* :: Documentação de software :: Manual do Utilizador::CedricArt



**Grupo:** 7

**Autores:** Maria Spínola & Ricardo Reis

**Curso:** Programação em Java

Índice

[Introdução 3](#_Toc22879)

[O programa "CedricArt" foi desenvolvido para: 3](#_Toc28168)

[Este programa foi concebido para ser utilizado por: 3](#_Toc1530)

[Este manual destina-se a: 3](#_Toc10381)

[Requisitos do Sistema 5](#_Toc32013)

[Sistema Operacional (SO): 5](#_Toc6293)

[Hardware: 5](#_Toc20946)

[Conexão com a Internet: 5](#_Toc15820)

[Instalação e Configuração 6](#_Toc16701)

[Uso do Programa 15](#_Toc30082)

[FAQs 21](#_Toc7899)

[1. Qual é o propósito do programa "CedricArt"? 21](#_Toc2692)

[2. Como faço para acessar informações sobre uma exposição específica? 21](#_Toc8667)

[3. Posso obter detalhes sobre artistas individuais usando o programa? 21](#_Toc22680)

[4. Como voltar para o menu principal? 22](#_Toc23304)

[5. Existe suporte técnico disponível para questões relacionadas ao programa? 22](#_Toc18671)

[Documente o Código Fonte (se aplicável): 23](#_Toc9223)

[Arquitetura do Programa 23](#_Toc22281)

[A arquitetura é composta por três partes principais: 23](#_Toc6422)

[Bibliotecas Usadas 23](#_Toc2529)

[O programa utiliza as seguintes bibliotecas e frameworks: 23](#_Toc31256)

[Documentação de Classes e Métodos 24](#_Toc27486)

[1. ExhibitionInfo Class: 24](#_Toc29879)

[2. Exhibitions Class: 24](#_Toc6233)

[3. Galeries Class: 24](#_Toc10465)

[4. Main Class: 25](#_Toc5554)

[Modelo de manual técnico - Sistema CedricArt 26](#_Toc496)

[Introdução 26](#_Toc15189)

[Diagrama de Classes 26](#_Toc4660)

[Descrição do Diagrama de Classes: 26](#_Toc19486)

[Diagrama de Pacotes 27](#_Toc19392)

[Descrição do Diagrama de Pacotes: 27](#_Toc18923)

[Conclusão 28](#_Toc2213)

[Modelo de relatório Técnico para o projeto 28](#_Toc20340)

[Enunciado do Problema: 28](#_Toc9730)

[Sites de Referência: 28](#_Toc17814)

[Análise Substantivo/Verbo: 29](#_Toc20624)

[Desenho da Solução: 29](#_Toc14384)

[Descrição Estratégia Tecnológica: 29](#_Toc22164)

[Conclusões: 30](#_Toc28366)

[Arquitetura: 30](#_Toc12894)

[Diagrama de Classes: 30](#_Toc20395)

[Persistência dos Dados: 30](#_Toc22969)

[Resultado Alcançado: 30](#_Toc2011)

[Requisitos Não Implementados: 30](#_Toc515)

# Introdução

O manual do utilizador destina-se a fornecer informações abrangentes sobre a utilização do programa "CedricArt", um sistema relacionado à gestão de exposições artísticas, artistas, obras de arte e galerias. Este programa visa simplificar a administração e visualização de exposições de arte, oferecendo uma plataforma intuitiva para utilizadores visitantes das Galerias.

## O programa "CedricArt" foi desenvolvido para:

* Informar os visitantes das galerias, que interagem com a aplicação CedricArt o acesso a informações detalhadas sobre artistas, exposições, obras de arte e galerias.
* Oferecer uma interface amigável para os utilizadores navegarem pelas exposições e obras de arte associadas.

.

### Este programa foi concebido para ser utilizado por:

* Gestores de Exposições: Responsáveis pela administração e organização de exposições artísticas.
* Artistas: Criadores de obras de arte cujas peças são exibidas em exposições.
* Utilizadores Finais: Entusiastas de arte que desejam explorar informações sobre exposições e obras de arte.

### Este manual destina-se a:

* Utilizadores Finais: Pessoas interessadas em explorar exposições e obras de arte sem conhecimento técnico avançado.
* Gestores de Exposições e Parceiros: Utilizadores envolvidos na administração e organização de exposições artísticas.
* Técnicos: Programadores que precisam entender o código fonte e a estrutura do programa para possíveis modificações ou extensões.

O manual foi estruturado para atender às necessidades de cada grupo, fornecendo informações relevantes de acordo com o nível de conhecimento técnico esperado. Utilizadores finais encontrarão orientações simples para explorar o programa, enquanto gestores de exposições e programadores terão detalhes técnicos e informações sobre a estrutura do código.

# Requisitos do Sistema

Para garantir a execução adequada do programa "CedricArt", é necessário atender aos seguintes requisitos de sistema:

## Sistema Operacional (SO):

* + Recomendado: Windows 10, macOS Catalina (ou superior), ou alguma distribuição Linux moderna.

## Hardware:

* + Processador: Dual-core 2 GHz ou superior.
  + Memória RAM: 4 GB ou mais.
  + Espaço em Disco: Mínimo de 500 MB de espaço livre.

## Conexão com a Internet:

* Uma conexão ativa com a internet é necessária para a funcionalidade de importação de dados da REST API Artsy, assim como da visualização das imagens das obras de arte, etc.

Certifique-se de que o sistema atenda a esses requisitos para garantir uma experiência de utilização eficiente do programa "CedricArt". Em caso de dúvidas ou dificuldades na configuração do ambiente, consulte a documentação específica para cada requisito ou entre em contacto com o suporte técnico.

# Instalação e Configuração

Nesta secção encontra as instruções detalhadas sobre como instalar e configurar o programa.

1. Configurar os seguintes parâmetros, caso seja necessário, no ficheiro “CedricArt.config” que encontra na directoria “resources\config” (ver tabela seguinte):

|  |
| --- |
| #  # Database *config*  DB\_SERVER=*localhost*  DB\_PORT=*3306*  DB\_NAME=*artsydb*  DB\_USER=*root*  DB\_PWD=*123*  #  # IS\_FAST\_ARTSY\_LOAD *from* Artsy  IS\_FAST\_ARTSY\_LOAD=*false*  #  # Artsy Api *config*  CLIENT\_ID=*9f14b9f06093d764ddf7*  CLIENT\_SECRET=*a8f0f836b1832d14bdc0e3e4c75559c4*  XAPP\_TOKEN=*https*://*api*.*artsy*.*net*/*api*/*tokens*/*xapp\_token*  MAX\_ATTEMPTS\_API=*5*  #  # CedricArt API *config*  CEDRIC\_ART\_API\_HOST=*http*://*localhost*:*4567* |

Sendo que estes são os parâmetros para a base de dados

# Database *config*

DB\_SERVER=*localhost*

DB\_PORT=*3306*

DB\_NAME=*artsydb*

DB\_USER=*root*

DB\_PWD=*123*

Caso seja necessário importar “todos” os dados da API Artsy (ver <https://artsy.github.io/blog/2014/09/12/designing-the-public-artsy-api/>) então colocar IS\_FAST\_ARSTY\_LOAD = false

Caso seja necessário importar dados “suficientes”, então colocar IS\_FAST\_ARSTY\_LOAD = true

NOTA: Ao colocar IS\_FAST\_ARSTY\_LOAD = false a importação dos dados da API Artsy é “profunda”, e logo consome muito mais tempo

# IS\_FAST\_ARTSY\_LOAD *from* Artsy

IS\_FAST\_ARTSY\_LOAD=*false*

Configurar os dados da “aplicação” necessária para usar a REST API da Artsy

Nota: O parâmetro MAX\_ATTEMPTS\_API é para as situações de “response.code() == 429” da API (O código de resposta HTTP 429 significa "Too Many Requests". Esse erro é usado quando um utilizador excedeu o limite de uso de uma API ou website, normalmente definido para prevenir o abuso e garantir a disponibilidade do serviço para todos os usuários.”

