

Relatório



Bora Lá: Explorador de Eventos Locais

Trabalho realizado por:

José Antunes

Luís Gomes

Sara Virgínia

Professora

Marta Martinho

Escola Superior de Tecnologia
Instituto Politécnico do Cávado e do Ave

RESUMO

Este relatório descreve um projeto multidisciplinar desenvolvido no âmbito do curso de requalificação Upskill de Power Platform. O projeto tem como objetivo aplicar e desenvolver conhecimentos adquiridos em várias disciplinas do curso, indo desde o planeamento, criação de uma base de dados (back-end), até ao desenvolvimento de uma aplicação/página Web (front-end), implementando no processo metodologias de Agile.

O conceito do projeto consistiu no desenvolvimento de um sistema (Web APP) que permite a utilizadores descobrirem eventos que irão decorrer nas suas proximidades. Todos os eventos seriam aceites, mas o foco seriam eventos que pela sua natureza mais local, como uma noite de Quiz no bar do bairro, têm dificuldade em obter visibilidade e chegar ao conhecimento de possíveis interessados.

O sistema foi desenvolvido com sucesso, desde levantamento de requisitos, elaboração de esquemas, desenvolvimento da base de dados completa, criação do Web service para comunicação com a mesma, e por fim desenvolvimento da APP para interação com os utilizadores. Todas as fases foram apoiadas na metodologia Agile, para um desenvolvimento mais eficaz. A elaboração do Web Service (API) e da APP, pela sua complexidade e tempo disponível, foram apenas realizadas em parte para prova de conceito.

Consideramos que o principal objetivo deste projeto, nomeadamente a aquisição de conhecimentos introdutórios nestas áreas de forma prática e activa, foi concluído com sucesso.

ABSTRACT

This report describes a multidisciplinary project developed as part of the Upskill Power Platform requalification course. The project aims to apply and develop knowledge acquired in various course disciplines, ranging from the development of a web application/front-end to its database/back-end, implementing Agile methodologies throughout the process.

The project concept involved creating a system (Web APP) that enables users to discover events taking place in their vicinity. While all events are accepted, the focus is on local events, such as a neighborhood bar quiz night, which struggle to gain visibility and reach potential interested parties due to their local nature.

The system was successfully developed, encompassing requirements gathering, schema design, complete database development, creation of a web service for communication, and ultimately, the development of the app for user interaction. All phases were supported by Agile methodology for more effective development. The creation of the web service (API) and the app, due to their complexity and time constraints, were only partially completed for proof of concept.

We consider that the primary objective of this project, particularly the acquisition of introductory knowledge in these areas through practical and active means, has been successfully achieved.

ÍNDICE

●	5
1. Introdução.....	1
○ 1.1 Aplicação Bora Lá.....	1
○ 1.2 Instigadores do projecto.....	3
○ 1.3 Objetivos do Projeto.....	4
2. Requisitos.....	5
2.1. Personas.....	5
2.2. User Stories.....	8
2.3. Requisitos Funcionais.....	10
2.4. Requisitos Não Funcionais.....	12
2.5. Prioritização de requisitos (técnica MoSCoW).....	12
2.6. Wireframes.....	14
3. Metodologias e Tecnologias de Trabalho.....	17
3.1. A Nossa Metodologia de Trabalho.....	17
3.1.1. Agile.....	17
3.1.1.1. Scrum.....	18
3.1.1.2. Funções dos Membros da Equipa.....	19
3.1.1.3. Ferramentas do Scrum.....	24
3.1.1.4. Fases de um Sprint.....	27
3.1.2. Tecnologias de Trabalho.....	29
3.2.1. Desenvolvimento da App em Angular.....	29
3.2.1.1 HTML, CSS e Typescript.....	30
3.2.1.3 Routing e Navegação.....	31
3.2.1.4 Formulários e Validação.....	32
3.2.1.5 Backend – Comunicação e Troca de Informação.....	33
3.2.1.6 Bootstrap.....	34
3.2.2 API em C# e ASP.NET Core Web API.....	35
3.2.2.1 Comunicação entre backend e frontend.....	36
3.2.2.2 Lógica Programacional.....	37
3.2.2.3 Autenticação e Autorização.....	38
3.2.2.4 Tratamento de Erros.....	39
3.2.3 Base de Dados (SQL Server).....	46
3.2.3.1 Modelação de Dados.....	47
3.2.3.1.1 Tabelas.....	48
3.2.3.1.2 Relações.....	49
3.2.3.2 Linguagem SQL (Structured Query Language).....	49
3.2.3.2.1 Indexação e Desempenho.....	50
3.2.3.2.2 Segurança e Acessos.....	50

3.2.3.2.3 Autenticação.....	51
3.2.3.2.4 Autorização.....	51
4. Conclusão.....	53
5. ANEXOS.....	54
Anexo A.....	55

1. Introdução

Este trabalho surge no âmbito de um curso de requalificação UpSkill, que promove a formação em Windows Power Platform. Apesar das metodologias e tecnologias aqui aplicadas não estarem diretamente relacionadas com Power Platform, ter conhecimentos das mesmas pode ser importante em algumas fases do nosso estágio, ou até em trabalhos futuros. As tecnologias usadas, desde a linguagem de programação (C#), o SQL Server para criação e gestão da base de dados, a API REST CORE.NET, foram escolhidas por serem as mais comuns em ambientes de desenvolvimento Windows, sendo assim as mais prováveis de encontrar no contexto do estágio e trabalho.

A primeira tarefa foi um brainstorming entre os elementos do grupo para chegar a uma ideia de um aplicativo ou sistema interessante. Várias ideias surgiram, sendo que algumas, apesar de muito interessantes, foram postas de lado devido a limitações como a impossibilidade ou dificuldade de obtenção de dados (por exemplo dados de trânsito). A ideia que se revelou vencedora foi o desenvolvimento de uma plataforma que permitisse a um utilizador descobrir que eventos irão decorrer num futuro próximo, na sua zona. Optamos por esta ideia porque além de interessante e com potencial de atender a muita gente, parecia ser também a que teria menos limitações em termos de obtenção de dados.

No ponto seguinte é descrita a ideia em pormenor.

- **1.1 Aplicação Bora Lá**

A aplicação Bora Lá surge da necessidade de estarmos a par dos eventos a decorrer perto de nós. Quantas vezes não perdemos um evento interessante por apenas sabermos da sua existência depois do mesmo ter ocorrido? Ou quantas vezes que temos vontade de fazer algo diferente mas acabamos a ir ao local de sempre por não saber o que está a decorrer.

É verdade que esta ideia não é de todo inovadora, já há muito tempo que existem aplicações/websites com agendas de eventos, como por exemplo o Viral Agenda (<https://www.viralagenda.com>), o Wikinight (<https://wikinight.eu>), ou até

mesmo os eventos no Facebook. Estes serviços têm no entanto quase sempre associados três problemas principais, que muitas vezes estão relacionados entre si. Um deles é de não conseguirem ganhar visibilidade e adesão de utilizadores. Outro é só apresentarem eventos de grande dimensão ou que são promovidos por entidades profissionais públicas ou privadas, deixando de lado iniciativas locais e de pequenos negócios. Por fim, um grave problema neste tipo de iniciativas é arranjar forma de alimentar a plataforma com a informação necessária.

De forma a resolver estes problemas decidimos que o foco desta plataforma serão os eventos locais, por exemplo uma noite de Karaoke no bar do bairro, um brunch acompanhado de declamação de poesia, um workshop de olaria no pequeno atelier perto de nossa casa, etc. Não rejeitamos eventos de grandes dimensões, mas acreditamos que estes eventos intimistas acabam por ter um interesse cada vez maior numa era dos grandes números, dos likes e das visualizações, o que pode levar a uma maior adesão de utilizadores. Por forma a alimentar a plataforma com a informação sobre os eventos e os espaços, existiria uma aposta inicial em comerciais que iriam fazer uma prospecção dos espaços existentes e contactar os gerentes, e pessoas que promovem eventos, para aderirem à Bora Lá e publicitar na plataforma os seus eventos e espaços.

O modelo de negócio consistiria na aplicação de uma mensalidade aos promotores de eventos que quisessem maior visibilidade. Note-se que quando falamos em promotores de eventos, falamos em pessoas que promovem eventos em vários espaços, ou em donos ou gerentes de espaços que promovem os eventos no seu espaço. Qualquer promotor se poderia registar sem ter de pagar, mas além da categoria grátis, podiam passar para a standard ou a premium, por uma mensalidade de 100€ e 200€ respectivamente.

Este modelo tem como desvantagem a necessidade, pelo menos inicial, de obrigar a uma prospecção por promotores, mas tem a grande vantagem de resolver a questão do fornecimento de dados, que passaria para o lado dos promotores, ainda que necessitasse de verificação. Com o aumento do número de utilizadores da aplicação, a necessidade de prospecção iria diminuir pois seriam os próprios promotores a ter interesse em juntarem-se à plataforma.

Numa segunda fase, para aumentar as fontes de rendimento, também podia existir uma versão premium para os utilizadores, que garantiria por exemplo notificações por mensagem, ou a reserva dos melhores lugares para assistir a um evento, etc. A hipótese de ter outro tipo de publicidade paga também não estaria descartada.

- **1.2 Instigadores do projecto**

Os instigadores ou partes interessadas na aplicação Bora Lá é qualquer pessoa que tenha vontade de descobrir os eventos que pode participar nas suas redondezas. Muitas vezes acabamos por não fazer nada, ou ficar aborrecidos por ter perdido alguma atividade que seria muito interessante para nós, por não saber o que está a acontecer. Os promotores do evento também seriam parte interessada para uma maior divulgação dos seus eventos ou espaços com objetivo de cativar mais clientes.

Outras partes interessadas seriam:

- Nós, como desenvolvedores e gestores deste projeto.
- Funcionários que sejam necessários contratar, como p.ex. comerciais.
- Empresas Locais: Negócios locais interessados em atrair clientes para eventos específicos.
- Turistas: Visitantes que desejam explorar eventos locais durante a sua estadia.
- Parceiros de Marketing: Colaboradores que podem promover a aplicação através de canais de marketing adicionais.
- Entidades Governamentais: Pode haver requisitos regulamentares ou de conformidade relacionados com a divulgação de eventos.
- Empresas de Tecnologia Externas: Fornecedores de serviços externos que podem estar envolvidos, como serviços de nuvem ou fornecedores de mapas.

- Associações Locais: Organizações comunitárias que podem beneficiar da promoção de eventos locais.
- Meios de Comunicação Locais: Jornais, rádios ou blogs locais que podem querer integrar ou destacar eventos.

○ **1.3 Objetivos do Projeto**

O objetivo principal deste projeto é o de aplicar e desenvolver conhecimentos na área do desenvolvimento de software, desde o back-end até ao front-end, e aplicar a metodologia Agile para agilizar este processo. Para tal, optamos pela criação de uma plataforma de descoberta de eventos nas redondezas de utilizadores, a que chamamos de Bora Lá. Uma vez que o projeto surge numa formação relacionada com tecnologias da Microsoft, todo o desenvolvimento assentou em tecnologias mais usadas em ambientes da Microsoft. Assim os objetivos mais específicos foram:

- Reuniões de brainstorming para definição das orientações gerais da plataforma;
- Estudo da implementação Agile mais adequada ao projeto (usamos a framework Scrum, com Sprints de uma semana);
- Levantamento de requisitos e definição do possível público alvo (estudo de “personas”);
- Estudo do aspecto visual da aplicação Web (criação de wireframes);
- Estudo dos dados e entidades necessárias, e criação de um esquema de relação de entidades;
- Criação e modelagem de uma base de dados em SQL Server;
- Inserção de dados na base de dados.
- Criação de uma API, Web Server REST em CORE.NET usando o Visual Studio e C# para programação;
- Criação de uma aplicação Web de página única, usando a framework Angular, com programação em HTML, CSS, JavaScript e TypeScript;
- Elaboração de relatório final;

2. Requisitos

Requisitos funcionais são as bases sobre as quais o desenvolvimento de um sistema é construído. São as necessidades e funcionalidades identificadas pelos stakeholders, expressas de forma clara e concisa, na forma de user stories por exemplo. Esses requisitos são caracterizados pela sua flexibilidade, estando sujeitos a ajustes ao longo do processo de desenvolvimento. Priorizados e entregues em incrementos menores ao longo de iterações, o principal propósito dos requisitos funcionais é assegurar que o sistema ou produto desenvolvido atende às necessidades e expectativas do utilizador final, fornecendo funcionalidades e características específicas que agregam valor à sua experiência de uso.

2.1. Personas

As personas são representações fictícias de diferentes tipos de utilizadores que interagem com o produto em desenvolvimento. Elas são criadas para oferecer à equipa de desenvolvimento uma compreensão mais profunda das necessidades, expectativas e comportamentos dos utilizadores finais. Ao desenvolver personas, a equipa pode explorar uma variedade de características demográficas, comportamentais e motivacionais dos utilizadores, bem como identificar os seus objetivos principais ao utilizar o produto. Essa compreensão detalhada das personas facilita a conceção de soluções que sejam verdadeiramente centradas no utilizador, garantindo que o produto final seja intuitivo, útil e capaz de satisfazer as necessidades específicas do público-alvo.

PERSONA 1

Tiago, o eterno aborrecido



*Idade: 27
Estado civil: Solteiro
Profissão: Comercial
Salário: 900€
Residência: partilha apartamento no centro de Santo Tirso.*

Estilo de vida:

O Tiago aborrece-se facilmente. Não é capaz de passar muito tempo em casa, e então gosta muito de festas, ou sair à noite, de forma a conviver e conhecer pessoas.

PERSONA 2

Maria, a mãe à procura de atividades infantis



*Idade: 35
Estado civil: Casada
Profissão: Advogada
Salário: 1200€
Residência: Casa própria no centro do Porto.*

Estilo de vida:

A Maria procura sempre que possível levar os filhos a eventos perto da sua residência. Gosta também da possibilidade de planear a ida a eventos em família num raio de 600km de casa, programando a conjugação das atividades pretendidas e das férias atempadamente.

PERSONA 3

António, o dono de um bar



Idade: 40

Estado civil: Casado

Profissão: Empresário

Salário: 1500€

Residência: Casa própria na periferia de Braga.

Estilo de vida:

O António dedica a maior parte do seu tempo ao crescimento do seu bar. Procura compreender o seu lugar no mercado para perceber quais as melhores formas de promoção de forma a cativar clientes para ampliar o seu negócio.

PERSONA 4

**Inês, a influencer
e promotora de eventos de stand-up comedy**



Idade: 25

Estado civil: Solteira

Profissão: Empresário

Salário: 1500€

Residência: Partilha apartamento no centro de Lisboa.

Estilo de vida:

A Inês é muito ativa socialmente e organiza eventos de stand-up com artistas que ainda são pouco conhecidos. Utiliza plataformas como o Facebook e o Instagram para promover estes eventos mas acha que são ferramentas um pouco limitadas para cativar os interessados nestes tipo de eventos.

