

3ª Fase – Fase Final do Projeto

Laboratório de Aplicações Web e Bases de Dados

Autores:

Gustavo Santos al70800

Ricardo Silva al70672

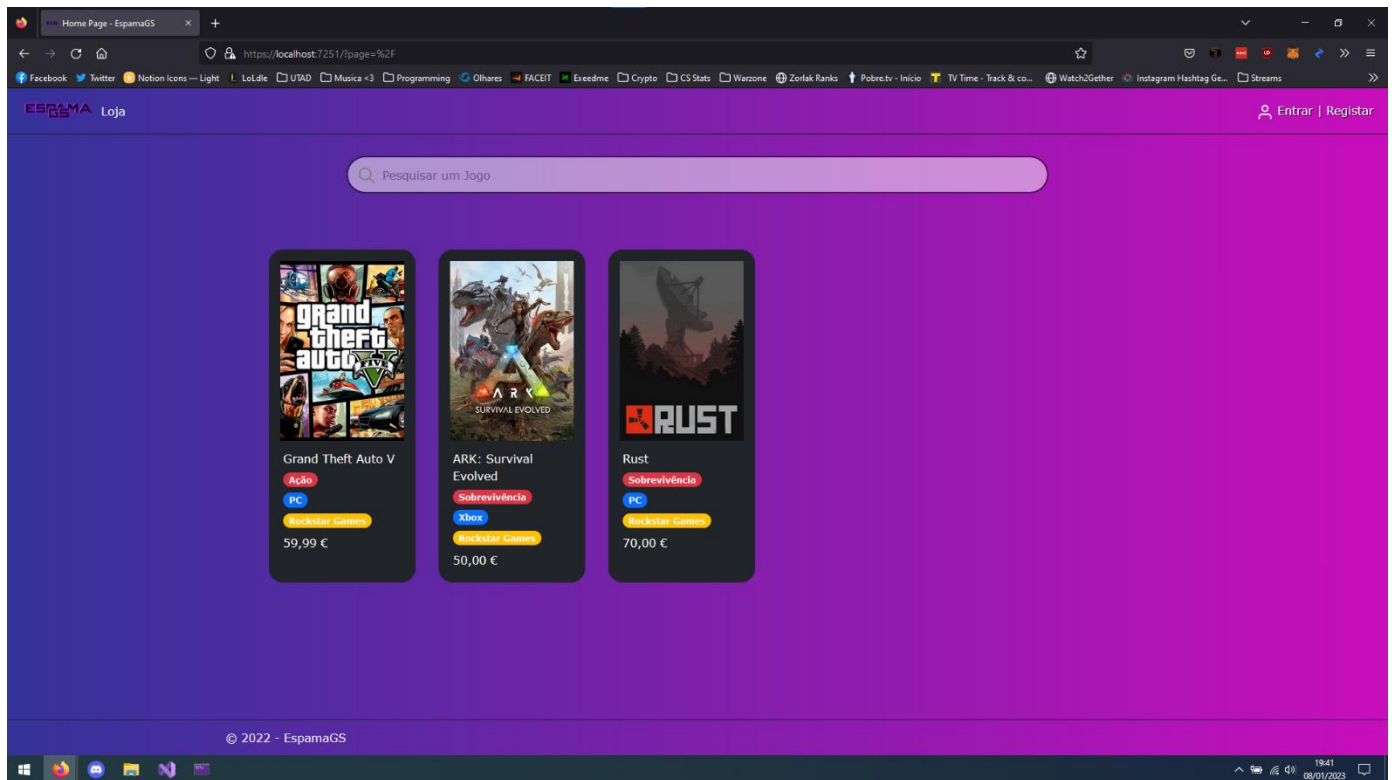
Tiago Pedras al70664

Vasco Teixeira al74107

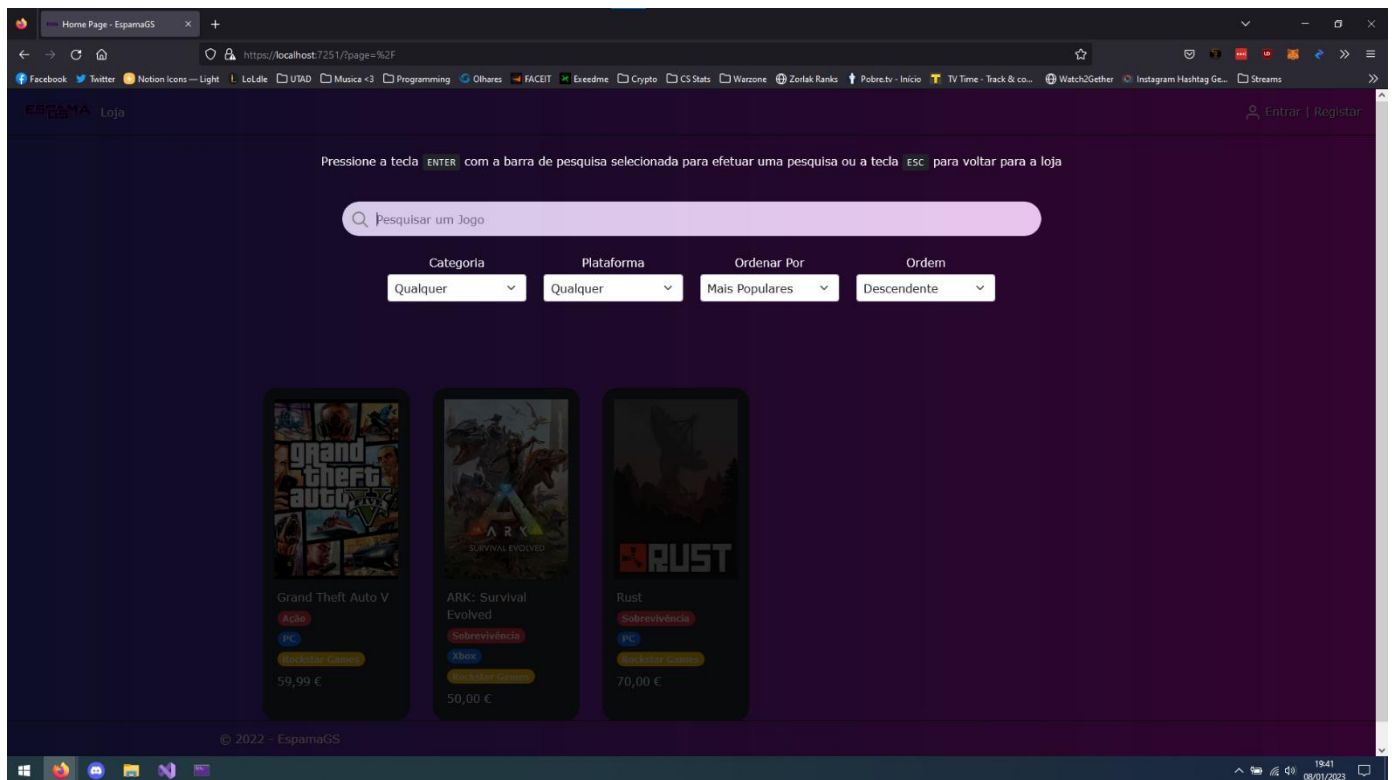
Introdução

Nesta terceira e última entrega, apresentámos o projeto final do trabalho da loja de jogos, neste relatório iremos apresentar e explicar todas as funcionalidades do nosso trabalho, tal como, as views de cada uma.

