



Universidade do Minho

Escola de Engenharia

Licenciatura em Engenharia informática

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2023/2024

Leilões On-Line

Augusto Campos, a93320

Carlos Alberto Silva, a93199

Tiago Alexandre Silva, a93182

Tiago Pinheiro da Silva, a93285

6 de novembro de 2023

Data de Recepção	
Responsável	
Avaliação	
Observações	

Leilões On-Line

Augusto Campos, a93320
Carlos Alberto Silva, a93199
Tiago Alexandre Silva, a93182
Tiago Pinheiro da Silva, a93285

24 de janeiro de 2024

Resumo

No âmbito desta unidade curricular, Laboratórios de Informática IV, foi-nos proposto o desenvolvimento de um software sobre leilões online.

Existe uma óbvia divisão de etapas, podendo classificá-las em 3, muito distintas:

Definição e Fundamentação; Especificação; e Construção.

Na primeira etapa definimos os princípios essenciais, atendendo as necessidades e objetivos específicos do projeto, elaboramos um plano que nos ajude a desenvolver o mesmo e que descreva como será concebido o software.

Numa segunda fase, deveremos examinar, detalhar e especificar os requisitos do nosso sistema, a fim de criar um documento de especificação que sirva de base para toda a vida do projeto.

Na terceira e última fase, iremos implementar as ideias e princípios definidos nas fases anteriores, respeitando as funcionalidades a que nos comprometemos fazer.

Área de Aplicação: Engenharia de Software, Desenvolvimento de Aplicações, Processos de Pesquisa, Apresentação Pública, Escrita de Relatórios, Bases de Dados.

Palavras-Chave: Leilões, Planeamento, Software, Requisitos, Modelação, Sistema.

Índice

Conteúdo

Resumo	3
Índice	4
1 Definição do Sistema	6
1.1 Contextualização.....	6
1.2 Fundamentação	6
1.3 Objetivos.....	7
1.4 Viabilidade	7
1.5 Recursos a utilizar	7
1.6 Equipa de trabalho.....	8
1.7 Plano de execução do trabalho	9
2 Definição de Requisitos	11
2.1 Apresentação da estratégia e método	11
2.2 Descrição geral dos requisitos levantados	11
2.2.1 Requisitos Funcionais	11
2.2.2 Requisitos Não Funcionais	15
2.3 Validação dos requisitos estabelecidos.....	16
3 Especificação e Modelação do Software	17
3.1 Apresentação geral da especificação	17
3.2 Aspetos estruturais	17
3.3 Aspetos comportamentais.....	18
Use cases de Utilização Geral	19
Use cases do Admin.....	20
Use Cases do Utilizador	20
Use Cases do Utilizador Padrão (Cliente)	21

4 Conceção do Sistema de Dados	22
4.1 Apresentação geral da estrutura do sistema de dados	22
4.2 Descrição detalhada dos vários elementos de dados e seus relacionamentos.....	23
4.2.1 Utilizador	24
4.2.2 Lance	24
4.2.3 Leilão.....	25
4.2.4 WatchList.....	25
5 Esboço dos Interfaces do Sistema	26
5.1 Estrutura geral das interfaces do sistema	26
5.2 Caracterização das interfaces.....	26
6. Implementação e Desenvolvimento de Software	30
6.1 Descrição do Processo de Implementação	30
Ferramentas Utilizadas	30
6.2 Estruturas da Camada de Aplicação	31
.....	31
7. Conclusão	32

Índice de Figuras

1 Definição do Sistema

1.1 Contextualização

Na sociedade contemporânea, a dinâmica da venda e leilão de arte passou por uma transformação notável. Impulsionada pela globalização e pela digitalização, a comercialização de obras de arte transcendeu as barreiras geográficas e sociais. Plataformas online revolucionaram o mercado, permitindo que artistas independentes alcancem um público mais amplo, e ao mesmo tempo as casas de leilão tradicionais também adotaram o ambiente virtual. Além disso, a valorização da arte contemporânea e a crescente demanda por arte digital trouxeram desafios e oportunidades únicas. A arte agora é negociada em um cenário global, onde a autenticidade, a proveniência e a acessibilidade desempenham papéis cruciais na determinação de seu valor. Essa metamorfose na venda de arte reflete a evolução cultural e tecnológica de nossa sociedade, promovendo uma nova era de criatividade e consumo artístico. Tendo isso em vista, a "artVault" pretende avançar com o desenvolvimento de uma plataforma de leilões de arte online com destaque para obras de autoria portuguesa.

1.2 Fundamentação

A artVault surge como uma resposta à necessidade urgente de uma plataforma de leilões de arte online dedicada exclusivamente às obras de arte portuguesas. Portugal possui uma rica tradição artística, mas muitas vezes as obras excepcionais permanecem ocultas. Através da artVault, pretendemos não apenas conectar artistas talentosos aos entusiastas da arte, mas também fornecer uma plataforma segura e confiável para a venda, compra e promoção das obras de arte mais destacadas do país.

A plataforma compromete-se com a autenticidade, transparência e responsabilidade. Com uma equipe de especialistas em arte e tecnologia de ponta, garantimos a verificação rigorosa das obras e a identificação de artistas genuínos. A artVault é mais do que uma plataforma de leilões; é uma comunidade dedicada à celebração da rica herança artística de Portugal. Através de parcerias estratégicas, educação artística e um ambiente de mercado justo, a artVault tem a visão de se tornar o destino preferido para colecionadores, investidores e amantes da arte que buscam tesouros únicos da arte portuguesa.

1.3 Objetivos

O principal objetivo da artVault é promover e preservar a riqueza da arte portuguesa, proporcionando uma plataforma de destaque para artistas locais e as suas obras. Queremos dar visibilidade a talentos emergentes e obras de artistas consagrados, tornando-as acessíveis a um público global.

Buscamos criar um mercado de arte online seguro e transparente, onde compradores e vendedores possam interagir com confiança. Os nossos objetivos incluem a verificação rigorosa de autenticidade, a simplificação do processo de compra e venda, e a promoção da ética e integridade no mercado de arte português.

1.4 Viabilidade

A viabilidade económica da plataforma artVault é respaldada por diversos fatores-chave.

Em primeiro lugar, a crescente demanda global por arte portuguesa, muitas vezes subestimada, oferece um vasto mercado de potenciais compradores e colecionadores. Além disso, o ambiente de compras online em constante expansão favorece a adoção da plataforma, facilitando o alcance de um público internacional. A artVault procura uma estrutura de negócios sustentável, gerando receita por meio de comissões sobre vendas e parcerias estratégicas que agregam valor.

Com uma gestão financeira sólida, planos de crescimento estratégicos e uma equipa experiente, estamos bem posicionados para alcançar a viabilidade económica a longo prazo, ao mesmo tempo em que promovemos e valorizamos a riqueza da arte portuguesa.

1.5 Recursos a utilizar

Para o desenvolvimento de uma aplicação desta dimensão será indispensável alocar recursos de maneira eficiente. Teremos então recursos que irão abranger aspectos humanos, de software e de hardware. Este projeto será então supervisionado e desenvolvido por 4 alunos de Engenharia Informática da Universidade do Minho.

A nível de ferramentas e tecnologia, através do Google Sheets, do Google Docs e do GitHub, conseguimos nos coordenar de uma maneira eficaz, adjacente ao Discord e ao Outlook, que nos permitiu manter uma comunicação contínua e consistente entre os elementos, resultando num desenvolvimento produtivo e coeso. O Visual Paradigm desempenhou um papel fundamental no design do sistema, permitindo que a equipa mapeasse visualmente a estrutura do software, assim como o Figma, que foi importante para o design da interface do utilizador, aprimorando a estética e utilidade desta nossa aplicação. Na parte de implementação, prevemos utilizar o Visual Studio e C#. Estes ambientes de desenvolvimento

provaram-se eficientes e fiáveis; e o SQL Server, como um sistema de gerenciamento de dados, garantindo o armazenamento seguro e a recuperação dos mesmos.

