

**UNIVERSIDADE FERNANDO PESSOA
ENGENHARIA INFORMÁTICA**

**Gabriel Fernandes, 40344 (50%)
Frederico Pinto, 39822 (50%)**

**PROJETO ENGENHARIA INFORMÁTICA E LABORATÓRIO
DE PROGRAMAÇÃO
ENTREGA INTERMÉDIA**

2021/2022

Resumo

Neste projeto pretendemos desenvolver um website responsivo, cujo tema é uma livraria online, semelhante às conhecidas Bertrand, Wook, entre outras...

O website pode ser dividido em três partes : Utilizadores, Pagamentos, Artigos.

Relativamente aos Utilizadores, teremos funcionalidades para registar utilizadores novos, Log In's e edição das informações dos mesmos(morada, contacto, método de pagamento associado,etc...). Teremos também um "Vendedor", que qualquer usuário pode ser, passando por um processo de registo, onde o usuário-vendedor pode vender os seus próprios artigos na loja. Utilizadores poderão, também, subscrever os seus autores preferidos, para estarem a par de novos lançamentos.

Quanto aos pagamentos, o utilizador pode escolher um método de pagamento, definir a morada de entrega, e consultar a todo o momento o estado do envio. Vendas serão registradas, verificadas e ,se for o caso, confirmadas.

Por fim, artigos representam todo o tipo de artigos vendidos na loja. Por agora, apenas Ebooks e Livros estão disponíveis, porém não descartamos a possível venda de outro tipo de artigos num futuro breve. Artigos poderão ser pesquisados e consultados no site por qualquer tipo de utilizador, inclusive aqueles que não se encontram registados(caso queiram efetuar a compra, terão que iniciar sessão). Outros tipos de pesquisa poderão ser feitos, gêneros/categorias, autores, editores. Estes artigos podem ser avaliados por clientes da loja, e adicionados às suas listas de desejos para saberem quando os seus artigos favoritos estão em promoção

Índice

Resumo	2
Índice	3
Introdução	4
Requisitos funcionais	4
Requisitos não funcionais	5
Diagrama casos de uso	6
Diagrama de classes	7
Diagramas de sequência	8
Search Item	8
Buy Item	9

Introdução

O objetivo deste trabalho é construir uma livraria online 100% funcional. Com todas as funcionalidades necessárias para o seu funcionamento. Com a conclusão do projeto, o resultado deverá ser todo um sistema capaz de dar resposta a vários pedidos (compras, registos, logins, pesquisas), e que gere este fluxo de informações com o auxílio de uma base de dados, de forma a fornecer um serviço online de uma loja de livros. Pretendemos oferecer tudo que uma livraria online oferece, assim como, algumas funcionalidades a mais.

Requisitos funcionais

ID	Identificação	Objetivo
1	Registar/Iniciar sessão	Cada utilizador deve poder se registar no site, e posteriormente iniciar sessão
2	Pesquisa de livros	Deverá ser possível pesquisar livros pelo seu título, por autores e gêneros
3	Ordenar resultados	Deverá ser possível ordenar resultados de uma pesquisa por preço e data
4	Filtrar resultados	Deverá ser possível filtrar resultados de uma pesquisa por preço, género, idioma, ano
5	CRUD	Deverá ser possível inserir, editar e remover livros do sistema
6	Disponibilidade	Artigos com stock terão que ter um sistema que gere a disponibilidade. Livros sem stock não podem ser comprados e livros reservados não podem contar na disponibilidade
7	Reserva de livros	Um cliente pode reservar um livro
8	Subscrição de autores	Um utilizador pode subscrever um autor, e receber notificações de novos lançamentos do mesmo via email
9	Lista de desejos	O utilizador pode adicionar um artigo à sua lista de

		desejos para receber uma notificação(por email e/ou telemóvel) de uma promoção neste artigo
10	Avaliações	Cada artigo contém avaliações, realizadas por usuários, e podem ter comentários
11	Marketplace	Cada usuário pode se registrar como vendedor e colocar artigos à venda
12	Exibir detalhes	Cada artigo deve poder ser consultado
13	FAQ	Deverá existir uma FAQ para consulta de qualquer tipo de utilizador
14	Vendedores	Para um utilizador virar um vendedor, terá que passar por um processo de registo onde será confirmada a sua identidade e elegibilidade
15	Promoções	Promoções podem ser registradas no sistema por ADM, estas podem ser ativadas e desativadas. Deverá ser possível manter promoções anuais, como por exemplo de Natal, Verão
16	Ebooks	Deve existir um sistema de leitura online