Essas situações podem acontecer, em especial, ao executar a class run.FourthLoadPartners (ver passo 3, mais abaixo)

# Artsy Api *config*

CLIENT\_ID=*9f14b9f06093d764ddf7*

CLIENT\_SECRET=*a8f0f836b1832d14bdc0e3e4c75559c4*

XAPP\_TOKEN=*https*://*api*.*artsy*.*net*/*api*/*tokens*/*xapp\_token*

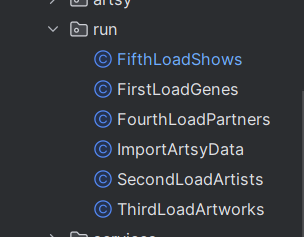
MAX\_ATTEMPTS\_API=*5*

1. Correr o script de criação da base de dados e das tabelas.

Ver tabela seguinte, ou então o ficheiro “Grupo 7 - Artsy-DB-EN.sql” que encontra na directoria “resources\database”

|  |
| --- |
| DROP Database if exists artsydb;    create database artsydb;  use artsydb;  CREATE TABLE User  (  name VARCHAR(150),  Email VARCHAR(100) NOT NULL,  id\_User INT NOT NULL AUTO\_INCREMENT,  username VARCHAR(100) NOT NULL,  password VARCHAR(100) NOT NULL,  PRIMARY KEY (id\_User) ,  UNIQUE (Email),  UNIQUE (username)  );  CREATE TABLE Gallerist  (  start\_at DATE NOT NULL,  end\_at DATE,  id\_Gallerist INT NOT NULL,  PRIMARY KEY (id\_Gallerist),  FOREIGN KEY (id\_Gallerist) REFERENCES User(id\_User)  );  CREATE TABLE Coordinator  (  start\_at DATE NOT NULL,  end\_at DATE,  id\_Coordinator INT NOT NULL,  PRIMARY KEY (id\_Coordinator),  FOREIGN KEY (id\_Coordinator) REFERENCES User(id\_User)  );  CREATE TABLE Artist  (  location VARCHAR(250),  hometown VARCHAR(250),  id\_Artist CHAR(40) NOT NULL,  name VARCHAR(500),  biography VARCHAR(2500),  slug VARCHAR(500),  birthyear VARCHAR(50) NOT NULL,  deathyear VARCHAR(50),  thumbnail VARCHAR(500),  url VARCHAR(250),  nationality VARCHAR(50) NOT NULL,  PRIMARY KEY (id\_Artist)  );  CREATE TABLE Gene  (  id\_Gene CHAR(40) NOT NULL,  name VARCHAR(500) NOT NULL,  description VARCHAR(2500),  PRIMARY KEY (id\_Gene)  );  CREATE TABLE Exhibition\_Status  (  Id\_Exhibition\_Status INT NOT NULL AUTO\_INCREMENT,  Status VARCHAR(15) NOT NULL,  PRIMARY KEY (Id\_Exhibition\_Status)  );  CREATE TABLE Partner  (  region VARCHAR(250),  url VARCHAR(250),  name VARCHAR(500) NOT NULL,  id\_Partner CHAR(40) NOT NULL,  website VARCHAR(500),  id\_Gallerist INT NOT NULL,  id\_Coordinator INT NOT NULL,  PRIMARY KEY (id\_Partner),  FOREIGN KEY (id\_Gallerist) REFERENCES Gallerist(id\_Gallerist),  FOREIGN KEY (id\_Coordinator) REFERENCES Coordinator(id\_Coordinator)  );  CREATE TABLE Artwork  (  title VARCHAR(500) NOT NULL,  date VARCHAR(100) NOT NULL,  id\_Artwork CHAR(40) NOT NULL,  thumbnail VARCHAR(500),  url VARCHAR(250),  created\_at TIMESTAMP NOT NULL,  updated\_at TIMESTAMP NOT NULL,  id\_Partner CHAR(40),  PRIMARY KEY (id\_Artwork),  FOREIGN KEY (id\_Partner) REFERENCES Partner(id\_Partner)  );  CREATE TABLE Exhibition  (  end\_at TIMESTAMP,  start\_at TIMESTAMP NOT NULL,  thumbnail VARCHAR(500),  description VARCHAR(2500),  id\_Exhibition CHAR(40) NOT NULL,  name VARCHAR(500) NOT NULL,  url VARCHAR(500),  id\_Partner CHAR(40) NOT NULL,  Id\_Exhibition\_Status INT NOT NULL,  PRIMARY KEY (id\_Exhibition),  FOREIGN KEY (id\_Partner) REFERENCES Partner(id\_Partner),  FOREIGN KEY (Id\_Exhibition\_Status) REFERENCES Exhibition\_Status(Id\_Exhibition\_Status)  );  CREATE TABLE Artwork\_Gene  (  id\_Artwork CHAR(40) NOT NULL,  id\_Gene CHAR(40) NOT NULL,  FOREIGN KEY (id\_Artwork) REFERENCES Artwork(id\_Artwork),  FOREIGN KEY (id\_Gene) REFERENCES Gene(id\_Gene)  );  CREATE TABLE Created\_By  (  id\_Artist CHAR(40) NOT NULL,  id\_Artwork CHAR(40) NOT NULL,  FOREIGN KEY (id\_Artist) REFERENCES Artist(id\_Artist),  FOREIGN KEY (id\_Artwork) REFERENCES Artwork(id\_Artwork)  );  CREATE TABLE Exhibition\_Artwork  (  id\_Artwork CHAR(40) NOT NULL,  id\_Exhibition CHAR(40) NOT NULL,  FOREIGN KEY (id\_Artwork) REFERENCES Artwork(id\_Artwork),  FOREIGN KEY (id\_Exhibition) REFERENCES Exhibition(id\_Exhibition)  );  INSERT INTO User (Email,username,password) values ('gallerist@cedricArt.pt', 'gallerist', 'password' );  INSERT INTO User (Email,username,password) values ('coordinator@cedricArt.pt', 'coordinator', 'password' );  insert into Gallerist (id\_Gallerist,start\_at) values (1, '2023-11-20');  insert into Coordinator (id\_Coordinator,start\_at) values (2,'2023-11-21');  insert into Exhibition\_Status(Status) values ('upcoming'), ('running'), ('closed'), ('current'); |

