

Licenciatura em Tecnologias da Informação

Desenvolvimento Web Multiplataforma

Worksheet 1

Keywords

Laravel; MVC; Routes; Views

This worksheet assumes that you have the tools necessary to implement the solutions using the Laravel framework and that Laravel is installed globally on your system. If you do not yet meet the minimum requirements, consult the official Laravel documentation: <https://laravel.com/docs/12.x/>.

Theme

We aim to create a web application to manage conferences. The application will dynamically handle different web conferences, including the registration process and the management of speakers and sponsors. The required data structure is outlined below.

Table 1: Database Structure

Table	Fields	Description
conferences	acronym, name, description, location, conferenceDate	Conference details
speakers	name, affiliation, biography, photo, socialNetworks, pageLink, speakerType, isActive	Speaker details
speakerType	description	Information about the speaker type
sponsors	name, logo, category, isActive	Sponsor details
users	firstname, lastname, username, userType, email, password, description, lastLogin, isActive, creationDate, changeDate	User registration data
userType	description	Information about the user type
confParticipant	conferenceID, participantID, registrationDate	Participant registration table
confSpeaker	conferenceID, speakerID	Association table between speakers and conference
confSponsor	conferenceID, sponsorID	Association table between sponsors and conference

Functional Requirements

- FR1: The system must allow access to conference information.
- FR2: The system must allow user registration for a conference.
- FR3: Administrators must be able to manage speakers and sponsors.
- FR4: Users must be able to update their information.

Non-Functional Requirements

- NFR1: The database must be protected against unauthorized access.
- NFR2: The application must function correctly in major browsers.

Tasks

1. Based on the above description, data structure, and requirements:

- a) Define the functional requirements in detail.
- b) Define the relational database model.
- c) Design wireframes for your application.
- d) If not already done, create a Laravel project for the application.
- e) Explore the following files: .env, database.php, web.php.

2. Taking into account what you defined in the previous step:

- a) Create the routes for your application.
- b) Create the views and assign them to the routes.
- c) Test the routes and views created.

I. Resolução da Ficha

1.a) Requisitos Funcionais Detalhados

Atores do Sistema:

- **Visitante (anónimo):** consulta informação pública de conferências, oradores e sponsors.
- **Utilizador autenticado (participante):** pode gerir o seu perfil, inscrever-se em conferências, consulta/cancela inscrições.
- **Administrador:** gere conferências, oradores, tipos de orador, sponsors, associações e participantes.
- **(Opcional) Orador/Sponsor autenticado:** pode atualizar os seus próprios dados (se a política permitir).

Mapeamento (tabela **users**): userType define o perfil (e.g., Admin, Participant, Speaker, Sponsor).

FR1 – Acesso à informação das conferências

Objetivo: qualquer utilizador deve conseguir ver dados completos de conferências e respetivas relações.

FR1.1 – Listagem de conferências

- O sistema apresenta lista com: *acronym, name, location, conferenceDate* (tabela *conferences*).
- Filtros: por data, localização e pesquisa por *name/acronym*.
- Ordenação: por *conferenceDate* (padrão) e por *name*.

Critérios de aceitação

- A listagem devolve apenas registos existentes.
- Pesquisa "contém" (case-insensitive) por name/acronym.
- Paginação quando o total excede o limite configurado.

FR1.2 – Detalhe da conferência

- Página de detalhe mostra: *name, acronym, description, location, conferenceDate*.
- Mostra **Oradores ativos** (tabelas *confSpeaker* → *speakers*, com *isActive = true*) com: *name, affiliation, biography, photo, socialNetworks, pageLink, speakerType*.

- Mostra **Sponsors ativos** (tabelas *confSponsor* → *sponsors*, com *isActive = true*) com: *name*, *logo*, *category*.

Critérios de aceitação

- Relações *many-to-many* corretas via *confSpeaker* e *confSponsor*.
- Oradores/Sponsors inativos **não** aparecem.
- Se não existirem oradores/sponsors, é apresentada mensagem adequada ("Ainda sem oradores/sponsors").

FR2 – Registo de utilizadores numa conferência

Objetivo: permitir que um utilizador se inscreva numa conferência e gere as suas inscrições.

FR2.1 – Criar conta / Autenticar

- Registo com: *firstname*, *lastname*, *username*, *email*, *password* e (opcional) *description*.
- Autenticação via *email/password*.
- Atualização de *lastLogin* no sucesso do login.

Critérios de aceitação

- *email* e *username* únicos.
- Password guardada com *hash*.
- Campos obrigatórios validados; mensagens de erro claras.