2.2. User Stories

As User Stories representam uma abordagem fundamental no desenvolvimento de software, permitindo expressar requisitos de forma clara e concisa. São essenciais para definir as funcionalidades de um sistema a partir da perspetiva do utilizador que necessita delas. Cada User Story é uma descrição breve e direta do que o utilizador deseja alcançar (**objectivo**) e porquê (**razão**), sendo formulada em linguagem acessível e centrada nas necessidades do utilizador. Essa metodologia coloca o utilizador no centro do processo de desenvolvimento, proporcionando uma compreensão profunda das suas necessidades e das razões subjacentes, o que facilita a criação de soluções relevantes e eficazes.

USER STORY 1

Como **pessoa aborrecida numa sexta à noite**, quero consultar que eventos interessantes e gratuitos tenho perto de mim no momento, para me divertir.

É sexta à noite, e estou aborrecido. Quero fazer alguma coisa, talvez conviver, talvez assistir a algo, mas passei a semana a trabalhar e não tive tempo de andar a pesquisar o que iria acontecer na minha cidade. Pego no telemóvel e abro a APP Bora Lá, e vejo os vários eventos prestes a acontecer. Que aplicação fantástica, consigo definir o tipo de evento (música, poesia, até eventos gastronómicos), se é gratuito ou não, e até quanto perto de mim será. A minha dúvida agora é mesmo qual escolher, de tanta coisa fixe a acontecer que não fazia ideia.

USER STORY 2

Como **pessoa a visitar outra cidade**, quero consultar eventos, mesmo que pagos, num raio de X km, para poder passar um ótimo serão em família.

Planeei uma visita a Óbidos porque vi com antecedência que iriam ocorrer uma série de eventos que me interessavam durante esta semana. Experimentei aquela APP que o meu amigo falou, a Bora Lá, e bastou-me definir a localização e nos filtros selecionar Eventos Familiares que logo algumas opções foram sugeridas.

USER STORY 3

Como **dono de um bar**, quero inserir um evento no sistema, para averiguar o interesse que desponta nos utilizadores do mesmo.

Ao utilizar a aplicação Bora Lá, podemos facilmente introduzir informações sobre os eventos que estamos a planear organizar, permite-nos avaliar o interesse dos utilizadores e adaptar a programação de acordo com as preferências do público-alvo. Esta funcionalidade não só simplifica o processo de divulgação de eventos, mas também proporciona uma visão clara do impacto das iniciativas promocionais, ajudando a maximizar o retorno sobre o investimento em eventos.

USER STORY 4

Como **promotor de eventos**, quero compreender que tipo de evento atrai mais público, para rentabilizar o meu trabalho.

Através da utilização da aplicação Bora Lá, podemos realizar testes eficazes para determinar quais tipos de eventos são mais atrativos para o seu público-alvo. Ao analisar os dados e feedback dos utilizadores fornecidos pela aplicação, podemos ajustar a nossa estratégia de programação e escolher os eventos que têm maior probabilidade de gerar interesse e participação. Esta abordagem baseada em dados não só ajuda a otimizar os recursos e o orçamento como também a fortalece a relação entre os promotores de eventos e o seu público, criando experiências mais envolventes e gratificantes para todos os envolvidos.

USER STORY 5

Como **gestor de um teatro amador**, quero inserir a estreia da nossa nova peça no sistema, para que os utilizadores fiquem a saber da mesma.

Para a próxima semana gostava de dinamizar o meu pequeno bar, com uma noite de Stand Up Comedy. No entanto é sempre um investimento significativo para as habituais receitas do meu espaço. Se soubesse que teria uma boa lotação isso não seria problema, mas é difícil chegar às pessoas. Se forem como eu, muitas das

vezes só sabem do evento depois dele acontecer, vendo os stories do Instagram dos amigos. Acho que vou testar a APP Bora Lá, que noutro dia me ligaram a perguntar se não queria anunciar lá os nossos eventos de forma gratuita. Realmente foi muito fácil criar conta e introduzir a informação do evento. Tem havido muita gente a falar da APP, talvez seja aquilo que procurava para finalmente os nossos eventos começarem a esgotar sempre.

2.3. Requisitos Funcionais

Existem dois tipos de utilizadores: o Visitante e o Promotor.

User:

- deve poder ter a opção de entrar sem login;
- deve poder editar um perfil (foto, nome, interesses);
- deve poder pesquisar os eventos ativos;
- deve poder avaliar e classificar o espaço, promotor, evento;
- deve poder consultar reviews de eventos passados;
- deve poder visualizar uma lista dos eventos ativos;
- deve poder seleccionar um evento para consultar informações detalhadas (data, hora, local, preço);
- deve poder filtrar a pesquisa ou a lista dos eventos por nome do evento, descrição, género, local, data, hora, (distância), nome do Promotor;
- deve poder visualizar os eventos ativos no mapa (interactivo);
- pode indicar preferências de categorias de eventos.

Promotor:

- deve poder inserir / editar / eliminar eventos;
- deve poder catalogar os eventos por nome do evento, tipo de evento, Data/hora do evento, local do evento, nome do Promotor, nome do local do evento, para facilitar as filtragens;
- deve poder registrar espaços e suas características (p.e. capacidade espaço);

- caso seja responsável por um espaço, deve poder associar-se ao mesmo no registo;
- deve poder consultar as reviews de um dado evento e responder às reviews;
- deve poder personalizar a página do Espaço (versão paga);
- pode efetuar um pagamento para usufruir de vantagens consoante uma categoria: Premium, Gold,

Requisitos gerais (todos os utilizadores):

- devem poder registar-se;
- devem poder fazer login através de conta Google, Apple ou Facebook (o login cria um perfil base).

Requisitos de sistema:

- deve enviar notificações consoante as preferências do utilizador;
- deve poder enviar notificações quando os eventos são actualizados/modificados;
- deve poder implementar sistema de controlo de acessos e autorizações (p.e. autorização para registar / editar / eliminar um evento);
- deve existir a possibilidade de denúncia e registo de erros;
- pode definir a frequência de envio de notificações ao utilizadores sobre um determinado evento e de acordo com a categoria de Promotor.
- verificar se um espaço já existe através da inserção de coordenadas/morada/outro.

Requisitos do administrador do sistema:

- deve ser capaz de alterar qualquer informação/estrutura/outro do site/aplicação;
- deve ser o único capaz de banir utilizadores ou promotores;
- deve ser capaz de rever denúncias.

2.4. Requisitos Não Funcionais

Língua:

Português de Portugal.

Plataforma:

- Web App;
- deve ser compatível com dispositivos móveis.

Desempenho:

- interface rápida e responsiva apresentação de resultados em menos de 2 segundos;
- interface deve ser intuitiva e fácil de navegar.

Segurança:

- proteger os dados dos utilizadores e cumprir com o RGPD.

Escalabilidade:

- app deve ser escalável para lidar com o aumento do número de eventos e utilizadores.

2.5. Prioritização de requisitos (técnica MoSCoW)

Quando enfrentamos a impossibilidade de implementar todos os requisitos acordados de uma só vez, é indispensável estabelecer prioridades sobre quais são as funcionalidades mais cruciais para a primeira versão do produto. Para isso, utilizamos a técnica de priorização "**MoSCoW**", para identificar e ordenar os requisitos, desde os mais cruciais até aos menos relevantes.

Must have

- eventos que se vão realizar, organizados por local, hora e tipo;
- espaços associados aos eventos;
- registo dos utilizadores e promotores.

Should have

- validação da utilização do Espaço e validação do Evento;
- os utilizadores deveriam poder indicar preferências;
- os utilizadores deveriam poder ter a opção de receber notificações sobre eventos próximos;
- os utilizadores deveriam poder ter a opção de receber notificações personalizadas com base nas suas preferências;
- integração de Mapa: a aplicação deveria poder ter integrado um mapa para mostrar a localização exata dos eventos;
- monetização através do pagamento de mensalidades por parte dos promotores para terem maior visibilidade.

Could have

- terceiro utilizador - Space Owner;
- dividir os Promotores em Promotor de Eventos e Responsável pelo Espaço;
- partilha de eventos nas redes sociais;
- avaliação/feedback de eventos;
- sistema deve poder gerir reservas e compra de bilhetes;
- permitir a utilização a pessoas com necessidades especiais (p.e. inviduais).
- integração com Mapa: a navegação para o local do evento poderia ser facilitada;
- a monetização através de publicidade não será já implementada.

Won't have

- Venda de bilhetes para grandes eventos;
- Promotores de grandes eventos não terão direito a categorias pagas para não prejudicar a visibilidade dos eventos mais pequenos.

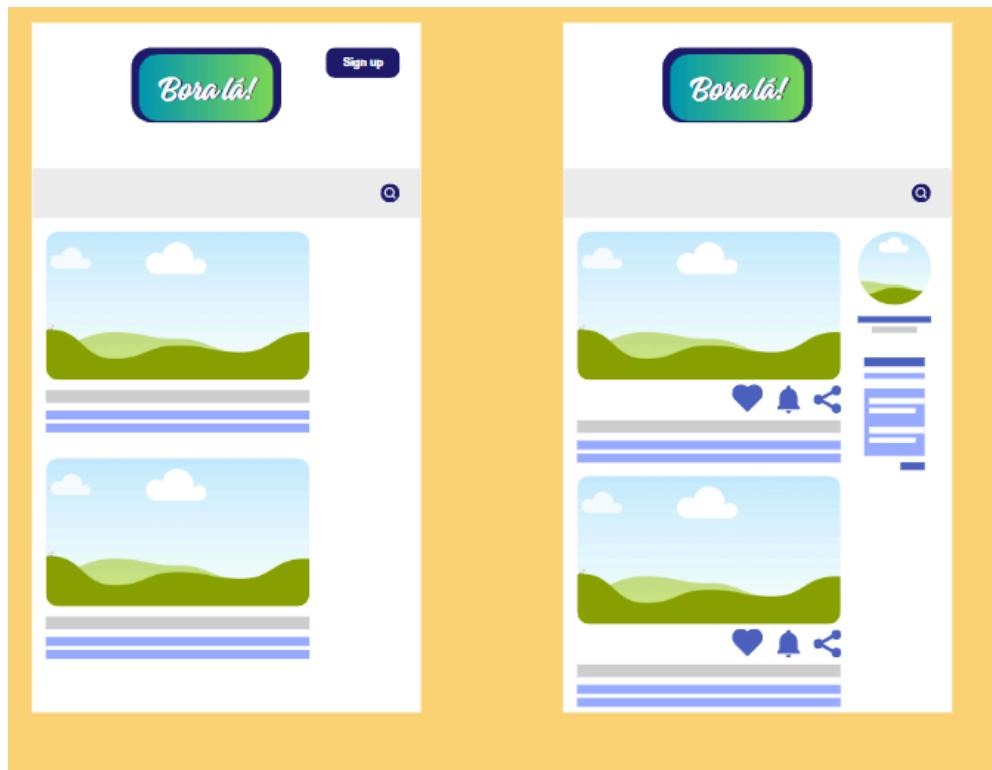
2.6. Wireframes

Os wireframes procuram representar o layout base do projeto de web design, sem detalhes estéticos. São criados na fase inicial do processo para facilitar a idealização da organização do conteúdo e a navegação. Permitem aos designers e aos developers entenderem a hierarquia da informação e testarem a usabilidade do projeto antes de avançarem no projecto. Estes esboços são fundamentais para comunicar as ideias entre as equipas, garantindo uma base sólida para o desenvolvimento de interfaces intuitivas e eficazes.

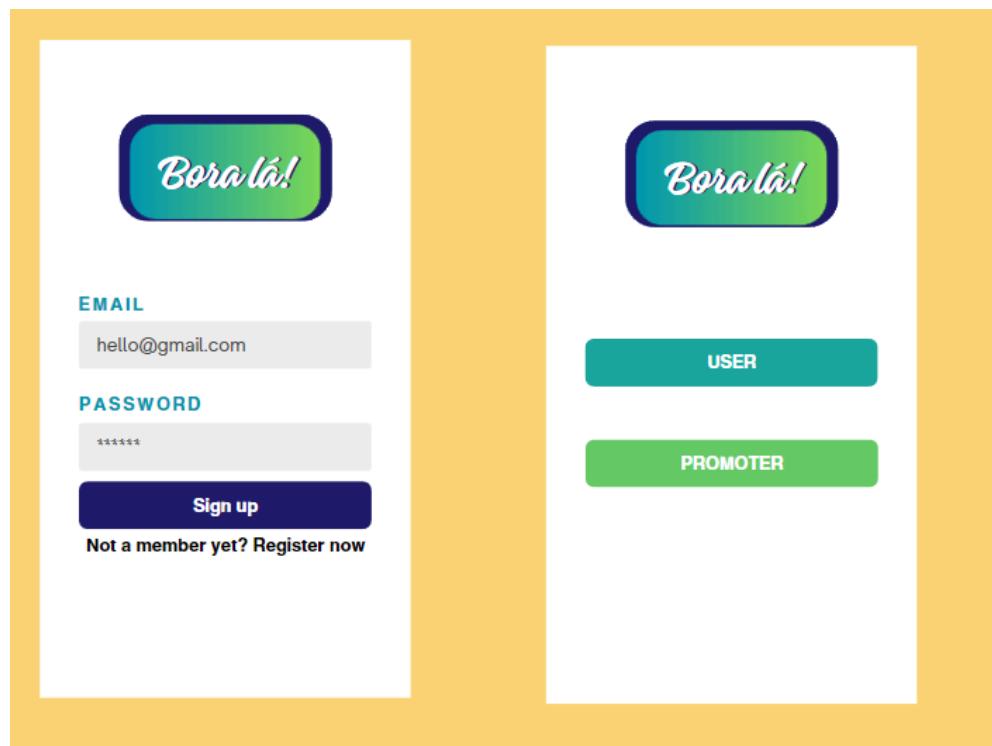
De seguida, apresentamos as wireframes base idealizadas para a app Bora Lá. Existirá um menu principal onde já poderão ser visualizados os Eventos e respectivos Espaços. Caso o utilizador pretenda obter mais detalhes ou guardar Favoritos, terá que fazer login, assim como o Promotor, caso pretenda obter mais informações como feedback dos utilizadores ou adicionar, editar ou eliminar Eventos, deverá fazer igualmente login.



