lima','01245392303','Rua da misericordia','Jordão','Recife','PE','34183256','1994-02-20','Fernanda@academy.com');

**Insert** **into** recepcionista **values** (3,'Raquel Oliveira','45612103294','Rua da Conquista','Derby','Recife','PE','34982661','1989-11-09','Raqueal@academy.com');

**Insert** **into** recepcionista **VALUES** (4,'Fabricia Manoela','10320378029','Rua da Historia','Fundão','Recife','PE','31920293','1990-12-30','Farbicia@academy.com');

**Insert** **into** recepcionista **VALUES** (5,'Amanda Souza','90283848908','Rua do mundo','Agua Fria','Recife','PE','34412017','1997-06-07','Amanda@academy.com');

**commit**;

**INSERT** **INTO** login **VALUES** ('Ana@academy.com','\*\*\*\*\*\*\*\*\*\*','2019-08-14');

**INSERT** **INTO** login **VALUES** ('Fernanda@academy.com','\*\*\*\*\*\*\*\*\*\*','2018-05-15');

**INSERT** **INTO** login **VALUES** ('Raqueal@academy.com','\*\*\*\*\*\*\*\*\*\*','2020-01-13');

**INSERT** **INTO** login **VALUES** ('Farbicia@academy.com','\*\*\*\*\*\*\*\*\*\*','2017-12-25');

**INSERT** **INTO** login **VALUES** ('Amanda@academy.com','\*\*\*\*\*\*\*\*\*\*','2018-08-30');

**commit**;

**INSERT** **INTO** atividade **VALUES**(1,'Esteira','2020-01-16','2020-05-30',3,1,2);

**INSERT** **INTO** atividade **VALUES**(2,'Musculação','2020-01-01','2020-03-30',2,1,1);

**INSERT** **INTO** atividade **VALUES**(3,'Dança','2019-03-20','2020-04-31',4,3,3);

**INSERT** **INTO** atividade **VALUES**(4,'Natação','2019-01-16','2020-03-25',5,4,5);

**INSERT** **INTO** atividade **VALUES**(5,'Artes Marciais','2019-05-16','2020-05-16',1,5,4);

**COMMIT**;

SELECTS

1- **SELECT** cliente.nome, atividade.tipoativfis

**FROM** atividade **INNER** **JOIN** cliente **ON** (atividade.codcli = cliente.codcli)

**WHERE** cliente.nome = 'José Manoel'

**AND** **MONTH**(datainicio) = '1'

**OR** **MONTH**(datafim) = '1'

**AND** **YEAR**(datainicio) = '2020'

**ORDER** **BY** 1

2- **SELECT** instrutor.nome, **COUNT**(cliente.codcli)

**FROM** instrutor **inner** **JOIN** cliente **ON** (instrutor.codcli = cliente.codcli)

**group** **BY** instrutor.nome

**ORDER** **BY** 1

3- **SELECT** \* **FROM** instrutor

**ORDER** **BY** instrutor.nome

4- **SELECT** nome, cpf\_cnpj, rg, **FLOOR**(**DATEDIFF**(**NOW**(), datanasc) / 365)

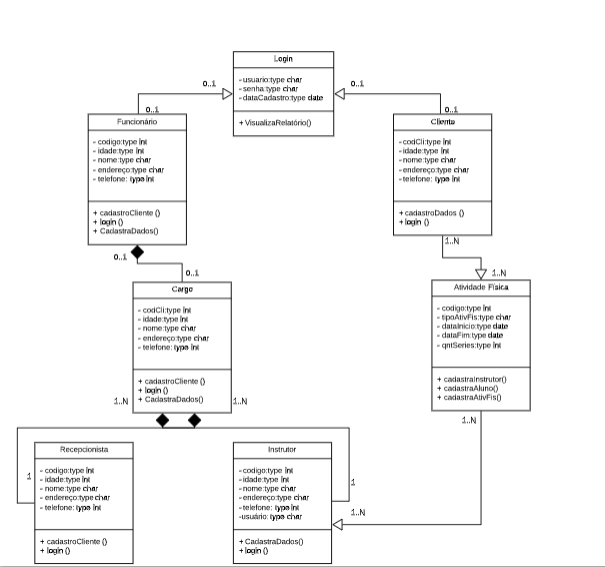
**FROM** cliente

5- **SELECT** \* **FROM** recepcionista

1. Análise e Design

## Diagrama de Classes

Será apresentado o diagrama de classes básicas e deverá ser informada qual ferramenta foi utilizada e apresentada a imagem final do diagrama.