FR2.2 – Inscrição numa conferência

- Utilizador autenticado pode inscrever-se em qualquer conferência.
- Cria registo em *confParticipant* com *conferenceID*, *participantID* (= *users.id*), *registrationDate* (agora).

Restrições

- **Única inscrição** por (utilizador, conferência). Chave única lógica: (*conferenceID*, *participantID*).

Critérios de aceitação

- Ao inscrever, se já existir registo, o sistema informa "Já estás inscrito".
- Após inscrever, mostra confirmação e a inscrição surge em "As minhas inscrições".

FR2.3 – Gestão de inscrições

- O utilizador vê as minhas conferências (de *confParticipant*) com *registrationDate*.
- Pode cancelar a inscrição (remoção do registo em *confParticipant*).

Critérios de aceitação

- Apenas o próprio pode cancelar a sua inscrição.
- A lista é ordenada por *conferenceDate* decrescente ou *registrationDate* (configurável).

FR3 – Gestão de oradores e sponsors (Admin)

Objetivo: admins mantêm catálogo de oradores, tipos e sponsors, e fazem associações às conferências.

FR3.1 – CRUD de Oradores (*speakers*)

- Criar/Editar: *name*, *affiliation*, *biography*, *photo (upload)*, *socialNetworks*, *pageLink*, *speakerType (FK)*, *isActive (boolean)*.
- Listar/Filtrar por *name*, *affiliation*, *speakerType*, *isActive*.
- Ativar/Desativar orador (*toggle isActive*).
- Remover orador (soft delete opcional; se não houver soft delete, bloquear remoção quando existirem associações).

Critérios de aceitação

- *speakerType* deve existir em *speakerType*.
- Upload de *photo* aceite apenas em tipos e tamanhos permitidos.
- Oradores inativos não aparecem no FR1.2.

FR3.2 – Gestão de Tipos de Orador (*speakerType*)

- CRUD de *description*.
- Impedir remoção se houver oradores com esse tipo (ou exigir reassociação).

FR3.3 – Associação Orador ↔ Conferência (*confSpeaker*)

- Admin pode associar/desassociar oradores a conferências.
- Evitar duplicados (chave única: (*conferenceID*, *speakerID*)).

FR3.4 – CRUD de Sponsors (*sponsors*)

- Criar/Editar: *name*, *logo (upload)*, *category*, *isActive*.

- **Listar/Filtrar** por *name*, *category*, *isActive*.
- **Ativar/Desativar** sponsor.
- **Remover** sponsor (mesmas regras de associação).

FR3.5 – Associação Sponsor ↔ Conferência (confSponsor)

- Admin associa/desassocia sponsors a conferências.
- Evitar duplicados (*conferenceID*, *sponsorID*).

FR3.6 – Gestão de Conferências (conferences)

- CRUD com: *acronym* (único), *name*, *description*, *location*, *conferenceDate*.
- Pesquisa e filtros por data/local.

FR3.7 – Gestão de Participantes

- Admin pode consultar participantes inscritos por conferência (*join confParticipant ↔ users*).
- Pode exportar listagem (CSV/Excel) (*opcional, mas recomendado*).
- (Opcional) Cancelar inscrição de um participante específico.

FR4 – Atualização de dados do utilizador

Objetivo: utilizadores podem manter os seus dados atualizados.

FR4.1 – Editar perfil

- Campos editáveis pelo próprio: *firstname*, *lastname*, *username*, *email*, *description*, alteração de password.
- Campos **não** editáveis pelo utilizador: *userType*, *isActive*, *lastLogin*, *creationDate*, *changeDate*.

Critérios de aceitação

- Validação de unicidade de email/*username* ao editar.
- *changeDate* atualizado automaticamente.
- Reautenticação exigida para alterar email/password.

FR4.2 – Visualizar atividade

- Ver *lastLogin* (só leitura).

- Ver lista de inscrições (ver FR2.3).

Requisitos Transversais Derivados (ainda funcionais)

Estes decorrem do domínio/estrutura de dados e suportam os FR acima.

FRT1 – Autenticação & Sessão

- Registo, *login*, *logout*, recuperação de password por email.
- Atualização de *lastLogin* no sucesso do login.

FRT2 – Autorização por perfis

- **Admin:** acesso total às secções de gestão.
- **Participante:** apenas operações sobre o próprio perfil e inscrições.
- **Visitante:** leitura pública de conferências, oradores e sponsors.
- **(Opcional) Orador/Sponsor:** pode editar os seus próprios dados (se *userType* o permitir).