Requisitos não funcionais

ID	Identificação	Objetivo
1	Línguas	O site deverá ter suporte de várias linguagens. O utilizador pode definir a linguagem e o site carrega com o código selecionado
2		

Diagrama casos de uso

No diagrama de casos de uso temos 4 atores primários e 1 ator secundário.

Visitor, e um visitante, qualquer pessoa que aceder ao site pode realizar aquelas operações sem precisar iniciar sessão.

Portanto, Logged User é um utilizador registado que iniciou a sessão. Para um utilizador poder avaliar um artigo, ele tem que possuir o artigo a avaliar. Um utilizador pode ainda registrar-se como vendedor, passando por um processo de verificação, para avaliar se o utilizador é elegível.

Por conseguinte, um Vendedor é um tipo de utilizador que já passou pelo processo de verificação. O vendedor herda todas as funcionalidades do Logged User, e possui ainda a opção de vender os seus próprios artigos.

Admin é um ator que representa os administradores do site, estes podem executar operações relativamente à gestão. Gerir artigos, editar informações, eliminar(ou desativar) artigos, bem como inserir e ativar/desativar promoções.

Por ora, o ator secundário Attendant, apenas verifica as informações de um utilizador que pretende se tornar um vendedor, aprovando ou recusando, dependendo de regras a serem criadas para este processo.