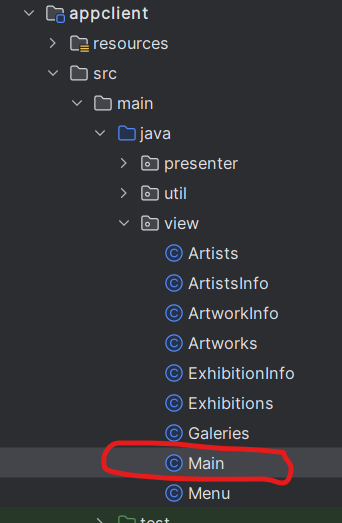
1. Executar as seguintes classes que se encontra no módulo “appimport”, e pela seguinte ordem:
2. run.FirstLoadGenes
3. run.SecondLoadArtists
4. run.ThirdLoadArtworks
5. run.FourthLoadPartners
6. run.FifthLoadShows



Cada uma dessas classes irá importar os dados respetivos da API Artsy para a base de dados local. Caso aconteça algum erro, voltar a executar os passos 2 e 3).

|  |
| --- |
| NOTA: Também é possível importar os dados diretamente para a base de dados, sem executar o passo 2.  Para isso, importar os dados que estão directoria “resources\database”:   * LightDataDump: Caso apenas se pretenda dados idênticos ao IS\_FAST\_ARSTY\_LOAD = true (ver passo 2) * FullDataDump: Caso apenas se pretenda dados idênticos ao IS\_FAST\_ARSTY\_LOAD = false (ver passo 2) |

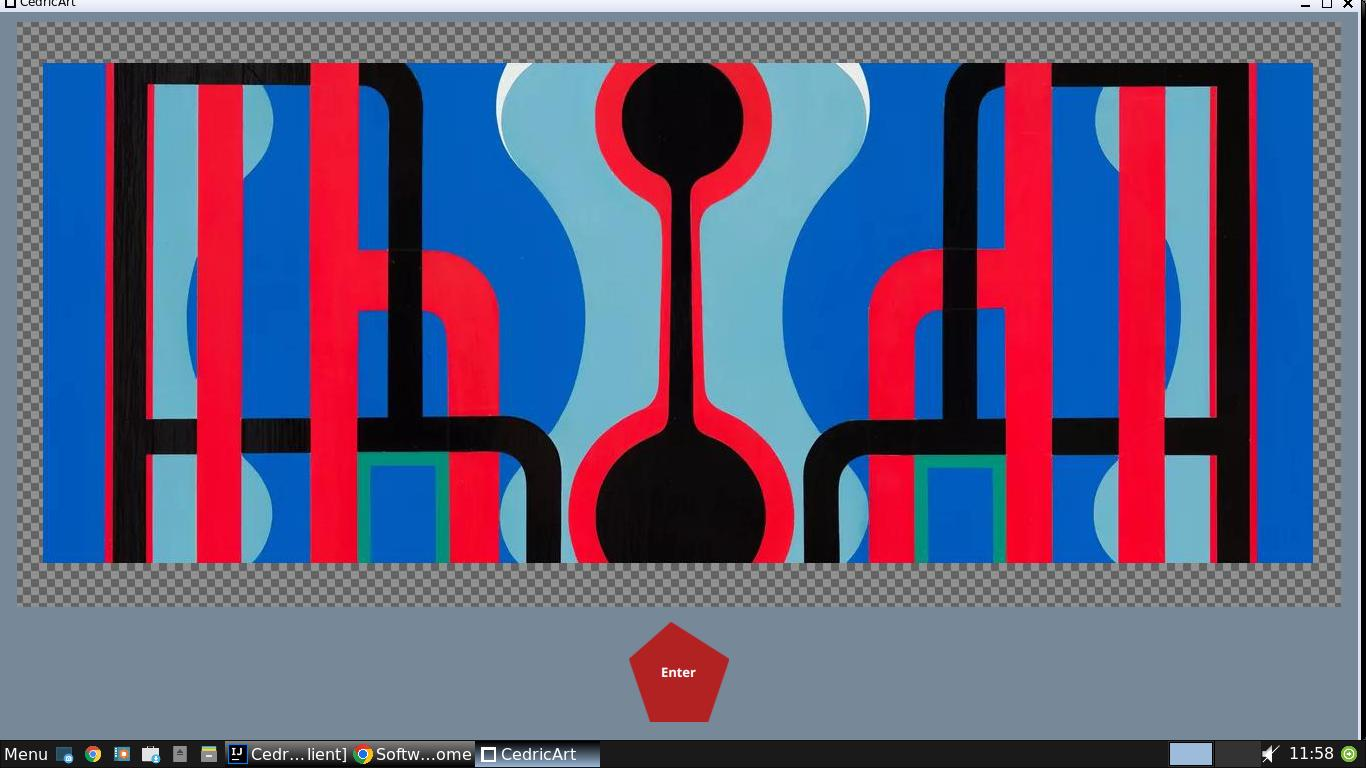
Após esses 3 passos serem executados com sucesso (significa que os dados foram importados da Artsy para a base de dados local), então podemos instalar o módulo “appclient” e disponibilizar o programa para os visitantes – ver “Uso do Programa” (próxima secção), executando a seguinte class “view.main”