FRT3 – Consistência de dados

- Unicidade: *users.email*, *users.username*, *conferences.acronym*.
- *Foreign keys:*
 - *speakers.speakerType* → *speakerType.id*
 - *users.userType* → *userType.id*
 - *confParticipant.conferenceID* → *conferences.id*
 - *confParticipant.participantID* → *users.id*
 - *confSpeaker.conferenceID* → *conferences.id*
 - *confSpeaker.speakerID* → *speakers.id*
 - *confSponsor.conferenceID* → *conferences.id*
 - *confSponsor.sponsorID* → *sponsors.id*

FRT4 – Estados "ativos"

- *speakers.isActive*, *sponsors.isActive*, *users.isActive* controlam visibilidade e acesso.
- Itens inativos não aparecem nas listagens públicas (FR1).

1.b) Base de Dados Relacional

Entidades nucleares: Conference, User, Speaker, Sponsor + tipos (*userType*, *speakerType*).

Associações:

- Conference ↔ Speaker via *confSpeaker* (N:N).
- Conference ↔ Sponsor via *confSponsor* (N:N).
- Conference ↔ User via *confParticipant* (inscrição; N:N).

Flags de atividade (isActive) em *users*, *speakers*, *sponsors* → útil para visibilidade.

Atributos chave: *acronym*, *conferenceDate*, biografia do orador, etc.

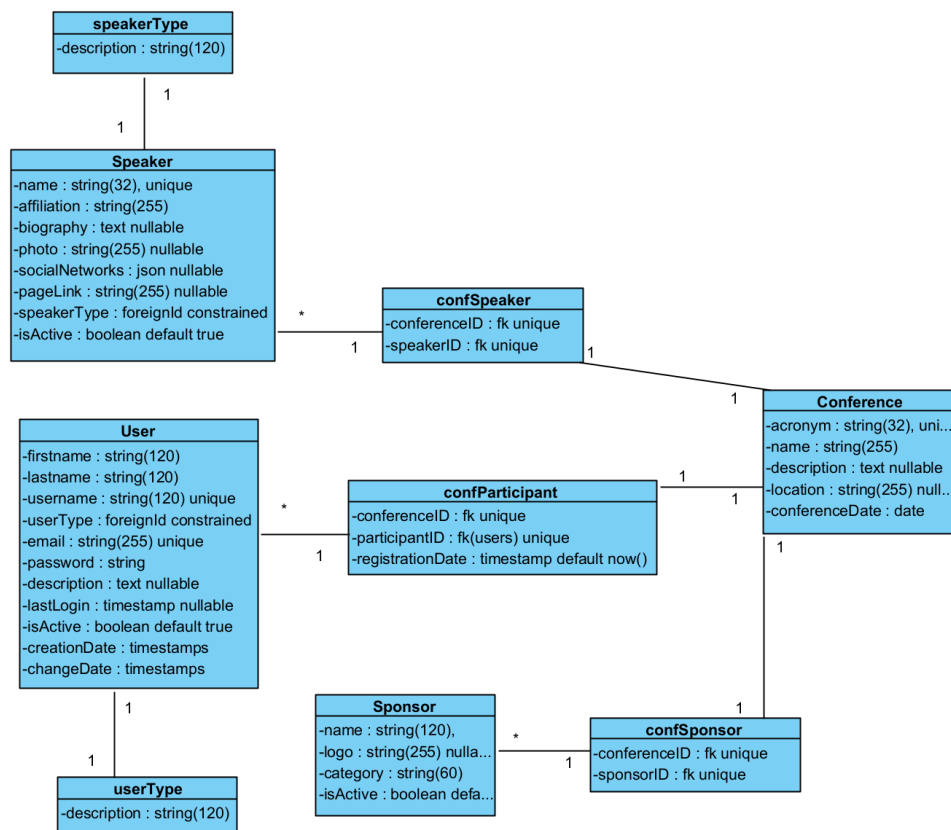
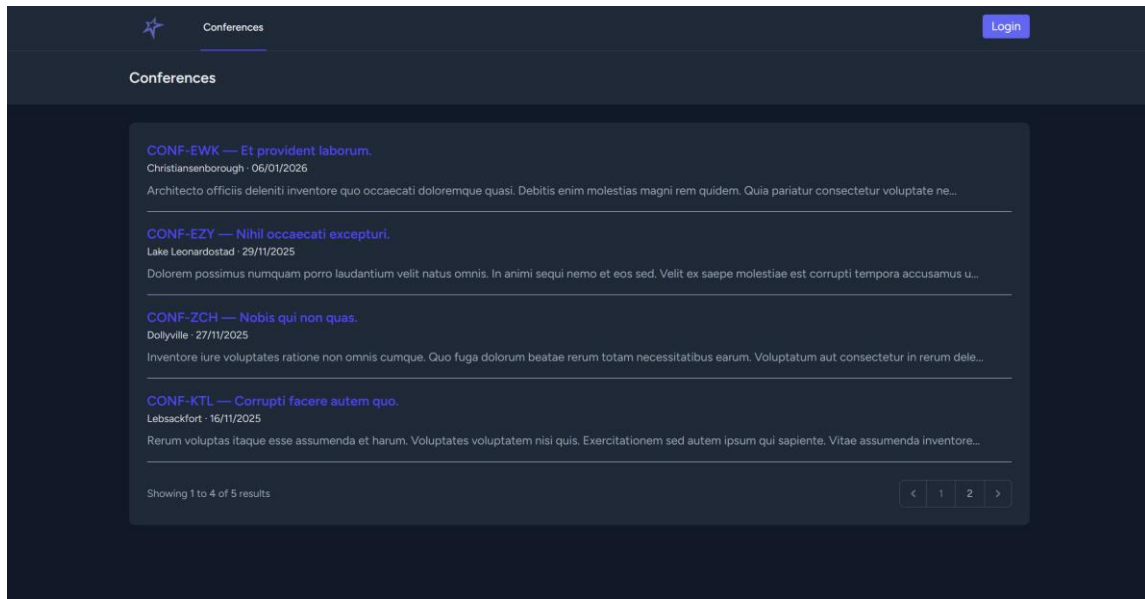