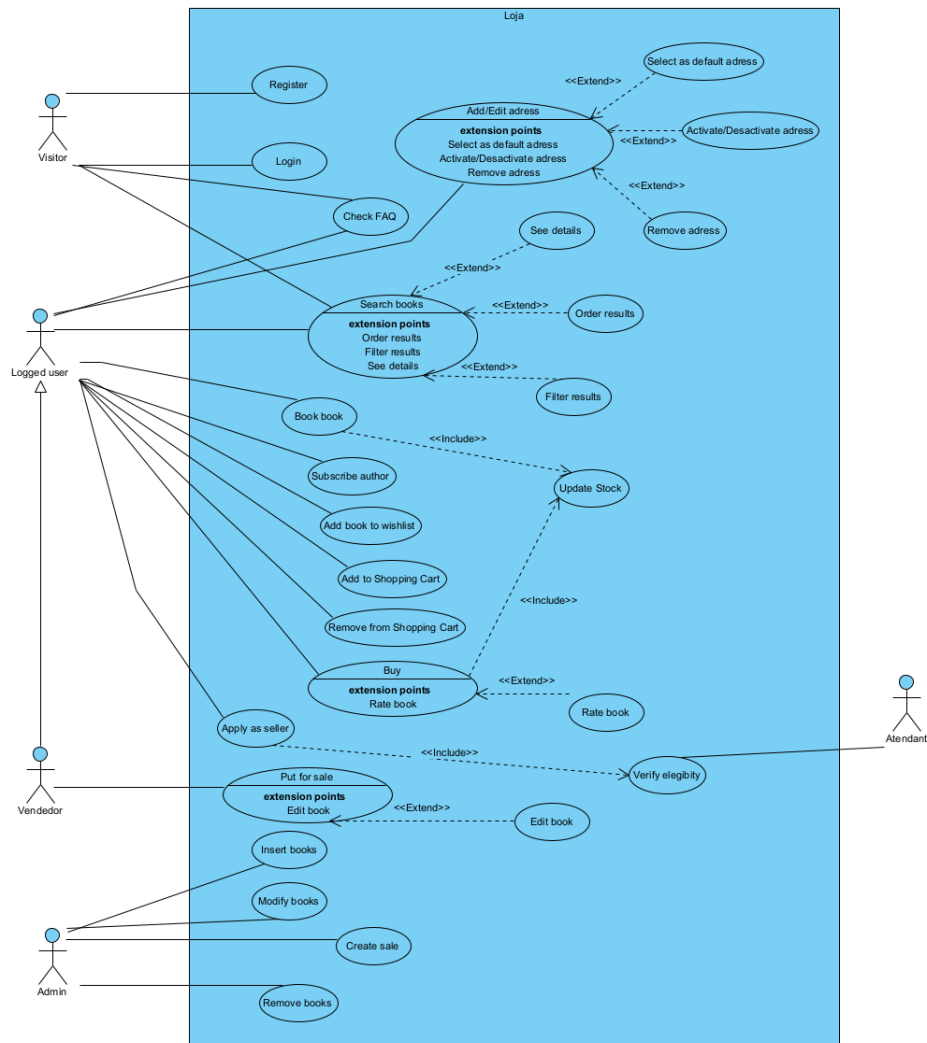
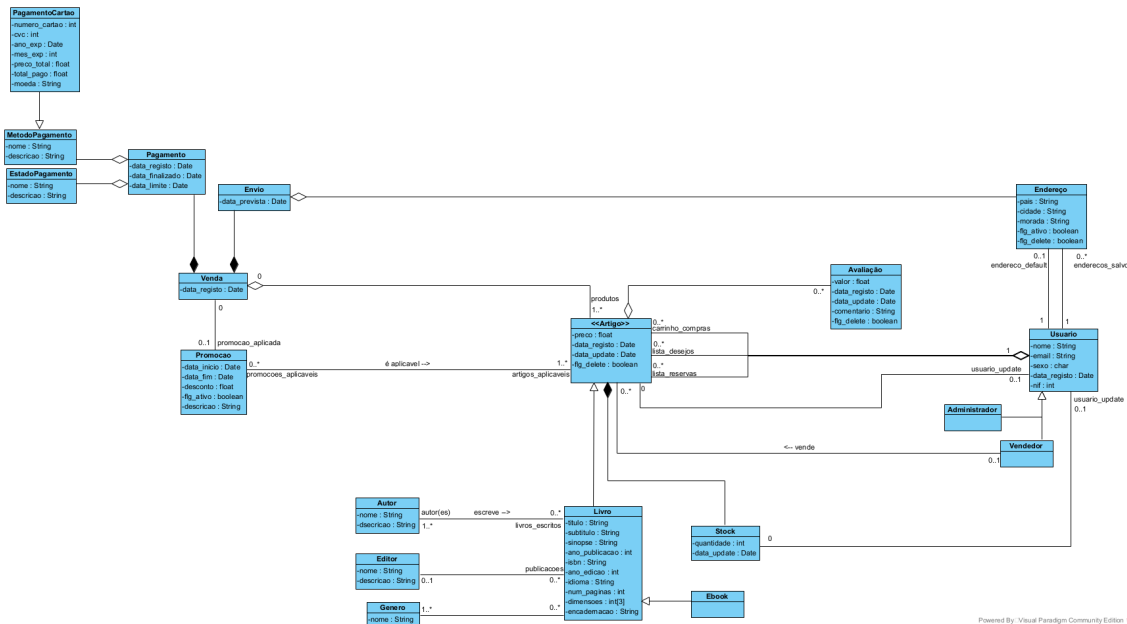


Diagrama de classes



A classe **Utilizador**, representa todos os utilizadores registados no site. Cada utilizador possui um carrinho de compras, lista de desejos e reservas, que consistem em arrays da classe **Artigo**. O utilizador possui também, um array da classe **Endereco**, onde se encontram todos os endereços do utilizador. O `endereco_default` é o endereço predefinido.

Dois tipos de utilizadores existem(para além do utilizador normal), **Vendedor** e **Administrador**. O **Vendedor** pode vender artigos. Neste momento, existem dois tipos de artigos, **Livros** e **Ebooks**(que são um tipo de **Livro**).