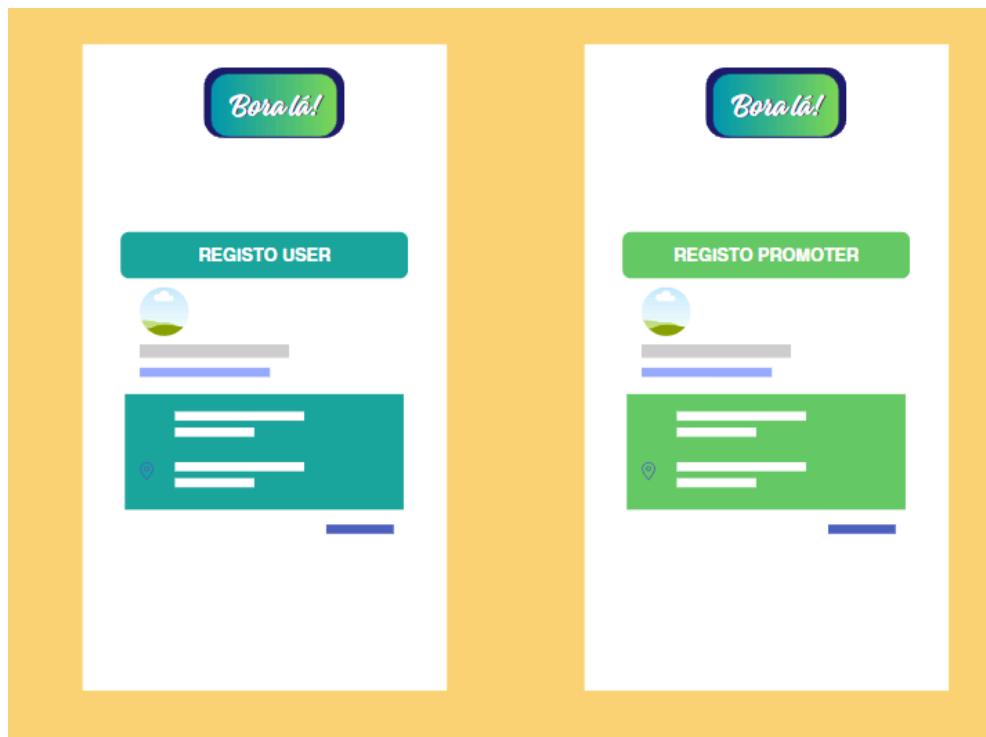
Wireframes: Vista geral da estrutura da app



Wireframes: Página Principal e Página após Login (com menu lateral do Utilizador ou do Promotor)



Wireframes: Menu de Login e Menu de escolha entre Utilizador e Promotor



Wireframes: Menu de Registo do Utilizador e Menu de Registo do Promotor

3. Metodologias e Tecnologias de Trabalho

3.1. A Nossa Metodologia de Trabalho

3.1.1. Agile

Ao longo do tempo, as metodologias de trabalho maioritariamente utilizadas em desenvolvimento de software têm vindo a evoluir de metodologias mais rígidas e “tradicionais” para metodologia mais flexíveis ou dinâmicas, facilmente adaptáveis.

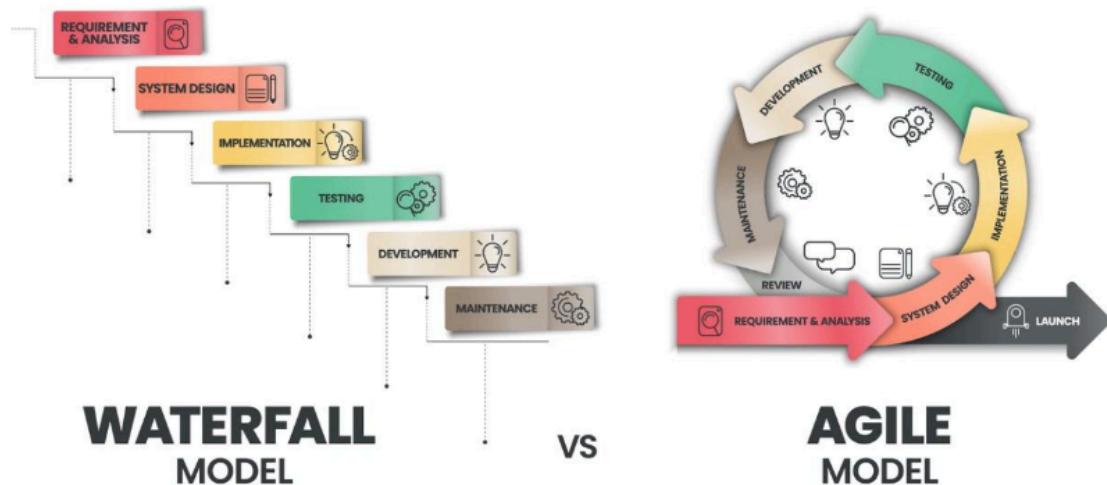
Dois excelentes exemplos disto são as metodologias Waterfall e Agile.

A metodologia Waterfall é linear e sequencial, dividindo o desenvolvimento em fases: conceção, design, implementação, testes e manutenção. Cada fase depende do sucesso da anterior, e apenas é lançada a app no final deste processo.

Tudo isto torna-a uma metodologia ineficaz no novo mundo do desenvolvimento de software, onde é necessária rapidez, agilidade e adaptação constantes. Naturalmente, esta metodologia tem vindo a ser substituída pela metodologia Agile.

A metodologia Agile encontra-se em direta oposição à anterior, enfatizando iterações curtas e colaboração próxima entre a equipa, permitindo uma resposta ágil às mudanças nos requisitos. O foco está na entrega contínua de valor ao cliente, com revisões regulares e adaptação do plano. A cultura de melhoria contínua e comunicação eficaz são pilares do Agile, garantindo a entrega consistente de software de alta qualidade.

A próxima figura ilustra a imensa diferença entre estas duas metodologias:



Metodologia Waterfall VS Metodologia Agile

Não queremos, com isto, dizer que se deve escolher sempre Agile em vez de Waterfall: deve-se ter em conta na decisão o tipo de projeto e até mesmo o produto que se pretende desenvolver.

Após esta análise, o grupo decidiu avançar com a utilização da metodologia Agile no nosso projeto; mais concretamente, o projeto foi desenvolvido na sua totalidade com base no **Scrum**.

3.1.1.1. Scrum

O **Scrum** é um framework de gestão de projetos que facilita a aplicação dos princípios da metodologia Agile, sendo amplamente adotado devido à sua estrutura organizada para as equipas gerirem o seu trabalho. Esta framework assenta em diversas reuniões, ferramentas e papéis que facilitam e promovem a gestão eficiente do projeto.

Uma das principais características do **Scrum** é sua abordagem iterativa e incremental. O projeto é dividido em iterações chamadas Sprints, que normalmente têm curta duração. Durante cada Sprint, a equipa concentra-se em desenvolver as funcionalidades mais importantes, visando entregar um produto potencialmente utilizável e valioso.

A colaboração contínua com o cliente é encorajada, permitindo que a equipa receba feedback valioso e faça ajustes conforme necessário. Assim, o **Scrum** prioriza a adaptação às mudanças, em vez de aderir rigidamente a planos e previsões.

Tudo isto faz com que, ao contrário das metodologias tradicionais, seja possível entregar ou demonstrar ao cliente uma versão completamente operacional e potencialmente pronta a ser lançada do produto requisitado no final de cada Sprint.

3.1.1.2. Funções dos Membros da Equipa

Como mencionado anteriormente, existem diversos papéis ou funções desempenhadas pelos elementos de uma equipa que utilize o **Scrum**. Os principais são: Product Owner, Scrum Master e Scrum/Development Team.

Todos os elementos do grupo desempenharam e encarnaram alguns aspectos de todos estes papéis que iremos aprofundar a seguir. Mas, na tentativa de aproximar o nosso projeto o mais possível da realidade do mercado de trabalho, havia um elemento responsável por cada papel.

Product Owner

O Product Owner é um papel fundamental no framework **Scrum**. É responsável por representar os interesses e as necessidades do cliente ou do utilizador final dentro da equipa de desenvolvimento. O Product Owner desempenha um papel crucial na definição das funcionalidades e das prioridades do produto, garantindo que o trabalho da equipa esteja alinhado com as expectativas do cliente e com os objetivos do negócio.

As principais responsabilidades do Product Owner incluem:

1. **Definir e gerir o backlog do produto:** O backlog do produto é uma lista priorizada de todas as funcionalidades, melhorias e correções de bugs necessárias para o produto. O Product Owner é responsável por elaborar e manter este backlog, garantindo que as funcionalidades mais importantes e de maior valor para o cliente sejam desenvolvidas primeiro.
2. **Priorizar as tarefas:** Com base no feedback do cliente, nas necessidades do negócio e outros fatores relevantes, o Product Owner define a ordem de prioridade das tarefas no backlog do produto. Isto orienta o trabalho da equipa de desenvolvimento, garantindo que se concentram os esforços nas funcionalidades mais cruciais e valiosas.
3. **Participar ativamente nas reuniões do Scrum:** O Product Owner colabora de perto com a equipa de desenvolvimento, participando nas reuniões com o resto da equipa. Ele fornece orientações e esclarecimentos sobre os requisitos do produto, esclarecendo dúvidas e ajustando as prioridades consoante necessário.
4. **Tomar decisões sobre o produto:** O Product Owner é autorizado a tomar decisões finais sobre o produto, incluindo a aceitação ou rejeição do trabalho realizado pela equipa de desenvolvimento. Ele é o ponto de contacto principal para questões relacionadas com o produto, garantindo que a visão do cliente é sempre tida em conta durante o processo de desenvolvimento.

Em resumo, o Product Owner desempenha um papel crucial na garantia do sucesso do projeto **Scrum**, representando os interesses do cliente e fornecendo direção e orientação à equipa de desenvolvimento ao longo do ciclo de vida do produto.

No nosso projeto, o cargo de Product Owner ficou à responsabilidade do José.

Scrum Master

O Scrum Master é um papel vital no framework **Scrum**, responsável por facilitar a implementação e a prática do **Scrum** dentro da equipa de desenvolvimento. O Scrum Master atua como um líder de serviço, trabalhando para remover obstáculos, facilitar a colaboração e promover uma cultura de melhoria contínua.

As principais responsabilidades do Scrum Master incluem:

1. **Facilitar reuniões:** O Scrum Master lidera as reuniões do **Scrum**, como o planeamento de sprint, as revisões de sprint e as reuniões diárias, garantindo que sejam realizadas de forma eficaz e produtiva. Ele ajuda a equipa a manter o foco nos objetivos do sprint e a resolver quaisquer impedimentos que possam surgir.
2. **Remover obstáculos:** O Scrum Master trabalha para identificar e eliminar quaisquer obstáculos que estejam a impedir o progresso da equipa. Isso pode envolver resolver problemas de comunicação, fornecer recursos adicionais ou interceder em conflitos internos.
3. **Proteger a equipa:** O Scrum Master atua como um escudo protetor para a equipa, protegendo-a de distrações externas e pressões excessivas. Ele ajuda a equipa a manter um ritmo sustentável de trabalho, evitando o excesso de carga de trabalho e o esgotamento.
4. **Fomentar a melhoria contínua:** O Scrum Master promove uma cultura de melhoria contínua, incentivando a equipa a refletir sobre seu processo de trabalho e identificar maneiras de se tornar mais eficiente e eficaz. Ele facilita sessões de retrospectiva de sprint, onde a equipa pode analisar o que correu bem, o que correu mal e como podem melhorar no futuro.
5. **Educar e orientar:** O Scrum Master atua como um mentor e treinador para a equipa de desenvolvimento, ajudando-os a entender e a implementar os princípios e práticas do **Scrum**. Ele fornece orientação sobre como aplicar o **Scrum** de forma eficaz e como resolver quaisquer desafios ou questões que possam surgir.

Em resumo, o Scrum Master desempenha um papel essencial na facilitação do processo **Scrum**, garantindo que a equipa de desenvolvimento tenha as condições necessárias para alcançar os seus objetivos e entregar valor de forma consistente ao cliente. Ele é um facilitador, mentor e defensor da equipa, trabalhando para maximizar sua eficácia e sucesso.

No nosso projeto, o cargo de Scrum Master ficou entregue ao Luís.

Scrum / Development Team

A Equipa de Desenvolvimento, ou Scrum Team, é composta por profissionais multifuncionais que são responsáveis por transformar os itens do backlog do produto em incrementos potencialmente entregáveis de funcionalidades ou produtos no final de cada sprint.

Algumas características e responsabilidades da Equipa de Desenvolvimento incluem:

1. **Multifuncionalidade:** A Equipa de Desenvolvimento é composta por indivíduos com habilidades diversas indispensáveis à completa realização do trabalho necessário para entregar as funcionalidades. Isto inclui (entre outros) programadores, designers, testers e analistas.
2. **Auto-organização:** A Equipa de Desenvolvimento é auto-organizada, o que significa que eles têm a autoridade e autonomia para decidir como realizar o trabalho e alcançar os objetivos do sprint. Eles organizam-se e gerem o seu tempo para cumprir as metas estabelecidas pelo Product Owner.
3. **Colaboração:** A colaboração é fundamental dentro da Equipa de Desenvolvimento. Eles trabalham juntos para resolver problemas, partilhar conhecimento e garantir que as funcionalidades entregues atendam aos padrões de qualidade e aos requisitos do cliente.
4. **Responsabilidade pela entrega:** A Equipa de Desenvolvimento é responsável por entregar os incrementos de funcionalidade ao final de cada

sprint. Eles trabalham para alcançar as metas do sprint e garantir que o produto esteja num estado potencialmente entregável no fim de cada iteração.

5. **Foco na melhoria contínua:** A Equipa de Desenvolvimento está sempre à procura de maneiras de melhorar seu processo de trabalho e aumentar a sua eficiência. Eles participam ativamente nas reuniões de retrospectiva de sprint para identificar o que funcionou bem, o que não funcionou e como podem melhorar no futuro.

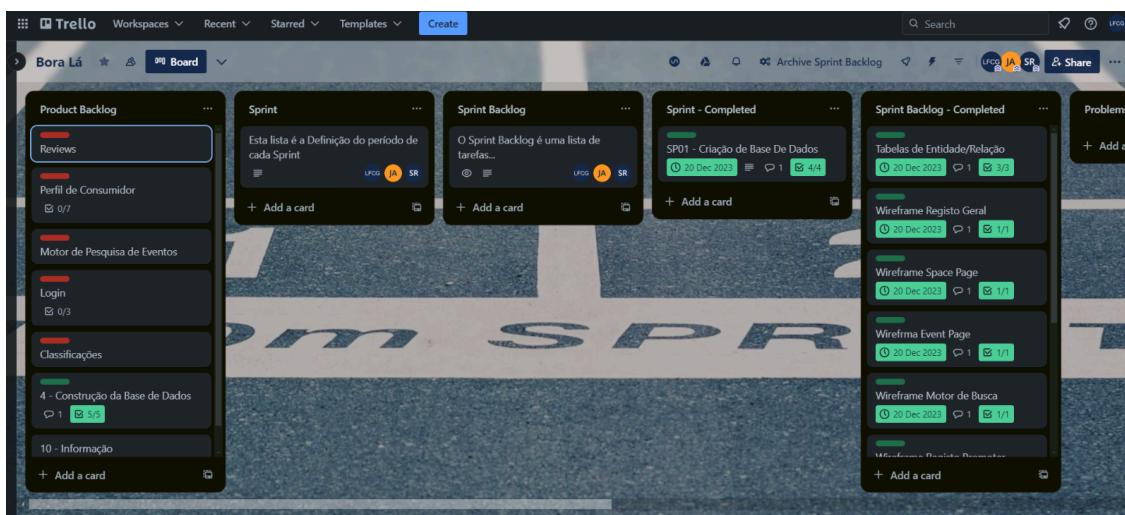
Em resumo, a Equipa de Desenvolvimento é o grupo de profissionais responsável por transformar os requisitos do produto em incrementos de funcionalidade entregáveis, trabalhando de forma colaborativa, auto-organizada e focada na entrega de valor ao cliente.