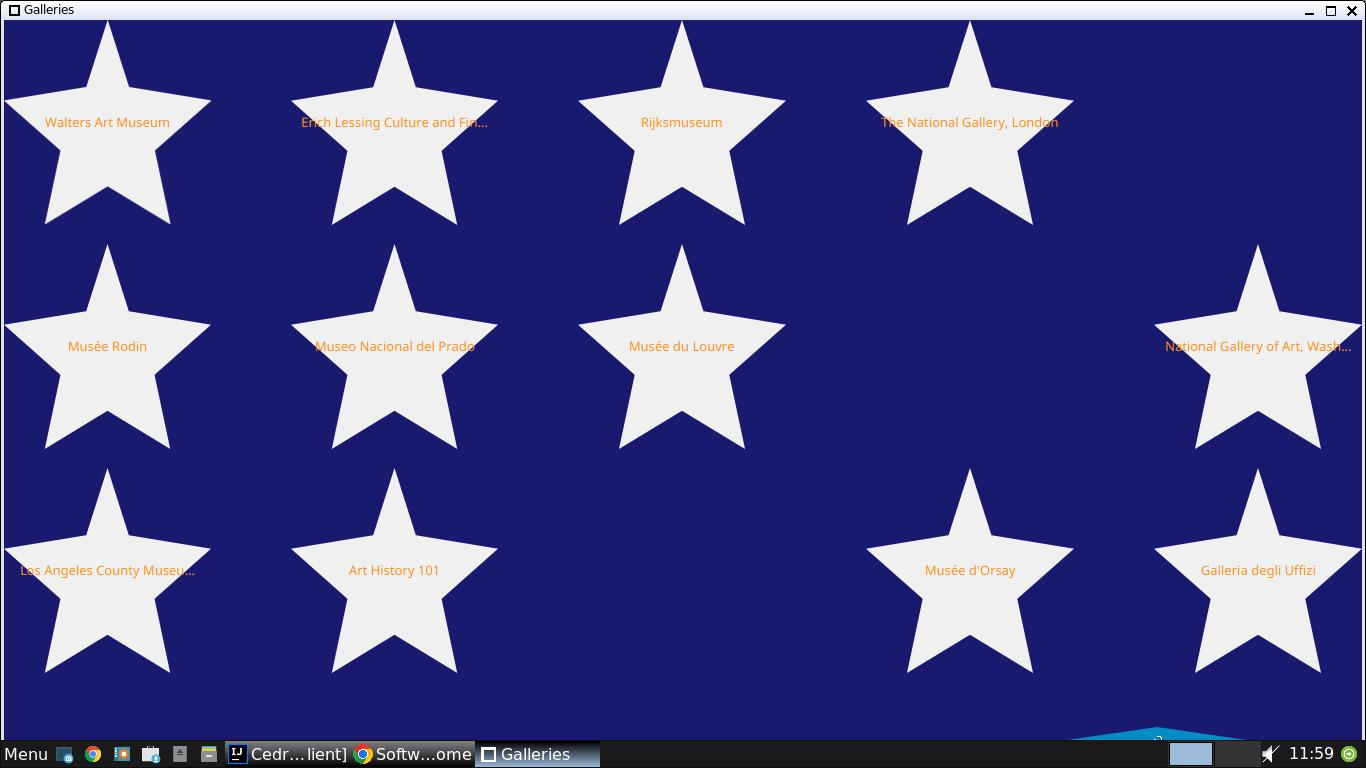
# Uso do Programa

Esta seção descreve como usar o programa, incluindo suas principais funcionalidades.

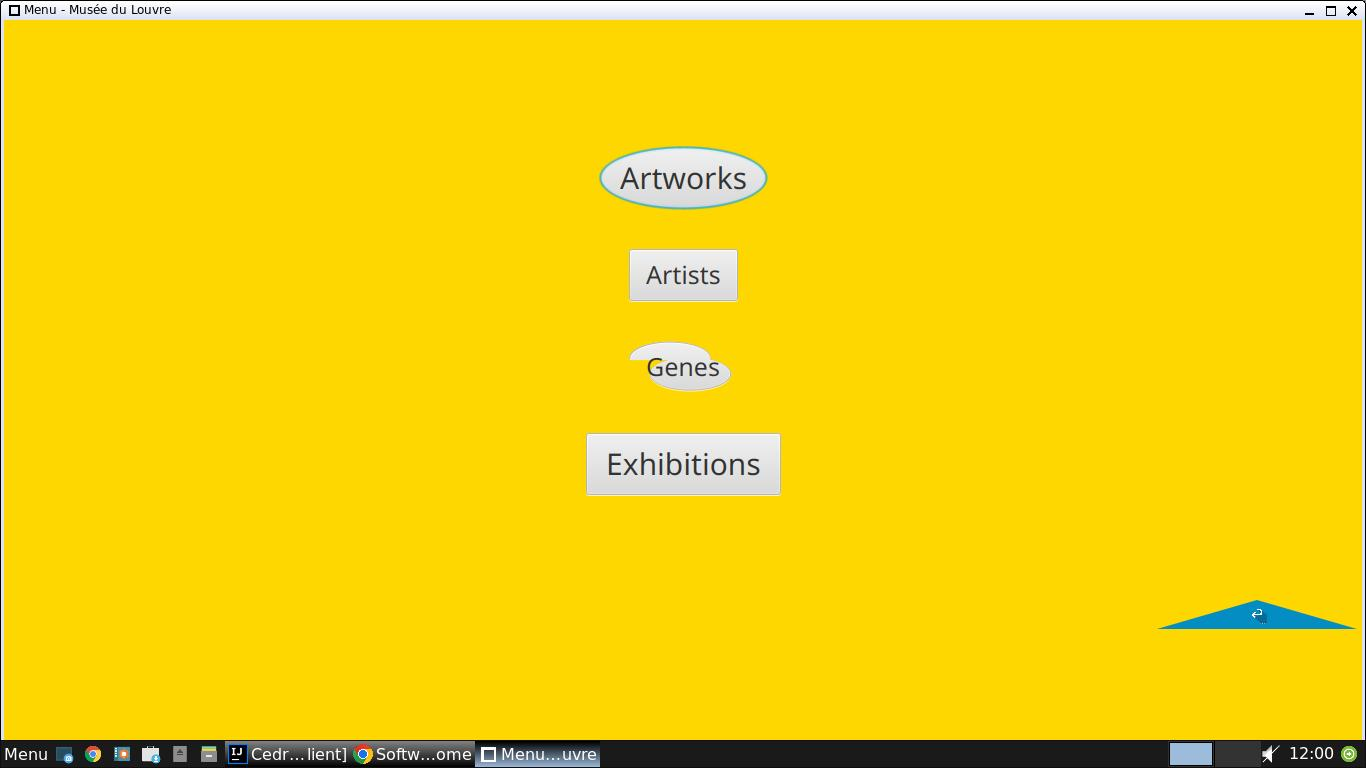
1 - Menu principal: Clicar no botão “Enter” para aceder às Galerias disponíveis.



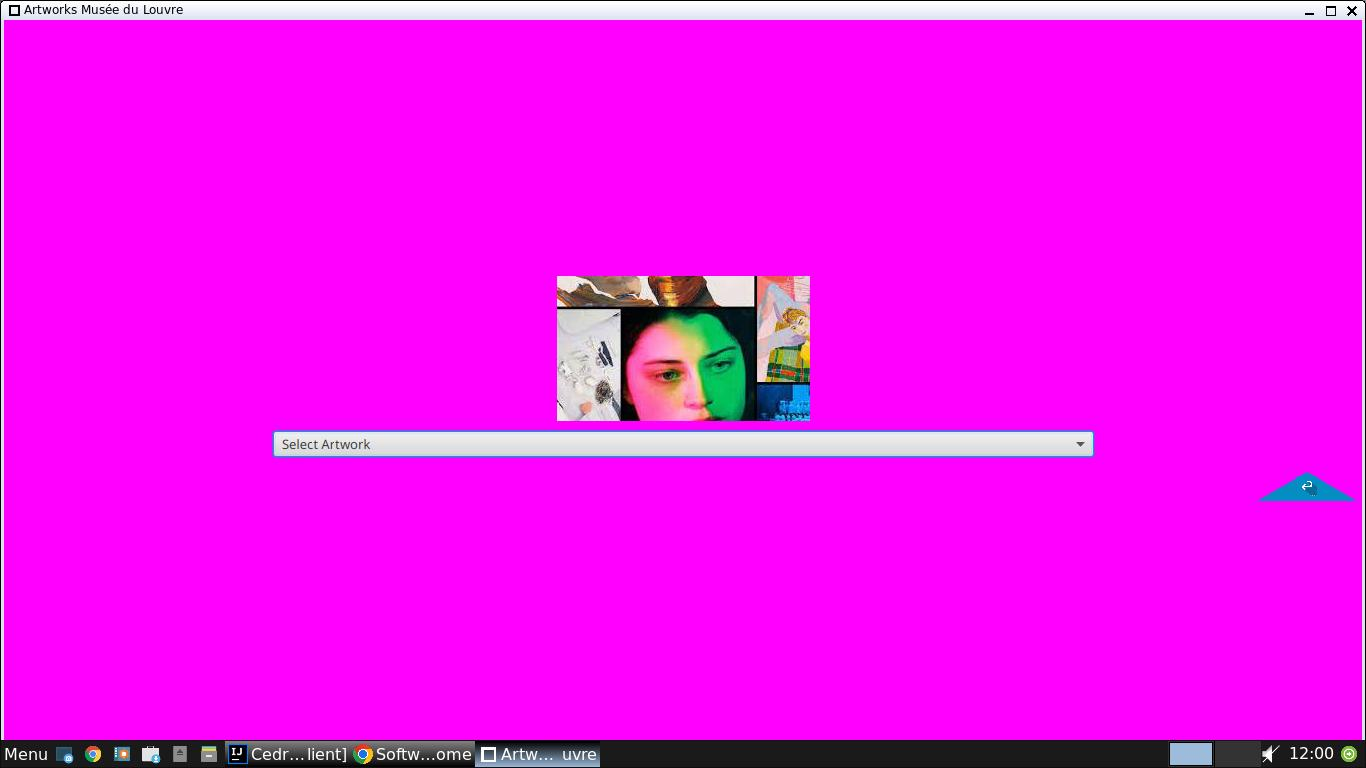
2 - Escolher a Galeria pretendida e clicar respectivamente.



