**

**ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO**

INSTITUTO POLITÉCNICO DA GUARDA

trabalho de Programação para a internet

Website eventorg2027

|  |  |
| --- | --- |
| **Curso** | **Engenharia Informática** |
| **Unidade Curricular** | Programação para a Internet |
| **Ano Letivo** | 2020/2021 |
| **Docente** | Noel Lopes |
| **Data** | 03/02/2021 |
| **Alunos** | Marcelo Lopes nº 1012179  João Aires nº 1700318 |

**Índice**

[1.](#_30j0zll) Definição do Projeto 2

[2.](#_1fob9te) Lista de Atores 3

[3.](#_3znysh7) Funcionalidades de aplicações semelhantes 4

[4.](#_2et92p0) Diagrama Casos de Uso 4

[5.](#_tyjcwt) Descrição de Casos de Uso 5

[5.1.](#_3dy6vkm) Caso de uso Criar Conta 5

[5.2.](#_1t3h5sf) Caso de uso Reservar Lugar no Evento 6

[Cartão User Story e Cartão Test Case 8](#_4d34og8)

[5.3.](#_2s8eyo1) Cartão User Story e Test Case, Reservar lugar num evento 8

[5.4.](#_17dp8vu) Cartão User Story e Test Case, Criar Conta 8

[6.](#_3rdcrjn) Diagrama de Classes 9

[7.](#_26in1rg) Diagrama de Sequência 11

[7.1.](#_lnxbz9) Diagrama de Sequência – Reservar lugar 11

[8.](#_35nkun2) Semântica de Classes 12

[8.1.](#_1ksv4uv) Eventos 12

[8.2.](#_44sinio) Localidade 14

[9.](#_3j2qqm3) Diagrama de Estados 16

[9.1.](#_1y810tw) Aprovação da Reserva 16

[10.](#_4i7ojhp) Diagrama de Componentes 17

[11.](#_2xcytpi) Diagrama de Instalação 18

[12.](#_1ci93xb) Diagrama de Pacotes de Casos de Uso 19

[13.](#_3whwml4) Casos de Teste 20

[13.1. Teste de Criar Conta 20](#_2bn6wsx)

[14.](#_3as4poj) Conclusão 21

1. Definição do Projeto

Com este projeto inserido na disciplina de Programação para a Internet pretendemos criar um site de Eventos Musicais e de Teatro e tratar de toda a parte da Engenharia de Software.

Vamos criar um site que dê a conhecer informações sobre Eventos Musicais e de Teatro e os respetivos cuidados a ter durante os tempos de pandemia.

Para tal, a nosso site irá apresentar Eventos, Tipo de Eventos, Localidades e Organizadores, vai ser possível os utilizadores criarem uma conta e ter acesso a todos os eventos disponíveis, vai ser também possível a reserva de um lugar nesse respetivo evento.

Assim sendo, o objetivo será conseguir desenvolver uma forma de gerir e reservar lugar em eventos mais facilmente, tudo um único lugar, numa única aplicação.

1. Lista de Atores

O nosso projeto tem como atores: Cliente e Administrador. A tabela abaixo apresenta os objetivos para os seus respetivos atores.

|  |  |
| --- | --- |
| Ator | *Objetivo* |
| Cliente | Consultar Informação sobre Eventos. |
|  | Consultar Eventos. |
|  | Reservar lugar nos respetivos eventos. |
| Administrador | Adicionar Eventos, Tipos de Eventos, Localidades, Organizadores |
|  | Editar Eventos, Tipos de Eventos, Localidades, Organizadores |
|  | Eliminar Eventos, Tipos de Eventos, Localidades, Organizadores |

*Tabela 1 - Atores/Objetivos*

1. Funcionalidades de aplicações semelhantes

Na tabela 2, verificamos as funcionalidades do nosso projeto comparando-as com outras do distrito da Guarda.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Funcionalidades  Aplicações | Redes Sociais | Login | Reserva | Multi-idiomas | Gerir Lotação | Lista de Localidades |
| EventOrg2027 | 🗶 | ✓ | ✓ | 🗶 | ✓ | ✓ |
| beira.pt | ✓ | 🗶 | 🗶 | 🗶 | 🗶 | ✓ |
| visitportugal.com | 🗶 | ✓ | 🗶 | ✓ | 🗶 | 🗶 |
| mun-guarda.pt | ✓ | 🗶 | 🗶 | 🗶 | 🗶 | ✓ |

Tabela 2 – Funcionalidades de aplicações

1. Diagrama Casos de Uso

Este diagrama tem como objetivo dar-nos a conhecer os atores do sistema assim como as suas interações com os diferentes casos de uso.

A fronteira do sistema limita os casos que dizem respeito ao sistema e a herança é um princípio que permite que os atores partilhem as funcionalidades. Neste caso, o administrador consegue interagir com todos os casos de uso, pois existe herança do Cliente para o administrador.