No nosso projeto, o cargo de Scrum Team ficou à responsabilidade da Sara, como senior developer; deve mencionar-se que este foi o papel em que mais nos interiorizamos e participamos como grupo.

3.1.1.3. Ferramentas do Scrum

O Scrum, por causa da sua natureza, tem uma implementação muito visível e alargada. É muito natural as equipas recorrerem a boards (tanto online como offline), porque este tipo de visualização é muito útil a dispor de forma organizada as diversas ferramentas do framework.

Na figura abaixo, apresentamos uma imagem do board utilizado pelo nosso grupo para implementar este framework ao nosso projeto:



Trello

Product Backlog

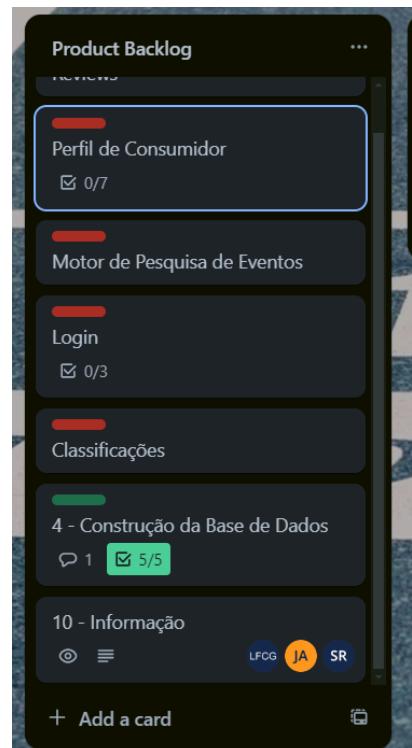
O Product Backlog é uma ferramenta central no Scrum, representando uma lista dinâmica e priorizada de todas as necessidades, requisitos e melhorias que devem ser desenvolvidos para atingir os objetivos do negócio. Esta lista é criada e gerida pelo Product Owner em colaboração com os Stakeholders/Clientes, que fornecem feedback/opiniões e percepções/insights sobre as suas necessidades.

Essa lista está sempre sujeita a alterações à medida que o projeto avança e novas informações são adquiridas. O Product Owner pode adicionar, modificar ou remover itens do backlog para refletir as mudanças nas prioridades do negócio ou

no feedback recebido. Essas mudanças podem ocorrer durante reuniões com a equipa ou como resultado do feedback dos Stakeholders.

O Product Backlog é fundamental para manter a equipa focada nas funcionalidades mais importantes e que adicionam valor ao produto. Ele fornece uma visão clara das necessidades do negócio e serve como um guia para a equipa de desenvolvimento ao longo de todo o projeto. Em resumo, é uma ferramenta essencial para garantir que o trabalho da equipa esteja alinhado com os objetivos do negócio e as necessidades dos utilizadores finais.

Na figura à direita conseguimos visualizar precisamente esta ordem de prioridade referida.



Sprint Backlog

O Sprint Backlog é uma lista de itens do Product Backlog que a equipa de desenvolvimento seleciona para trabalhar durante um sprint específico. Esta lista é detalhada e específica, incluindo tarefas, atividades e funcionalidades que serão desenvolvidas durante o sprint.

Após o planeamento do sprint, a equipa colabora para decidir quais itens do Product Backlog serão incluídos no Sprint Backlog, levando em consideração a capacidade da equipa, as prioridades definidas pelo Product Owner e o objetivo do sprint.

No final do sprint, todas as tarefas do Sprint Backlog devem ser concluídas, resultando em melhorias funcionais e potencialmente entregáveis no projeto. Se houver itens não concluídos, eles são devolvidos ao Product Backlog para serem reavaliados e priorizados em futuros sprints.

Na figura abaixo podemos visualizar os itens completos de um Sprint Backlog (ou, por outras palavras, o incremento).



Incremento

O Incremento é o resultado tangível e funcional de cada ciclo de trabalho no Scrum, conhecido como Sprint. É o produto em desenvolvimento que é constantemente atualizado e aprimorado ao longo das iterações. A cada Sprint, a equipa de desenvolvimento trabalha para agregar valor ao produto, implementando novas funcionalidades, corrigindo defeitos ou fazendo melhorias.

Este Incremento é um marco significativo no processo iterativo e de aperfeiçoamento do projeto, pois proporciona valor real aos Stakeholders e permite que eles acompanhem o progresso do produto. A equipa empenha-se em garantir que o Incremento seja de alta qualidade, atenda aos critérios de aceitação definidos e esteja pronto para ser entregue e utilizado pelo cliente.

Com a conclusão de cada Sprint, o Incremento acumula as mudanças e melhorias feitas ao longo do tempo, refletindo o progresso contínuo do projeto. Essa abordagem iterativa permite uma evolução gradual e adaptativa do produto, à medida que as necessidades do cliente e do mercado são melhor compreendidas e atendidas.

3.1.1.4. Fases de um Sprint

Como tem vindo a ser mencionado, um Sprint possui várias fases. As fases de um sprint podem variar ligeiramente dependendo da implementação específica do Scrum em cada contexto, mas geralmente seguem um processo semelhante, composto por:

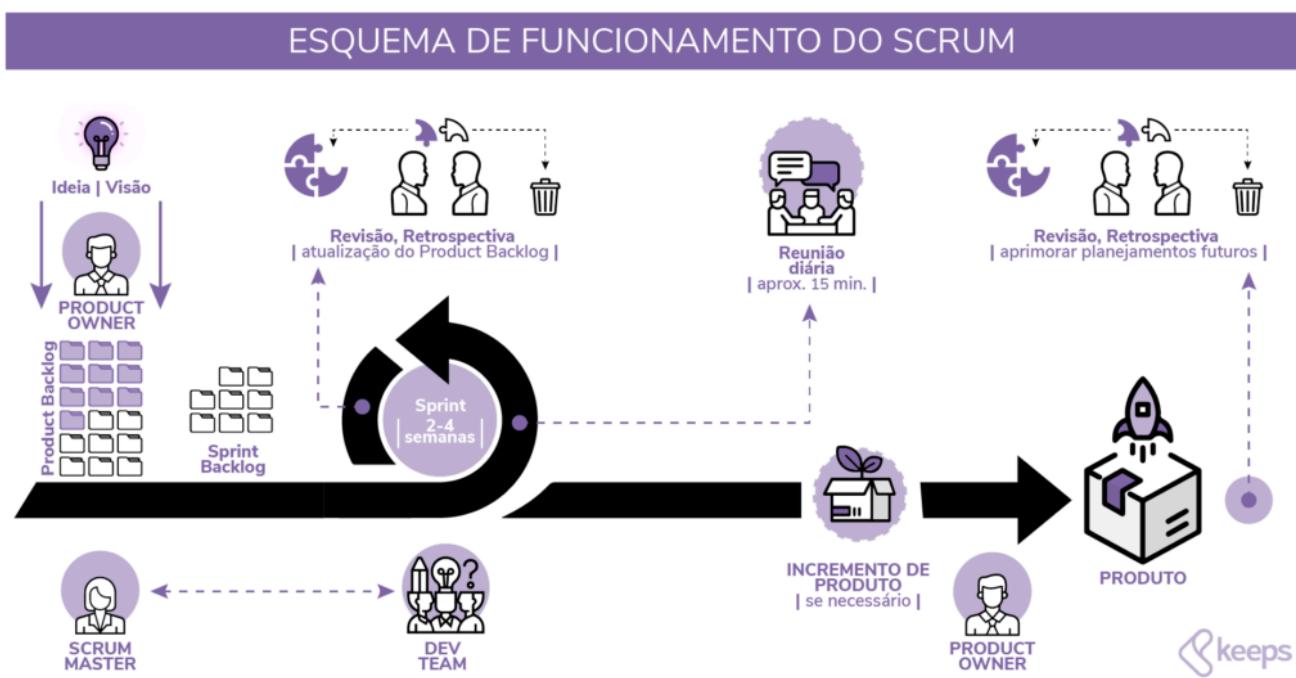
1. **Planeamento do Sprint:** Esta é a primeira fase do sprint, onde a equipa de desenvolvimento e o Product Owner se reúnem para definir os objetivos do sprint e selecionar as funcionalidades a serem desenvolvidas. Durante esta reunião, a equipa estima o esforço necessário para concluir cada funcionalidade e compromete-se a entregar um conjunto específico de tarefas no final do sprint.
2. **Desenvolvimento:** Após o planeamento do sprint, a equipa de desenvolvimento começa a trabalhar nas tarefas definidas no sprint backlog. Eles colaboram para implementar as funcionalidades selecionadas, seguindo as práticas de desenvolvimento ágil, como a programação em pares e a integração contínua. Durante esta fase, é importante que a equipa mantenha uma comunicação próxima e transparente, atualizando-se regularmente sobre o progresso e identificando quaisquer obstáculos que possam surgir.
3. **Revisões diárias:** Durante todo o sprint, a equipa realiza reuniões diárias curtas, conhecidas como reuniões diárias ou stand-ups, para sincronizar o trabalho e discutir o progresso e os obstáculos. Estas reuniões são uma oportunidade para a equipa identificar quaisquer problemas ou desafios que possam estar a enfrentar e colaborar na resolução deles.
4. **Revisão do Sprint:** No final do sprint, a equipa de desenvolvimento realiza uma revisão do sprint, onde demonstra as funcionalidades desenvolvidas ao

Product Owner e outros Stakeholders. Durante esta revisão, o Product Owner fornece feedback sobre o trabalho realizado e verifica se as funcionalidades atendem aos critérios de aceitação definidos. Esta é também uma oportunidade para a equipa refletir sobre o seu desempenho e identificar áreas de melhoria para futuros sprints.

5. **Retrospetiva do Sprint:** Após a revisão do sprint, a equipa realiza uma retrospetiva do sprint, onde reflete sobre o seu processo de trabalho e identifica o que correu bem, o que poderia ser melhorado e quais ações podem ser tomadas para melhorar no futuro. Esta é uma oportunidade crucial para a equipa aprender e adaptar-se continuamente, promovendo uma cultura de melhoria contínua.

Estas são as principais fases de um sprint no Scrum, cada uma desempenhando um papel importante no processo de desenvolvimento ágil e na entrega de valor ao cliente de forma iterativa e incremental.

A figura abaixo ilustra este processo de uma forma fácil de seguir.



3.2. Tecnologias de Trabalho

O nosso projeto recorreu a diversas ferramentas, tecnologias e programas. Segue-se agora, antes da apresentação do projeto propriamente dito, uma breve descrição do processo de trabalho em cada uma dessas ferramentas.

3.2.1. Desenvolvimento da App em Angular

A web app BoraLá foi desenvolvida com recurso à framework Angular. Esta foi a framework escolhida porque permite gerar uma interface muito interativa, através do recurso a métodos como data binding (vinculação de dados), definição de rotas entre elementos do projeto (routing), criação de serviços, entre outros.

A vinculação de dados permite que a interface do utilizador seja atualizada automaticamente com base nos dados armazenados no back-end, proporcionando uma experiência de utilizador mais fluida. Além disso, o Angular organiza o código de forma modular, facilitando a manutenção e a escalabilidade da aplicação.

O uso do Angular CLI, ou Command Line Interface, simplifica o processo de desenvolvimento, permitindo a criação rápida e eficiente de novos componentes, serviços e outros artefactos. Na estrutura Angular, a camada de apresentação é baseada em componentes, onde cada componente representa uma parte específica da interface do utilizador.

Cada componente é composto por um arquivo TypeScript, que contém a lógica do componente, e um arquivo HTML, que define a estrutura visual. O estilo do componente pode ser personalizado utilizando arquivos CSS. Essa abordagem modular e orientada a componentes facilita a criação, organização e manutenção de aplicações web complexos.

3.2.1.1 HTML, CSS e Typescript

No contexto de Angular, HTML, CSS e TypeScript desempenham papéis distintos e complementares:

1. **HTML (HyperText Markup Language):** No contexto do Angular, o HTML é fundamental para a construção da interface do utilizador (UI) das aplicações. É por meio do HTML que são definidos os elementos visuais, como botões, formulários, imagens e texto, e sua estruturação na página. No Angular, os arquivos HTML são usados para criar os moldes dos componentes, onde são definidos os elementos da interface e a forma como eles são apresentados ao utilizador. Além disso, o Angular introduz uma série de diretivas e interpolação de dados diretamente no HTML, permitindo a criação de interfaces dinâmicas e interativas.
2. **CSS (Cascading Style Sheets):** O CSS desempenha um papel crucial no Angular ao permitir a estilização dos elementos HTML. Por meio do CSS, é possível definir a aparência visual dos elementos, incluindo cores, fontes, tamanhos, margens e espaçamentos. Isso possibilita a criação de interfaces atraentes e intuitivas para os utilizadores. No Angular, os estilos CSS podem ser aplicados diretamente aos elementos HTML nos templates dos componentes ou de forma mais global, afetando a aparência de toda a aplicação.
3. **TypeScript:** TypeScript é a linguagem de programação principal utilizada no Angular. É um superset de JavaScript que adiciona recursos avançados, como tipagem estática, interfaces, classes e módulos, ao JavaScript padrão. No Angular, o TypeScript é usado para escrever a lógica dos componentes, serviços e outros artefactos da aplicação. Ele oferece vantagens significativas, como verificação de tipos em tempo de compilação, que ajudam a reduzir erros e melhorar a manutenção do código. Além disso, o TypeScript fornece uma experiência de desenvolvimento mais rica.

Em conjunto, HTML, CSS e TypeScript formam os pilares do desenvolvimento de aplicações web modernos com Angular, possibilitando a criação de interfaces

ricas, dinâmicas e responsivas, além de oferecer uma experiência de desenvolvimento produtiva e eficiente. Parte superior do formulário

3.2.1.2 Serviços

Nas aplicações Angular, os serviços são como "caixas de ferramentas" que encapsulam a lógica de negócios e operações não relacionadas diretamente com a interface do utilizador. Eles desempenham um papel crucial na partilha de dados e funcionalidades entre diferentes componentes, além de fornecer uma maneira de interagir com APIs, que, por sua vez, se comunicam com o backend e a base de dados.

Para se comunicar com a API, o Angular utiliza requisições HTTP. Para simplificar e tornar esse processo mais eficiente, o Angular oferece o módulo HttpClient. Esse módulo permite fazer chamadas HTTP de forma fácil e assíncrona, facilitando a interação com APIs externas e a obtenção e envio de dados entre o frontend e o backend da aplicação.

Em suma, os serviços em Angular são essenciais para organizar a lógica de negócios, compartilhar dados e funcionalidades entre os diferentes componentes e facilitar a comunicação com APIs externas para acesso a dados e recursos do servidor.