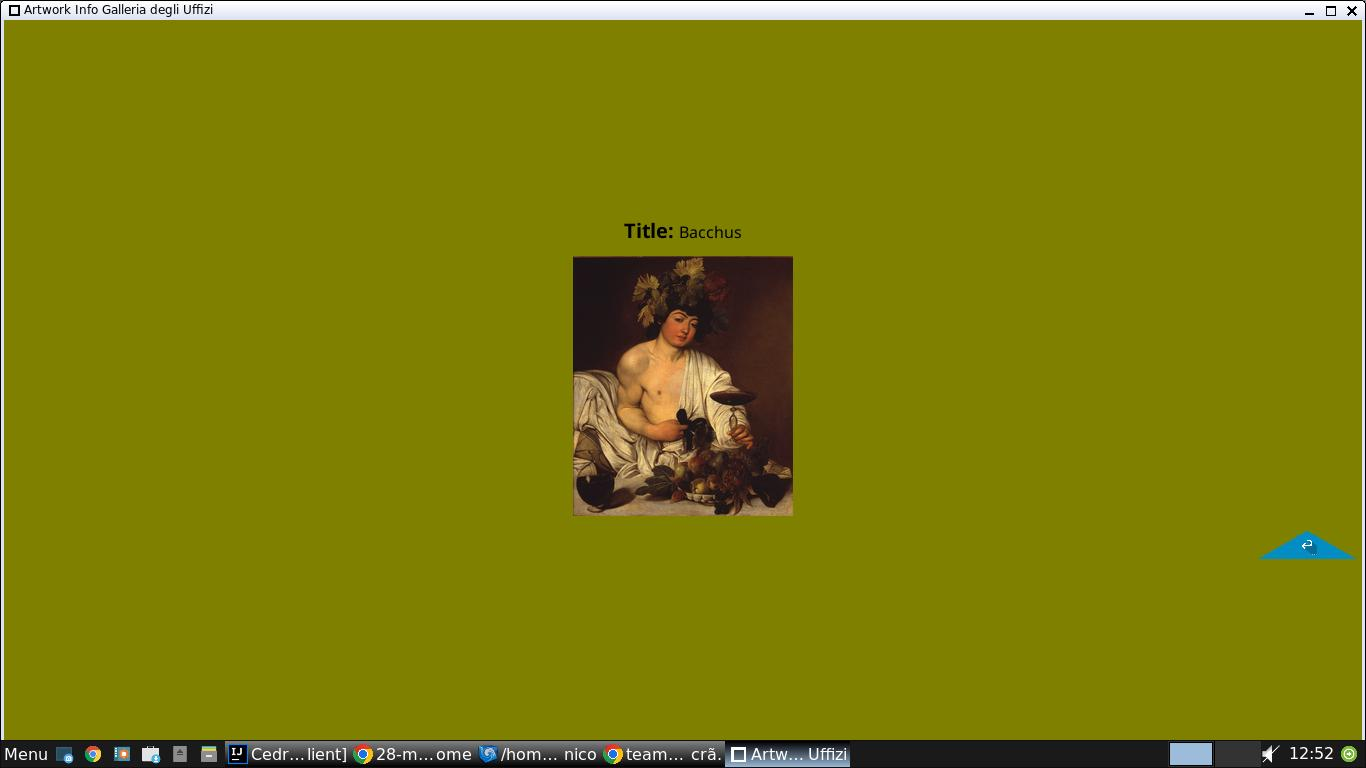
3 - Escolher entre “Artworks”, “Artists”, “Genes”(em desenvolvimento) e “Exhibitions” e clicar respetivamente.



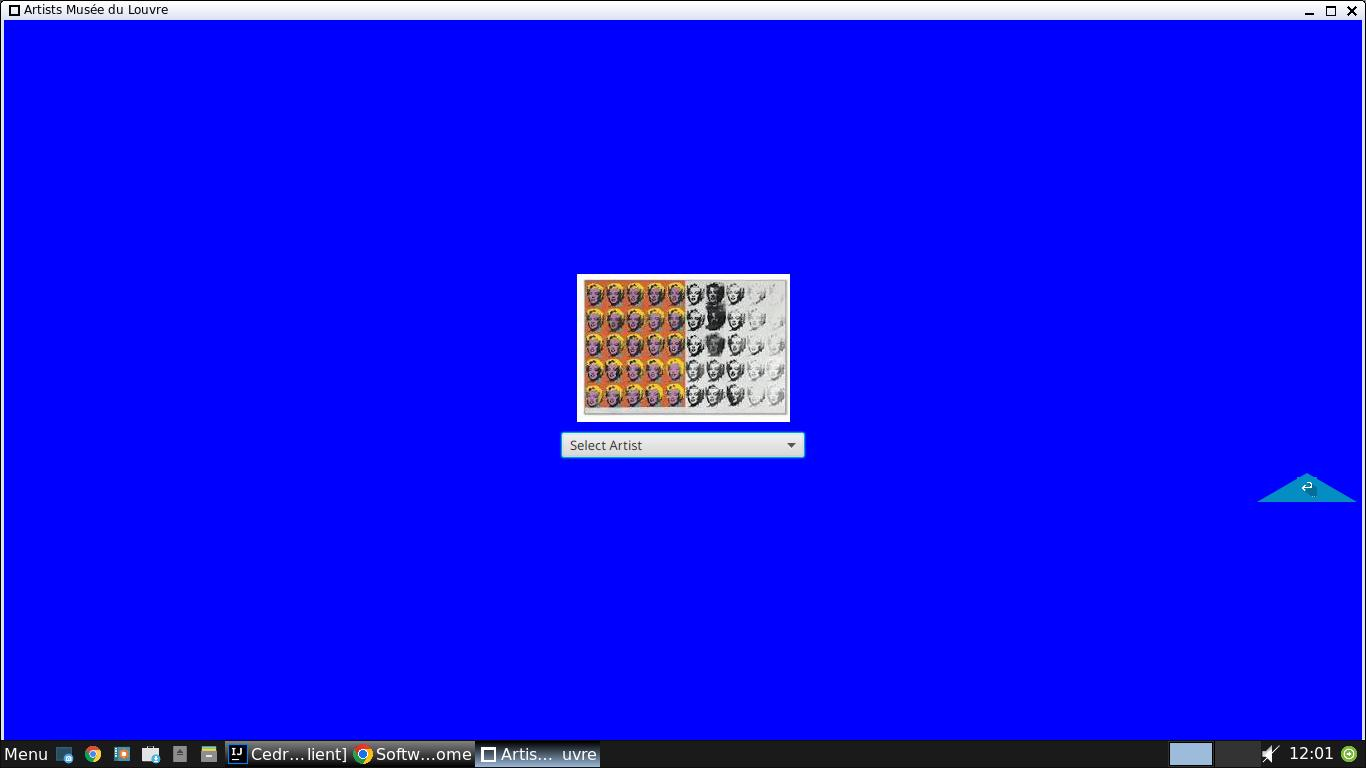
4 - Ao selecionar a opção “Artworks” irá surgir todas as Obras de Arte que existem na Galeria previamente selecionada. Clicar na comboBox e clicar na Obra de Arte pretendida.