Apresentação de todas as funcionalidades

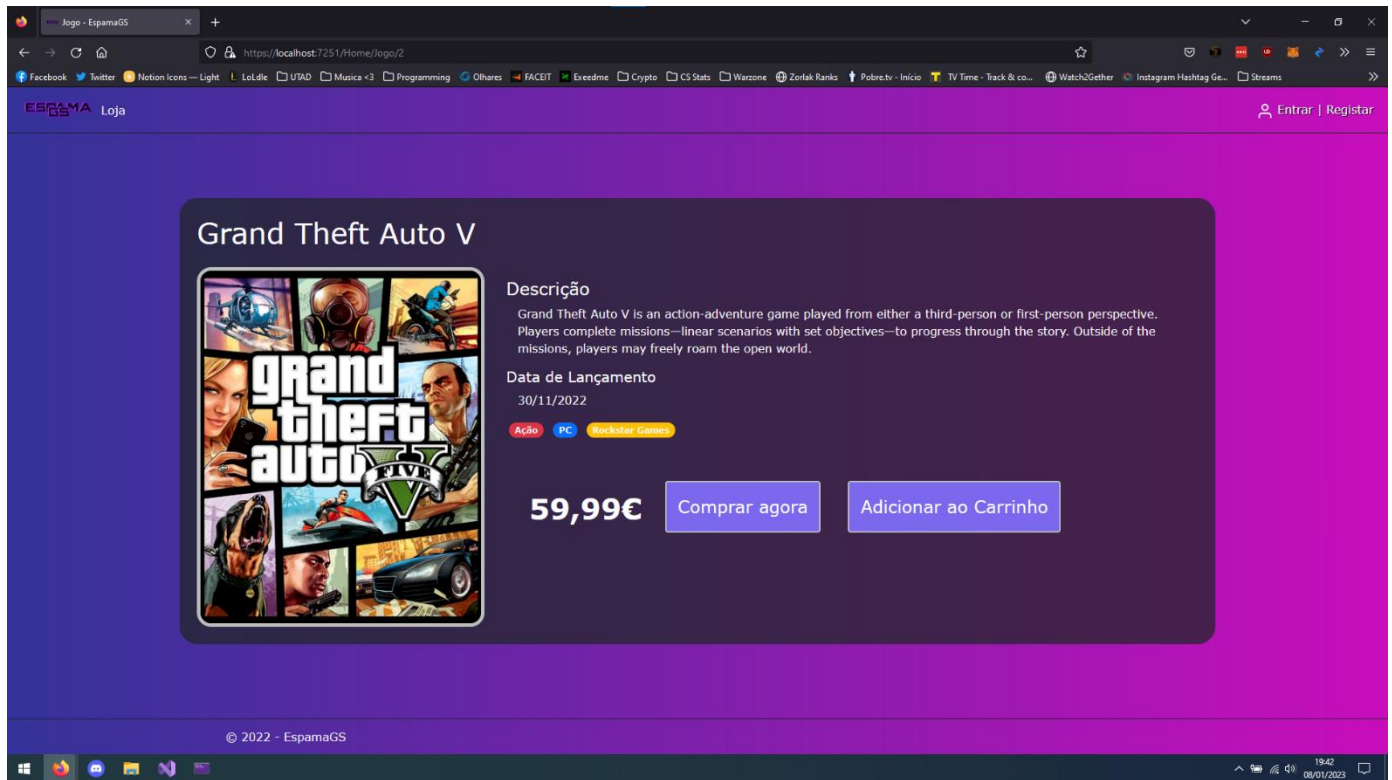


Nesta view podemos ver, a apresentação de todos os jogos que estão listados para venda no nosso site, tal como, o respetivo preço, categoria, plataforma e desenvolvedora. E também podemos ver a barra de pesquisa que será aprofundada mais à frente.

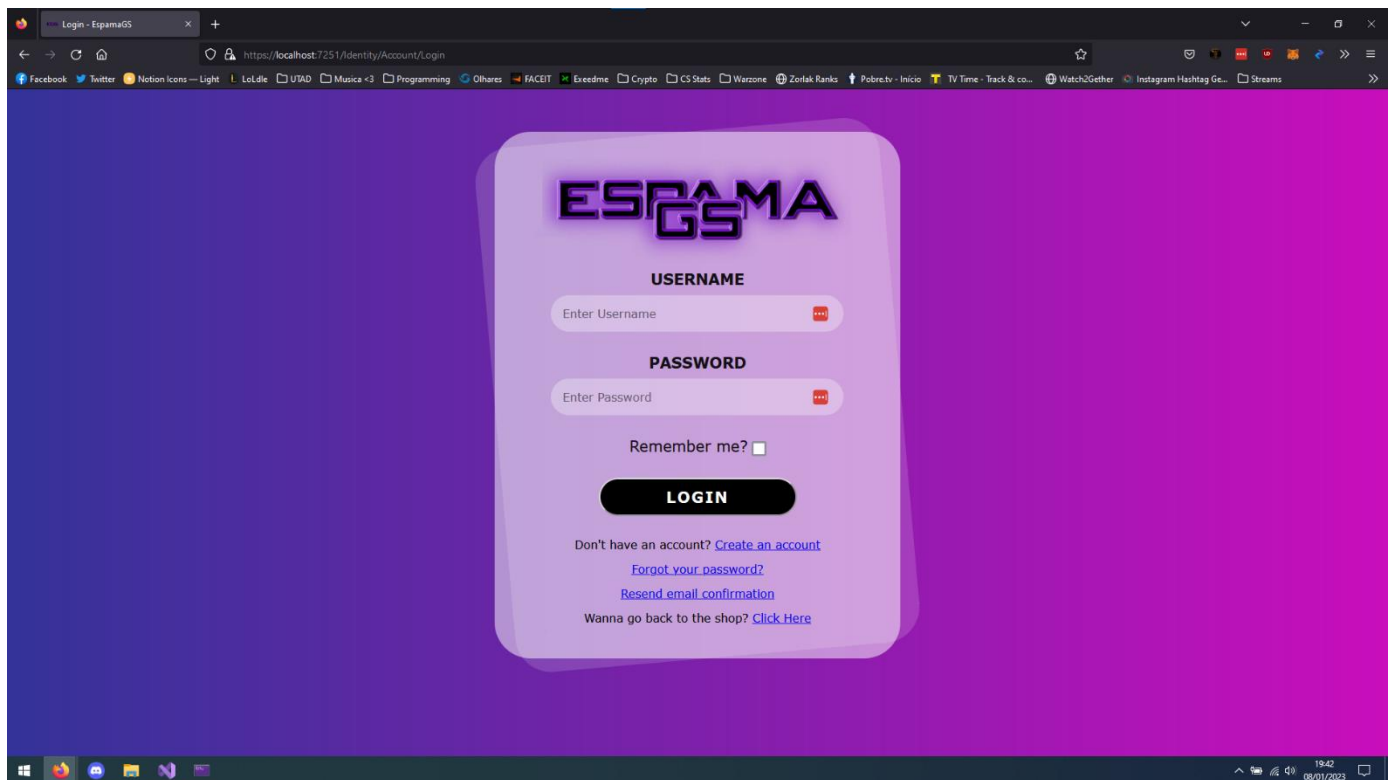


Aqui, tal como foi referido anteriormente, conseguimos ver as funcionalidades da nossa barra de pesquisa, onde conseguimos filtrar certos parâmetros para facilitar a pesquisa de todo o tipo de jogos listados no site.

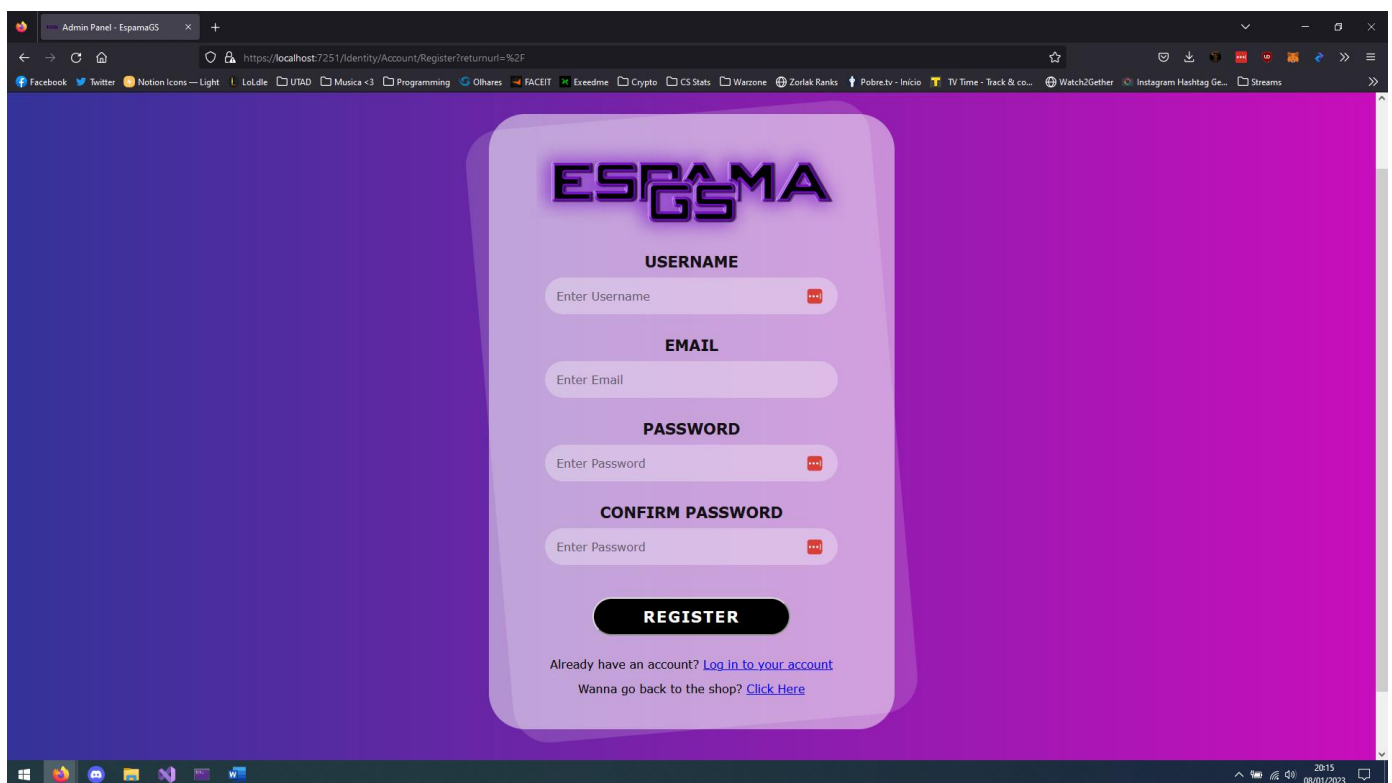
Conseguimos alterar, tal como é visto, a categoria, plataforma, e também ordenar (por ordem decrescente e crescente) os jogos mais populares, pelo seu preço e data de lançamento.



