MOVIE SHARK

1- RF - Requisitos Funcionais

- RFx1 O sistema deve guardar o cadastro de filmes, designando um código único para cada um, armazenar status, título, diretor (ES), produtor (ES), data de lançamento, local e duração.
- RFx2 O sistema deverá permitir a catalogação e alteração do cadastro de filmes pelos usuários com permissão de "Gestão de Estoque" ou "Administrador".
- RFx3 O sistema deve cadastrar exemplares de filmes, designando um código único para cada um, armazenar status e local de armazenamento físico na locadora.
- RFx4 O sistema deverá permitir o cadastro e alteração do cadastro de exemplares pelos usuários com permissão de "Gestão de Estoque" ou "Administrador".
- RFx5 O sistema deverá cadastrar os locais da locadora onde os filmes são armazenados, atribuindo código único para cada um, armazenar status corredor, lado do corredor, coluna e prateleira.
- RFx6 O sistema deverá permitir o cadastro e alteração do cadastro de locais da locadora pelos usuários com permissão de "Gestão de Estoque" ou "Administrador".

2- Requisitos Não-Funcionais

- RNFx1 O sistema deve ser estável não possuindo bugs que impossibilitem o uso do software e se manter operante durante toda utilização pelo usuário.
- RNFx2 O sistema deverá se comunicar com o banco de dados.
- RNFx3 O sistema deverá ser de fácil aprendizado e responsivo de acordo com as ações do usuário
- RNFx4 Cada usurário será cadastrado e fornecido um login e senha por segurança.
- RNFx5 O sistema deverá se comunicar com o banco de dados
- RNFx6 O banco de dados será produzido em MySQL.

3- Esquema de Requisitos

- 3.1 Caso de Uso
- 3.2. Descrição dos Casos de Uso

4- Incrementos dos Analistas

5- Modelo ER

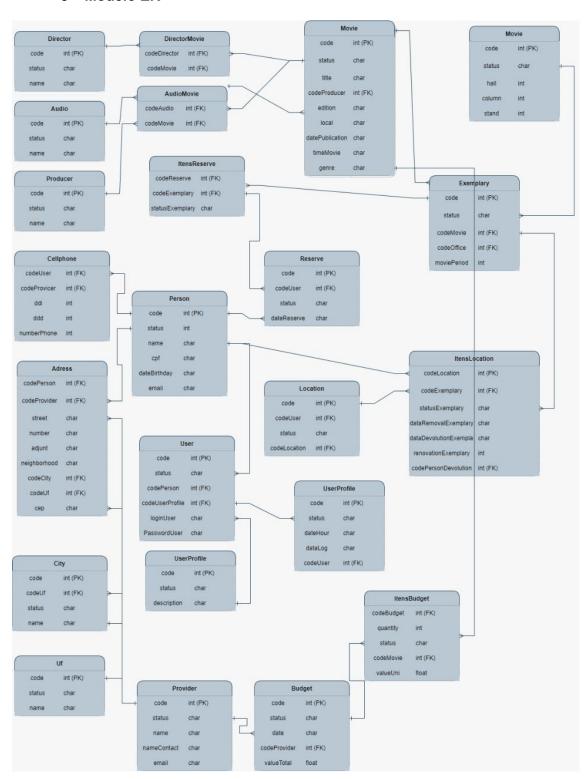


DIAGRAMA DE CLASSES

AUDIO

EXEMPLARY

code : int

status : char

codeMovie : int

codeOffice : int

+ create() : boolean + recordBD() : boolean

+ readBD(): boolean

ADRESS

codePerson : int

street: char

number : char
 adjunt : char

codeCity : int

cep : char

codeProvider : int

neighborhood : char

+ create() : boolean + recordBD() : boolean + readBD() : boolean

DIRECTOR

- code : int
- status : char name : char
- + create(): boolean
- + recordBD() : boolean + readBD() : boolean

MOVIE

- code : int
- status : char
- title : char
- codeDirector : char codeAudio: char
- codeProducer : char
- edition : char
- local : char
- datePublication : char
- timeMovie : char
- genre : char
- + create() : boolean + recordBD() : boolean + readBD(): boolean

PERSON

- code : int
- status : charname : char
- cpf : int dateBirthday : char
- email : char numberPhone : int
- + create(): boolean

- + recordBD() : boolean + readBD() : boolean

USERPROFILE

- code : int
- status : char description : char

- + create() : boolean + recordBD() : boolean + readBD() : boolean

USER

- code : int
- codeUserProfile : int
- status : char codePerson : int
- loginUser : char
 passwordUser : char
- + create() : boolean + recordBD() : boolean + readBD() : boolean

- + createHash(): char

PRODUCER

- code : int status : char code : int status : char name : char name : char
- + create(): boolean + create(): boolean + recordBD() : boolean + readBD() : boolean
 - + recordBD() : boolean + readBD() : boolean

OFFICE

- code : int
- status : char
- hall: int column : int
- + create() : boolean + recordBD() : boolean
- + readBD(): boolean

CELLPHONE

codePerson : int

- codeProvider : int

numberPhone : int

+ create(): boolean

+ recordBD(): boolean + readBD(): boolean

ddd: int

RESERVE

- code : int
- status : char name : char
- dateReserve + create() : boolean
- + recordBD() : boolean + readBD() : boolean

ITENSRESERVE

- codeReserve : int
- codeExemplary : int statusExemplary: char
- + create() : boolean
- + recordBD() : boolean

CITY

code : int

status : char

codeUser: int

name : char

codeStates: int

+ create(): boolean

+ recordBD() : boolean + readBD() : boolean

+ readBD() : boolean

LOCATION

- code : int status : char
- name : char
- + create(): boolean
- + recordBD() : boolean + readBD() : boolean

ITENSI OCATION

- codeLocation : int
- codeExemplary : int
- statusExemplary: char
- dateRemovalExemplary : char
- dateDevolutionExemplary : char
- renovationExemplary : char
- codePersonDevolution : int
- + create() : boolean + recordBD() : boolean

+ readBD() : boolean

STATES

- code : int
- status : char name : char
- + create(): boolean
- + recordBD() : boolean + readBD() : boolean

NOTIFICATIONS

- code : int
- codeUserProfile : int
- status : char
- codeUser : int textNotifications : char

+ create() : boolean + recordBD() : boolean

+ readBD(): boolean

LOGIN

- code : int
- codeUser : int
- status : char
- dateHour : char dataLog : char
- + create(): boolean
- recordBD() : boolean + readBD() : boolean

PROVIDER

- code : int
- status : char
- name : char - nameContact : char
- email : charnumberPhone : int
- + create(): boolean + recordBD() : boolean + readBD() : boolean

BUDGET

- code : int
- status : char date : char
- codeProvider : int valueTotal : float
- + create(): boolean
- + recordBD() : boolean + readBD() : boolean

ITENSBUDGET

- codeBudget : int
- status : char

- + create(): boolean + recordBD(): boolean + readBD(): boolean

- codeMovie: int
- quantity: intvalueUni : float