5 - Irá aparecer a Informação da Obra de Arte previamente selecionada: Title e Image.



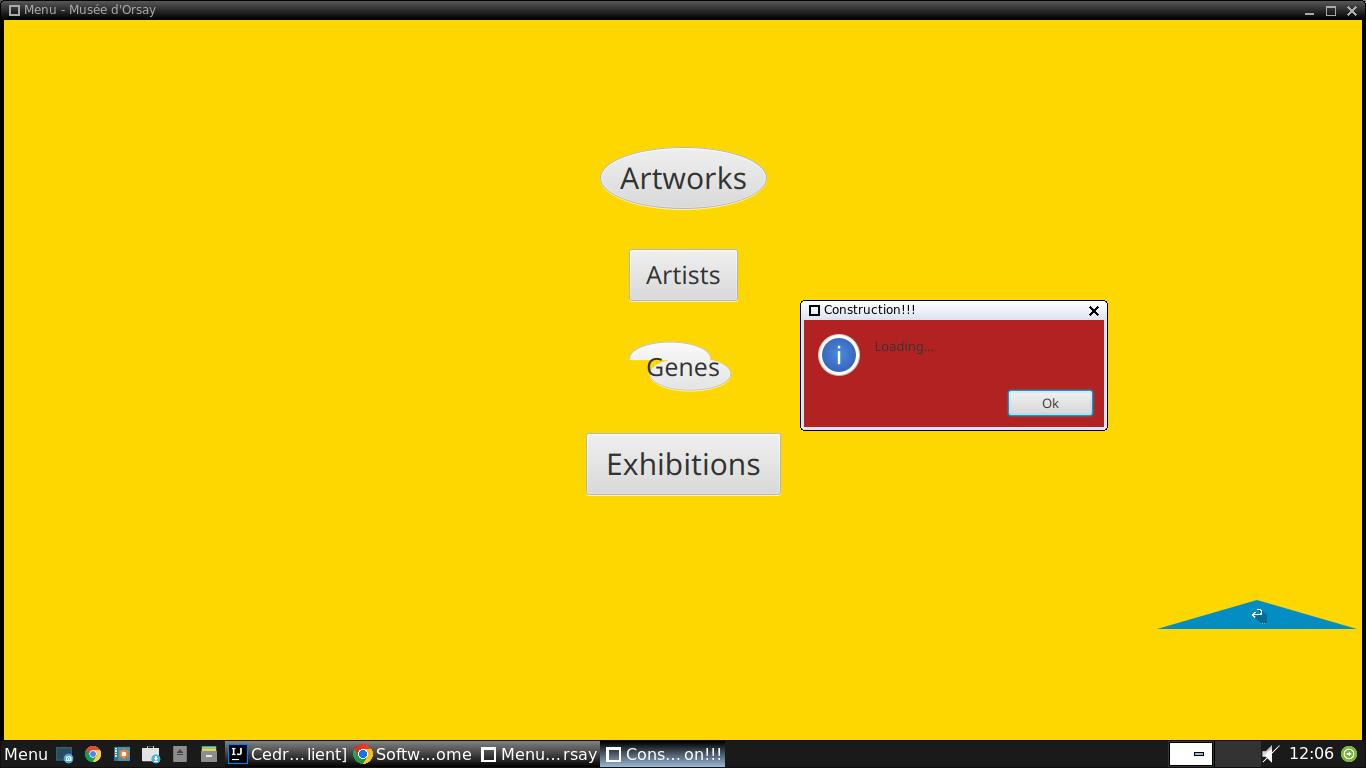
6 - Ao selecionar a opção “Artists” irão surgir todos os Artistas que existem na Galeria previamente selecionada. Clicar na comboBox e clicar no Artista pretendido.



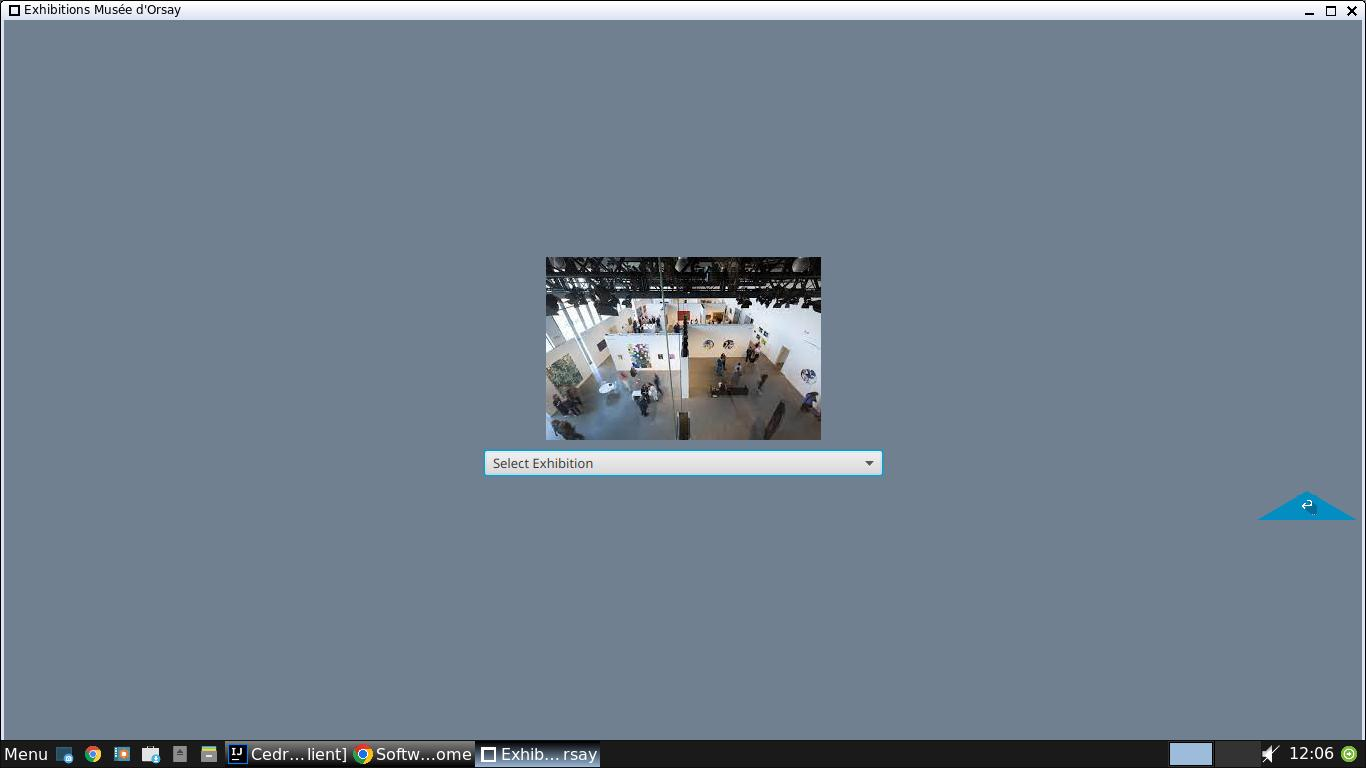
7 - Irá aparecer a Informação do Artista previamente selecionado: Name, Hometown, Biography e Imagem.