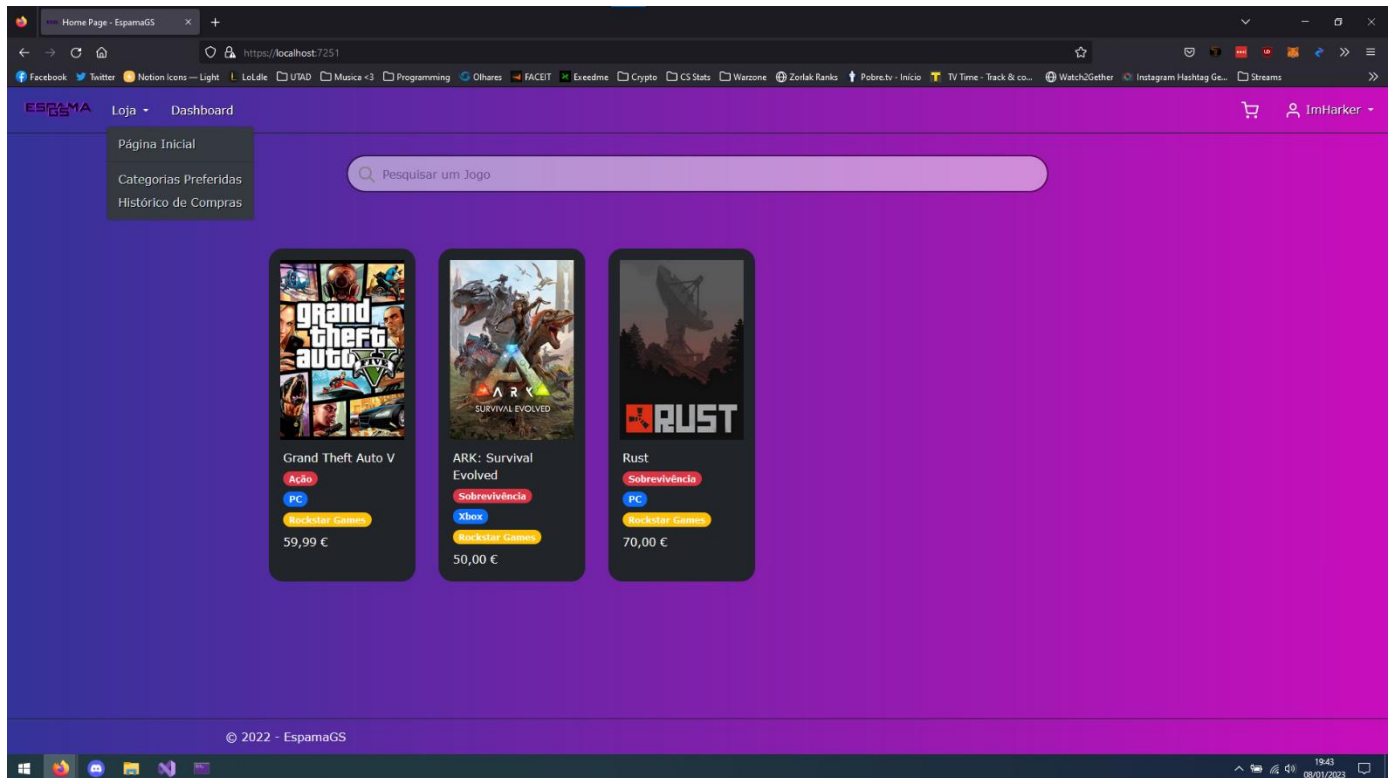
Nesta página, tal como é apresentada em cima, conseguimos ver já os detalhes específicos de cada jogo, tal como, o seu preço, categoria, plataforma e desenvolvedora (que também era possível ver na página da listagem de todos os jogos) , uma breve descrição do jogo, em que fala do que se trata e do seu objetivo, a data de lançamento, e já temos os botões de comprar agora, que nos reencaminha diretamente para a página final da compra, e o botão de adicionar ao carrinho que nos permite continuar a comprar outros jogos, enquanto este já foi selecionado anteriormente.



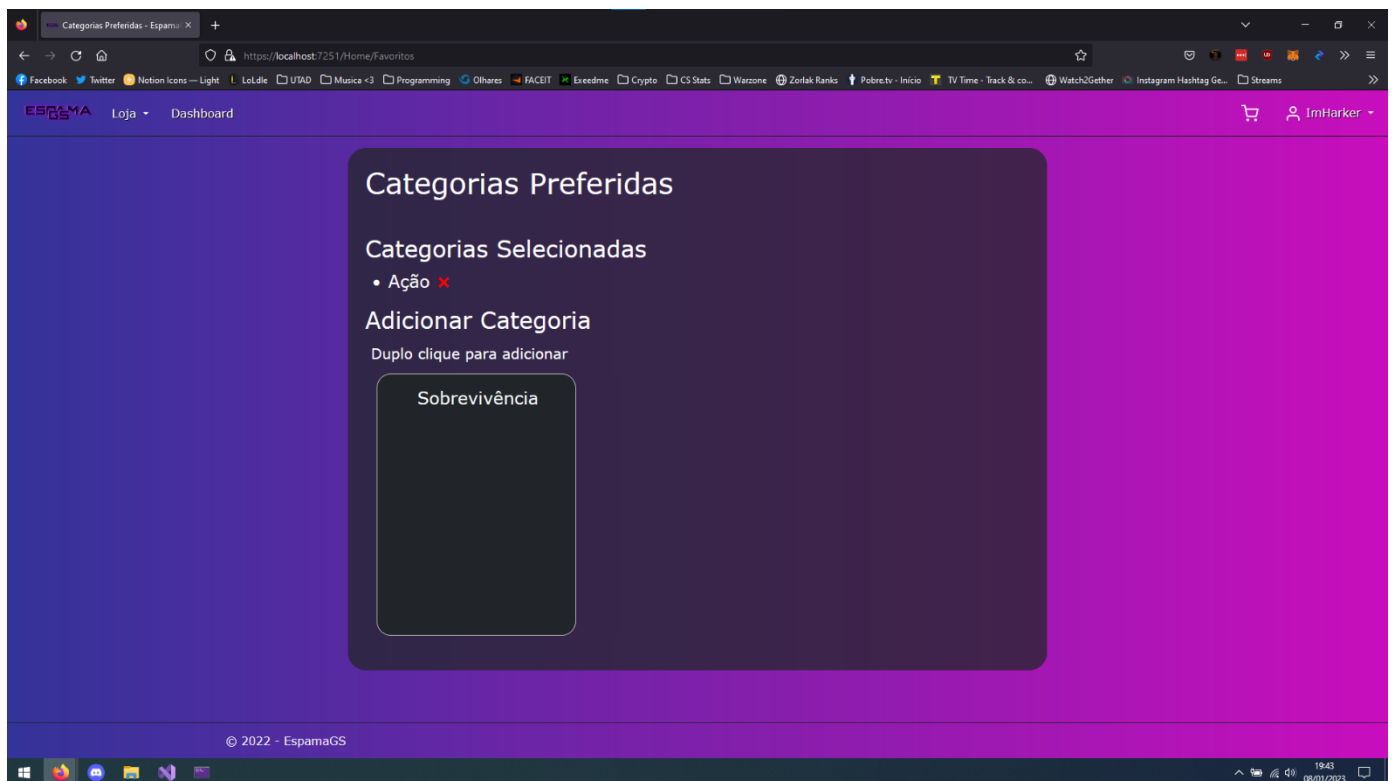
Nesta imagem já nos é apresentado o login, que permite o acesso tanto a utilizadores como a funcionários e administradores.