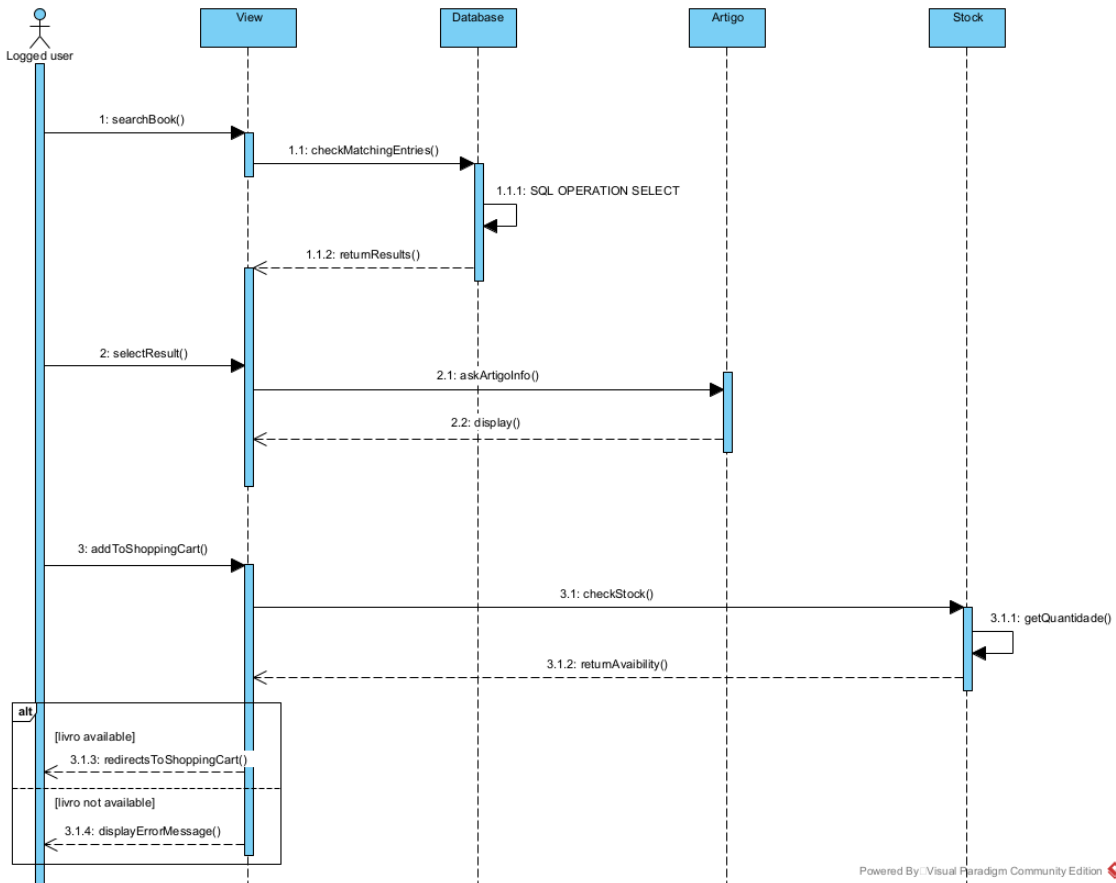
Um **Livro** é escrito por um ou mais **Autor**, pertence a 1 ou mais **Gênero** e pode (ou não) ser publicado por um **Editor**. Cada artigo possui um **Stock**.

Promoções podem ser aplicadas, uma por venda. **Vendas** são criadas a partir do carrinho de compras, nelas serão guardadas os produtos e a promoção aplicada, ao invés de guardar o preço calculado no carrinho, para não haver possibilidade de comprar artigos sem stock ou pagar um artigo com um preço que já não está disponível no site(por promoções).

Envio será a morada de entrega, **EstadoPagamento**(o estado do pagamento), **MétodoPagamento**(as diferentes formas de pagamento com taxas associadas).