Diagram

Description automatically generated

*Figura 1 - Diagrama de Casos de Uso*

1. Descrição de Casos de Uso

As tabelas seguintes (Tabela 3, Tabela 4, Tabela 5) demonstram como deveria decorrer o caso de uso para que o sistema funcione da maneira pretendida, contendo um caminho principal, do qual deveria resultar o funcionamento normal do sistema, mas pode ser necessário tomar rotas alternativas como o caminho alternativo indica.

* 1. Caso de uso Criar Conta

|  |  |
| --- | --- |
| Nome | Criar Conta. |
| Descrição | Este caso de uso tem como objetivo descrever o processo de criar uma conta de voluntario. |
| Tamanho | S. |
| Pré-Condição | Login Válido. |
| Caminho Principal | 1. O utilizador seleciona a opção “Criar Conta”; 2. O sistema apresenta uma página com todos os campos que o utilizador tem de preencher; 3. O utilizador preenche todos esses campos e de seguida pressiona o botão Registar; 4. O sistema guarda a conta do utilizador. |
| Caminhos Alternativos | 1. Erro a presentar a página; 2. Os campos obrigatórios não foram todos preenchidos; |

*Tabela 3 - Caso de uso Criar Conta*

* 1. Caso de uso Reservar Lugar no Evento

|  |  |
| --- | --- |
| Nome | Reservar lugar no evento. |
| Descrição | Este caso de uso tem como objetivo descrever o processo de reservar um lugar em um determinado evento. |
| Tamanho | S. |
| Pré-Condição | Login Válido. |
| Caminho Principal | 1. O utilizador seleciona a opção “Evento”; 2. O sistema apresenta uma página com todos os eventos; 3. O utilizador seleciona um evento; 4. O utilizador carrega no botão de reserva; 5. O sistema guarda a reserva do lugar; |
| Caminhos Alternativos | 1. Erro a apresentar a página; 2. O utilizador pode já estar inscrito, o evento pode já ter ocorrido ou a lotação pode ter excedido o número máximo de inscrições; |

*Tabela 4 - Caso de uso reservar lugar no evento.*

Cartão User Story e Cartão Test Case

User Story – Contem o que o sistema de software pode fazer e também expressa o valor que ele fornecerá a um utilizador do sistema.

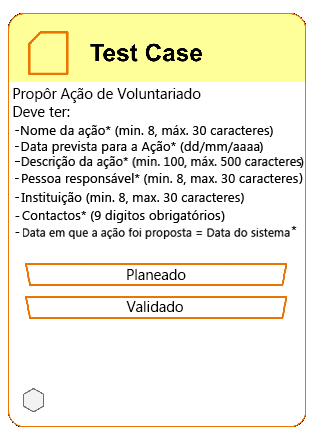
Test Case – Define as entradas de teste e resultados esperados para avaliar se o User Story foi completamente e corretamente implementado.

* 1. Cartão User Story e Test Case, Reservar lugar num evento

Graphical user interface, text, application

Description automatically generated

*Figura 3 - Test Case, Reservar lugar num evento*



Reservar lugar no evento

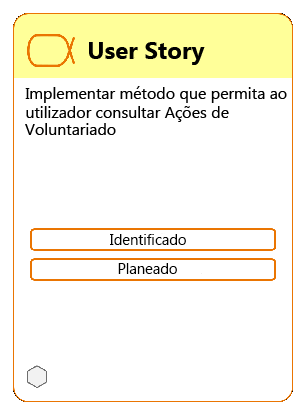
O cliente poderá reservar o lugar nesse determinado evento.

Implementar método que permita ao cliente reservar um lugar num determinado evento.

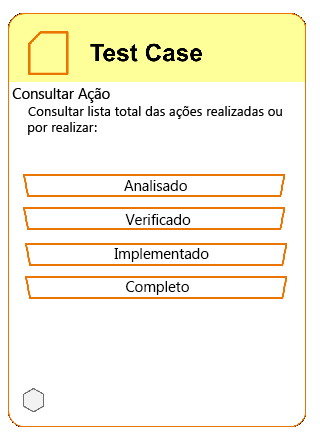
*Figura 2 - User Story, Reservar lugar num evento*

* 1. Cartão User Story e Test Case, Criar Conta

*Figura 5 - Test Case, Criar conta*



Implementar método que permita ao cliente criar uma conta.



Criar conta

O cliente poderá criar uma conta

*Figura 4 - User Story, Criar conta*

1. Diagrama de Classes

Um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetoDiagram

Description automatically generated

*Figura 8 - Diagrama de Classes*

1. Diagrama de Sequência

O Diagrama de Sequência tem como objetivo principal representar graficamente o comportamento de uma funcionalidade, considerando a interação entre todos os componentes de software relacionados ao seu uso.