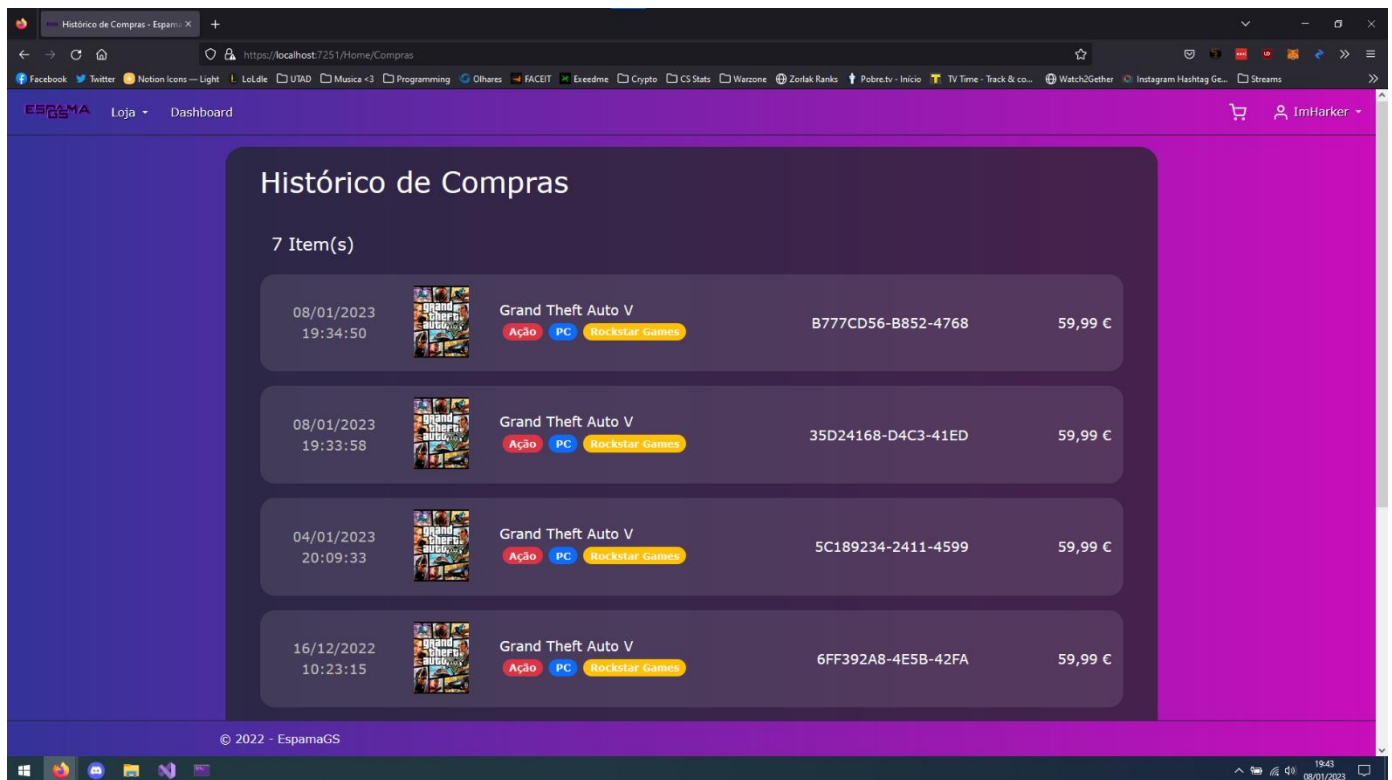
Aqui podemos ver a página em que é permitido aos utilizadores criar conta no site.



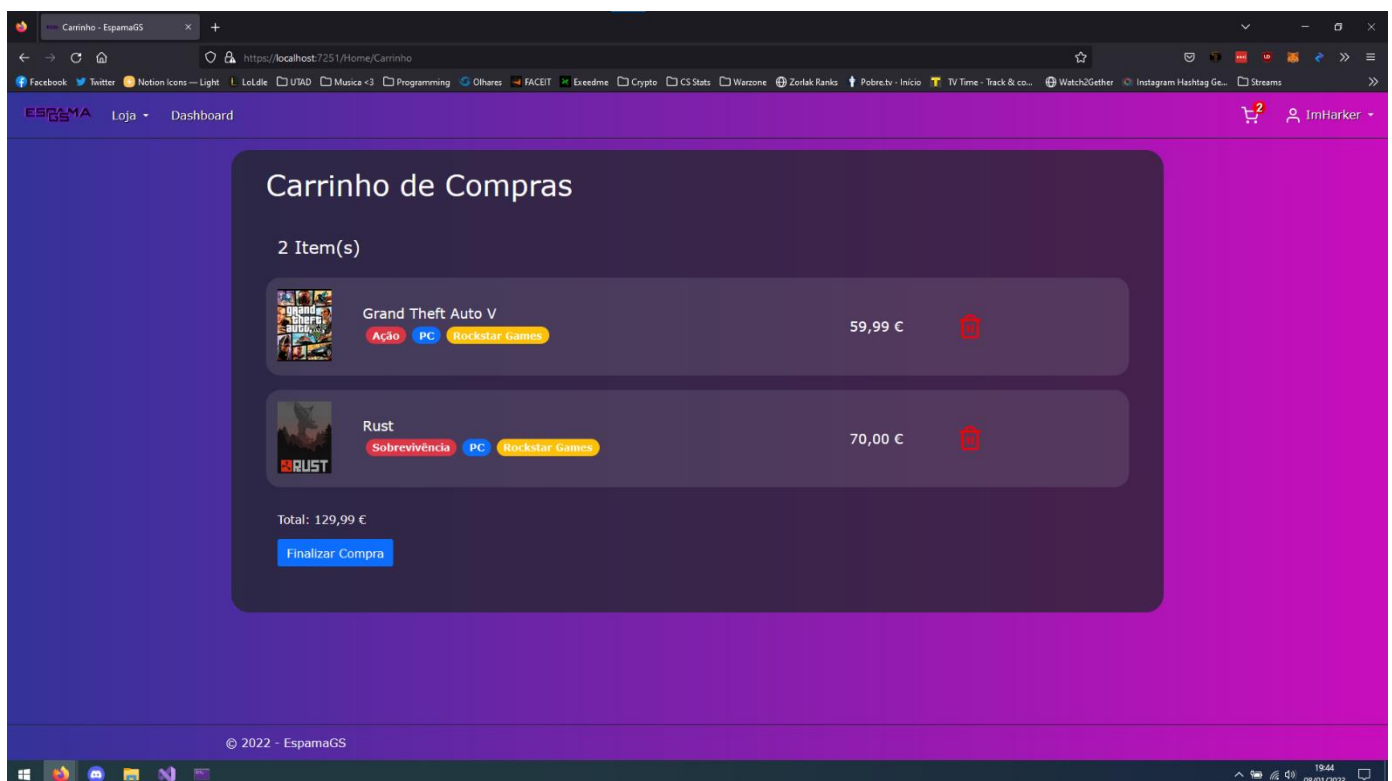
Esta é a página que nos é apresentada se estivermos autenticados, a opção Dashboard apenas é apresentada a funcionários e administradores.