3.2.1.3 Routing e Navegação

No Angular, o roteamento e a navegação são responsáveis por gerenciar a transição entre as diferentes partes de uma aplicação web, permitindo que os utilizadores naveguem entre páginas ou componentes de forma fluida e eficiente.

O roteamento refere-se à definição de rotas ou URLs para diferentes partes da aplicação. Isso é feito utilizando o módulo de roteamento do Angular, onde são configuradas as rotas correspondentes a cada componente da aplicação. Por exemplo, uma rota pode ser configurada para direcionar os utilizadores para a página inicial, outra rota para um formulário de contato e assim por diante.

Já a navegação está relacionada à transição entre essas rotas ou URLs. O Angular fornece diretivas, como o **router-outlet**, para exibir o conteúdo correspondente à rota atual na interface do utilizador. Além disso, ele oferece métodos de navegação programática, como **routerLink** ou o serviço **Router**, que permitem que os utilizadores naveguem entre as diferentes partes da aplicação clicando em links ou botões, ou através de interações programáticas.

Em resumo, o roteamento e a navegação no Angular são essenciais para criar aplicações web com múltiplas páginas ou estados, permitindo que os utilizadores naveguem entre elas de forma intuitiva e eficiente, seja por meio de URLs ou de interações na interface do utilizador.

3.2.1.4 **Formulários e Validação**

No Angular, a criação e validação de formulários na interface do utilizador são suportadas de forma abrangente. Existem dois principais tipos de formulários disponíveis: os Template-Driven Forms (Formulários Dirigidos por Modelo) e os Reactive Forms (Formulários Reativos).

Os formulários template-driven são mais simples de implementar e são definidos diretamente no template HTML do componente. A validação é especificada utilizando diretivas do Angular, como required, minLength, maxLength, entre outras. Além disso, os dados do formulário podem ser acessados utilizando two-way data binding, tornando a interação com o formulário direta e fácil de entender.

Por outro lado, os formulários reativos oferecem mais flexibilidade e são controlados programaticamente. Eles são criados utilizando a API de formulários reativos do Angular e são mais adequados para cenários complexos ou que exigem um maior controlo sobre a validação e manipulação de dados. Com os formulários reativos, é possível definir validações customizadas, reagir a eventos de forma mais granular e realizar operações assíncronas com os dados do formulário.

Em resumo, o Angular oferece opções flexíveis para a criação e validação de formulários, permitindo aos desenvolvedores escolherem a abordagem mais adequada para cada situação específica. Seja utilizando formulários template-driven para casos simples e diretos, ou formulários reativos para cenários mais complexos e exigentes, o Angular oferece ferramentas poderosas para lidar com todas as necessidades de formulários na interface do utilizador.

3.2.1.5 **Backend – Comunicação e Troca de Informação**

o Angular, a comunicação entre a aplicação frontend e a API é essencial para que a aplicação funcione corretamente. O Angular utiliza o HttpClient para enviar solicitações HTTP ao servidor e receber as respostas correspondentes.

Essas solicitações podem conter informações importantes, como cabeçalhos de autenticação ou tokens de segurança, para autenticar o utilizador e aceder os dados necessários no backend. As respostas do servidor são tratadas de forma assíncrona, permitindo que os componentes do Angular atualizem dinamicamente a interface do utilizador com base nas informações recebidas.

Essa interação entre o frontend e o backend possibilita que a aplicação aceda, manipule e apresente dados de forma eficiente e responsiva. Com o Angular, os desenvolvedores podem criar aplicações web modernos e poderosos, que atendem às necessidades dos utilizadores e oferecem uma experiência de utilização de alta qualidade.

Em resumo, o Angular capacita os desenvolvedores a criar aplicações interativas, utilizando uma variedade de recursos, como componentes, templates, serviços, rotas de navegação, formulários e comunicação com o backend, para fornecer uma experiência de utilização rica e envolvente.

3.2.1.6 Bootstrap

O Bootstrap é uma biblioteca open source de CSS que simplifica o desenvolvimento da interface de utilizador, oferecendo uma ampla gama de componentes e estilos pré-criados. Esta framework traz diversos benefícios:

1. **Responsividade:** O Bootstrap é conhecido por sua capacidade de adaptação a diferentes tamanhos de ecrã, garantindo uma experiência de utilizador consistente em dispositivos desktops, tablets e smartphones.
2. **Grid System:** Ele utiliza um sistema de grelha que facilita a criação de layouts de página responsivos, permitindo uma organização estruturada e flexível do conteúdo, independentemente do tamanho do ecrã.
3. **Componentes Prontos para Uso:** O Bootstrap oferece uma vasta gama de componentes reutilizáveis, como botões, formulários, barras de navegação, alertas, entre outros. Esses componentes são estilizados e prontos para uso imediato, acelerando o desenvolvimento da interface de utilizador e garantindo uma aparência consistente em toda a aplicação.
4. **Estilos e Temas Personalizáveis:** O Bootstrap fornece estilos pré-definidos que podem ser facilmente personalizados para atender às necessidades visuais específicas de cada aplicação.
5. **Facilidade de Integração:** Ele pode ser facilmente integrado com outras bibliotecas e frameworks, como o Angular. Isso permite utilizar classes CSS do Bootstrap em componentes Angular, simplificando o processo de estilização da interface de utilizador.
6. **Compatibilidade Cross-Browser:** O Bootstrap é construído para ser compatível com os principais navegadores web, garantindo consistência na aparência e comportamento da aplicação em diferentes plataformas.

Em resumo, a combinação do Bootstrap com o Angular oferece uma abordagem eficiente para o desenvolvimento de interfaces responsivas, atraentes e fáceis de usar. Isso permite aproveitar a flexibilidade e a reutilização dos componentes do Bootstrap em todo o projeto, resultando numa experiência de utilização de alta qualidade para os utilizadores.

3.2.2 API em C# e ASP.NET Core Web API

A API deste projeto foi desenvolvida utilizando C# e ASP.NET Core Web API. Esta API é responsável por receber e processar solicitações vindas do Frontend (Angular), executando a lógica de negócios e interagindo com o Backend (Base de Dados).

Implementada como uma API RESTful, segue o estilo de arquitetura REST (Representational State Transfer) para comunicação com serviços web. O ASP.NET Core Web API oferece um ambiente robusto e seguro para criar APIs RESTful. Através de controladores, definem-se endpoints que correspondem às operações CRUD (Create, Read, Update and Delete) para adicionar, consultar, atualizar e remover dados da Base de Dados.

A linguagem C# é utilizada para implementar a lógica de negócios necessária para processar solicitações, validar dados e aplicar regras de negócios no acesso à Base de Dados SQL Server.

A utilização do padrão de injeção de dependências no ASP.NET Core simplifica a organização do código e aumenta a testabilidade dos componentes do backend, facilitando a manutenção e o desenvolvimento do projeto.

Nas figuras abaixo podemos observar o resultado do desenvolvimento desta API.

```

ReportTypeDTO <-
{
    type: "object",
    properties: {
        id: {
            type: "integer",
            format: "int32"
        },
        name: {
            type: "string"
        },
        description: {
            type: "string"
        },
        matable: {
            type: "boolean"
        }
    }
}
  
```

Swagger - Reports Types

```

PromoterCategories 1.0 OAS3
https://localhost:7203/swagger/v1/swagger.json

PromoterCategories
GET /api/PromoterCategories
POST /api/PromoterCategories
GET /api/PromoterCategories/{id}
PUT /api/PromoterCategories/{id}
DELETE /api/PromoterCategories/{id}
GET /api/PromoterCategories/Count

schemas
Credential
{
    idCredential: integer($int32)
    email: string
    available: true
    password: string
    multitable: true
    role: string
    multitable: true
    promoter: Promoter > {...}
}
  
```

Swagger - Promoter Categories

```

BoraLa.api 1.0 OAS3
https://localhost:7100/swagger/v1/swagger.json

EventTypes
GET /api/EventTypes
POST /api/EventTypes
GET /api/EventTypes/{id}
PUT /api/EventTypes/{id}
DELETE /api/EventTypes/{id}
GET /api/EventTypes/GetSubtypes/{eventTypeID}
  
```

Swagger - Event Types

3.2.2.1 Comunicação entre backend e frontend

No processo de comunicação entre o Frontend em Angular e a web API em ASP.NET, são estabelecidas requisições HTTP. O Angular envia solicitações para a API RESTful, indicando o tipo de operação (como GET, POST, PUT, DELETE) e, se necessário, os dados relevantes. Por sua vez, a API processa essas solicitações, executa as operações correspondentes na Base de Dados através de protocolos TCP/IP e devolve as respostas obtidas de volta para o Frontend. Essa troca de

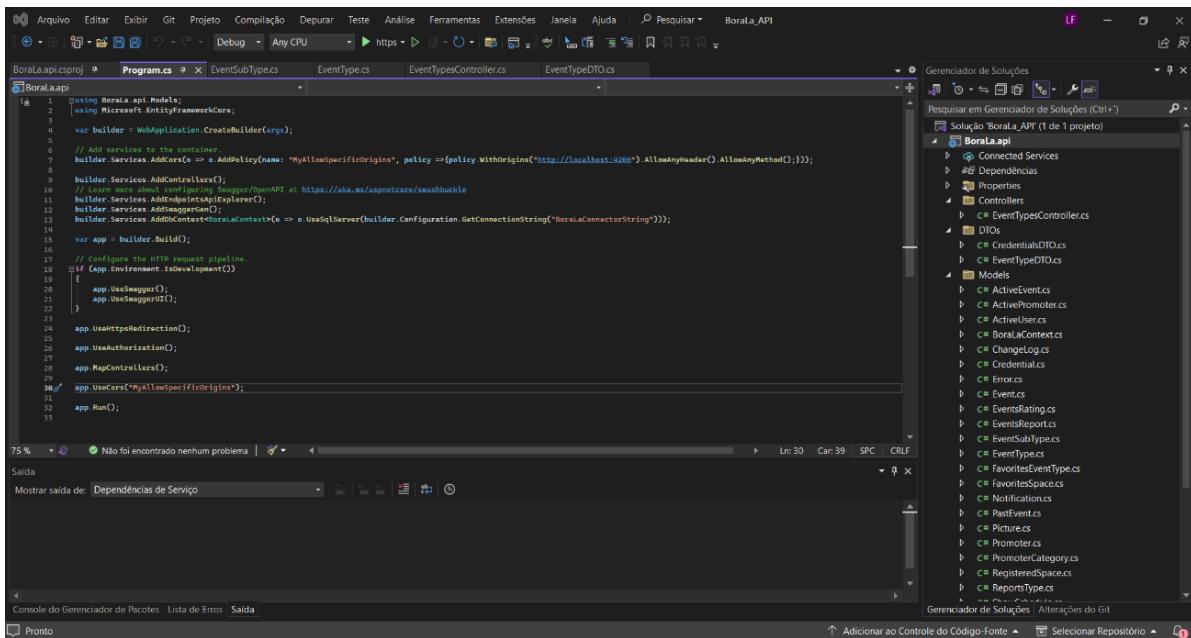
informações permite que o Frontend e o Backend interajam de forma eficiente, garantindo que os dados sejam atualizados e sincronizados conforme necessário.

3.2.2.2 Lógica Programacional

No backend, representado pela ASP.NET Core Web API, a lógica de negócios desempenha um papel fundamental. Ela é responsável por processar as solicitações recebidas do Frontend e realizar as operações necessárias na base de dados. Isso inclui uma variedade de tarefas, como validações, cálculos, processamento de dados e aplicação de regras específicas relacionadas à gestão dos recursos da aplicação.

A organização eficiente da lógica de negócios no backend é crucial para garantir que a aplicação mantenha uma estrutura coesa e bem definida, facilitando a manutenção e a escalabilidade do código. Além disso, ao manter essa lógica no backend, aumentamos a segurança do sistema, pois podemos controlar e validar o acesso às operações antes que elas sejam executadas na base de dados. Isso ajuda a prevenir possíveis vulnerabilidades e garante a integridade dos dados manipulados pela aplicação. Em resumo, a correta implementação da lógica de negócios no backend é essencial para o bom funcionamento e segurança de uma aplicação.

Na imagem seguinte podemos ver um excerto do que mencionamos na nossa explicação em cima.



3.2.2.3 Autenticação e Autorização

Em sistemas de gestão de aplicações, a segurança é uma prioridade essencial. Para isso, são implementados mecanismos de autenticação e autorização no backend, assegurando que apenas utilizadores autorizados tenham acesso às funcionalidades de gestão e possam executar determinadas operações.

O ASP.NET Core Web API oferece ferramentas integradas para implementar esses sistemas de autenticação e autorização. Existem diversas estratégias disponíveis para autenticar os utilizadores, como o uso de JSON Web Tokens (JWT), cookies de autenticação ou tokens de autenticação. Geralmente, o processo de autenticação ocorre no início da interação do utilizador com a aplicação, enquanto a autorização é verificada para cada solicitação subsequente, garantindo que apenas utilizadores autorizados possam realizar operações específicas.

Essas medidas de segurança são fundamentais para proteger os dados sensíveis e garantir a integridade das operações realizadas na aplicação. Ao implementar adequadamente a autenticação e autorização no backend, é possível criar um ambiente seguro e confiável para a gestão de informações e recursos.

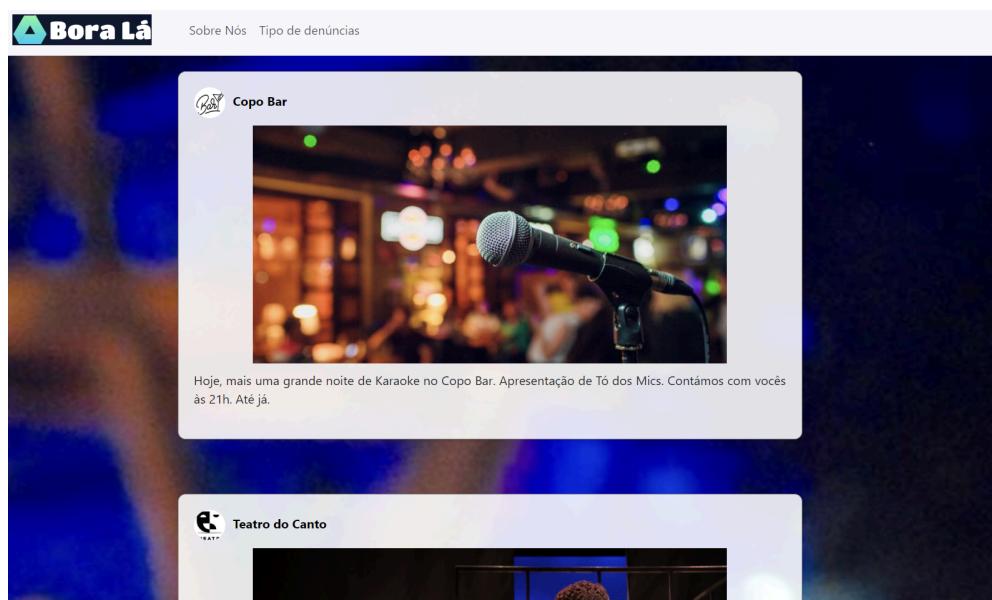
3.2.2.4 Tratamento de Erros

No desenvolvimento de aplicações, lidar com erros de forma eficaz é crucial para garantir a robustez do sistema. O ASP.NET Core Web API oferece recursos para capturar e gerenciar erros de maneira eficiente, garantindo que o frontend receba respostas apropriadas em caso de falhas ou exceções.

