

Ecommerce de artes

Aluno: Leonardo Lage

Instituto Infnet

Projeto de Bloco: Desenvolvimento Java

Apresentação Pessoal

Estou no primeiro período no INFNET

- Busco minha primeira oportunidade na área de Desenvolvimento.
- Meus conhecimentos em desenvolvimento de sistemas são por enquanto:
 - Desenvolvimento de sistemas RESTful em Java.
 - Integração destes sistemas com a parte Web.
- Minha expectativa após a conclusão do curso de Java é de poder fazer deploy deste sistema daqui a um tempo. Depois do término da faculdade desejo já estar encaminhado no mercado como Desenvolvedor Web.



Ecommerce de artes

Sistema web pensado para
a divulgação e venda de produtos artísticos.

Ecommerce de artes

- No sistema proposto as principais funcionalidades são:
- 1. O serviço de usuários, que posta o usuário com nome, email, cep, foto e endereço.
- 1.1 (O endereço é buscado por uma API e é introduzido automaticamente a partir do cep.)
- 2. O serviço de obras de arte, que posta obras de arte com nome da obra, preço, foto e nome do artista.
- 2.1 O nome do artista se conecta com o serviço de usuários a partir do email.
- 3. O sistema Web, que conecta com os dois serviços a partir de um FeignClient e apresenta a view com Jsp.



localhost:8083

Users Table

[Click here to see all registered users!](#)

[Click here to see all registered artworks!](#)

Jogando esta uri no browser podemos ter acesso à aplicação!

>

Nome	Email	Telefone	CEP	Foto	Endereço	Update	Delete
Baudi Ferra	baudiferra@gmail.com	21887338	20270133	uploads/profile2122.jpg	cep=20270-133, logradouro=Rua do Matoso, complemento=até 128 - lado par, bairro=Praça da Bandeira, localidade=Rio de Janeiro, uf=RJ}	Update	Delete
Jorge aragao	jorig@hotmail.com	2192821920	20261242	uploads/mark.jpg	cep=20261-242, logradouro=Avenida Paulo de Frontin, complemento=de 238 a 516 - lado par, bairro=Praça da Bandeira, localidade=Rio de Janeiro, uf=RJ}	Update	Delete

Ao clicar em ver os usuários registrados, se deparará com a tabela de usuários!

Marcia Lage	marcialagecerqueira@gmail.com	21994331773	20511000	uploads/obras-arte-pixel-arte_5 (1).jpg	cep=20511-000, logradouro=Avenida Maracanã, complemento=de 983 a 1529 - lado ímpar, bairro=Tijuca, localidade=Rio de Janeiro, uf=RJ}	Update	Delete
-------------	-------------------------------	-------------	----------	---	--	------------------------	------------------------

Create User

Para cadastrar um novo usuário basta clicar no hyperlink "Create User".

O mesmo se aplica a atualização e excluir. Para atualizar dados do usuário ou deletar este usuário é preciso clicar em "Update" ou "Delete".

Artworks

>

Id	Nome	Preço	Foto	Artista	Update	Delete
	Vida	1000000.0	uploads/mark.jpg	User{nome=Leonardo Lage , email=leolagecerqueira@gmail.com, telefone=21998909363}	Update	Delete

YesYess

125000.0

uploads/pixelart-480x480.jpg

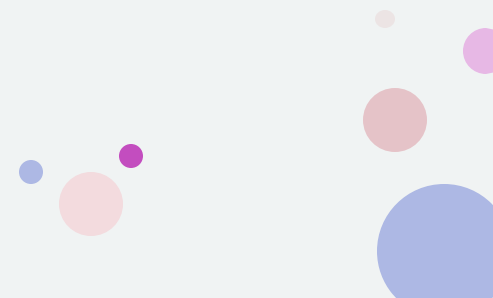
User{nome=Leonardo Lage ,
email=leolagecerqueira@gmail.com, telefone=21998909363}

[Update](#)

[Delete](#)

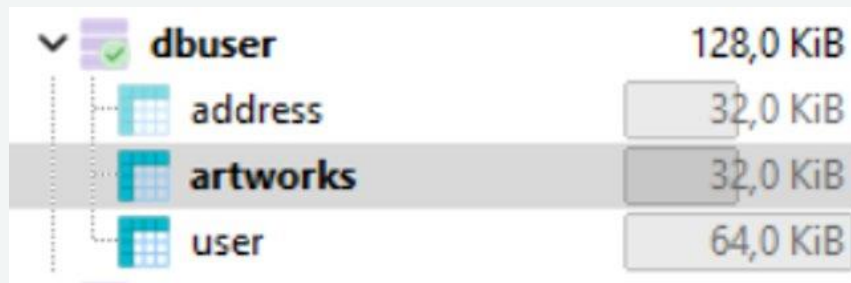
[Create Artworks](#)

Ocorre o mesmo princípio para as artworks



Ecommerce de artes – Banco de Dados

- No sistema desenvolvido se utilizou o banco de dados MariaDB.
- No banco temos as tabelas User, Address e Artworks.
- Esta mostrada a direita é o mapeamento da tabela User.
- Utilizou-se as anotações JPA para fazer o mapeamento.