* 1. Diagrama de Sequência – Reservar lugar

Diagram

Description automatically generated

*Figura 10 - Diagrama de Sequência, Reservar lugar*

1. Semântica de Classes

A semântica de classes ajuda a identificar atributos e operações por análise dos vários cenários em que a classe participa, identificando-se assim as responsabilidades da classe.

* 1. Eventos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Eventos | | | | | |
| Nome do Campo | Tipo de dados | Descrição | Valores Válidos | Formato | Restrições |
| ID\_Evento  (PK) | Numeração automática | Número sequencial que identifica cada evento. | Maior que 0. | - | Gerado pelo sistema / Não alterável |
| Nome do Evento | Nome | Nome que identifica o evento. | Conjunto de caracteres não nulo. | Até 50 caracteres. | Obrigatório / Alterável |
| Descrição | String | Descrição do evento. | Conjunto de caracteres não nulo. | Até 500 caracteres. | Obrigatório / Alterável |
| Lotação | Inteiro | Conjunto de itens referentes à lotação. | Conjunto de caracteres, não nulo . | - | Obrigatório / Alterável |
| Data da Realização | Data | Data prevista para ser realizado o evento. | - | “dd-mm-yyyy” | Obrigatório / Alterável |
| Hora da Realização | Data | Hora prevista para ser realizado o evento. | - | “hh:mm” | Obrigatório / Alterável |
| Localidade\_Id  (FK) | Numeração automática | Número sequencial que identifica cada localidade. | Maior que 0. | - | Gerado pelo sistema / Não alterável |
| TipoEventos\_Id  (FK) | Numeração automática | Número sequencial que identifica cada tipo de evento. | Maior que 0. | - | Gerado pelo sistema / Não alterável |
| Organizador\_Id  (FK) | Numeração automática | Número sequencial que identifica cada organizador. | Maior que 0. | - | Gerado pelo sistema / Não alterável |

*Tabela 6 - Dicionário de Dados, Eventos*

|  |  |
| --- | --- |
| Operações | |
| Nome | Descrição |
| CriarEvento() | Operação que permite criar Evento   1. Sistema gera o ID\_Evento; 2. Administrador insere o nome que identifica a Evento; 3. Adiciona uma descrição, referente ao Evento; 4. Insere a lotação para esse evento; 5. Insere a data para a realização da Evento; 6. Insere a hora para a realização da Evento; 7. Escolhe a respetiva localidade; 8. Escolhe o tipo de evento; 9. Escolhe o organizador; 10. Por fim, adiciona o evento; |
| EditarEvento() | Operação que permite editar o Evento   1. Selecionar Nome do Evento– Editar; 2. Selecionar Descrição – Editar; 3. Selecionar Lotaçao – Editar; 4. Selecionar Data – Editar; 5. Selecionar Hora – Editar; 6. Selecionar Localidade – Editar; 7. Selecionar Tipo de Evento – Editar; 8. Selecionar Organizador – Editar; |
| EliminarEvento() | Operação que permite apagar Evento   1. Selecionar – Eliminar Evento; |
| AcompanhamentoAção() | Operação que permite validar o estado da Evento;   1. Se Evento estiver agendada – Agendada; 2. Se Evento estiver a ser realizada – Em andamento; 3. Se Evento estiver Concluída – Concluída; |

*Tabela 7 - Operações, Eventos*

|  |
| --- |
| Casos de Uso em que a classe participa |
| * Registar Evento * Atualizar Evento * Consultar Evento * Consultar Evento |

*Tabela 8 - Casos de Uso em que a Classe participa, Eventos*

* 1. Localidade

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Localidade | | | | | |
| Nome do Campo | Tipo de Dados | Descrição | Valores Válidos | Formato | Restrições |
| ID\_Localidade  (PK) | Numeração Automática | Número sequencial, que identifica cada Localidade. | Maior que 0. | Até 6 dígitos. | Gerado pelo Sistema/Não Alterável |
| Nome | String | Nome da Localidade. | Conjunto de caracteres de A-Z e a-z. | Até 50 caracteres. | Obrigatório/Não Alterável |
| Descrição | String | Descrição da localidade. | Conjunto de caracteres não nulo. | Até 500 caracteres. | Obrigatório / Alterável |
| População | Inteiro | Conjunto de itens referentes à população. | Conjunto de caracteres, não nulo . | - | Obrigatório / Alterável |