Através do uso inteligente destes recursos acima enumerados, foi conseguido o sucesso a nível do desenvolvimento do software pretendido.

1.6 Equipa de trabalho

Para este projeto, foi estruturada uma equipa de trabalho composta por diversos profissionais com habilidades diversas, tendo cada elemento realizado um papel específico e essencial nas suas respetivas áreas, tendo sido a colaboração de todos fundamental para o sucesso do desenvolvimento do software.

Equipa constituída por 4 (quatro) alunos de Licenciatura em Engenharia Informática (LEI) da Universidade do Minho.



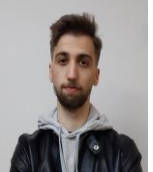

Identificação	Função	Retrato
Augusto César Oliveira Campos A93320	Engenheiro de Software	
Carlos Alberto Fernandes Dias da Silva A93199	Engenheiro de Software	
Tiago Pinheiro da Silva A93285	Engenheiro de Software	
Tiago Alexandre Ferreira Silva A93182	Engenheiro de Software	

Tabela 1 – Tabela de Composição da Equipa

1.7 Plano de execução do trabalho

Este planeamento surge da necessidade de organizar as etapas, e ter uma melhor perceção do tempo que cada uma durará, com a data de entrega em mente, para um melhor e mais responsável desenvolvimento deste sistema de leilões.

Embora não especificado no diagrama, a divisão de tarefas não foi feita apenas temporalmente, mas também por elementos, obtendo no final uma igual contribuição entre estes.

O diagrama usado é um diagrama de *Gantt*.

Leilões On-Line

Plano de Execução do Projeto

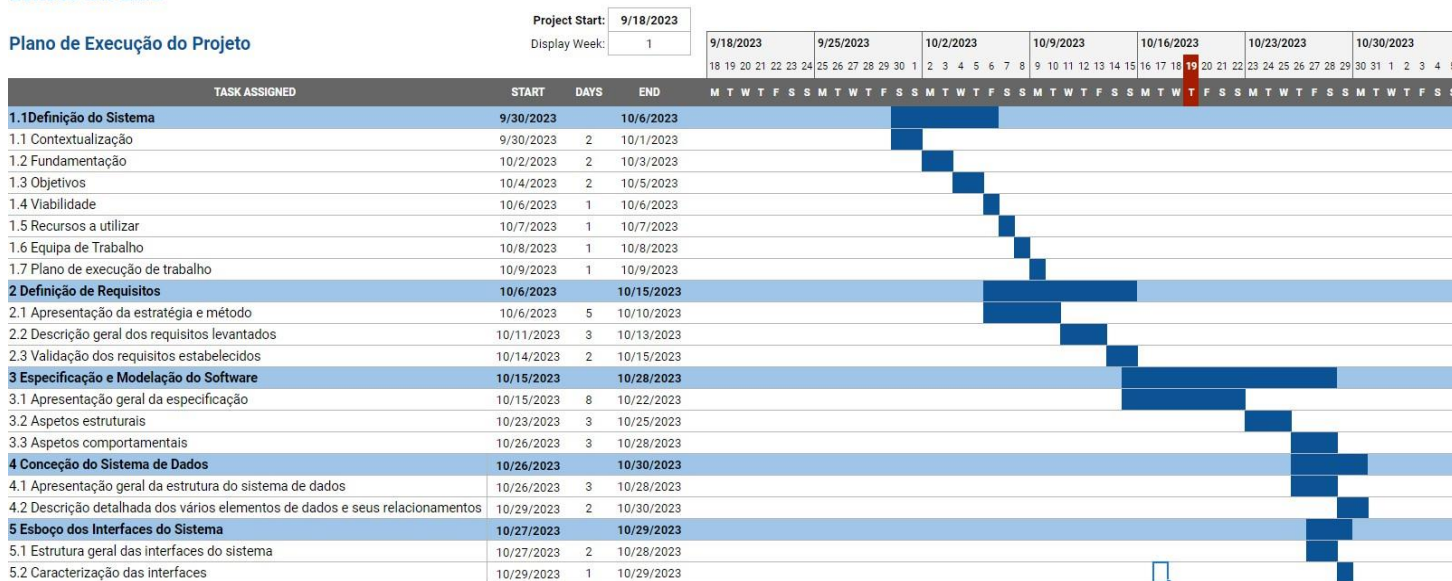


Figura 1 – Diagrama de GANTT Fase 1

Leilões On-Line

Plano de Execução do Projeto

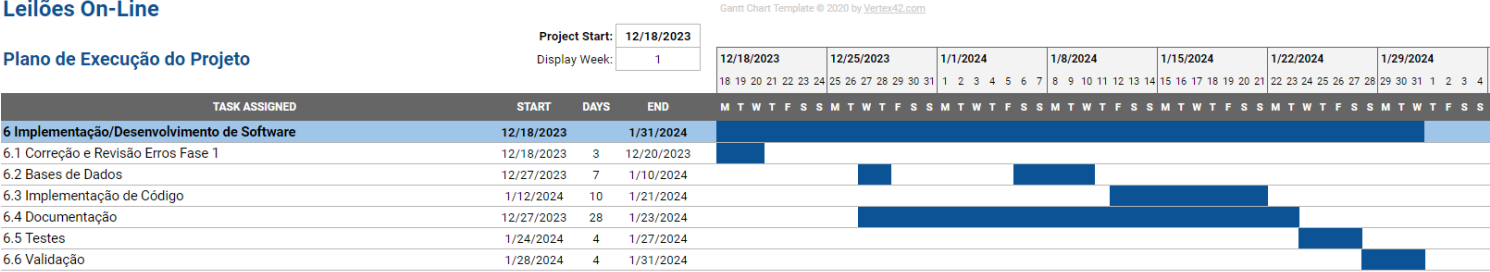


Figura 2 – Diagrama de GANTT Fase 2

2 Definição de Requisitos

2.1 Apresentação da estratégia e método

O método de levantamento e de análises de requisitos é algo fundamental no desenvolvimento de software por diversas razões. É através deste processo que a equipa entenderá as necessidades, objetivos e expectativas do seu cliente. Permite também que haja uma “linha” por onde seguir e desenvolver, sem deixar margem para um descontrolo desproporcionado durante o desenvolvimento, seja de funções que devem ser implementadas ou de tempo que será necessário para as tais. Para levantamento destes requisitos, utilizamos então Entrevistas, Observações e Análise de Documentos.

Primeiramente foram realizadas várias entrevistas com o cliente de forma a entender as necessidades. As entrevistas foram conduzidas individualmente e em grupo, de forma formal e/ou informal, e também foram feitas a clientes habituais de leilões, para que melhor pudéssemos entender este mundo.

Por seguida, as observações, de leilões, tanto on-line como presenciais, que nos forneceram valiosas informações sobre as necessidades não só do cliente, mas de um leilão em si, dos pilares que este precisa para ser um verdadeiro leilão e

Por último, realizou-se a análise de documentos, que se assemelha ao ponto anterior, a nível de estudo de como um leilão pode e deve funcionar, e as dinâmicas nele inseridas.

2.2 Descrição geral dos requisitos levantados

2.2.1 Requisitos Funcionais

RF01: Registo de Utilizador

- Requisitos de utilizador:
 1. A aplicação deverá permitir que um utilizador se registe na plataforma.
 2. As contas de artistas apenas devem ser confirmadas após revisão manual de um administrador autenticado.
- Requisitos do sistema:
 1. O sistema deverá solicitar o nome, CC, NIF, morada, username, password, e-mail, tipo de conta (padrão, artista).
 2. O sistema não deverá permitir utilizadores com o mesmo CC, NIF, username ou email.
 3. O sistema deverá atribuir um ID único a cada utilizador.