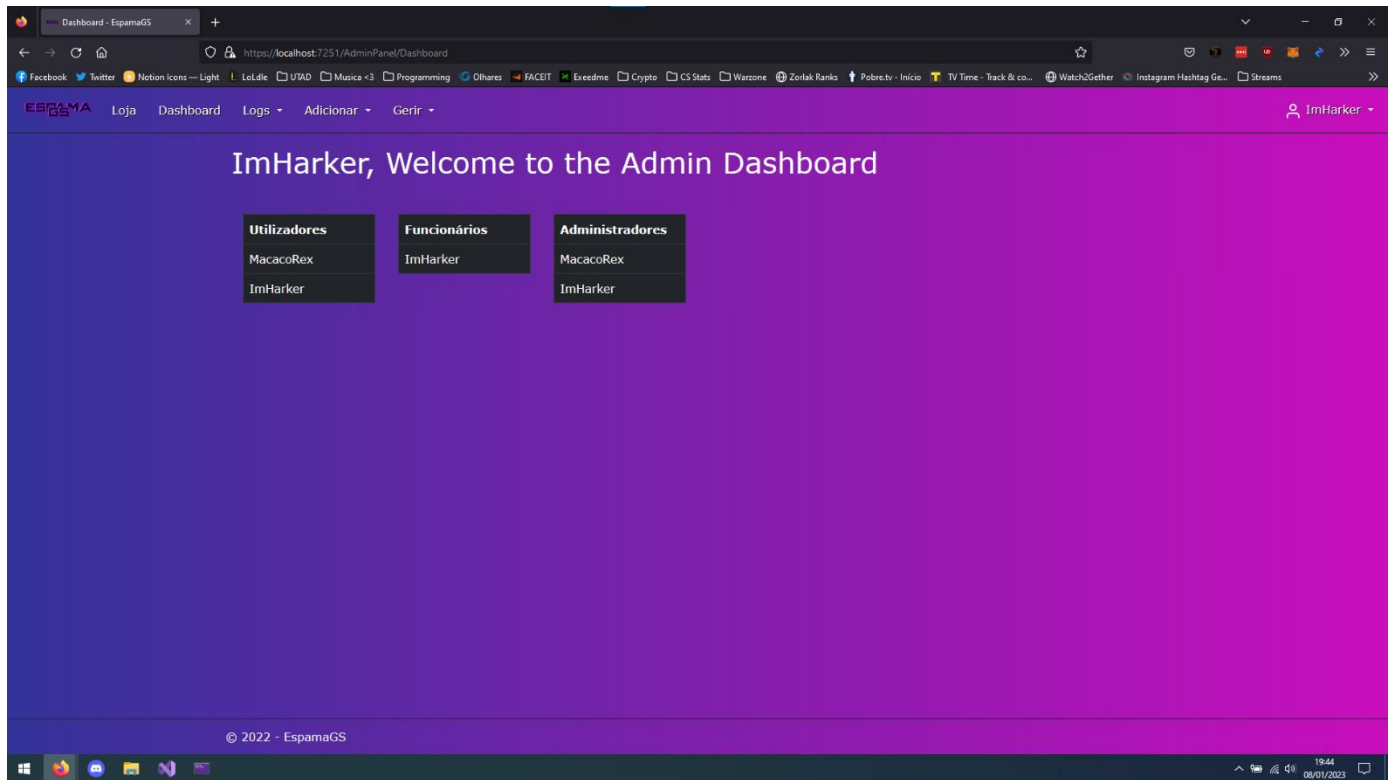
Aqui é apresentada a página que permite aos utilizadores seleccionar as suas categorias de preferência, tal como remover as mesmas.



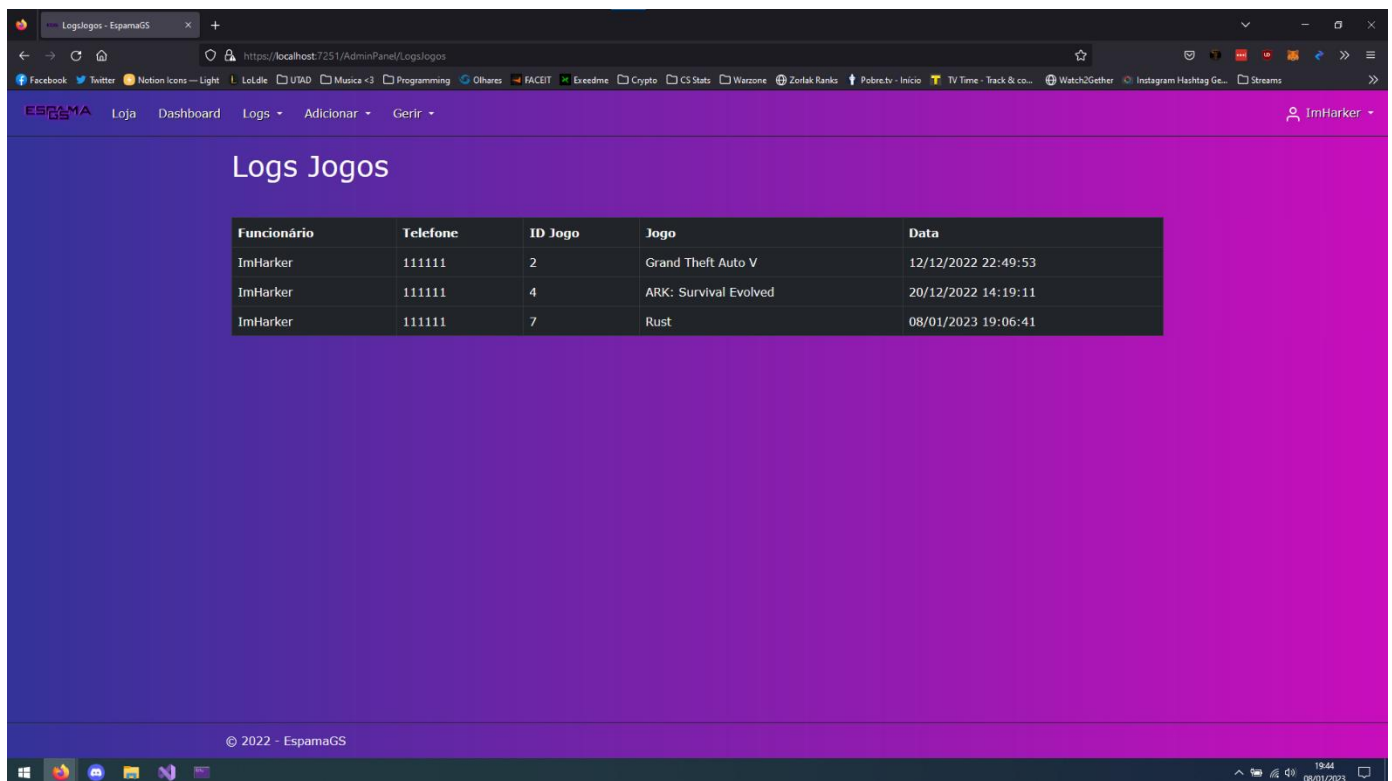
O histórico de compras, que nos apresenta os jogos que o utilizador já comprou, tal como, as datas de compra dos mesmos, o preço a que foi comprado na altura e a KEY de ativação de cada jogo.



Aqui já podemos ver o carrinho de compras do utilizador, que permite que cada utilizador veja o preço final de todos os jogos seleccionados, tal como excluir algum jogo da lista e finalizar a sua compra.



Página inicial do Dashboard, que os administradores e funcionários têm acesso, que permite verificar que contas estão registadas como utilizador, funcionário e administrador do site.



Logs dos jogos, que permite verificar quando os jogos foram adicionados e que funcionário os adicionou.

Logs Funcionarios

Administrador	Novo Funcionário	Tipo	Data
ImHarker	ImHarker	Funcionário	01/01/1970 00:00:00
ImHarker	ImHarker	Administrador	01/01/1970 00:00:00
ImHarker	MacacoRex	Administrador	08/01/2023 19:10:38

© 2022 - EspamaGS

Logs dos funcionários, que permite ver quem adicionou certos funcionários ou administradores e a data que as ações foram realizadas.

Logs Compras

Data	Cliente	ID Jogo	Jogo	Preço	Key
13/12/2022 12:28:35	ImHarker	2	Grand Theft Auto V	59,99 €	287C2E68-EA64-4672
13/12/2022 12:30:49	ImHarker	2	Grand Theft Auto V	59,99 €	1730429C-13CD-49EE
16/12/2022 09:58:58	ImHarker	2	Grand Theft Auto V	59,99 €	5C2C8E4F-4E45-4880
16/12/2022 10:23:15	ImHarker	2	Grand Theft Auto V	59,99 €	6FF392A8-4ESB-42FA
04/01/2023 20:09:33	ImHarker	2	Grand Theft Auto V	59,99 €	5C189234-2411-4599
08/01/2023 19:33:58	ImHarker	2	Grand Theft Auto V	59,99 €	35D24168-D4C3-41ED
08/01/2023 19:34:50	ImHarker	2	Grand Theft Auto V	59,99 €	B777CD56-B852-4768

© 2022 - EspamaGS

Logs das Compras, que permite apenas aos administradores, ver todas as compras realizadas no site, tal como o cliente que a realizou, a KEY de cada um dos jogos, a data da compra e o preço que foi pago por cada cliente.

A screenshot of a web browser displaying a form titled "Adicionar Funcionários". The form is centered on a purple gradient background. It contains three input fields: "USERNAME" (a text box), "TELEFONE" (a text box), and "TIPO DE FUNCIONÁRIO" (a dropdown menu with "Funcionário" selected). Below these fields is a black button with the text "ADICIONAR" in white. The browser's address bar shows "https://localhost:7251/AdminPanel/AddFuncionarios". The Windows taskbar at the bottom shows the date and time as 19:45 on 08/01/2023.

Aqui conseguimos ver a página que permite aos administradores adicionar novos funcionários ou administradores, em que é necessário registar o número telefónico e o tipo de funcionário para cada pessoa.