É fundamental que as respostas fornecidas ao frontend em caso de erros sejam claras e informativas, proporcionando aos utilizadores uma experiência amigável. Além disso, registrar os erros de forma adequada no backend é essencial para monitorar e resolver problemas, possibilitando uma manutenção mais eficiente da aplicação.

Dessa forma, ao implementar um tratamento de erros eficiente, garantimos uma experiência de utilização mais suave e confiável, enquanto facilitamos a identificação e correção de problemas no sistema. Isso contribui para a criação de aplicações robustas e de alta qualidade.

Tendo tudo isto em conta, e obedecendo à lógica de programação, terminamos com algo do género que podemos ver nas figuras abaixo.



Bora Lá

Sobre Nós | Tipo de denúncias

Sobre Nós

Bem-vindo à **Bora Lá**, a tua principal fonte para descobrir e participar nos eventos mais exclusivos e envolventes perto de ti, especialmente nos aconchegantes espaços que tornam Portugal tão especial.

Somos uma plataforma dedicada a conectar utilizadores e promotores de eventos de forma única e autêntica. Nosso foco está nos eventos íntimos e vibrantes que acontecem em bares, cafés e restaurantes locais, onde a verdadeira essência da cultura portuguesa ganha vida.

O Que Oferecemos:

- Descreve Eventos Locais:** Explore uma ampla variedade de eventos acontecendo na tua área, desde concertos acústicos até noites temáticas em pequenos bares e restaurantes charmosos.
- Conexões Autênticas:** Conectamos promotores de eventos apaixonados e utilizadores que procuram experiências autênticas. Os eventos são cuidadosamente curados para garantir momentos especiais e memoráveis.
- Notificações Personalizadas:** Cria listas dos teus eventos favoritos e recebe notificações personalizadas sempre que algo especial estiver prestes a acontecer perto de ti.
- Visibilidade Exclusiva para Promotores:** Oferecemos opções para promotores destacarem os seus eventos, aumentando a visibilidade e alcançando um público mais amplo. Saiba mais sobre as nossas opções de promoção para destacar o teu evento.

Junte-se à Comunidade:

A **Bora Lá** é mais do que uma plataforma, é uma comunidade de amantes de eventos que valorizam experiências autênticas. Se és um promotor de eventos ou um entusiasta à procura de novas experiências, estás no lugar certo.

Descobre, conecta e celebra a cultura local connosco. Bora lá explorar e criar memórias inesquecíveis!

Equipa Bora Lá

Copyright © Bora Lá 2024

Bora Lá

Sobre Nós | Tipo de denúncias

Tipos de Denúncias

Ver tudo | Criar novo | Procurar



Copyright © Bora Lá 2024

Bora Lá

Sobre Nós | Tipo de denúncias

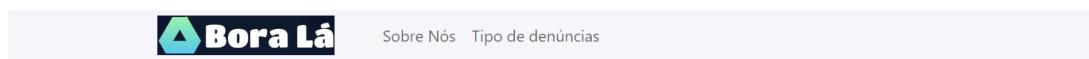
Tipos de Denúncias

Ver tudo | Criar novo | Procurar

ID	Tipo de Denúncia	Descrição	Editar	Apagar
1	Conteúdo Ofensivo	Conteúdo ofensivo, quer através de palavras ou ações.	Editar	Apagar
3	Bullying	Conteúdo que atenta contra o bem estar de alguém.	Editar	Apagar
4	Conteúdo de nudez ou sexual	Conteúdo de caráiz sexual ou com nudez explícita.	Editar	Apagar
5	Violência Gráfica	Exibição ou conteúdo com violência gráfica como mutilações.	Editar	Apagar
6	Informações Falsas	Divulgações de informações/conteúdo falso ou difamatório.	Editar	Apagar
7	SPAM	Conteúdo sem interesse, ou não relacionado com o evento indicado.	Editar	Apagar
8	Exploração infantil	Conteúdo que atenta aos direitos da criança.	Editar	Apagar
9	Exploração animal	Conteúdo que atenta aos direitos dos animais...	Editar	Apagar
10	Teste POST	Teste de inserção via API Post	Editar	Apagar
12	Conteúdo sem Nexo??	Teste de Post via FrontEnd.	Editar	Apagar

Lista do tipo de Denúncias. Total = 10

Copyright © Bora Lá 2024



Eliminar Tipo de Denúncia

ID do Tipo:

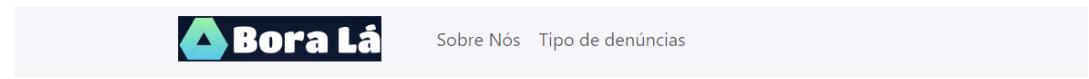
1

[Apagar](#)

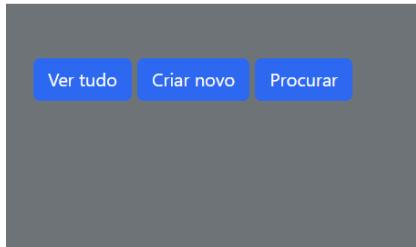
ID	Tipo de Denúncia	Descrição
1	Conteúdo Ofensivo	Contúdo ofensivo, quer através de palavras ou ações.

Informação do tipo de denúncia selecionado

Copyright © Bora Lá 2024



Tipos de Denúncias



Procurar Tipo de Denúncia via ID

ID:

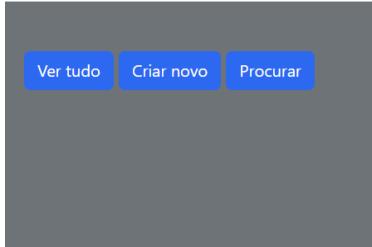
Id do tipo de denúncia a procurar

[Procurar](#)

Copyright © Bora Lá 2024



Tipos de Denúncias



Nome do tipo de denúncia

Conteúdo abusivo

Descrição

Conteúdo que ...

[Criar novo](#)

Copyright © Bora Lá 2024

Bora Lá Sobre Nós Tipo de denúncias

Editar Tipo de Denúncia

ID do Tipo: 1

Tipo: Conteúdo Ofensivo

Descrição: Contúdo ofensivo, quer através de palavras ou ações.

Atualizar

Copyright © Bora Lá 2024

BORA LÁ!



UPSKILL Programação WEB 2
Sara Virginia

Categorias dos Promotores dos Eventos



UPSKILL Programação WEB 2
Sara Virginia

Categorias dos Promotores dos Eventos

ID	Categoria do Promotor	Descrição	Mensalidade	Ações
TOTAL DE CATEGORIAS: 10				
1	egseg	sda	10 €	
2	Exhibitions	Organizing art exhibitions and cultural events	15 €	
3	Sports	Managing sports events and tournaments	20 €	
4	Conferences	Arranging conferences and business meetings	12,5 €	
5	Festivals	Hosting festivals and outdoor events	18 €	
6	teste	teste	50 €	
7	Magia	Espectáculos de sala	18 €	
10	Circo	Actividades ligadas às artes circenses	5 €	
13	Teatro de Rua	Espectáculos adequados à realização no exterior	20 €	
51	dasd	saddsa	3 €	

Categorias dos Promotores dos Eventos

ID do Promotor: 5

Categoria do Promotor: Festivals

Descrição: Hosting festivals and outdoor events

Valor da mensalidade: 18

Update

UPSKILL Programação WEB 2
Sara Virginia

Categorias dos Promotores dos Eventos

Categoria do Promotor:

Descrição:

Valor da mensalidade:

Create

UPSKILL Programação WEB 2
Sara Virginia

Categorias dos Promotores dos Eventos

ID5 Festivals

Hosting festivals and outdoor events

18 € / mês

Delete

UPSKILL Programação WEB 2
Sara Virginia

BoraLá Categorias dos Promotores dos Eventos

ID do Promotor
insira o ID a consultar

+ Informações

ID5 Festivals
Hosting festivals and outdoor events
18 € / mês



BoraLá Home Tipos de Evento & Sign In

Estás em casa aborrecido e não sabes o que fazer?
Estás a planejar uma viagem mas não sabes o que vai estar a acontecer durante as tuas bem merecidas férias?
Estás à procura de uma plataforma para promover o teu evento ou espaço?

Bem vindo à solução para os teus problemas!

Bem vindo à BoraLá!

A BoraLá é uma plataforma de promoção de eventos fácil de utilizar e que te diz tudo o que precisas de saber!
Explora os eventos a acontecer na tua área neste preciso momento, ou dá uma espreitadela no futuro e planeia uma viagem onde não te faltará entretenimento.
De concertos a noites de quiz, passando por tudo o que há no meio. Está tudo aqui!
Cria uma conta para te manteres a par das novidades na tua área, ou salta diretamente para explorar os eventos que estão disponibilizados!

Vais mesmo ficar em casa? BoraLá!

Explore Events

BoraLá Home Tipos de Evento & Sign In

EVENTS

[View All Event Types](#) [Create Event Type](#)

Copyright © BoraLá 2023
José do Egípto, Sara Rocha and Luís Gomes

BoraLá Home Tipos de Evento [Sign In](#)

All Event Types

- 1 - Concert**
 - [View Details](#)
 - [Update Event Type](#)
 - [Delete](#)
- 2 - Exhibition**
 - [View Details](#)
 - [Update Event Type](#)
 - [Delete](#)
- 3 - Quiz**
 - [View Details](#)
 - [Update Event Type](#)
 - [Delete](#)
- 4 - Conference**
 - [View Details](#)
 - [Update Event Type](#)
 - [Delete](#)

BoraLá Home Tipos de Evento [Sign In](#)

Event Type Generator

Event Type Name
Insira aqui o nome do tipo de evento a criar; Deve ser curto e abrangente.

[Create](#)

[View All Event Types](#) [Create Event Type](#)

Copyright © BoraLá 2023
José do Egípto, Sara Rocha and Luís Gomes

BoraLá Home Tipos de Evento [Sign In](#)

Event Type ID
1

Event Type Name
Concert

Associated Event Subtypes (Name, ID)
Rock Concert, 1
Jazz Concert, 2

Copyright © BoraLá 2023
José do Egípto, Sara Rocha and Luís Gomes

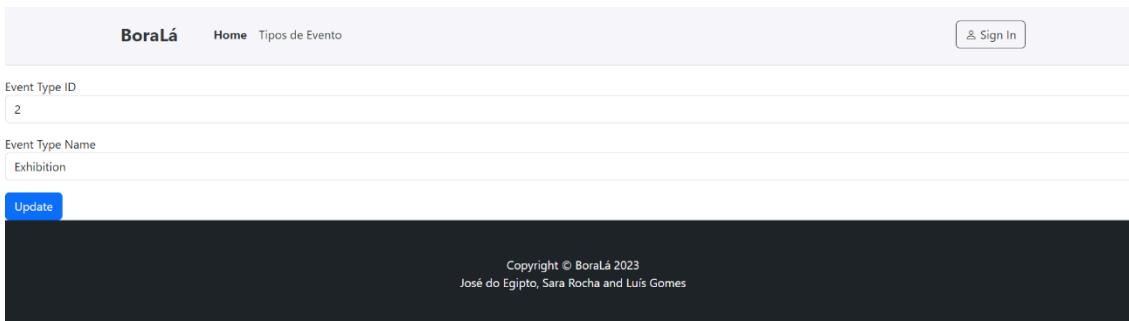
BoraLá Home Tipos de Evento [Sign In](#)

Event Type ID
1

Event Type Name
Concert

[Delete Event Type](#)

Copyright © BoraLá 2023
José do Egípto, Sara Rocha and Luís Gomes



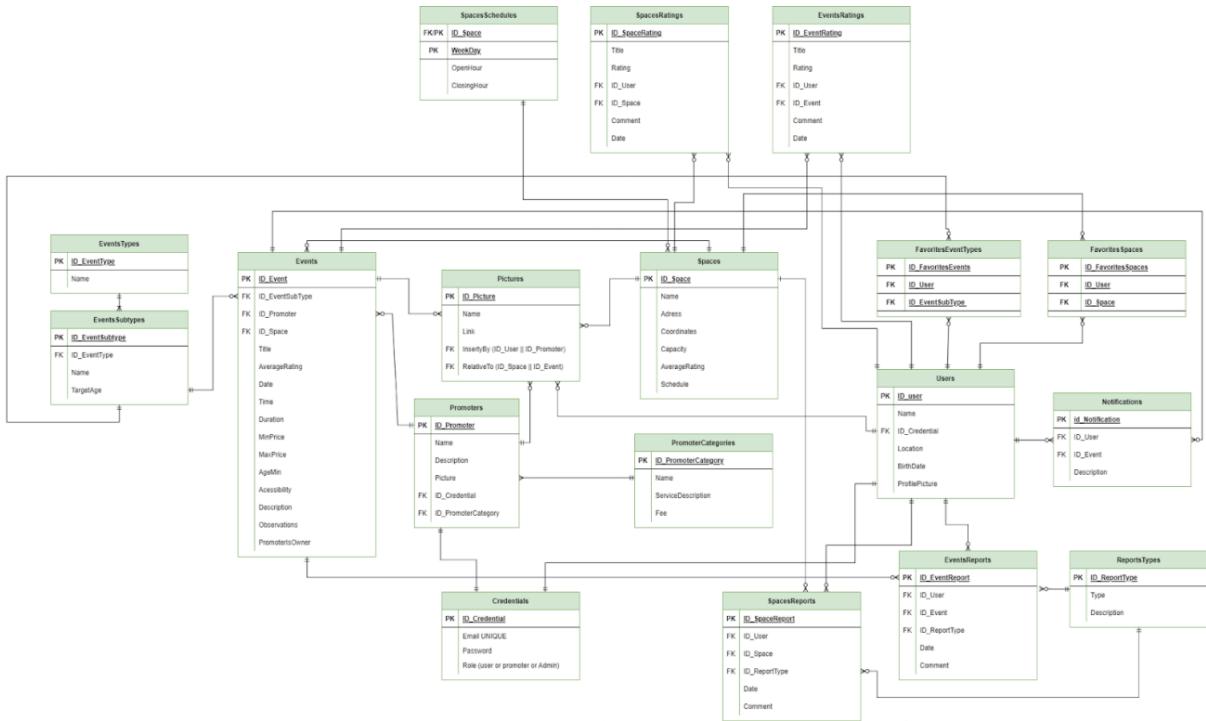
3.2.3 Base de Dados (SQL Server)

A base de dados foi desenvolvida utilizando SQL Server e é responsável pelo armazenamento e gestão dos dados da aplicação. Foram criadas tabelas e estabelecidos relacionamentos entre elas para organizar os dados de acordo com as necessidades do negócio. Através da API RESTful (ASP.NET Core Web API), é possível interagir com a base de dados para realizar operações de leitura (consultas) e escrita (inserções, atualizações e exclusões) nos dados.