RF02: Autenticação na Aplicação

- Requisitos de utilizador:
 1. A aplicação deverá permitir que um utilizador faça login.
- Requisitos do sistema:
 1. O sistema deverá validar as credenciais introduzidas pelo utilizador.
 2. Se as credenciais introduzidas forem válidas, o utilizador efetua o login com sucesso.
 3. Se as credenciais introduzidas não forem válidas, o sistema deverá apresentar uma mensagem de erro.

RF03: Criação de Leilões

- Requisitos de utilizador:
 1. A aplicação deverá permitir que um artista registre um Leilão na plataforma.
 2. O artista deve estar autenticado na aplicação.
 3. O artista deve indicar o nome do produto a leiloar, o tipo de leilão, uma data de fim, um preço inicial que será colocado no preço de referência, o preço reservado, submeter uma imagem e dimensões, e, opcionalmente, pode fornecer uma descrição.
- Requisitos do sistema:
 1. O sistema deverá registar o Leilão, atribuindo-lhe um ID único.

RF04: Cancelar um Leilão

- Requisitos de utilizador:
 1. A aplicação deverá permitir que um artista cancele um Leilão que ele tenha iniciado previamente.
- Requisitos do sistema:
 1. O artista deve estar autenticado na aplicação.
 2. O sistema deve anular todos os Lances que tenham sido feitas no Leilão.
 3. O sistema deve notificar Utilizadores que tenham o leilão na sua Watchlist, ou que já tenham efetuado um Lance no leilão.

RF05: Finalizar leilão

- Requisitos do sistema:
 1. O sistema deveria finalizar um leilão quando o tempo do leilão terminar.
 2. O sistema deve garantir que o valor da bid vencedora é igual ou maior ao preço reservado quando este foi indicado na criação do leilão.
 3. O sistema deve notificar o Artista e o detentor da bid vencedora.
 4. O sistema deve informar o departamento de pagamentos com as informações necessárias para a realização da transferência.
 5. O sistema deve notificar Utilizadores que tenham o leilão na watchlist.

RF06: Visualizar um Leilão

- Requisitos de utilizador:
 1. A aplicação deve permitir a um utilizador visualizar todas as informações sobre um Leilão.
- Requisitos do sistema:
 2. O sistema deve garantir que o utilizador autenticado é um Artista ou comprador.
 3. O sistema deve obter as informações do leilão inseridas pelo artista no momento em que criou o leilão.
 4. Se o utilizador autenticado for um comprador, o sistema deve mostrar o nome do produto, tempo restante no leilão e valor da última bid realizada e histórico de bids do leilão.
 5. Se o utilizador autenticado for um artista, o sistema deve mostrar o nome do produto, tempo restante no leilão, valor da última bid realizada, histórico de bids do leilão e se for o seu próprio leilão deveria visualizar a opção de Cancelar um Leilão.

RF07: Visualizar o Catálogo de Leilões

- Requisitos de utilizador:
 1. A aplicação deve permitir a um utilizador visualizar os Leilões que estão em curso.
- Requisitos do sistema:
 1. O sistema deve garantir que o utilizador autenticado é um Artista ou comprador.

2. O sistema deve obter os leilões que ainda estão em curso, isto é, os leilões cujo tempo para ser leilado ainda não terminou.
3. O sistema deve mostrar ao utilizador autenticado os diversos leilões em curso, mostrando o nome do produto, tempo restante no leilão e valor da última bid realizada.

RF08: Fazer um lance num Leilão

- Requisitos de utilizador:
 1. A aplicação deve permitir a um comprador efetuar um lance num Leilão que esta em curso.
- Requisitos do sistema:
 1. O sistema deve garantir que o utilizador autenticado é um comprador.
 2. O sistema deve verificar qual é o tipo de leilão.
 3. Quando o tempo restante do leilão for significativamente pequeno, este deve aumentar de modo que não seja possível surpreender outros licitadores, se for um leilão inglês.
 4. O sistema deve garantir que o valor do lance é maior em pelo menos 10% do valor do último lance, caso seja um leilão inglês.
 5. O sistema deve notificar o Utilizador que fez a última bid, caso este exista.
 6. O sistema deve adicionar o Leilão à "Watch List" do Utilizador.

RF09: Validação de Vendedor (Artista) por um Admin

- Requisitos de utilizador:
 1. A aplicação deve permitir a um admin visualizar todos os artistas em espera de validação de conta.
- Requisitos do sistema:
 1. O sistema deve garantir que o utilizador autenticado é um admin.
 2. O sistema deverá permitir ao administrador aprovar uma conta de artista caso considere que essa conta reúne condições para ser aprovada. Nesse caso, o sistema deve marcar a conta de vendedor como aprovada.
 3. O sistema deverá eliminar a conta de artista caso o administrador não a aprove.

RF10: Adicionar um leilão à "Watch List"

- Requisitos de utilizador:
 1. A aplicação deve permitir a um utilizador adicionar um leilão à sua Watch List.
- Requisitos do sistema:
 1. O sistema deve garantir que o utilizador está autenticado.
 2. O sistema deverá adicionar o Leilão selecionado na Watch List.

RF11: Configuração "Watch List"

- Requisitos de utilizador:
 1. A aplicação deve permitir a um utilizador visualizar todos os Leiloes que este adicionou à sua Watch List e remover leiloes da mesma.
- Requisitos do sistema:
 2. O sistema deve garantir que o utilizador está autenticado.
 3. O sistema deverá remover leiloes selecionados da Watch List.

2.2.2 Requisitos Não Funcionais

RNF01: Ambiente de execução do programa

A parte do sistema que interage com os utilizadores deverá executar a partir de um browser de internet, e a parte do armazenamento de dados e da lógica de negócio deverá correr em ambiente Microsoft Windows.

RNF02 - Ferramentas a usar no desenvolvimento do programa

Para a elaboração do sistema, usaremos a framework .NET e como resultado a linguagem de programação C#. Também deverá ser usado o sistema Microsoft SQL Server para a gestão da base de dados necessária para o funcionamento do sistema.

RNF03 - Tempo de execução das funcionalidades interativas

Todas as funcionalidades do sistema que incluam interação direta com um utilizador, deverão ser executadas, preferencialmente num tempo menor que 5 segundos, e obrigatoriamente em menos de 15 segundos.

RNF04 - As contas de administrador não são criadas pelo sistema

O sistema não deverá conseguir criar contas de administrador. Estas contas deverão ser criadas com a inserção manual na base de dados, por um perfil com privilégios para tal.

2.3 Validação dos requisitos estabelecidos

De forma a validar os requisitos levantados, para garantir que as necessidades do cliente sejam compreendidas e atendidas de maneira eficaz foram feitas reuniões detalhadas com o cliente. O objetivo é de confirmar que o software desenvolvido está alinhado com as expectativas do cliente, evitando retrabalho e más surpresas no final do projeto.

3 Especificação e Modelação do Software

3.1 Apresentação geral da especificação

A partir dos requisitos por nós definidos, é possível determinar a estrutura e o comportamento pretendido da aplicação.

Os requisitos servem como um guia para entender como a aplicação deve ser construída e o que ela deverá fazer. Eles descrevem as funcionalidades e os objetivos gerais da aplicação, permitindo que elementos da equipa tenham uma visão clara do que precisa ser alcançado.

Para alcançar este objetivo, fizemos uso da *Unified Modeling Language*, que nos auxiliou na criação de diagramas que detalham a estrutura e o comportamento do software.

Nos aspectos estruturais, criamos um modelo de domínio para representar as entidades do sistema e as relações entre elas.

Nos aspectos comportamental, definimos um conjunto de Use Cases que descrevem as funcionalidades principais do programa. Através destes *uses cases*, um Diagrama de Use Cases foi produzido.

3.2 Aspetos estruturais

Como antes referido, foi elaborado um modelo de domínio pois este oferece uma representação visual clara das entidades do sistema e das suas relações.