A screenshot of a web browser displaying a form titled "Adicionar Categoria". The form is centered on a purple gradient background. It contains one input field: "NOME" (a text box with the placeholder text "Enter Name"). Below this field is a black button with the text "ADICIONAR" in white. The browser's address bar shows "https://localhost:7251/AdminPanel/AddCategoria". The Windows taskbar at the bottom shows the date and time as 19:45 on 08/01/2023.

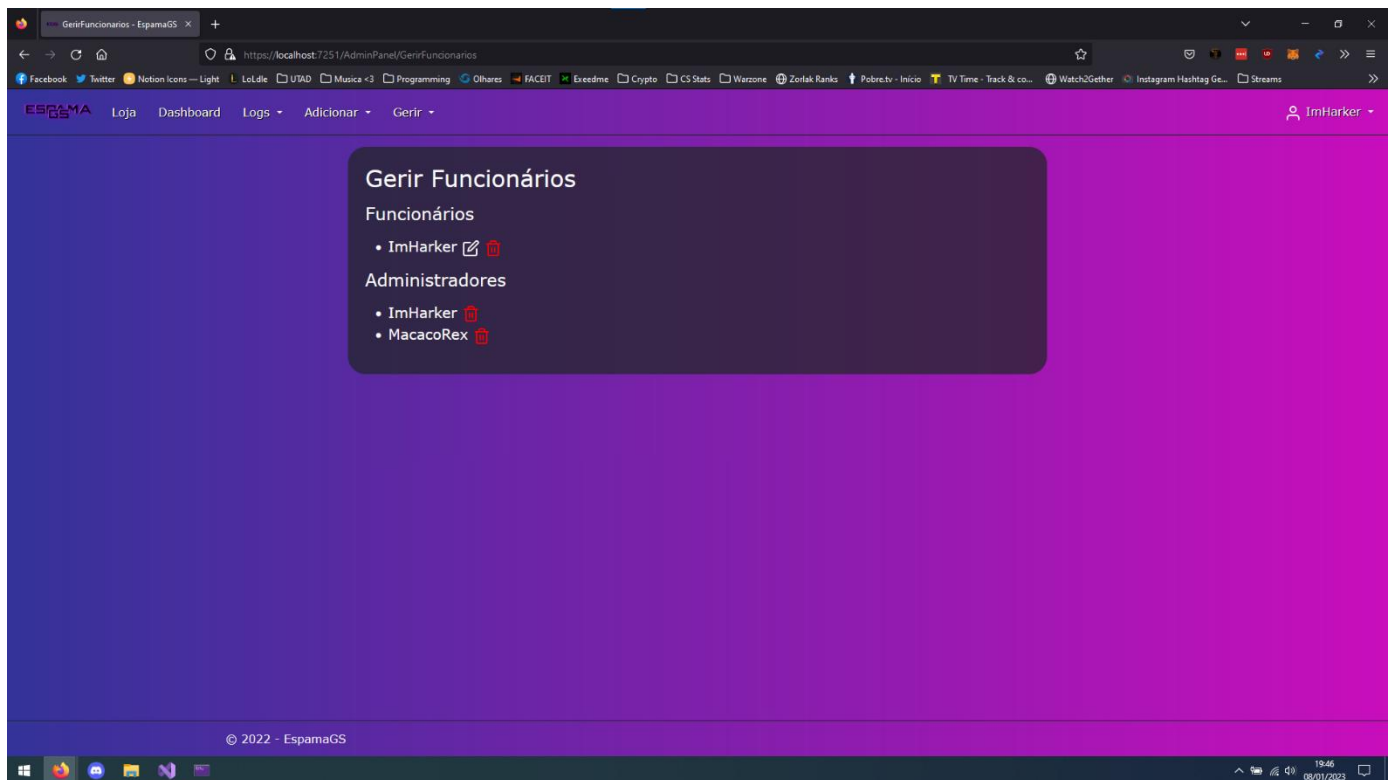
Esta página é idêntica para as categorias, plataformas e desenvolvedores, e permite-nos adicionar uma destas variáveis.

The image shows a web browser window displaying a form titled "Adicionar Jogo" (Add Game). The form is centered on a purple gradient background. The form fields are as follows:

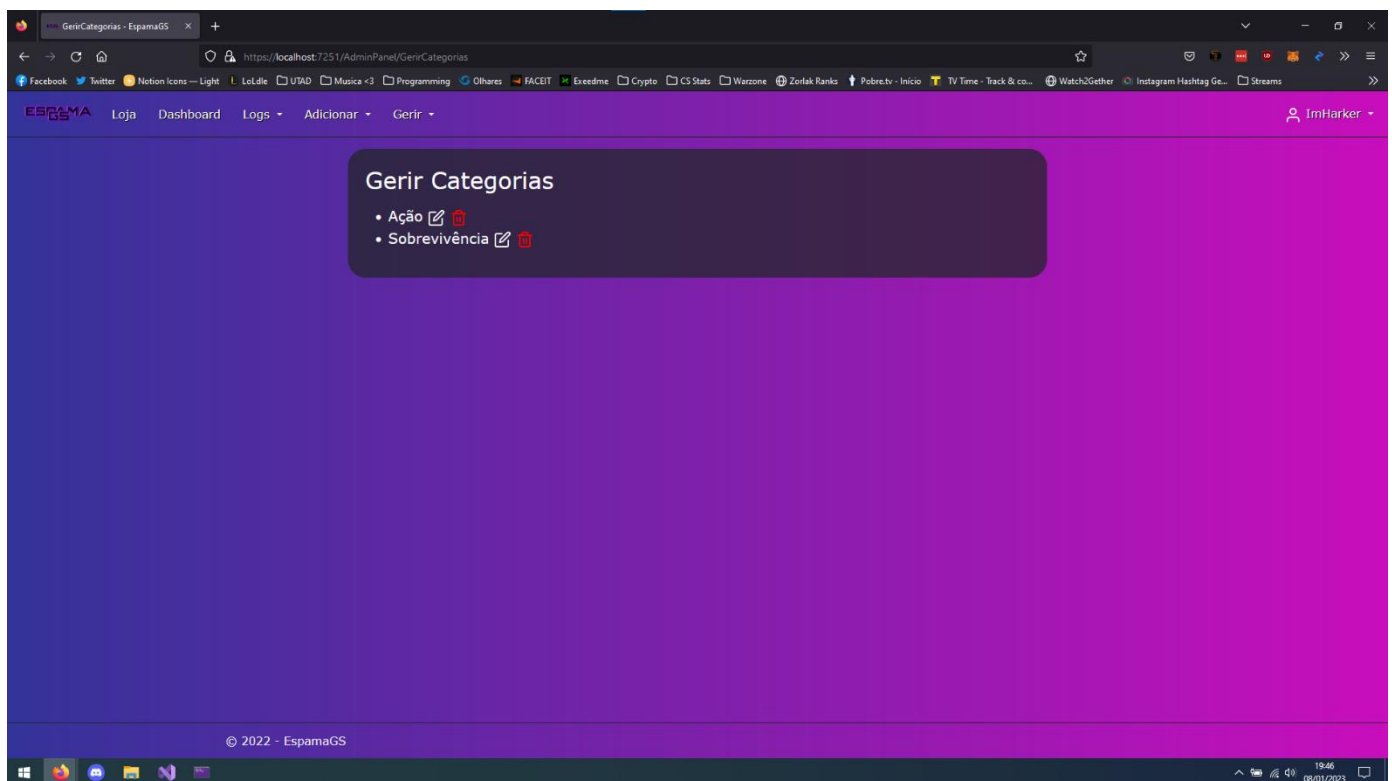
- TÍTULO**: Enter Title
- DESCRIÇÃO**: Enter Desc
- FOTO**: Explorar... Nenhum ...ionado.
- PREÇO**: Enter Price
- DATA DE LANÇAMENTO**: dd / mm / aaaa
- CATEGORIA**: Ação
- DESENVOLVEDORA**: Rockstar Games
- PLATAFORMA**: PC
- ADICIONAR** (button)

The browser window shows the URL "https://localhost:7251/AdminPanel/AddJogo" and the user "ImHarker". The footer of the page says "© 2022 - EspamaGS".