*Tabela 9 - Dicionário de Dados, Localidade*

|  |  |
| --- | --- |
| Operações | |
| Nome | Descrição |
| CriarLocalidade() | Operação que permite criar uma localidade   1. Sistema gera o ID\_Localidade; 2. O administrador preenche o campo Nome; 3. O administrador preenche o campo Descrição; 4. O administrador preenche o campo População; 5. Por fim, adiciona a localidade; |
| EditarLocalidade() | Operação que permite editar a localidade   1. Selecionar Nome- Editar; 2. Selecionar Descrição – Editar; 3. Selecionar População – Editar; |
| EliminarLocalidade() | Operação que permite apagar localidade   1. Selecionar – Eliminar Localidade; |
| ConsultarLocalidade() | Operação que permite consultar a localidade   1. Selecionar localidade; 2. Consultar localidade; |

*Tabela 10 - Operações, Localidade*

|  |
| --- |
| Casos de Uso em que a classe participa |
| * Criar Localidade * Atualizar Localidade * Consultar Localidade |

*Tabela 11 - Casos de Uso em que a Classe participa, Localidade*

1. Diagrama de Estados

O diagrama de estados serve para capturar comportamento dinâmico, mostrar os eventos que causam a transição de um estado para outro, assim como as ações que resultam de uma alteração de estado e tem como objetivo modelar o ciclo de vida de objetos.

* 1. Aprovação da Reserva

Diagram

Description automatically generated

*Figura 12 - Diagrama de Estados, Aprovação da Reserva*

1. Diagrama de Componentes

O Diagrama de Componentes permite descrever os diversos “pedaços” de software que são os programas fonte, bibliotecas ou programas executáveis, cujo objetivo é organizar o código fonte, construir uma release executável e especificar uma base de dados física.

Diagram

Description automatically generated

*Figura 14 - Diagrama de Componentes*

1. Diagrama de Instalação

O Diagrama de Instalação permite descrever a arquitetura do sistema em termos de hardware e a sua relação com os diferentes componentes de software.

Ilustra a arquitetura do sistema em termos de nós (nodes) que efetuam o processamento de componentes. Permite mostrar como o hardware estará organizado e como os componentes (software) estarão distribuídos, estabelecendo assim a sua relação física.

Diagram

Description automatically generated

*Figura 15 - Diagrama de Instalação*

1. Diagrama de Pacotes de Casos de Uso

O Diagrama de Pacotes mostra como os elementos do modelo estão organizados em pacotes, bem como as dependências entre pacotes. Muito usado para ilustrar a arquitetura de um sistema mostrando o agrupamento das classes ou utilizado em qualquer fase do processo para organizar os diagramas.

Text

Description automatically generated with medium confidence

*Figura 16 - Diagrama de Pacotes para Casos de Uso*

1. Casos de Teste

Um Caso de Teste é um conjunto de condições usadas para realizar testes de software, ajudando a identificar defeitos na estrutura interna do software, ou garantir que os requisitos do software que foram construídos sejam plenamente realizados.

Os testes neste ponto encontram-se a negativo pois ainda não foi desenvolvido até este ponto do software.

13.1. Teste de Criar Conta

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Caso de Teste Criar Conta | | | | | | |
| ID | Entrada | Observações | Resultados esperados | Resultados atuais | V | X |
| 1 | Botão  “Registar” | O Utilizador selecionou a opção “Criar Conta”. | Apresentar página com campos para preencher. | Apresenta página com campos para preencher. | V |  |
| 2 | Botão  “Registar” | O utilizador visualiza um erro. | Erro ao apresentar a página. | Erro ao apresentar a página. | V |  |
| 3 | Formulário para preencher | O utilizador seleciona os campos com dados referentes ao login. | Consegue preencher todos os campos. | Consegue preencher todos os campos. | V |  |
| 4 | Preenchimento do Formulário | Validações, verificar se os dados inseridos são corretos.  Ex: Inserir um nome, no campo “Nome”  (campo obrigatório). | Mensagem de erro, caso não sejam inseridos caracteres correntes no campo. | Consegue visualizar a mensagem de erro. | V |  |
| 5 | Botão “Registar” | O utilizador selecionar o botão Registar. | O sistema guardar a conta do Utilizador. | Guardou a Conta do Utilizador. | V |  |
| 6 | Notificação ao Utilizador | O Utilizador recebe uma confirmação de como a conta foi criada com sucesso. | Receber notificação. | Recebeu notificação. |  | X |

Tabela 21 – Teste Criar Conta

1. Conclusão

Na elaboração deste trabalho concluímos que desenvolver um software, exige rigor, dedicação, muito trabalho e um bom planeamento, de forma a que qualquer pessoa o conseguir compreender com relativa facilidade.

Conseguimos obter conhecimentos sobre C# e ASP.NET aplicando ao projeto do Eventos de Música e Teatro e assim vimos as vantagens que podem trazer, como por exemplo o desenvolvimento aprimorado, completo e eficaz.

Desta forma podemos minimizar falhas futuras na realização do projeto, de conceção de erros no software e também conseguir analisar e testar outros softwares de maneira mais lógica e racional.