Diagramas de sequência

Search Item



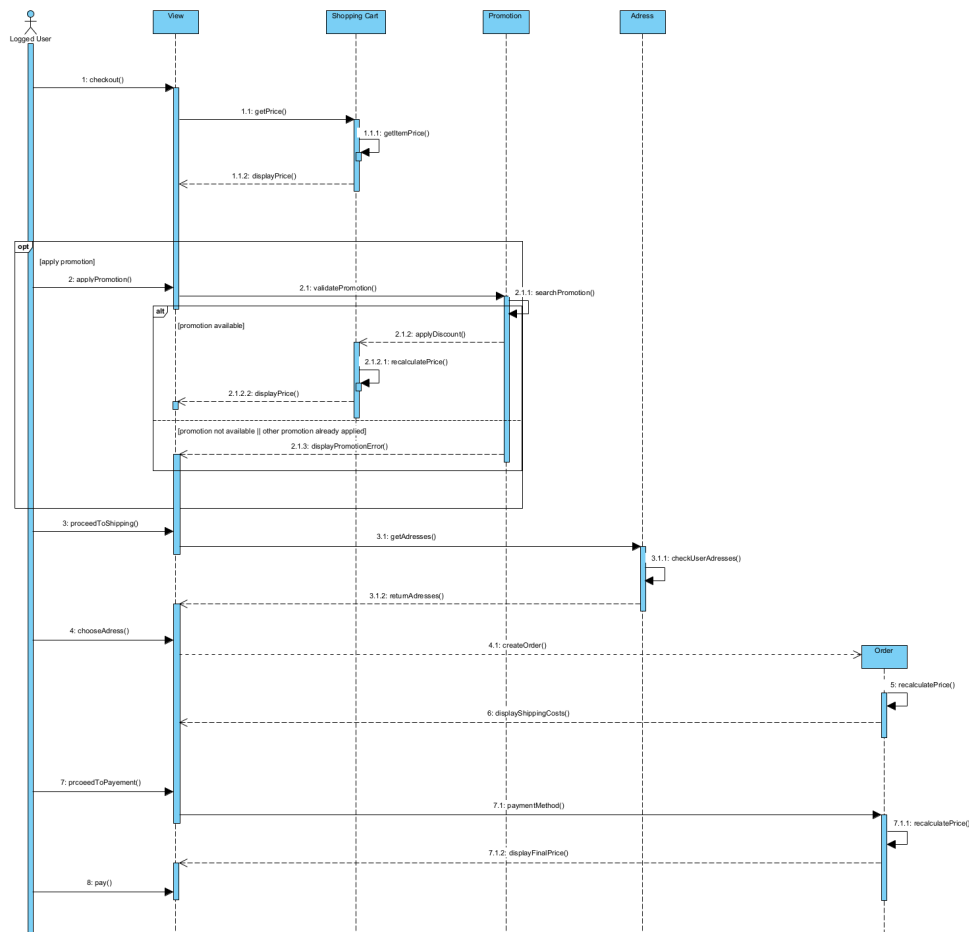
Neste diagrama está representado todas as interações no sistema desde a pesquisa de um artigo até adicioná-lo ao carrinho de compras.

Inicialmente, um utilizador logado procura um livro(por título, autor, gênero, etc...). A view(o que o utilizador vê, ou seja, a página web), pesquisa por qualquer combinação igual ao input do utilizador na database, que executa um SELECT de todos os artigos que contenham uma substring igual ao input.

Seguidamente, os resultados são retornados, e o utilizador pode escolher o livro. A view pede pelas informações do artigo selecionado, e apresenta ao usuário.

Por fim, o utilizador pode adicionar ao carrinho o artigo selecionado, onde a view confirma com o Stock deste artigo se está disponível, se sim ele é redirecionado para o carrinho de compras, se não é apresentado um erro.

Buy Item



Neste artigo está representado todas as interações do sistema desde o checkout no carrinho de compras, até ao pagamento de uma venda.

Inicialmente, um utilizador procede ao checkout do carrinho de compras. A view irá pedir pelo valor do carrinho, que calcula o preço de todos os produtos, e retorna o valor.

Opcionalmente, o utilizador pode aplicar um cupom. Este cupom será validado nas promoções. Caso a promoção seja válida, será aplicada no carrinho de compras, que recalcula o preço total com o desconto. Caso contrário ou caso o carrinho de compras já tenha outro cupom aplicado, será apresentado um erro.

Em seguida, o utilizador procede para o envio, onde define a morada para a compra ser entregue. A partir daqui, é criada uma Order(Venda), com dados de entrega, pagamento, itens e cupom aplicado. O preço é recalculado com o valor dos portes, e apresentado mais uma vez.

Por fim, o utilizador escolhe o método de pagamento. A Order recalcula mais uma vez os valores, apresenta e, finalmente, o utilizador pode pagar.