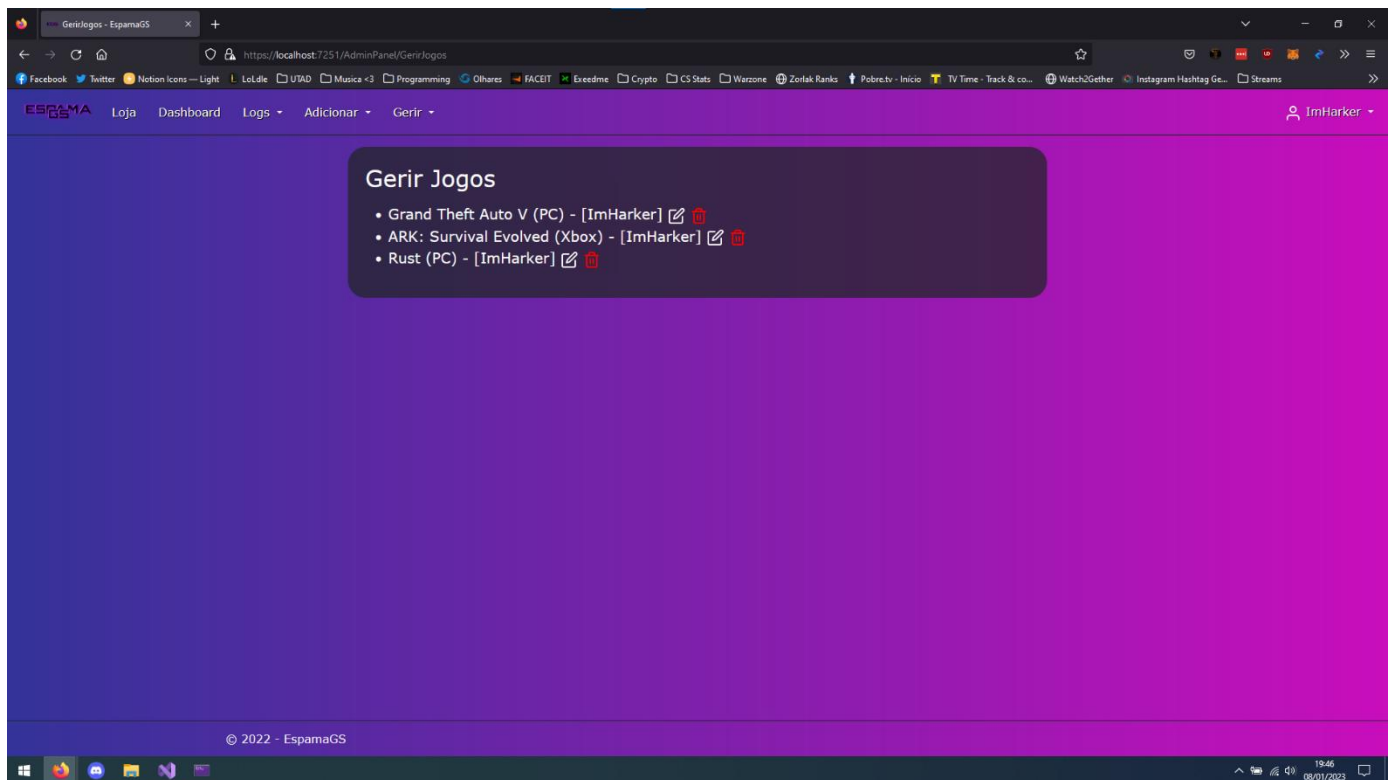
Aqui está apresentada a ação de adicionar um jogo, que permite a cada funcionário ou administrador acrescentar um jogo para venda no site, e vemos que podemos preencher vários campos, tais como o título do jogo, a descrição do mesmo, uma foto de capa do jogo, o seu preço, a data de lançamento, a categoria do jogo, a sua desenvolvedora e a plataforma do mesmo.



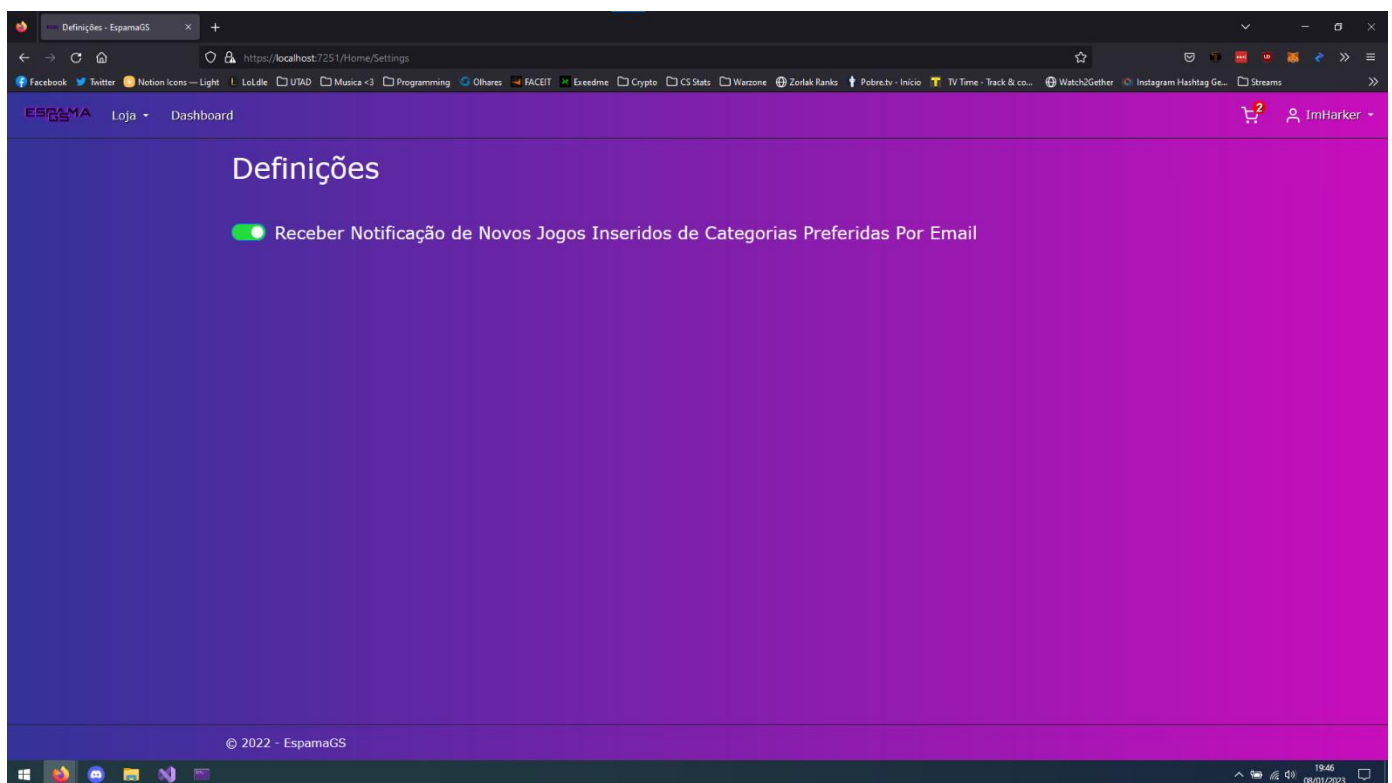
Aqui é apresentada a página que permite aos administradores gerir os funcionários que adicionaram no site.



Esta página é idêntica para as categorias, plataformas e desenvolvedoras, e permite gerir as mesmas que já foram adicionadas.

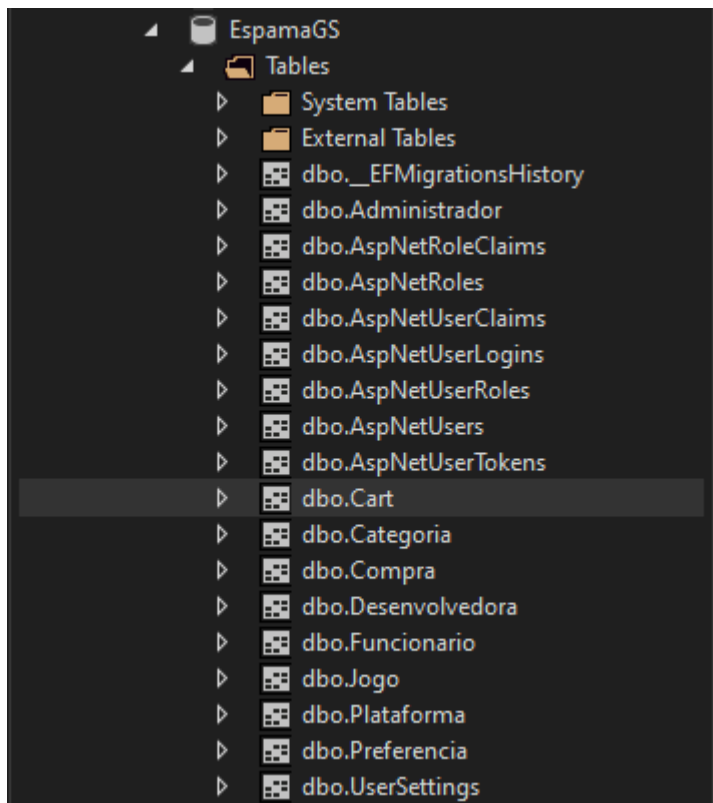


Gestão dos jogos, que permite aos funcionários e administradores gerir os jogos que adicionaram, em que podem alterar alguma das informações do mesmo, ou remover o jogo para venda no site.