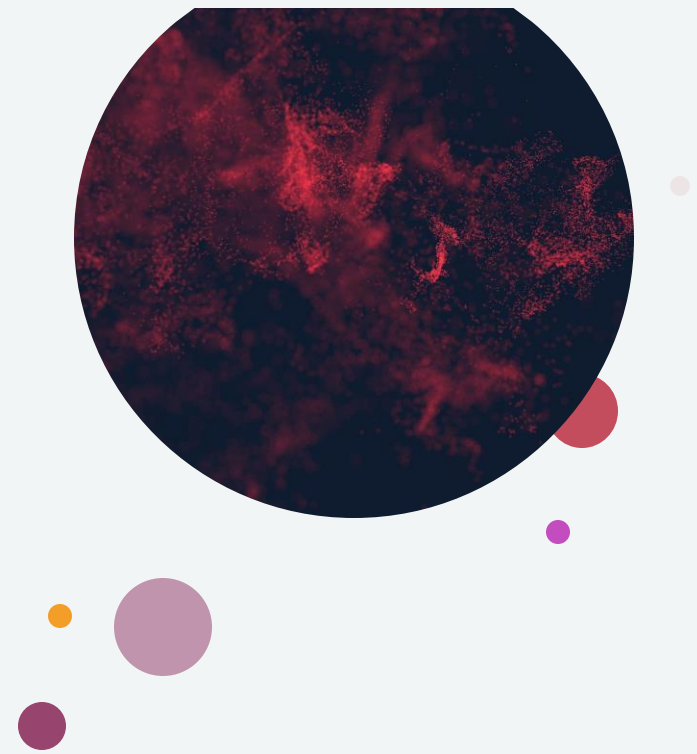


dbuser	128,0 KiB
address	32,0 KiB
artworks	32,0 KiB
user	64,0 KiB

```
public class User implements Serializable {  
  
    private static final long serialVersionUID = 1L;  
    @Basic(optional = false)  
    @NotNull  
    @Size(min = 1, max = 50)  
    @Column(name = "nome")  
    private String nome;  
    @Id  
    @Basic(optional = false)  
    @NotNull  
    @Size(min = 1, max = 50)  
    @Column(name = "email")  
    private String email;  
    @Basic(optional = false)  
    @NotNull  
    @Size(min = 1, max = 50)  
    @Column(name = "telefone")  
    private String telefone;  
    @Basic(optional = false)  
    @NotNull  
    @Size(min = 1, max = 100)  
    @Column(name = "cep")  
    private String cep;  
    @Basic(optional = false)  
    @NotNull  
    @Size(min = 1, max = 150)  
    @Column(name = "file_url")  
    private String fileUrl;  
    @JoinColumn(name = "cep_user", referencedColumnName = "cep")  
    @ManyToOne(optional = false)  
    private Address cepUser;  
}
```

Ecommerce de artes – Banco de Dados

- É preciso lembrar que o pacote Model (mapeamento) é o mesmo nos dois serviços e na aplicação Web. O único caso que não precisaria seria o User com Artworks, pois é o Artworks que chama o User.



Ecommerce de artes – Banco de Dados

nome 🔑	email 🔑	telefone	cep	file_url	cep_user 🔑
Baudi Ferra	baudiferra@gmail.com	21887338	20270133	uploads/profile2122.jpg	20270-133
Jorge aragao	jorig@hotmail.com	2192821920	20261242	uploads/mark.jpg	20261-242
Leonardo Lage	leolagecerqueira@gmail.com	21998909363	22050011	uploads/profile2122.jpg	22050-011
Marcia Lage	marcialagecerqueira@gmail.com	21994331773	20511000	uploads/obras-arte-pixel-arte_5 (1).jpg	20511-000