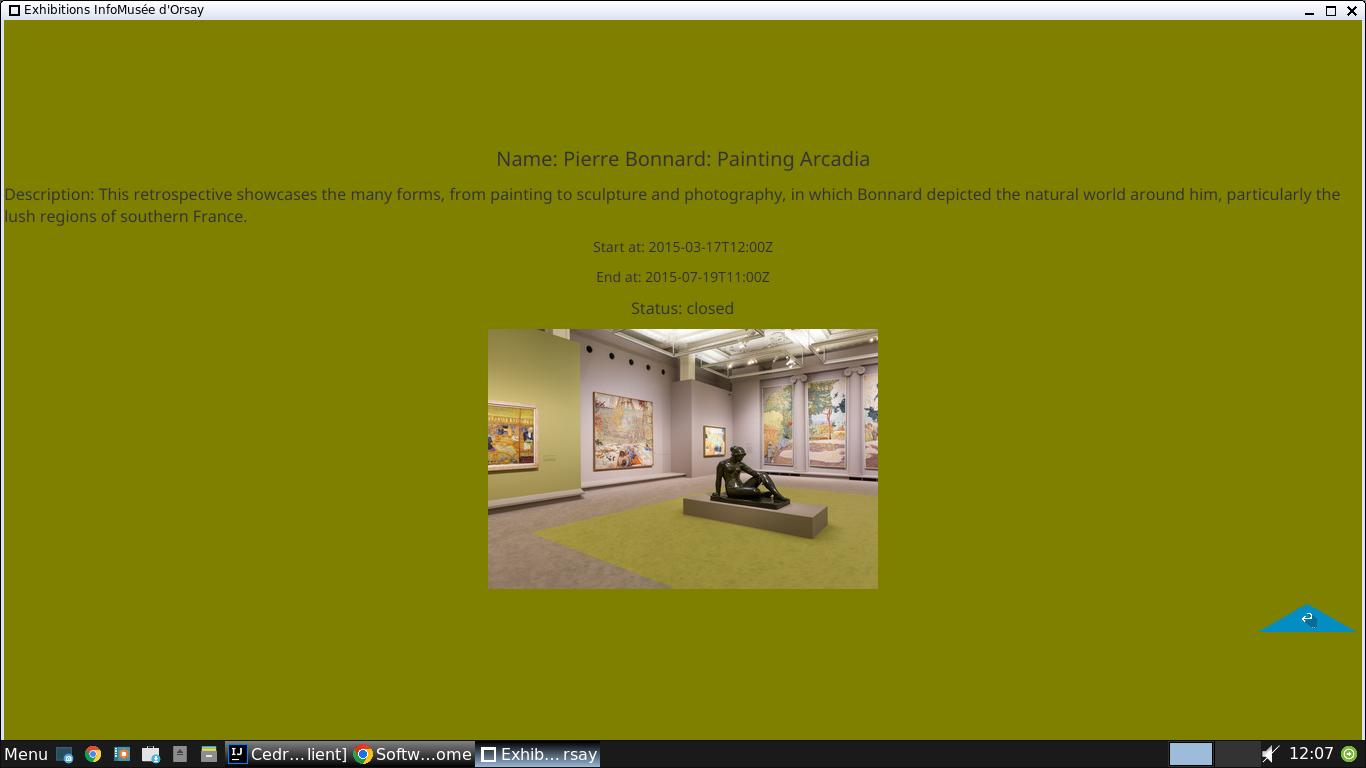
8 - Ao selecionar a opção “Genes” irá surgir a informação de página em construção.



9 - Ao selecionar a opção “Exhibition” irão surgir todas as Exibições que existem na Galeria previamente selecionada. Clicar na comboBox e clicar na Exibição pretendida.



10 - Irá aparecer a Informação da Exibição previamente selecionada: Name, Description, Start at, End at, Status e Image.



# 

# 

# 

# 

# 

# FAQs

## 1. Qual é o propósito do programa "CedricArt"?

O programa "CedricArt" foi desenvolvido para fornecer informações sobre exposições artísticas, artistas, galerias e obras de arte. Ele permite aos usuários explorar o mundo da arte, visualizar exposições e obter detalhes relevantes sobre o cenário artístico.

## 2. Como faço para acessar informações sobre uma exposição específica?

Para acessar informações sobre uma exposição específica, siga os seguintes passos:

* No menu principal, escolha a opção "Exhibitions".
* Selecione a exposição desejada na lista apresentada.
* Informações detalhadas sobre a exposição, como nome, descrição, datas e status, serão exibidas.

## 3. Posso obter detalhes sobre artistas individuais usando o programa?

Sim, o programa permite que você obtenha informações detalhadas sobre artistas individuais. Siga estas etapas:

* No menu principal, escolha a opção "Artists".
* Selecione o artista desejado na lista apresentada.
* Visualize informações sobre o artista, incluindo obras de arte associadas e detalhes biográficos.

## 4. Como voltar para o menu principal?

Em qualquer parte do programa, você pode retornar ao menu principal pressionando o botão de retorno. Geralmente, o botão de retorno é identificado por uma seta apontando para a esquerda ou por um ícone representando uma seta.

## 5. Existe suporte técnico disponível para questões relacionadas ao programa?

Sim, caso encontre dificuldades ou tenha perguntas relacionadas ao programa "CedricArt", entre em contacto com nosso suporte técnico pelo email cedric.grueau@estsetubal.ips.pt. Estamos aqui para ajudar a garantir uma experiência positiva de utilização.

Essas perguntas frequentes abordam alguns dos tópicos mais comuns relacionados ao programa "CedricArt". Se você tiver outras dúvidas, não hesite em entrar em contato com o suporte técnico.

# Documente o Código Fonte (se aplicável):

## Arquitetura do Programa

O programa "CedricArt" segue uma arquitetura orientada a objetos, implementada em Java usando o framework JavaFX para a interface gráfica do usuário (GUI). A estrutura geral do programa é organizada em classes, proporcionando modularidade e facilitando a manutenção e expansão do código.

## A arquitetura é composta por três partes principais:

* Modelo de Dados: Classes que representam entidades fundamentais, como Exhibition, Artist, Partner, e outras, armazenando e manipulando dados relacionados ao domínio do programa.
* Interface Gráfica do Usuário (GUI): Implementada usando JavaFX, proporciona uma interação intuitiva com o usuário. As classes associadas à GUI incluem ExhibitionInfo, Exhibitions, Artists, Galleries, entre outras.
* Lógica de Controle: Gerencia a lógica do programa e a interação entre o modelo de dados e a interface gráfica. Classes como CedricArtPresenter controlam a recuperação e manipulação de dados.