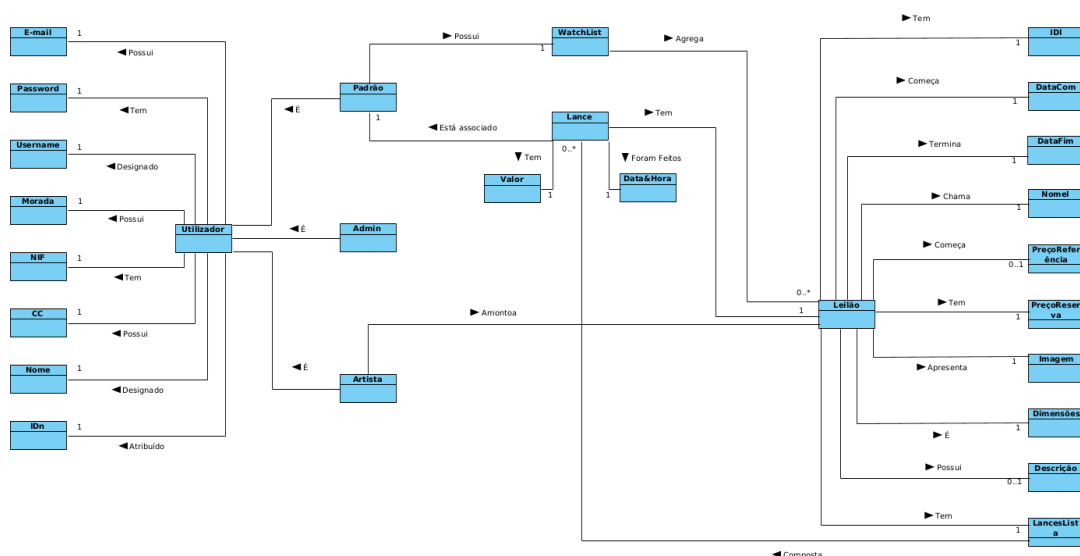


Figura 3 - Modelo de Domínio

Consideramos como principais entidades o Utilizador (e seus subtipos), o Leilão e o Lance.

O Utilizador Padrão poderá visualizar todos os leilões a decorrer no momento, os respetivos detalhes sobre este e fazer um lance sobre o mesmo. O Utilizador Padrão pode também adicionar leilões à sua “WatchList” que o permitirá seguir leilões em que está interessado e leilões nos quais licitou.

Sempre que um lance é feito, este é adicionado a lista de licitações dum leilão que permite ver o histórico de licitações feitas no mesmo.

Este modelo de domínio não nos permite distinguir entre os 2 tipos de leilões que serão implementados, no caso, o leilão inglês e o leilão de sealed-bid, sendo que ambos possuem algumas diferenças, por exemplo, num leilão inglês um Utilizador Padrão poderá fazer várias licitações no mesmo leilão, no entanto no leilão de sealed-bid o Utilizador Padrão poderá apenas fazer uma licitação, além de não ser possível ver o histórico de licitações até ao final do mesmo e não possuir um preço inicial.

3.3 Aspetos comportamentais

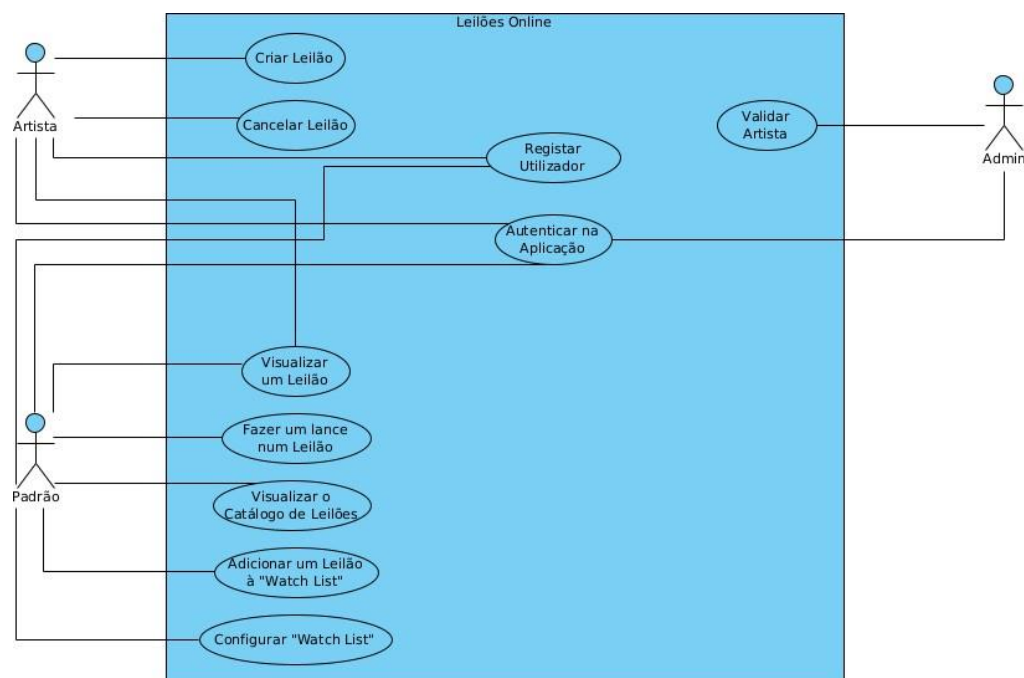


Figura 4 - Modelo de Use Cases

Como também antes referido, foram elaborados Use Cases e o consequente Diagrama de Use Cases, pois estes desempenham um papel vital na compreensão, documentação e comunicação dos requisitos do sistema.

Use cases de Utilização Geral

Autenticação na Aplicação

Para iniciar a sessão o utilizador deverá fornecer as suas credenciais (e-mail e password).

De seguida, o sistema verifica se o e-mail inserido está associado a alguma conta existente. Se a conta não existir, o sistema informa que a conta não existe. Quando existe, efetua-se a verificação da password e o utilizador fica autenticado caso esta esteja correta ou é informado que a password está errada.

Este use case satisfaz o **RF02**.

Registo de Utilizador

Para utilizar a plataforma, um utilizador precisa de estar registado. Para o efeito, o utilizador terá de fornecer os seguintes dados:

- Nome
- Número do cartão de cidadão
- NIF
- Nome de utilizador
- Password
- E-mail
- Morada
- Tipo de conta (padrão ou artista)

O sistema tem de garantir que o número do cartão de cidadão, NIF, nome de utilizador e e-mail não estão associados a nenhuma outra conta já existente. Caso um desses elementos não seja único, a aplicação informa o utilizador que a conta não pode ser criada por esse motivo.

Após a verificação da unicidade dos dados acima referenciados, o sistema cria de imediato a conta se esta for padrão atribuindo-lhe um identificador único. No caso de se tratar de uma conta de artista, esta ficará pendente de aprovação até que um admin a aprove posteriormente.

Este use case satisfaz o **RF01**.

Use cases do Admin

Validar Artista

Como referido anteriormente, o registo de contas de artista carece de uma aprovação realizada por um admin.

O sistema apresenta uma lista das contas que estão pendentes de aprovação. O utilizador pode então selecionar uma das contas para proceder a uma análise. A aplicação apresenta as informações da conta selecionada e, de seguida, decide se a conta será aprovada ou não.

Se for aprovada, a conta é ativada e pronta a ser usada pelo utilizador. Se não, a conta é descartada pelo sistema.

Este use case satisfaz o **RF09**.

Use Cases do Utilizador

Visualizar um Leilão

Ao se encontrar perante um Leilão, o Utilizador, autenticado, deverá poder escolher visualizar as informações sobre este mesmo. Então, perante a escolha de visualização, o sistema deverá apresentar o nome do produto do Leilão, o tempo restante deste mesmo e valor do último lance realizada e histórico de bids do leilão, caso este Utilizador seja Padrão.