1. Arquitetura do Software

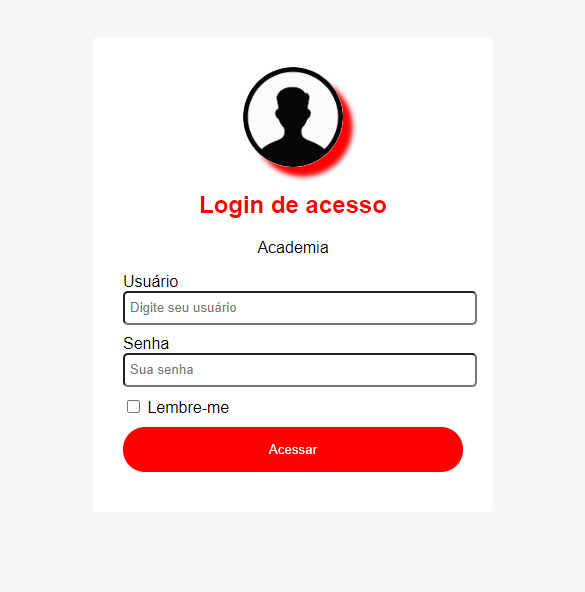
## Padrão de projeto

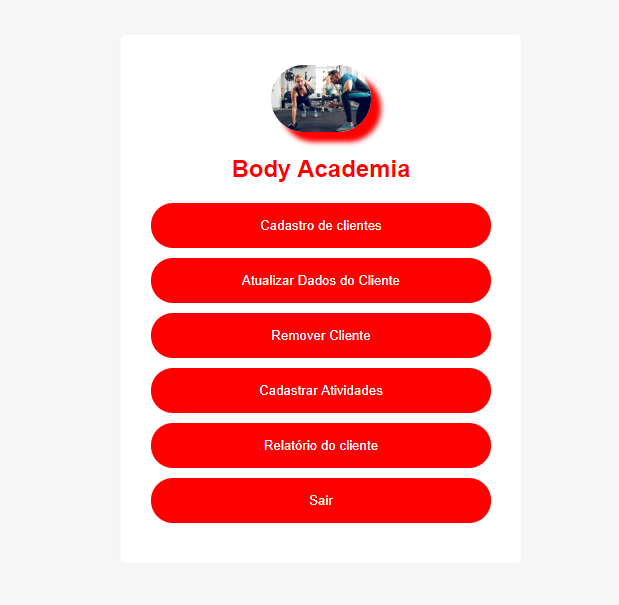
O projeto será apresentado em três camadas chamadas de MVC que são, view, model e control. A view será responsável por trazer as telas do projeto, a model por trazer as classes do projeto e por último a controller que será o meio entre as duas, que buscará as informações e dará ao que o usuário determinar.

## Protótipo

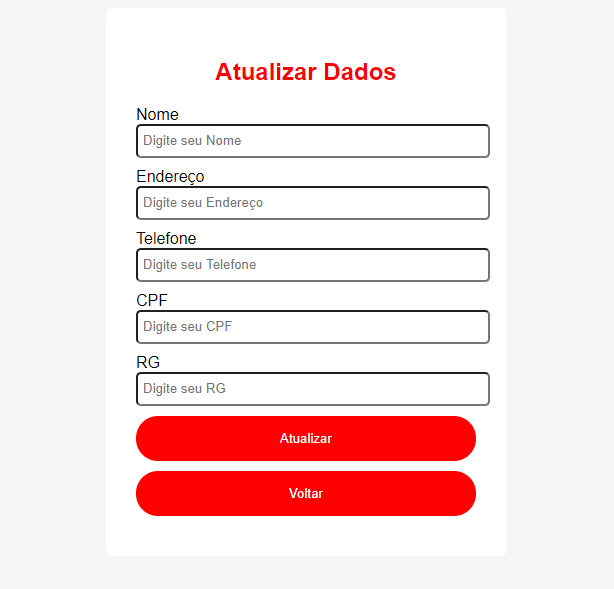
Serão apresentadas cada uma das telas do sistema para cada uma das camadas.

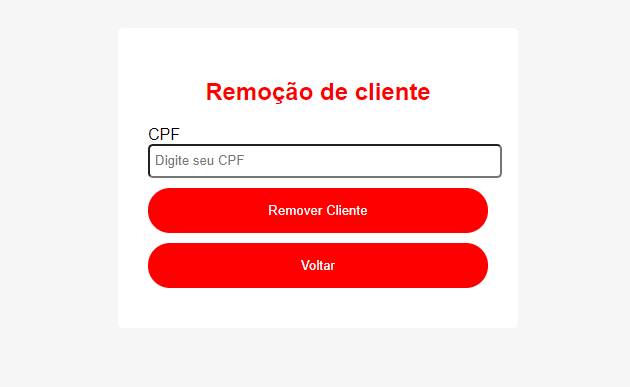
4.2.1.Telas e classes da camada View









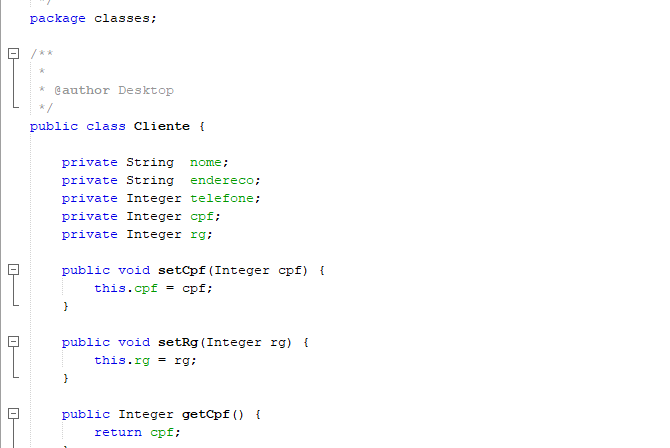




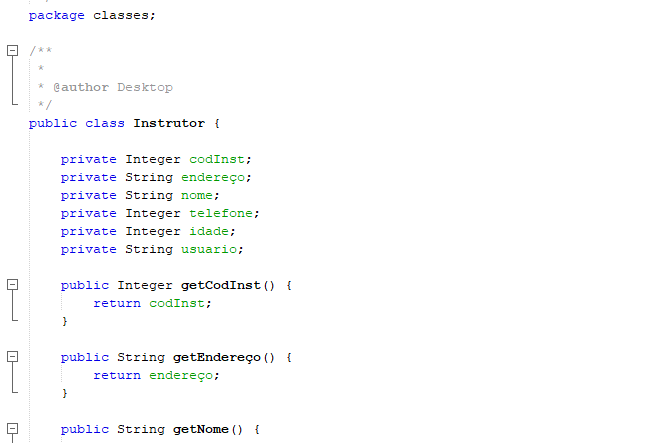


4.2.2.Classes da camada Model













4.2.3.Classes da camada Controller