O SQL Server oferece recursos avançados de gestão de dados, como integridade referencial, transações, índices e otimização de consultas, garantindo a eficiência e confiabilidade da base de dados. A separação em camadas, desde o frontend da aplicação até a API e a base de dados, permite um desenvolvimento mais organizado, modular e fácil de manter.

Além disso, o uso de tecnologias como Angular, C# e ASP.NET Core Web API proporciona um conjunto poderoso de ferramentas para construir aplicações web de alto desempenho e escaláveis, proporcionando uma experiência superior ao utilizador. Essa abordagem de desenvolvimento estruturado e a utilização de tecnologias robustas contribuem para a criação de sistemas confiáveis e eficazes.

Na figura abaixo é possível visualizar o desenho da base de dados, as suas entidades e relações.



3.2.3.1 Modelação de Dados

Na fase de estruturação dos dados, a modelagem desempenha um papel crucial na definição do esquema da base de dados, que será responsável por armazenar as informações da aplicação. Uma modelagem precisa é fundamental para assegurar que os dados sejam organizados de forma lógica, eficiente e mantendo a integridade referencial.

Essa etapa é essencial para garantir que a base de dados seja capaz de armazenar os dados de forma coerente, seguindo as necessidades e requisitos do sistema. Uma modelagem cuidadosa permite a definição de relacionamentos entre as diferentes entidades, a criação de chaves primárias e estrangeiras, e a especificação de restrições para garantir a consistência e integridade dos dados.

Em resumo, a modelagem adequada dos dados é um passo crítico no processo de desenvolvimento de uma aplicação, pois estabelece as bases para uma base de dados bem estruturada e eficiente.

3.2.3.1.1 Tabelas

Na modelagem de dados, as tabelas desempenham um papel fundamental, sendo os blocos de construção essenciais do sistema. Cada tabela representa uma entidade dentro da aplicação, como um utilizador, produto, pedido ou qualquer outra informação relevante. Dentro de uma tabela, cada coluna corresponde a um atributo ou campo que descreve uma característica específica da entidade, como nome, preço, data de criação, entre outros. Essa estrutura tabular permite organizar e armazenar os dados de forma eficiente e coerente, garantindo que as informações sejam gerenciadas de maneira clara e acessível.

Nas figuras abaixo conseguimos visualizar as entidades e atributos referidos.

Dicionário de Entidade - Promoters

NOME	DESCRÍÇÃO
Promoters	Promotores de eventos. São as pessoas responsáveis por inserir toda a informação sobre eventos a decorrer que sejam promotores do mesmo. Podem ser os donos/gestores dos espaços onde os eventos irão decorrer.

Dicionário de Atributos - Promoters

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Promoter	Número identificativo do promotor	INT IDENTITY (1,1)	>0		X		
Name	Nome do promotor	NVARCHAR(25)	String c/ no máx 25 caracteres				
Description	Descrição do promotor	NVARCHAR(1000)	limite de 1000 caracteres	X			
ID_Credential	Identificador das credenciais de acesso	INT	>0		X		Credentials
ID_PromoterCategory	Número identificativo da categoria do promotor	INT	> 0		X		PromoterCategories

3.2.3.1.2 Relações

No SQL Server, é possível criar relacionamentos entre tabelas para conectar diferentes entidades. Esses vínculos são estabelecidos utilizando chaves estrangeiras, que fazem referência à chave primária de outra tabela.

Esses relacionamentos são essenciais para manter a integridade referencial dos dados, prevenindo inconsistências e redundâncias. Em outras palavras, os relacionamentos garantem que as informações relacionadas sejam conectadas de maneira precisa e coerente, promovendo uma estrutura de dados organizada e confiável.

Na figura abaixo conseguimos visualizar uma entrada no dicionário de relações.

Dicionário de Relações - Promoters

Entidade		Relação	Entidade	Relação	Justificação
Promoters	Promove/cria	1...*	Events	1...1	
Promoters	Tem	0...*	Spaces	1...1	
Promoters	Tem	1...1	Credentials	1...1	Segurança
Promoters	Tem	1...1	Categories	0...*	

3.2.3.2 Linguagem SQL (Structured Query Language)

O SQL Server utiliza a Linguagem SQL (Structured Query Language) para realizar operações na base de dados. O SQL é uma linguagem de programação declarativa que permite à equipe de desenvolvimento enviar comandos para buscar, inserir, atualizar ou excluir dados nas tabelas.

As consultas SELECT são usadas para recuperar dados de uma ou mais tabelas. Com elas, é possível filtrar, ordenar e agrupar os resultados para obter as informações desejadas.

As operações de Inserção (INSERT), Atualização (UPDATE) e Exclusão (DELETE) são utilizadas para adicionar novos registros, atualizar informações existentes ou remover dados de uma tabela, respectivamente. Essas operações permitem que a equipe de desenvolvimento manipule os dados da base de dados de acordo com as necessidades da aplicação.

3.2.3.2.1 Indexação e Desempenho

O SQL Server disponibiliza recursos de indexação para aprimorar o desempenho das consultas e otimizar o acesso aos dados. Os índices são estruturas de dados que aceleram a recuperação de informações ao organizar os dados em uma ou mais colunas de forma ordenada. Essa organização ajuda a reduzir o tempo necessário para pesquisar dados e aumenta a eficiência da base de dados, especialmente em tabelas de grande volume.

Além disso, é crucial planejar e otimizar as consultas SQL para melhorar o desempenho da base de dados. O SQL Server inclui optimizadores de consulta que analisam as consultas enviadas e selecionam o plano de execução mais eficiente para recuperar os dados solicitados. Essas práticas ajudam a garantir um acesso mais rápido e eficiente aos dados, resultando em um desempenho aprimorado do sistema como um todo.

3.2.3.2.2 Segurança e Acessos

A segurança na camada de dados é de extrema importância, pois é onde as informações sensíveis da aplicação são armazenadas. O SQL Server disponibiliza

recursos de segurança sólidos para assegurar que somente utilizadores autorizados tenham acesso aos dados.

Nesta camada, são implementadas medidas para proteger os dados contra acessos não autorizados, garantindo a confidencialidade e integridade das informações armazenadas. O SQL Server oferece opções para configurar permissões de acesso, criptografar dados sensíveis e monitorar atividades de utilizadores, contribuindo para a manutenção de um ambiente seguro e protegido na gestão dos dados.

3.2.3.2.3 Autenticação

O SQL Server oferece a possibilidade de configurar diversos métodos de autenticação, incluindo autenticação do Windows, autenticação do SQL Server e autenticação baseada em certificados. Esses métodos permitem que os utilizadores se identifiquem e validem os seus acessos ao servidor de base de dados de acordo com as suas credenciais e permissões específicas.

Essa variedade de opções de autenticação permite uma abordagem flexível na gestão da segurança de acesso aos dados, adaptando-se às necessidades e políticas de segurança da organização. Cada método de autenticação possui características e requisitos próprios, proporcionando uma gama de escolhas para garantir a proteção dos dados e a integridade do sistema.

3.2.3.2.4 Autorização

Na camada de dados, é possível estabelecer permissões detalhadas para tabelas e objetos do banco de dados, garantindo que apenas utilizadores específicos possam acessar determinadas informações ou realizar operações específicas.

Além disso, o SQL Server oferece suporte à criptografia de dados, o que proporciona uma camada adicional de segurança ao proteger informações sensíveis armazenadas no banco de dados contra acessos não autorizados.

Dessa forma, a camada de dados desempenha um papel crucial em qualquer aplicação, assegurando a persistência e a gestão adequada dos dados. Com uma modelagem cuidadosa e a utilização eficiente da Linguagem SQL, podemos criar um banco de dados eficaz, seguro e capaz de atender às exigências da nossa aplicação.

4. Conclusão

Este projeto proporcionou uma valiosa oportunidade para aplicarmos e desenvolvermos os fundamentos adquiridos em diversas tecnologias e metodologias durante a fase inicial do curso UpSkill de Power Platform. Ao escolhermos o desenvolvimento de uma aplicação web centrada na descoberta de eventos locais, enfrentamos desafios e aprendemos lições cruciais. Desde o planeamento e criação da base de dados até à implementação do front-end e do back-end, cada fase do projeto contribuiu para a nossa compreensão mais profunda das tecnologias-chave, como C#, SQL Server, REST API CORE.NET e Angular.

A metodologia Agile, que adotamos desde o início, revelou-se essencial para um desenvolvimento eficiente e adaptável às mudanças necessárias ao longo do processo. Apesar das limitações de tempo, conseguimos realizar uma parte do Web Service (API) e da aplicação para provar o conceito, consolidando assim os nossos conhecimentos práticos.

Acreditamos que atingimos com êxito o principal objetivo deste projeto: adquirir conhecimentos introdutórios de forma prática e activa. Este trabalho não apenas melhorou as nossas habilidades técnicas, mas também nos preparou para enfrentar desafios semelhantes em ambiente profissional. Estamos confiantes de que as competências e experiências adquiridas serão valiosas para futuros projetos e oportunidades no vasto campo do desenvolvimento de software.

5. ANEXOS

Anexo A

Dicionários de Atributos, entidades e relacionamentos da base de dados

Dicionário de Entidade - Promoters

NOME	DESCRÍÇÃO
Promoters	Promotores de eventos. São as pessoas responsáveis por inserir toda a informação sobre eventos a decorrer que sejam promotores do mesmo. Podem ser os donos/gestores dos espaços onde os eventos irão decorrer.

Dicionário de Atributos - Promoters

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Promoter	Número identificativo do promotor	INT IDENTITY (1,1)	>0		X		
Name	Nome do promotor	NVARCHAR(25)	String c/ no máx 25 caracteres				
Description	Descrição do promotor	NVARCHAR(1000)	limite de 1000 caracteres	X			
ID_Credential	Identificador das credenciais de acesso	INT	>0			X	Credentials
ID_PromoterCategory	Número identificativo da categoria do promotor	INT	> 0			X	PromoterCategories

Dicionário de Relações - Promoters

Entidade		Relação	Entidade	Relação	Justificação
Promoters	Promove/cria	1...*	Events	1...1	
Promoters	Tem	0...*	Spaces	1...1	
Promoters	Tem	1...1	Credentials	1...1	Segurança
Promoters	Tem	1...1	Categories	0...*	

Dicionário de Entidade - PromoterCategories

NOME	DESCRIÇÃO
PromoterCategories	Categoria em que se insere a subscrição de um promotor (por exemplo, Gold, Silver, Free, etc...). Afeta algumas das funcionalidades do navegador (como, por exemplo, a página em que aparecem os eventos promovidos pelo promoter em causa).

Dicionário de Atributos - PromoterCategories

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_PromoterCategory	Identificador da categoria	INT IDENTITY (1,1)	>0		X		
Name	Nome da Categoria	NVARCHAR(50)	Máximo de 50 caracteres				
ServiceDescription	Breve descrição das vantagens da categoria	NVARCHAR(1000)	Máximo de 1000 carateres				
Fee	Custo mensal da subscrição do serviço	MONEY	>0	X			

Dicionário de Relações - PromoterCategories

Entidade		Relação	Entidade	Relação	Justificação
PromoterCategories	Está associada	0...*	Promoters	1...1	

Dicionário de Entidade - Spaces

NOME	DESCRÍÇÃO
Spaces	Espaços onde decorrerão os eventos. Podem ser desde pequenos cafés, bares, discotecas, restaurantes, teatros, bibliotecas, parques, etc. Por norma terão um promotor associado, podendo ser o dono do espaço (talvez o mais comum para pequenos espaços) ou alguém responsável pela gestão/promoção de eventos (em espaços maiores). Será sempre um promotor a registar o espaço, mas terá de haver uma verificação dos dados inseridos por nós.

Dicionário de Atributos - Spaces

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Space	Número identificativo do espaço	Int Identity(1,1)	> 0		X		Spaces
Name	Nome do espaço	NVARCHAR(150)	String c/ no máx 150 caracteres				
Adress	Morada do espaço	NVARCHAR(500)	String c/ no máx 500 caracteres				
Coordinates	Coordenadas do espaço	GEOGRAPHY	SRID 4326: Latitude: -90 a 90 (90 - Polo Norte, -90 para - Polo Sul) Longitude: -180 a 180 (0 para o meridiano de Greenwich)				

Dicionário de Atributos - Spaces

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
Capacity	Lotação do local	INT	> 0				
AverageRating	Média das classificações atribuídas por utilizadores - trigger	DECIMAL(2,1)	>= 0	X			
Schedule	Horário de funcionamento - vem de um trigger na tabela horários	XML		X			

Dicionário de Relações - Spaces

Entidade		Relação	Entidade	Relação	Justificação
Spaces	Tem	0...*	Pictures	1...1	
Spaces	Tem	0...*	SpaceReports	1...1	
Spaces	É favorito de	0...*	FavoriteSpaces	1...1	
Spaces	Tem	0...*	SpaceRatings	1...1	

Dicionário de Relações - Spaces

Entidade		Relação	Entidade	Relação	Justificação
Spaces	Tem	0...*	Events	1...1	
Spaces	Tem	0...*	Horário/Dia	1...1	

Dicionário de Entidade - Spaces Schedules

NOME	DESCRIÇÃO
SpacesSchedules	Tabela com os horários de funcionamento dos espaços. Para permitir uma informação detalhada permitimos indicar a hora de abertura e hora de fecho de cada dia.

Dicionário de Atributos - Spaces Schedules

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Space	Identificativo do espaço pertencente	Int Identity(1,1)	> 0		X		Spaces
WeekDay	Código numérico para os dias	INT	Entre 1 para domingo, e 7 para sábado.		X		
OpenHour	Hora de abertura	TIME	[0 : 24]				
CloseHour	Hora de fecho	TIME	[0 : 24]			X	

Dicionário de Relações - Space Schedules

Entidade		Relação	Entidade	Relação	Justificação
SpacesSchedules	Tem	0...*	Spaces	1...1	

Dicionário de Entidade - Events

NOME	DESCRIÇÃO
Events	Eventos que são inseridos no sistema pelos promotores. Podem ser de vários tipos, desde culturais a gastronómicos. São a entidade principal do sistema, e tudo gira à volta deles.