Caso nos encontremos perante um Utilizador Artista que pediu para visualizar um Leilão, o sistema deverá apresentar o nome do produto do Leilão, tempo restante no mesmo, o valor do último lance realizada, histórico de lances feitos neste Leilão e, se for o seu próprio Leilão, deverá ter a opção de Cancelar o Leilão.

Este use case satisfaz o **RF06**.

Visualizar o Catálogo de Leilões

Um Utilizador deverá, se autenticado, ser apresentado com a hipótese de visualizar todos os Leilões em curso.

Então, caso a data final não tenha sido expirada, o sistema deverá apresentar ao Utilizador, consoante o leilão selecionado, o nome do produto, o tempo restante no leilão e o valor do lance mais recente.

Este use case satisfaz o **RF07**.

Use cases do Utilizador Artista

Ambos os Use cases seguintes requerem que um utilizador do tipo Artista esteja previamente devidamente autenticado.

Criar Leilão

Quando um artista pretende leiloar uma peça este cria um leilão. Para o efeito, o artista terá de fornecer os seguintes dados:

- Nome do produto a leiloar
- Tipo de Leilão
- Preço inicial
- Preço reservado
- Imagem
- Dimensões
- Data de fim
- Descrição (opcional)

O leilão fica registado no sistema juntamente com a data de começo e um identificador único.

O preço inicial ficará guardado como preço referência no sistema visto que será um valor que oscilará durante um fluxo natural de um leilão à medida que licitações vão sendo efetuadas.

Este use case satisfaz o **RF03**.

Cancelar Leilão

O sistema deverá apresentar a opção de cancelamento de Leilão, no caso de este ter sido inicializado previamente.

Após o sistema deverá informar todos os Utilizadores deste Leilão recentemente cancelado, retirando-o das "Watchlist's" a que foi adicionado.

Este use case satisfaz o **RF04**.

Use Cases do Utilizador Padrão (Cliente)

Fazer um lance num Leilão

Um Utilizador Padrão deverá, se autenticado, ser apresentado com a hipótese de fazer um lance num leilão, caso a data de expiração deste não tenha sido ultrapassada. Depois, o sistema apenas poderá aceitar lances feitos com um valor de 110% do último lance feito, sendo qualquer outro lance descartado, se se tratar de um leilão inglês, caso seja um leilão *sealed-bid* não existe tal limitação no valor do lance.

No caso de um leilão inglês, quando um lance for efetuado com sucesso, a data de expiração deve ser atualizada caso esta seja significativamente pequena, depois o agora penúltimo Utilizador Padrão a fazer um lance deverá ser notificado que o seu lance foi ultrapassado.

Este use case satisfaz o **RF08**.

Adicionar um Leilão à "WatchList"

Um Utilizador Padrão deverá, se autenticado, ser apresentado com a hipótese de adicionar um Leilão a sua "Watchlist". Caso escolha esta opção, este leilão deverá ser, pelo sistema, adicionado a "Watchlist".

Este use case satisfaz o **RF10**.

Configurar "Watchlist"

Um Utilizador Padrão deverá, se autenticado, ser apresentado com a hipótese de visualizar os leilões da sua "Watchlist", ou remover estes mesmos.

Caso escolher visualizar, deve ser redirecionado pelo sistema para a página do Leilão. Caso escolha remover, este leilão deve ser removido da sua "Watchlist".

Este use case satisfaz o **RF11**.

4 Conceção do Sistema de Dados

Com o decorrer do tempo, o nosso sistema irá armazenar informação de diversos leilões, utilizadores e de milhares de lances e, por esse motivo, é de extrema importância implementar uma base de dados que consiga armazenar e organizar toda esta informação de forma eficiente. A base de dados é, também, útil para gerar relatórios com informações oportunas. Por exemplo, podem ser gerados relatórios sobre os artistas que mais venderam, essas informações podem ser usadas para ordenar o nosso catálogo, de modo que os leilões de artistas com mais vendas sejam os primeiros a aparecer no nosso catálogo.

Por último, as bases de dados contam com implementações muito eficientes de armazenamento de dados, o que permite uma consulta rápida dos mesmos permitindo aos utilizadores finais ter uma experiência de utilização mais agradável.

4.1 Apresentação geral da estrutura do sistema de dados

Depois de definirmos os modelos que apoiaram a implementação do nosso sistema, foi necessário planear uma base de dados que sustentará o seu funcionamento. A seguir a

analisar os requisitos recolhidos e com a ajuda dos modelos demonstrados na especificação do sistema, conseguimos criar o modelo lógico aqui apresentado:

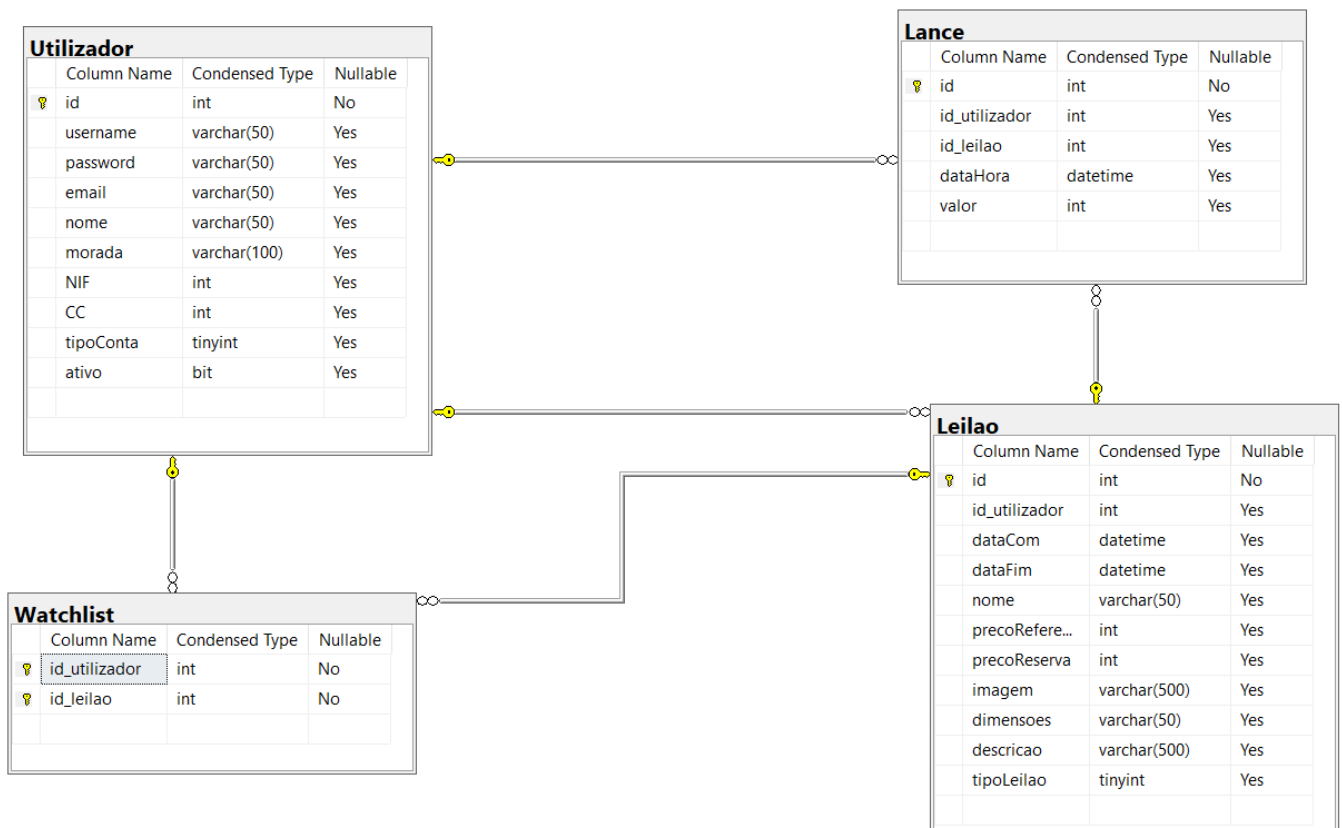


Figura 5 - Modelo Lógico

4.2 Descrição detalhada dos vários elementos de dados e seus relacionamentos

Neste ponto iremos aprofundar o estudo do nosso modelo lógico, apresentando tabelas que nos permitem entender cada entidade de uma melhor maneira, através de colunas com os seus respetivos atributos, tipos de dados descrição e exemplos.