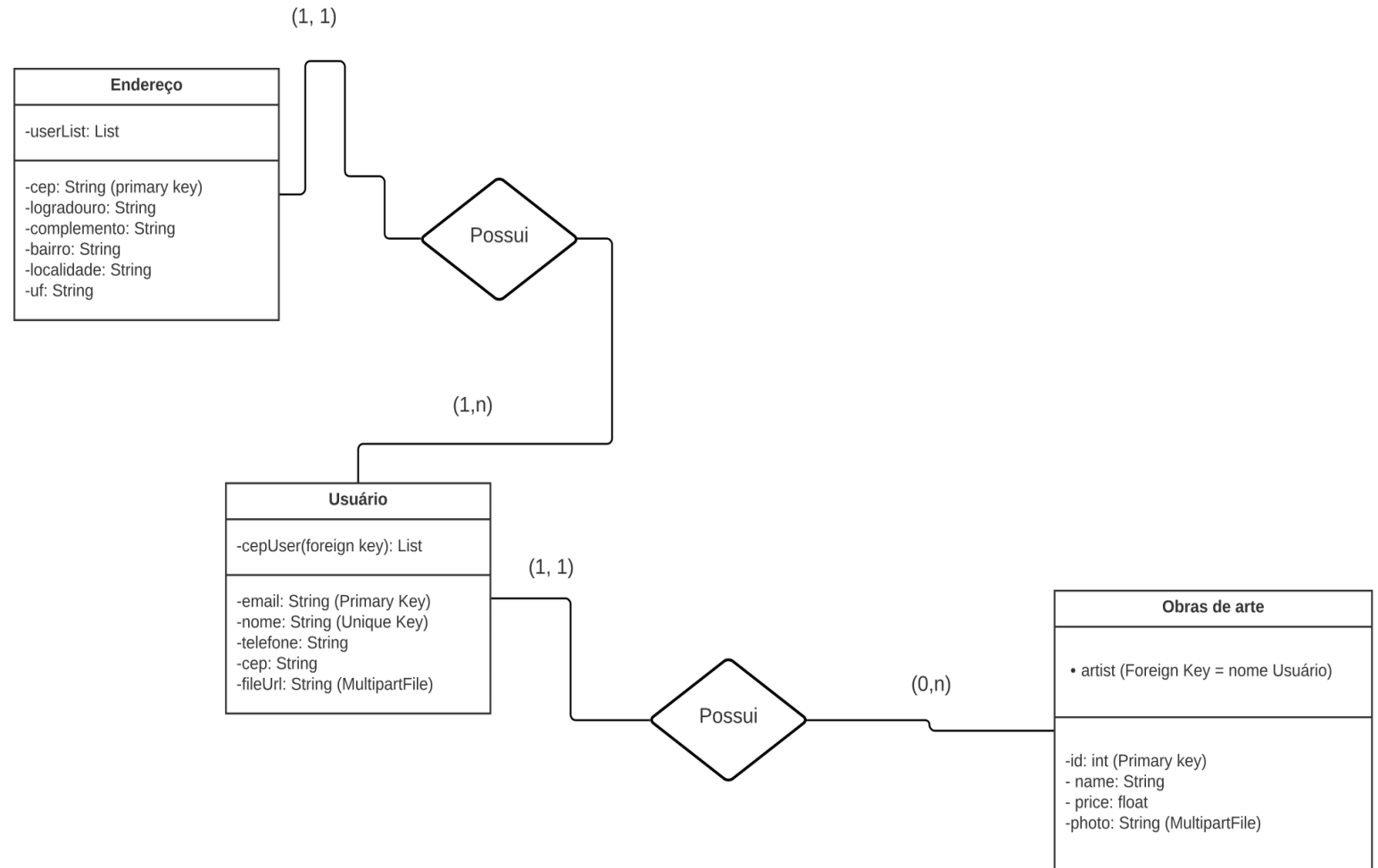
- Email é uma primary key e nome é uma unique key.
- Na hora de postar a arte, é preciso colocar o email do usuário vinculado a arte. Colocando a unique key, no banco de dados o parâmetro artist fica como o nome do usuário ao invés do email.

id 🔑	name	price	photo	artist 🔑
5	Vida	1.000.000	uploads/mark.jpg	Leonardo Lage
6	SextaFeira	10.000.000	uploads/pixelart-480x480.jpg	Leonardo Lage

Ecommerce de artes – Banco de Dados

No caso do endereço, o cep é puxado por uma API e retorna informações importantes como logradouro, complemento e uf.

Essas informações se conectam através do CepUser, usando chave estrangeira para se conectar o cep do usuário com cep do endereço para assim retornar o Json completo.



Ecommerce de artes - Pacotes

Client: Há um cliente artworks e client User para conectar aos 2 serviços Rest.

Controller:
Responsabilizam-se pela comunicação entre o cliente e os Jsp.

Service:
O único Service da web é feito exclusivamente para guardar arquivos(photo no caso de artworks e fileUrl no caso de User).

```

  client
    ArtworksClient.java
    UserClient.java
  controller
    ArtworksController.java
    UserController.java
  model
    Address.java
    Artworks.java
    User.java
  StorageService
    StorageService.java
    WebApplication.java
  > resources
  webapp \ WEB-INF \jsp
    createartwork.jsp
    createuser.jsp
    readartworks.jsp
    readuser.jsp
    updateartworks.jsp
    updateuser.jsp
```


Ecommerce de artes

Agradeço a atenção!

Aluno: Leonardo Lage

Instituto Infnet

Projeto de Bloco: Desenvolvimento Java