# Bibliotecas Usadas

## O programa utiliza as seguintes bibliotecas e frameworks:

* JavaFX: Utilizado para desenvolvimento da interface gráfica do usuário.
* Java Standard Library: Amplamente empregada para funcionalidades essenciais.

#### 

# Documentação de Classes e Métodos

## 1. ExhibitionInfo Class:

* Método start(Stage primaryStage) throws Exception:
* Inicializa a interface da janela de informações da exposição.
* Método updateInfo(Exhibition exhibition):
* Atualiza dinamicamente as informações da exposição exibidas na interface.

##### 

## 2. Exhibitions Class:

* Método start(Stage primaryStage) throws Exception:
* Inicializa a interface principal para visualização de exposições.
* Método fetchRandomExhibition(List<Exhibition> lExhibition):
* Preenche a lista de exposições exibidas na interface.
* Método listExhibitions(): List<Exhibition>:
* Recupera a lista de exposições associadas a um parceiro.

##### 

## 3. Galeries Class:

* Método start(Stage primaryStage) throws Exception:
* Inicializa a interface principal para visualização de galerias.
* Método openMenu(Partner partner):
* Abre o menu associado a uma galeria específica.

##### 

## 4. Main Class:

* Método start(Stage primaryStage) throws Exception:
* Ponto de entrada principal do programa.
* Método openGalleriesForm():
* Abre a interface principal após o botão "Enter" ser pressionado.

Essa documentação fornece uma visão geral da arquitetura, bibliotecas e principais métodos de algumas classes-chave no código-fonte. Para obter detalhes mais específicos, consulte o código-fonte e os comentários integrados.

# Modelo de manual técnico - Sistema CedricArt

## Introdução

O Manual Técnico do sistema CedricArt destina-se a programadores envolvidos na análise, extensão e correção do programa, bem como a stakeholders que participam na análise do sistema. Este documento fornece detalhes cruciais de análise e design, destacando os diagramas de classes e de pacotes, que são os principais artefactos visuais utilizados na disciplina de Programação Orientada a Objetos (POO).

## Diagrama de Classes

O diagrama de classes é uma representação visual das classes no sistema CedricArt e das relações entre elas. Abaixo está um exemplo resumido do diagrama de classes:

+---------------------+ +---------------------+ +----------------------+

| Exhibition | | Artwork | | Partner |

+---------------------+ +---------------------+ +----------------------+

| -id: int | | -id: int | | -id: int |

| -name: String | | -title: String | | -name: String |

| -startDate: Date | | -artist: Artist | | -location: Location |

| -endDate: Date | | -medium: String | | -email: String |

+---------------------+ +---------------------+ +----------------------+

| +getExhibitionInfo()| | +getArtworkInfo() | | +getPartnerInfo() |

+---------------------+ +---------------------+ +----------------------+

## Descrição do Diagrama de Classes:

* Exhibition:
  + A classe que representa uma exposição.
  + Atributos: id, name, startDate, endDate.
  + Métodos: getExhibitionInfo().
* Artwork:
  + A classe que representa uma obra de arte.
  + Atributos: id, title, artist, medium.
  + Métodos: getArtworkInfo().
* Partner:
  + A classe que representa um parceiro do CedricArt.
  + Atributos: id, name, location, email.
  + Métodos: getPartnerInfo().

## Diagrama de Pacotes

O diagrama de pacotes destaca a estrutura de pacotes do sistema CedricArt, proporcionando uma visão geral da organização do código.

+------------------+

| cedricart |

+------------------+

| -Exhibition |

| -Artwork |

| -Partner |

| -Location |

+------------------+

| +main(args: []) |

+------------------+

## Descrição do Diagrama de Pacotes:

* cedricart:
  + O pacote principal contém as classes Exhibition, Artwork, Partner e Location.
  + Método principal: main().

## Conclusão

O Manual Técnico fornece uma visão detalhada da estrutura interna do sistema CedricArt. Os diagramas de classes e de pacotes são ferramentas essenciais para compreender a arquitetura do programa. Para obter informações mais detalhadas sobre cada classe e método, consulte a documentação do código-fonte.

Este documento é vital para aqueles que desejam entender e contribuir para o desenvolvimento contínuo do sistema CedricArt. Para detalhes específicos de implementação, consulte o código-fonte disponível.

# Modelo de relatório Técnico para o projeto

Documentos Usados para a Análise do Problema:

## Enunciado do Problema:

* O enunciado do problema forneceu informações essenciais sobre os requisitos e objetivos do projeto "CedricArt". Ele definiu claramente o contexto do sistema, identificou as entidades principais (Exhibition, Artist, Partner, etc.) e especificou os principais fluxos de trabalho.

## Sites de Referência:

* Durante a análise do sistema, foram consultados diversos sites de referência relacionados a JavaFX, boas práticas de programação orientada a objetos e design de interfaces gráficas. Esses recursos foram úteis para a implementação eficiente da interface do usuário e a resolução de desafios específicos relacionados à tecnologia utilizada.

## Análise Substantivo/Verbo:

A técnica de análise substantivo/verbo, por meio de Cartas CRC (Classe-Responsabilidade-Colaborador), foi empregada para identificar as entidades principais do sistema, suas responsabilidades e as colaborações entre elas. Isso ajudou a estabelecer uma compreensão clara da estrutura do programa e a definir as interações entre as classes.

## Desenho da Solução:

O desenho da solução envolveu a criação de diagramas de classe e pacotes, os quais são os principais artefactos visuais na disciplina de Programação Orientada a Objetos (POO). Esses diagramas fornecem uma representação visual das relações entre as classes, seus atributos e métodos, bem como a organização em pacotes.

Exemplo de Diagrama de Classe JavaFX:

## Descrição Estratégia Tecnológica:

A estratégia tecnológica adotada inclui o uso do framework JavaFX para a construção da interface gráfica do usuário, aproveitando sua flexibilidade e recursos visuais. A arquitetura orientada a objetos foi escolhida para promover a modularidade e a manutenção do código.

## Conclusões:

## Arquitetura:

* A arquitetura orientada a objetos proporcionou uma estrutura modular, facilitando a expansão e manutenção do sistema. A separação clara entre modelo, interface gráfica e lógica de controle contribui para a flexibilidade.

## Diagrama de Classes:

* Os diagramas de classes fornecem uma visão abrangente das relações entre as entidades do sistema, facilitando a compreensão do código fonte.

## Persistência dos Dados:

* A persistência dos dados foi implementada por meio da classe CedricArtPresenter, garantindo a integridade das informações ao interagir com o banco de dados.

## Resultado Alcançado:

* O programa "CedricArt" atende aos requisitos especificados, proporcionando uma experiência intuitiva e eficiente.

## Requisitos Não Implementados:

* Todos os requisitos identificados foram implementados conforme o escopo definido. Não há requisitos significativos não implementados.

Este relatório técnico oferece uma visão geral da análise do sistema, destacando os documentos utilizados, a estratégia tecnológica adotada e os principais artefactos visuais gerados durante o processo de design. Para detalhes mais profundos, consulte os documentos anexos, incluindo diagramas completos e o código-fonte do projeto.