Esta descrição auxiliada dos exemplos irá permitir um melhor entendimento do que cada atributo de cada entidade nos pede

4.2.1 Utilizador

Entidade	Atributos	Tipos de Dados	Descrição	Exemplo
Utilizador	id	int	ID único do Utilizador	7
	email	varchar(50)	Email do Utilizador	pedrop@outlook.pt
	password	varchar(50)	Password do Utilizador	pedro123
	username	varchar(50)	Username do Utilizador	pedrop
	nome	varchar(50)	Nome do Utilizador	Pedro Pereira
	morada	varchar(100)	Morada do Utilizador	Rua do Pedro, nº12, 3ºESQ
	NIF	varchar(9)	NIF do Utilizador	123456789
	CC	varchar(9)	CC do Utilizador	987654321
	tipoConta	tinyint	Tipo de Conta do Utilizador	Artista
	ativo	bit	Indica se um Utilizador está ativo e/ou válido	1

4.2.2 Lance

Entidade	Atributos	Tipos de Dados	Descrição	Exemplo
Lance	id	int	ID único do Lance	7
	id_utilizador	int	ID do Licitador	10
	id_leilao	int	ID do Leilão Licitado	5
	dataHora	DateTime	Data e Hora do Lance	6/28/2023 16:58
	valor	int	Valor do Lance	200

4.2.3 Leilão

Entidade	Atributos	Tipos de Dados	Descrição	Exemplo
Leilão	id	int	ID único dum leilão	8734
	id_utilizador	int	ID do artista que lançou o Leilão	5034
	dataCom	Datetime	Data e Hora do começo do Leilão	6/28/2023 16:58
	dataFim	Datetime	Data e Hora de fim definido pelo artista	7/15/2023 16:58
	nome	varchar(50)	Nome da peça do Leilão	Mona Lisa
	precoReferencia	int	Preço da maior licitação ou valor inicial	130
	precoReserva	int	Preço mínimo para a consumação do Leilão	150
	imagem	varchar(500)	Path da imagem descritiva de uma peça	./Pictures/art/mona_lisa.jpg
	dimensões	varchar(50)	Dimensões de uma peça	100x60
	descrição	varchar(500)	Descrição da peça do Leilão	Pintura a óleo sobre madeira etc...
	tipoLeilao	tinyint	Leilão Inglês ou Sealed Bid	1

4.2.4 WatchList

Entidade	Atributos	Tipos de Dados	Descrição	Exemplo
WatchList	id_utilizador	int	ID do Utilizador à qual a Watchlist pertence	1934
	id_leilão	int	ID do leilão que está na watchlist	1456

5 Esboço dos Interfaces do Sistema

5.1 Estrutura geral das interfaces do sistema

Elaborou-se o seguinte esquema para expressar de forma clara o modo de transição de umas interfaces para as outras.

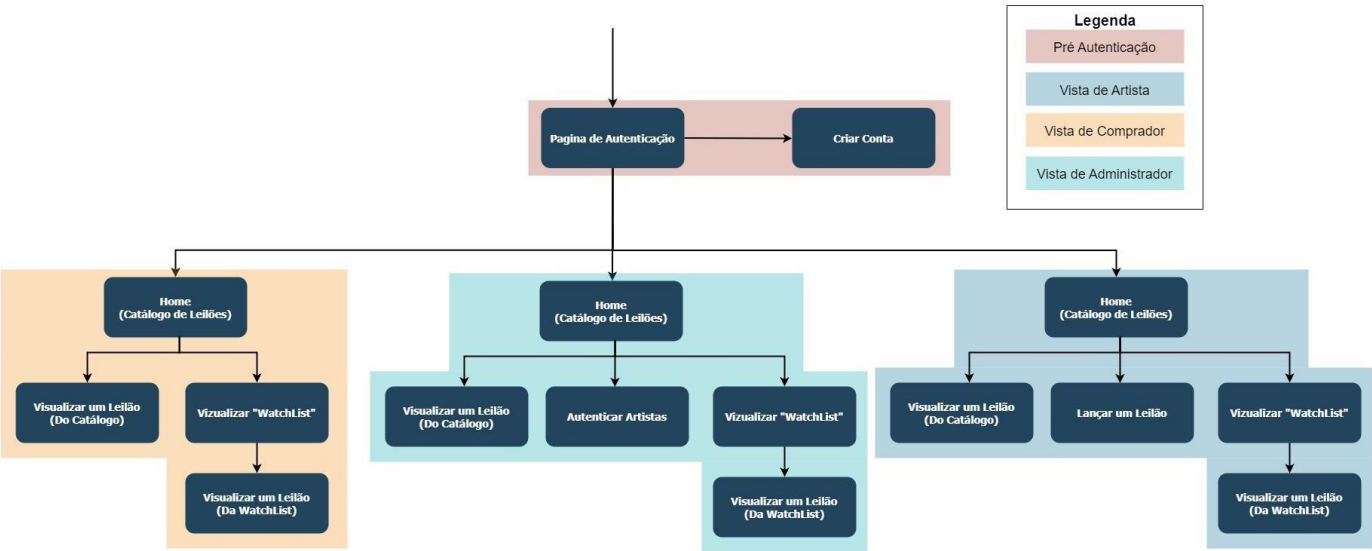


Figura 6 - Esquema das interfaces do Sistema

5.2 Caracterização das interfaces

Seguem-se alguns mockups das interfaces que consideramos que representam as funcionalidades principais da aplicação.

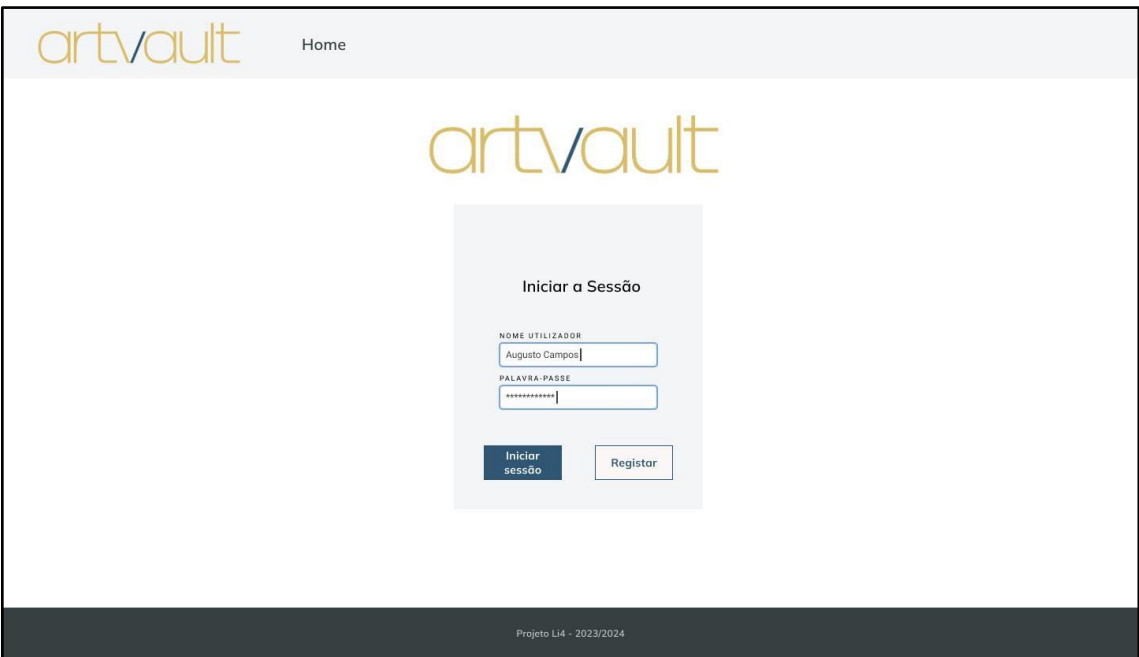


Figura 7 - Mockup Login

Este mockup (Figura 6) representa a página de início de sessão e permitirá satisfazer o requisito RF02 referente à Autenticação na Aplicação. Também é visível um botão para proceder ao registo de um novo usuário (requisito RF01) caso este não se tenha registado anteriormente.