Dicionário de Atributos - Events

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Event	Número identificativo do evento	INT IDENTITY (1,1)	> 0		X		Events
ID_EventSubtype	Número que identifica o subtipo do evento, e consequentemente o tipo também.	INT	> 0			X	EventsSubtypes
ID_Space	Número identificativo do espaço	INT	> 0			X	Spaces
Date	Data do Evento	DATETIME					Events
Time	Hora de início do evento.	TIME		X			Events

Dicionário de Atributos - Events

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
Duration	Tempo aproximado de duração do evento	TIME	> 15 (minutos)	X			
MinPrice	Preço mínimo do evento.	DECIMAL (7,2)	0 (€) (se grátils) ou mais				
MaxPrice	Preço máximo do evento	DECIMAL (7,2)	> 0 (€)				
MinAge	Identifica a idade mínima exigida pelo evento	INT	BETWEEN 1 AND 17	X			
Acessibility	Se o evento tem acesso para pessoas de mobilidade reduzida	BIT	0: Não tem 1: Tem Null: desconhecido	X			

Dicionário de Atributos - Events

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
Description	Descrição por escrito do evento e possíveis informações úteis.	NVARCHAR (2000)	Até 2000 caracteres				
Observations	Observações que corresponde a informação não apresentada noutros campos e possa ser importante.	NVARCHAR (1000)	Até 1000 caracteres				
AverageRating	Classificação média do evento. Provém de um TRIGGER cada vez que são adicionadas classificações	DECIMAL (2,1)	>= 0	X			
PromoterIsOwner	Indica se o promotor é o dono do espaço onde o evento é realizado	BIT	0: falso, 1: verdadeiro				
Name	Nome do evento	NVARCHAR (150)	Até 150 caracteres				

Dicionário de Relações - Events

Entidade		Relação	Entidade	Relação	Justificação
Events	Tem	1...1	Promoters	1...*	
Events	Tem	1...1	Spaces	1...*	
Events	Tem	0...*	Pictures	1...1	
Events	Tem	0...*	EventsRatings	1...1	
Events	Promove	0...*	Notifications	1...1	

Dicionário de Relações - Events

Entidade		Relação	Entidade	Relação	Justificação
Events	Tem	1...*	EventsSubType	1...1	
Events	Tem	0...*	EventsReports	1...1	

Dicionário de Entidade - Users

NOME	DESCRÍÇÃO
Users	Utilizadores da aplicação/página Web. São pessoas interessadas em descobrir eventos perto de si ou planejar uma saída num local onde irão visitar, mais tarde.

Dicionário de Atributos - Users

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_User	Número identificativo do utilizador	INT	IDENTITY (1,1)		X		Users
Name	Nome do utilizador	NVARCHAR	máximo de 100 caracteres				Users
Location	Localidade preferencial do utilizador (A ser utilizada quando o utilizador não tiver a permissão de partilha de localização ativada)	GEOGRAPHY	SRID 4326: Latitude: -90 a 90 (90 - Polo Norte, -90 para - Polo Sul) Longitude: -180 a 180 (0 para o meridiano de Greenwich)				Users
BirthDate	Data de nascimento do utilizador	DATE	ano-mês-dia				Users
ProfilePicture	Fotografia de perfil	NVARCHAR	máximo de 50 caracteres	X			Users

Dicionário de Relações - Users

Entidade		Relação	Entidade	Relação	Justificação
Users	Tem	1...1	Credentials	1...1	Segurança de Dados
Users	Classifica	1...*	Spaces Ratings	1...1	Avaliar Sucesso do Espaço
Users	Classifica	1...*	Events Ratings	1...1	Avaliar Sucesso do Evento
Users	Reporta	1...*	Spaces Reports	1...1	Feedbacks relevantes
Users	Reporta	1...*	Events Reports	1...1	Feedbacks relevantes
Users	Tem Favorito	1...*	Favorites Spaces	1...1	Listagem de Espaços Favoritos
Users	Tem Favorito	1...*	Favorites Event Types	1...1	Listagem de Eventos Favoritos
Users	Recebe	1...*	Notifications	1...1	Alerta de interesses

Dicionário de Entidade - Notifications

NOME	DESCRIÇÃO
Notifications	Notificações enviadas aos utilizadores sobre eventos em que possam estar interessados.

Dicionário de Atributos - Notifications

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Notification	Identificador da notificação	INT IDENTITY (1,1)	>0		X		Notification
ID_User	Identificador do Usuário a quem é dirigida a notificação	INT	>0			X	Users
ID_Event	Identificador do Evento sobre o qual o utilizador está a ser notificado	INT	>0			X	Events
Description	Conteúdo da notificação	NVARCHAR	Máximo de 100 caracteres	X			Notification

Dicionário de Relações - Notifications

Entidade		Relação	Entidade	Relação	Justificação
Users	1...*	Recebe	Notifications	1...1	Alerta de evento de interesse
Events	1...*	Envia	Notifications	1...1	Alerta de captação de clientes

Dicionário de Entidade - Credentials

NOME	DESCRIÇÃO
Credentials	Credenciais de acesso de um utilizador à aplicação (tanto User como Promoter)

Dicionário de Atributos - Credentials

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Credential	Identificador das credenciais	INT	>0		X		
Email	Email de registo/login do utilizador	NVARCHAR(50)	UNIQUE Máximo de 50 caracteres				
Password	Password de acesso do utilizador	NVARCHAR(20)	Máximo de 20 caracteres				
Role	Papel do utilizador (User, Promoter ou Admin)	NVARCHAR(20)	Máximo de 20 caracteres				

Dicionário de Relações - Credentials

Entidade		Relação	Entidade	Relação	Justificação
Users	Possui	1...1	Credentials	1...1	Segurança de Dados
Promoters	Possui	1...1	Credentials	1...1	Segurança de Dados

Dicionário de Entidade - Event Type

NOME	DESCRIÇÃO
Event Type	Tipo principal do evento (música, gastronomia, teatro, etc...)

Dicionário de Atributos - Event Type

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_EventType	Identificador do tipo de evento.	Int	>0		X		
Name	Breve descrição do tipo de evento	String	Máximo de 20 caracteres				

Dicionário de Relações - Event Type

Entidade		Relação	Entidade	Relação	Justificação
Event Type	Possui	1...*	Event Subtype	1...1	

Dicionário de Entidade - Event Subtype

NOME	DESCRIÇÃO
Event Subtype	Subtipos de evento inseridos e relacionados com o tipo de evento.

Dicionário de Atributos - Event Subtype

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_EventSubtype	Identificador do subtipo de evento	INT IDENTITY (1,1)	>0		X		
ID_EventType	Identificador do tipo de evento	INT	>0			X	EventTypes
Name	Breve descrição do subtipo de evento	NVARCHAR(50)	Máximo de 50 caracteres				
TargetAge	Alcance de idades do público alvo.	INT	>0	X			

Dicionário de Relações - Event Subtype

Entidade		Relação	Entidade	Relação	Justificação
Favourite EventTypes	Possui	0...*	Event Subtype	1...1	
EventTypes	Possui	1...*	Event Subtype	1...1	
Events	Possui	1...1	Event Subtype	1...*	

Dicionário de Entidade - ReportsTypes

NOME	DESCRIÇÃO
ReportsTypes	Tipos de Relatos, relacionados com os Eventos, Espaços e Utilizadores.

Dicionário de Atributos - ReportsTypes

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_ReportType	Identificador do Tipo de Relato	INT IDENTITY(1,1)	>0		X		ReportsTypes
Type	Tipo de Relato	NVARCHAR	Máximo de 50 caracteres				ReportsTypes
Description	Relato discriminado	NVARCHAR	Máximo de 100 caracteres				ReportsTypes

Dicionário de Relações - ReportsTypes

Entidade		Relação	Entidade	Relação	Justificação
ReportsTypes	1...*	Envia	EventsReports	1...1	Relatos limitados aos Eventos
ReportsTypes	1...*	Envia	SpacesReports	1...1	Relatos limitados aos Espaços

Dicionário de Entidade - SpacesReports

NOME	DESCRÍÇÃO
SpacesReports	Relatos relativos aos Espaços.

Dicionário de Atributos - SpacesReports

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_SpaceReport	Identificador do Relatório do Espaço	INT IDENTITY	>0		X		SpacesReports
ID_User	Identificador do Utilizador	INT	>0			X	Users
ID_Space	Identificador do Espaço	INT	>0			X	Spaces
ID_ReportType	Identificador do Tipo de Relatório	INT	>0			X	ReportsTypes
Date	Data do Relatório	DATE	ano-mês-dia				SpacesReports
Comment	Comentário do Utilizador	NVARCHAR	Máximo de 100 caracteres	X			SpacesReports

Dicionário de Relações - SpacesReports

Entidade		Relação	Entidade	Relação	Justificação
Spaces	1...*	Envia	SpacesReports	1...1	Recebe dados do Espaço
ReportsTypes	1...*	Envia	SpacesReports	1...1	Recebe tipos de Relatos

Dicionário de Entidade - EventsReports

NOME	DESCRIÇÃO
EventsReports	Relatos relativos aos Eventos.

Dicionário de Atributos - EventsReports

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_EventReport	Identificador do Relatório do Evento	INT IDENTITY(1,1)	>0		PK		EventsReports
ID_User	Identificador do Utilizador	INT	>0			FK	Users
ID_Event	Identificador do Evento	INT	>0			FK	Events
ID_ReportType	Identificador do Tipo de Relatório	INT	>0			FK	ReportsTypes
Date	Data do Relatório	DATE	Data				EventsReports
Comment	Comentário do Utilizador	NVARCHAR	Máximo de 100 caracteres	X			EventsReports

Dicionário de Relações - EventsReports

Entidade		Relação	Entidade	Relação	Justificação
Events	1...*	Envia	EventsReports	1...1	Recebe dados do Evento
ReportsTypes	1...*	Envia	EventsReports	1...1	Recebe tipos de Relatos

Dicionário de Entidade - FavoritesEventTypes

NOME	DESCRIÇÃO
FavoritesEventTypes	Listas dos (sub)tipos Favoritos de Eventos.

Dicionário de Atributos - FavoritesEventTypes

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_FavoritesEvents	Identificador dos Eventos Favoritos do Utilizador	INT IDENTITY(1,1)	>0		X		FavoritesEventTypes
ID_User	Identificador do Utilizador	INT	>0			X	Users
ID_EventSubType	Identificador do Subtipo de Evento	INT	>0			X	EventsSubTypes

Dicionário de Relações - FavoritesEventTypes

Entidade		Relação	Entidade	Relação	Justificação
EventSubTypes	1...*	Envia	Favourites EventTypes	1...1	Recebe dados do Subtipo de Eventos
Users	1...*	Envia	Favourites EventTypes	1...1	Recebe dados do Utilizador

Dicionário de Entidade - FavoritesSpaces

NOME	DESCRIÇÃO
FavoritesSpaces	Listas dos Espaços Favoritos.

Dicionário de Atributos - FavoritesSpaces

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_FavoriteSpaces	Identificador dos Espaços Favoritos do Utilizador	INT IDENTITY(1,1)	> 0	NOT NULL	X		FavoriteSpaces
ID_User	Identificador do Utilizador	INT	> 0	NOT NULL		X	Users
ID_Space	Identificador do Espaço	INT	> 0	NOT NULL		X	Spaces

Dicionário de Relações - FavoritesSpaces

Entidade		Relação	Entidade	Relação	Justificação
Spaces	1...*	Envia	Favourites Spaces	1...1	Recebe dados do Espaço
Users	1...*	Envia	Favourites Spaces	1...1	Recebe dados do Utilizador

Dicionário de Entidade - FavoritesSpaces

NOME	DESCRÍÇÃO
FavoritesSpaces	Listas dos Espaços Favoritos.

Dicionário de Atributos - FavoritesSpaces

Nome	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_FavoriteSpaces	Identificador dos Espaços Favoritos do Utilizador	INT IDENTITY(1,1)	> 0	NOT NULL	X		FavoriteSpaces
ID_User	Identificador do Utilizador	INT	> 0	NOT NULL		X	Users
ID_Space	Identificador do Espaço	INT	> 0	NOT NULL		X	Spaces

Dicionário de Relações - FavoritesSpaces

Entidade		Relação	Entidade	Relação	Justificação
Spaces	1...*	Envia	Favourites Spaces	1...1	Recebe dados do Espaço
Users	1...*	Envia	Favourites Spaces	1...1	Recebe dados do Utilizador

Dicionário de Entidade - Spaces Ratings

NOME	DESCRIÇÃO
SpacesRatings	Tabela com as classificações atribuídas pelos utilizadores aos espaços (de 1 a 5).

Dicionário de Atributos - Spaces Ratings

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_SpaceRating	Número identificativo da classificação	INT IDENTITY (1,1)	> 0		X		
Title	Título da classificação	NVARCHAR(150)	String de no máx. 150 caracteres				
Rating	Classificação atribuída	INT	Entre 1 e 5.				
ID_User	Identificação do utilizador que classificou	INT	>0		X		Users
ID_Space	Identificação do Espaço classificado	INT	>0		X		Spaces

Dicionário de Atributos - Spaces Ratings

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
Comment	Comentário do utilizador	NVARCHAR(1000)	> 0	X			
Date	Data de criação	DATE TIME	Obtida pelo GETDATE()				

Dicionário de Relações - Space Ratings

Entidade		Relação	Entidade	Relação	Justificação
SpacesRatings	Tem	1...1	Spaces	0...*	

Dicionário de Entidade - Events Ratings

NOME	DESCRIÇÃO
EventsRatings	Tabela com as classificações atribuídas pelos utilizadores aos eventos (de 1 a 5).

Dicionário de Atributos - Events Ratings

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_EventRating	Número identificativo da classificação	INT IDENTITY (1,1)	> 0		X		
Title	Título da classificação	NVARCHAR(150)	String de no máx. 150 caracteres				
Rating	Classificação atribuída	INT	Entre 1 e 5.				
ID_User	Identificação do utilizador que classificou	INT	>0		X		Users
ID_Space	Identificação do Espaço classificado	INT	>0		X		Spaces

Dicionário de Atributos - Events Ratings

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
Comment	Comentário do utilizador	NVARCHAR(1000)	> 0	X			
Date	Data de criação	DATE TIME	Obtida pelo GETDATE()				

Dicionário de Relações - Events Ratings

Entidade		Relação	Entidade	Relação	Justificação
EventRatings	Tem	1...1	Spaces	0...*	

Dicionário de Entidade - Pictures

NOME	DESCRIÇÃO
Pictures	Tabela os dados e links de imagens/fotos dos espaços e eventos.

Dicionário de Atributos - Pictures

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Picture	Número identificativo da imagem	INT IDENTITY (1,1)	> 0		X		
Name	Curto nome da imagem	NVARCHAR(50)	String de no máx. 50 caracteres				
Link	Indicação de onde está armazenada	VARCHAR(150)	String de no máx. 150 caracteres				
ID_User	Identificação do utilizador que adicionou a imagem	INT	>0 (nota: ID_User e ID_Promoter não podem ser nulos simultaneamente).	X		X	Users
ID_Promoter	Identificação do utilizador que adicionou a imagem	INT	>0	X		X	Spaces

Dicionário de Atributos - Pictures

Atributo	Descrição	Tipo de dados	Domínio	NULL	PK	FK	Entidade
ID_Space	Identificação do espaço a que pertence	INT	> 0 (nota: ID_User e ID_Promoter não podem ser nulos simultaneamente).	X		X	Spaces
ID_Event	Identificação do evento a que pertence	INT	> 0	X		X	Events

Dicionário de Relações - Pictures

Entidade		Relação	Entidade	Relação	Justificação
Pictures	Tem	0...1	Users	0...*	
Pictures	Tem	0...1	Promoters	0...*	
Pictures	Tem	0...1	Spaces	0...*	
Pictures	Tem	0...1	Events	0...*	