Aqui mostra as definições de cada utilizador do site, que permite que o mesmo receba notificações por email, quando um novo jogo é adicionada a uma categoria que o mesmo tem na lista de categorias favoritas.

Estrutura Final da Base de Dados



A base de dados tem as tabelas default por utilizar o sistema de identidade e as tabelas criadas por nós:

- Administrador

```
[Key]
[Column(name: "ID_UTILIZADOR")]
[StringLength(20)]
[Unicode(false)]
7 references
public string IdUtilizador { get; set; } = null!;
[Column(name: "ID_ADMIN")]
[StringLength(20)]
[Unicode(false)]
4 references
public string IdAdmin { get; set; } = null!;
[Column(name: "DATA_REGISTO", TypeName = "datetime")]
1 reference
public DateTime DataRegisto { get; set; }

[ForeignKey(name: "IdAdmin")]
[InverseProperty("InverseIdAdminNavigation")]
3 references
public virtual Administrador IdAdminNavigation { get; set; } = null!;
[InverseProperty("IdAdminNavigation")]
2 references
public virtual ICollection<Funcionario> Funcionarios { get; set; }
[InverseProperty("IdAdminNavigation")]
2 references
public virtual ICollection<Administrador> InverseIdAdminNavigation { get; set; }
```

- Cart

```
[Key]
2 references
public int Id { get; set; }
3 references
public int IdJogo { get; set; }
[StringLength(50)]
[Unicode(false)]
11 references
public string IdCliente { get; set; } = null!;

[ForeignKey(name: "IdJogo")]
[InverseProperty("Carts")]
18 references
public virtual Jogo IdJogoNavigation { get; set; } = null!;
```

- Categoria

```
[Key]
[Column(name: "ID")]
[JsonInclude]
8 references
public int Id { get; set; }
[Column(name: "NOME")]
[StringLength(20)]
[Unicode(false)]
[JsonInclude]
18 references
public string Nome { get; set; } = null!;

[InverseProperty("IdCategoriaNavigation")]
[JsonIgnore]
5 references
public virtual ICollection<Jogo> Jogos { get; set; }
[InverseProperty("IdCategoriaNavigation")]
[JsonIgnore]
2 references
public virtual ICollection<Preferencia> Preferencia { get; set; }
```

- Compra


```

[Key]
[Column(name: "ID")]
1 reference
public int Id { get; set; }
[Column(name: "ID_CLIENTE")]
[StringLength(20)]
[Unicode(false)]
4 references
public string IdCliente { get; set; } = null!;
[Column(name: "KEY_JOGO")]
[StringLength(18)]
[Unicode(false)]
4 references
public string Key { get; set; } = null!;
[Column(name: "ID_JOGO")]
5 references
public int IdJogo { get; set; }
[Column(name: "DATA_COMPRA", TypeName = "datetime")]
6 references
public DateTime DataCompra { get; set; }
[Column(name: "PRECO", TypeName = "money")]

public decimal Preco { get; set; }

[ForeignKey(name: "IdJogo")]
[InverseProperty("Compras")]

public virtual Jogo IdJogoNavigation { get; set; } = null!;

```

- Desenvolvedora
Idêntica à estrutura da Categoria

- Funcionario

```
[Key]
[Column(name: "ID_UTILIZADOR")]
[StringLength(20)]
[Unicode(false)]
13 references
public string IdUtilizador { get; set; } = null!;
[Column(name: "TELEFONE")]
[StringLength(9)]
[Unicode(false)]
6 references
public string Telefone { get; set; } = null!;
[Column(name: "ID_ADMIN")]
[StringLength(20)]
[Unicode(false)]
4 references
public string IdAdmin { get; set; } = null!;
[Column(name: "DATA_REGISTO", TypeName = "datetime")]
1 reference
public DateTime DataRegistro { get; set; }

[ForeignKey(name: "IdAdmin")]
[InverseProperty("Funcionarios")]
3 references
public virtual Administrador IdAdminNavigation { get; set; } = null!;
[InverseProperty("IdFuncionarioNavigation")]
2 references
public virtual ICollection<Jogo> Jogos { get; set; }
```

- Jogo

```
[Key]
[Column(name: "ID")]
[JsonInclude]
21 references
public int Id { get; set; }
[Column(name: "NOME")]
[StringLength(50)]
[Unicode(false)]
[JsonInclude]
23 references
public string Nome { get; set; } = null!;
[Column(name: "FOTO")]
[StringLength(50)]
[Unicode(false)]
[JsonInclude]
16 references
public string Foto { get; set; } = null!;
[Column(name: "DESCRICAO")]
[StringLength(1024)]
[Unicode(false)]
[JsonInclude]
9 references
public string Descricao { get; set; } = null!;
[Column(name: "PRECO", TypeName = "money")]
[JsonInclude]
18 references
public decimal Preco { get; set; }
```

```

[Column(name: "ID_CATEGORIA")]
[JsonInclude]
11 references
public int IdCategoria { get; set; }
[Column(name: "ID_DESENVOLVEDORA")]
[JsonInclude]
9 references
public int IdDesenvolvedora { get; set; }
[Column(name: "ID_PLATAFORMA")]
[JsonInclude]
10 references
public int IdPlataforma { get; set; }
[Column(name: "ID_FUNCIONARIO")]
[StringLength(20)]
[Unicode(false)]
[JsonIgnore]
5 references
public string IdFuncionario { get; set; } = null!;
[Column(name: "DATA_REGISTO", TypeName = "datetime")]
[JsonIgnore]
2 references
public DateTime DataRegistro { get; set; }
[Column(name: "DATA_LANCAMENTO", TypeName = "date")]
[JsonInclude]
11 references
public DateTime DataLancamento { get; set; }

```

```

[ForeignKey(name: "IdCategoria")]
[InverseProperty("Jogos")]
[JsonInclude]
17 references
public virtual Categoria? IdCategoriaNavigation { get; set; } = null!;
[ForeignKey(name: "IdDesenvolvedora")]
[InverseProperty("Jogos")]
[JsonInclude]
17 references
public virtual Desenvolvedora? IdDesenvolvedoraNavigation { get; set; } = null!;
[ForeignKey(name: "IdFuncionario")]
[InverseProperty("Jogos")]
[JsonIgnore]
5 references
public virtual Funcionario? IdFuncionarioNavigation { get; set; } = null!;
[ForeignKey(name: "IdPlataforma")]
[InverseProperty("Jogos")]
[JsonInclude]
21 references
public virtual Plataforma? IdPlataformaNavigation { get; set; } = null!;
[InverseProperty("IdJogoNavigation")]
[JsonIgnore]
2 references
public virtual ICollection<Cart>? Carts { get; set; }
[InverseProperty("IdJogoNavigation")]
[JsonIgnore]
2 references
public virtual ICollection<Compra>? Compras { get; set; }

```

- Plataforma
Idêntica à estrutura da Categoria
- Preferencia

```
[Key]
[Column(name: "ID_CLIENTE")]
[StringLength(20)]
[Unicode(false)]
5 references
public string IdCliente { get; set; } = null!;
[Key]
[Column(name: "ID_CATEGORIA")]
6 references
public int IdCategoria { get; set; }

[ForeignKey(name: "IdCategoria")]
[InverseProperty("Preferencia")]
4 references
public virtual Categoria IdCategoriaNavigation { get; set; } = null!;
```

- UserSettings

```
[Key]
4 references
public string Id { get; set; }

5 references
public Boolean EmailNotifications { get; set; }
```

Ainda é necessário fazer uma modificação sempre que a base de dados é criada, que é adicionar o ON DELETE CASCADE nas tabelas que referenciam o jogo como chave estrangeira

Organização do grupo no desenvolvimento do projeto

Durante o desenvolvimento deste extenso projeto, o nosso grupo organizou-se através da utilização de Kanban Boards e reuniões periódicas, que foi facilitado devido a habitar todos juntos, o que permitiu uma maior facilidade quando surgiram dúvidas ou erros no código.

Há medida que eram implementadas novas funcionalidades estas eram revistas por outros elementos do grupo e testadas.

Conclusão

Neste projeto conseguimos entender as dificuldades da realização de um site, neste caso de uma loja de jogos, e aprendemos a utilizar a framework ASP .NET CORE. Onde tirámos algumas ideias de uns sites de venda de jogos já existentes, que nos ajudou a planear as ideias e as views do site.