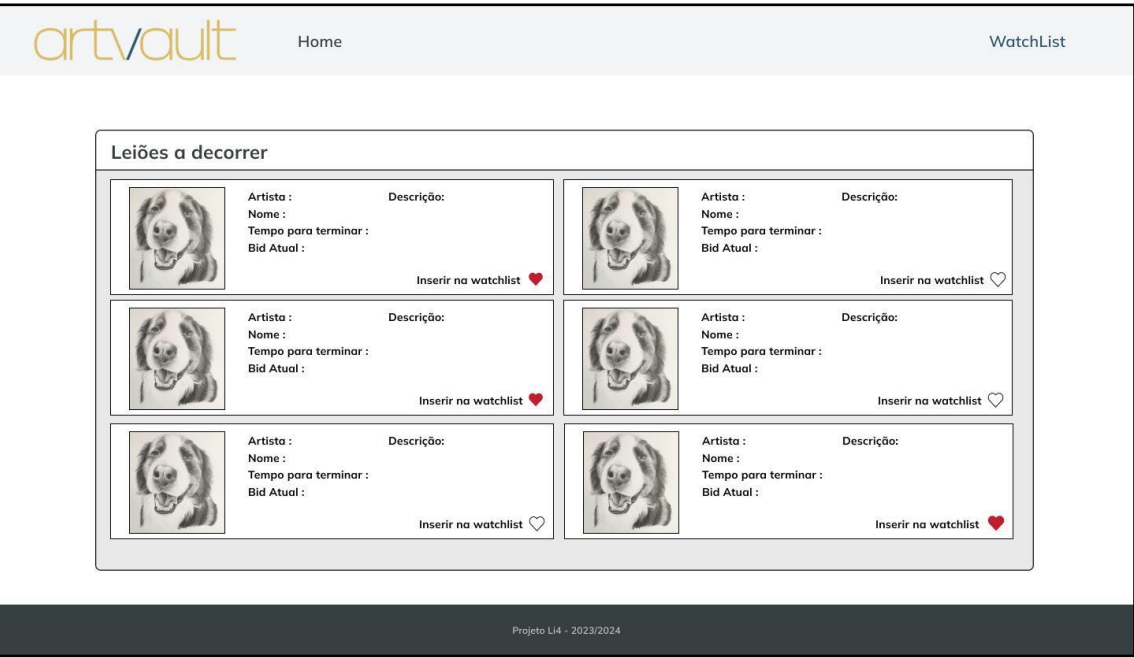


Figura 8 - Mockup Home Page

Esta página (Figura 7) permite a visualização do catálogo de leilões (requisito RF07) de uma forma simples e moderna mostrando as informações mais relevantes de cada leilão e é também neste ambiente que um utilizador pode adicionar um determinado leilão à sua WatchList pessoal (requisito RF10).

A partir desta página, o utilizador pode selecionar um leilão para saber mais informações do mesmo (requisito RF06) e, caso tenha interesse, fazer uma licitação (requisito RF08).

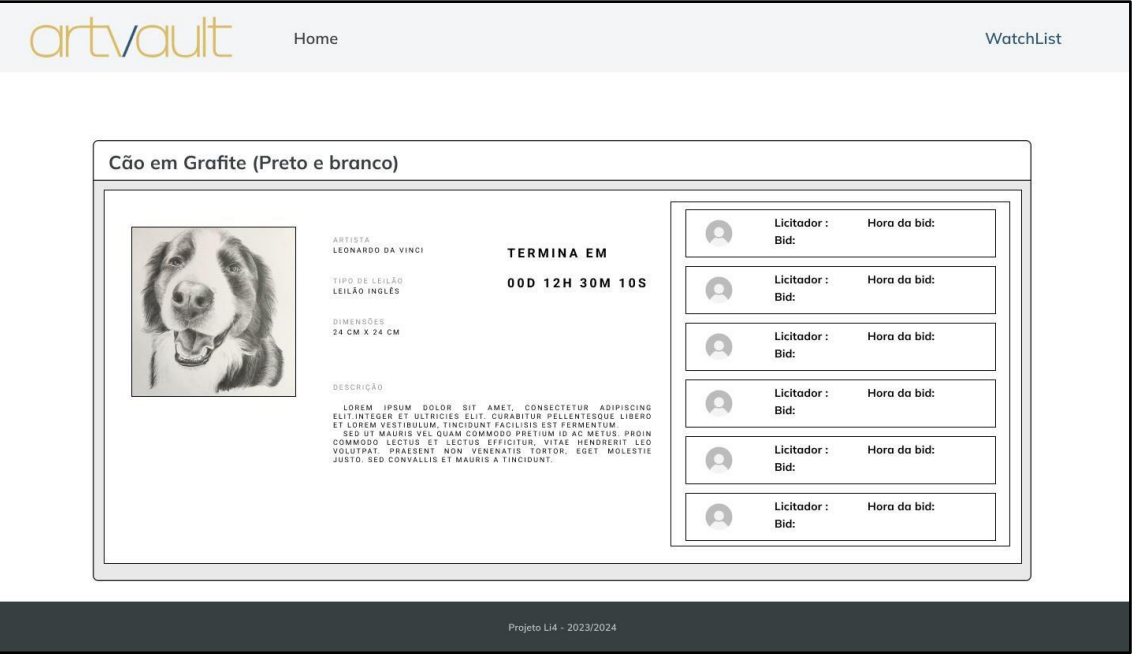


Figura 9 - Mockup Visualizar um Leilão

Aqui temos a página da visão geral de um leilão (Figura 8) (requisito RF06) na qual, para além dos elementos presentes no mockup, também haverá campos que permitam ao usuário fazer ações sobre esse leilão, nomeadamente fazer um lance num leilão (requisito RF07) ou cancelar um leilão (requisito RF04) sendo o utilizador padrão ou artista responsável pelo leilão, respetivamente.

artvault Home WatchList

Lançar um Leilão

NOME

DIMENSÕES

PREÇO RESERVA

PREÇO INICIAL

TIPO DE LEILÃO

Selecione o Tipo de Leilão

Leilão de Lance Fechado
Leilão Inglês

DATA DE FIM DO LEILÃO

06/24/2023 - 06/24/2023

Hora de Fim

LABEL

Descrição do Artigo

Projeto Li4 - 2023/2024

Figura 10 - Mockup Criar um Leilão

Este mockup (Figura 9) representa o espaço no qual o artista pode criar um leilão (requisito RF03).

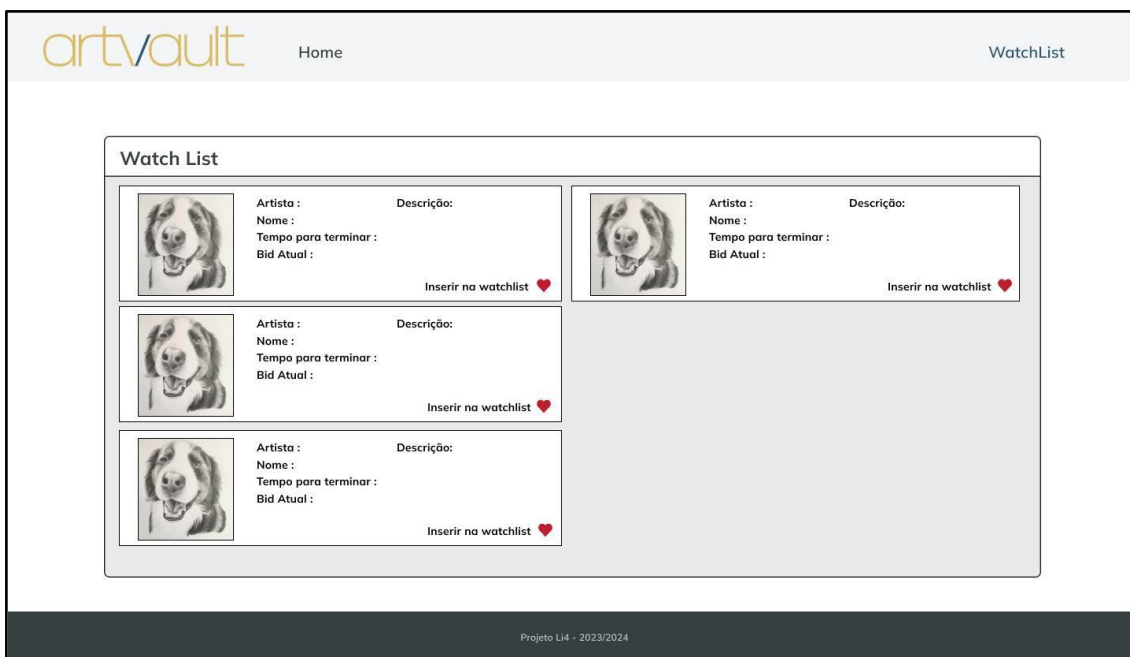


Figura 11 - Mockup WatchList

Nesta página (Figura 10) o utilizador padrão pode visualizar e gerir a sua WatchList (requisito RF11), bem como seleccionar um determinado leilão presente neste painel para uma visão mais detalhada desse mesmo leilão (requisito RF06).

6. Implementação e Desenvolvimento de Software

6.1 Descrição do Processo de Implementação

No desenvolvimento desta aplicação, a base de dados e a API `IDataBaseFacade` foram implementadas primeiro. Isso garantiu que os dados pudessem ser armazenados corretamente e que a comunicação entre a base de dados e o resto da aplicação funcionasse corretamente.

Em seguida, a lógica de negócio e a API `IArtvaultFacade` foram desenvolvidas. Isso incluiu a implementação dos métodos da lógica de negócio, validações e processamento de dados. Toda a lógica de negócio foi implementada em paralelo com o desenvolvimento da interface gráfica. Tomamos essa decisão, pois consideramos ser a melhor forma de ir explorando as ferramentas que eram novas para a nossa equipa de trabalho, testando pequenas funcionalidades da camada de negócio, tendo de imediato uma amostra de como tudo seria apresentado.

Ferramentas Utilizadas

Sistema de Gestão de Base de Dados:

- Microsoft SQL Server – Foi a ferramenta sugerida pela equipa docente e revelou-se prática e com bom desempenho.

Ambiente de Desenvolvimento:

- Microsoft Visual Studio – Um IDE que nos foi muito útil no desenvolvimento de código da aplicação graças aos seus recursos de debug. Esta ferramenta, em conjunto com a plataforma .NET e a framework ASP.NET e Blazor permitiu-nos também desenvolver a interface gráfica. Escolhemos Blazor face à sua capacidade de permitir combinar C# e HTML.

Ferramentas de Apoio à Gestão do Projeto:

- Git – Um recurso essencial para qualquer projeto de software em equipa.
- Visual Paradigm – Ferramenta que permite criar diagramas UML, sendo muito útil nas fases de especificação

6.2 Estruturas da Camada de Aplicação

No desenvolvimento da aplicação, utilizamos um modelo de três camadas: Camada de interface, Lógica de negócio e Camada de dados.

Para garantir uma comunicação simplificada entre a camada de lógica de negócio e a camada de acesso aos dados, foi utilizado o padrão DAO (Data Access Object) que providencia uma interface abstrata de acesso à base de dados. Essa interface foi encapsulada em `IDatabaseFacade`.

Na camada de lógica de negócio, foram implementadas todas as classes que constituem a funcionalidade principal do projeto. Esta funcionalidade apesar de estar dividida em várias classes é aglomerada numa classe maior `IArtVaultFacade`.

A Camada de Interface utiliza a `IArtVaultFacade` como caminho de comunicação com a funcionalidade do negócio.

A estrutura final da aplicação pode ser resumida no seguinte esquema:



7. Conclusão

Para garantir um planeamento e implementação rigorosos do projeto, o seu desenvolvimento foi dividido em três fases distintas. Cada fase focou-se em um aspecto central do projeto: definição, especificação e implementação.

Na fase de definição, foram definidos os objetivos do projeto, a sua viabilidade e os recursos necessários. Nesta fase, também foi contextualizado o problema que o projeto pretendia resolver.

Na fase de especificação, foram definidos os requisitos estruturais e comportamentais do sistema. Nesta fase, também foram elaborados diagramas UML para auxiliar a especificação do sistema.

Estas duas primeiras fases foram importantes para garantir que o projeto tivesse um planeamento sólido e bem estruturado. Isso permitiu uma melhor compreensão do que se queria com o software e, conseqüentemente, uma implementação mais eficaz.

Com a estrutura e requisitos do projeto especificados, passamos para a fase de implementação que consistiu no desenvolvimento do software tendo como suporte a especificação elaborada na fase anterior. Dado que a maior parte das ferramentas que nos foram indicadas eram novas para a nossa equipa de trabalho, esta fase não foi fácil. O arranque foi lento sendo que era primeira vez que trabalhávamos com ferramentas como o Visual Studio e até com C# que, apesar de ser uma linguagem orientada a objetos, não deixou de ser um desafio.

Apesar das dificuldades, o facto de termos elaborado uma forte especificação e planeamento da aplicação permitiu que não restassem dúvidas acerca do que precisaria de ser implementado, na medida que aliviou um pouco a dificuldade nesse aspeto em comparação com outros projetos de desenvolvimento de software onde a fase de especificação não existiu ou não havia sido feita por nós. Isso permitiu que nos pudessemos concentrar em aprender a utilizar as já referenciadas ferramentas.

No futuro, de modo a reter utilizadores e/ou atrair novos clientes, é natural o desenvolvimento de novas funcionalidades tais como o incorporar métodos de pagamento, novos tipos de leilão e áreas de utilizador mais ricas. De maneira geral, também será importante aumentar a segurança relativamente aos dados dos utilizadores. Para além disso, é necessário, com o crescimento da aplicação tanto a nível de usuários como de funcionalidades, um trabalho de manutenção do sistema de dados para permitir que a expansão da aplicação se possa dar organizadamente ao longo do seu tempo de vida.

Lista de Siglas e Acrónimos

SQL - Structured Query Language

CC - Cartão de Cidadão

NIF - Número de Identificação Fiscal

UML - Unified Modeling Language

API - Application Programming Interface

IDE - Integrated Development Environment

HTML - HyperText Markup Language

DAO - Data Access Object

Referências

Ian Sommerville, Software Engineering, 10ª Ed. (Abril de 2015) de <https://software-engineering-book.com/>

Site de Leilões Internacional (s.d.) de <https://www.catawiki.com/pt/>

Site de Leilões nacional (s.d.) de <https://www.economias.pt/e-leiloes/>

Documentação Sql Server (s.d.) de <https://learn.microsoft.com/pt-br/sql/?view=sql-server-ver16>

Documentação do padrão DAO (s.d.) de <https://www.geeksforgeeks.org/data-access-object-pattern/>