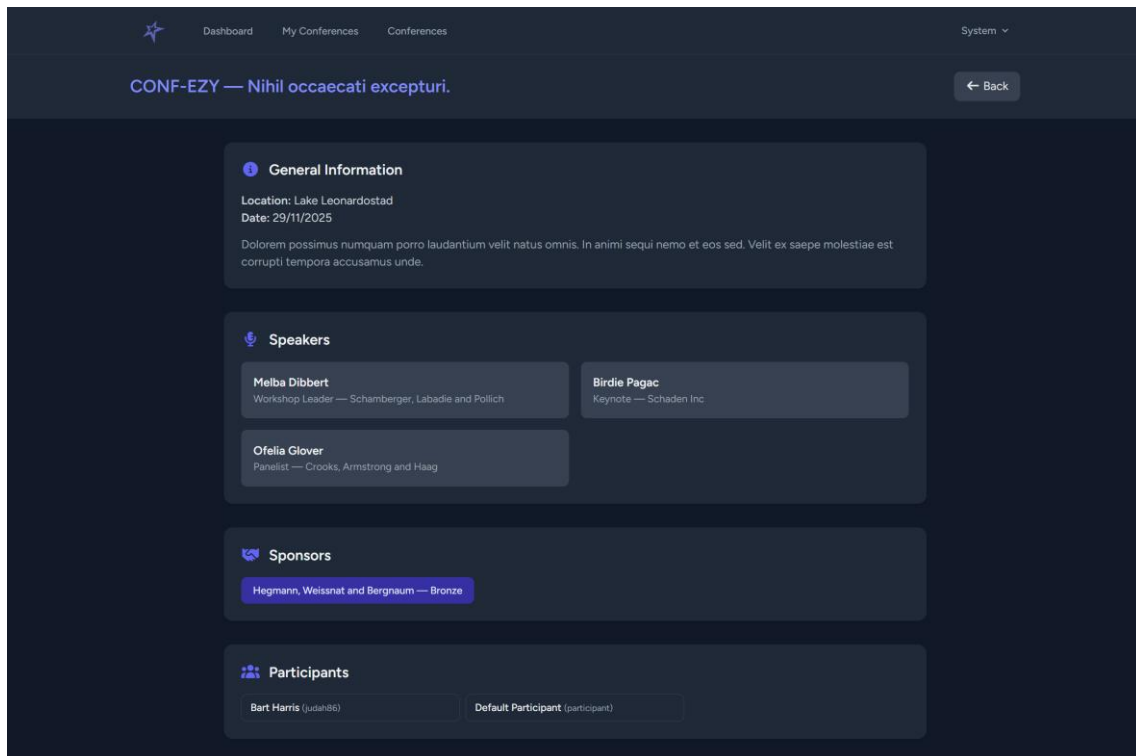
Figura 1: Base de Dados Relacional

1.c) Mockups do Projeto

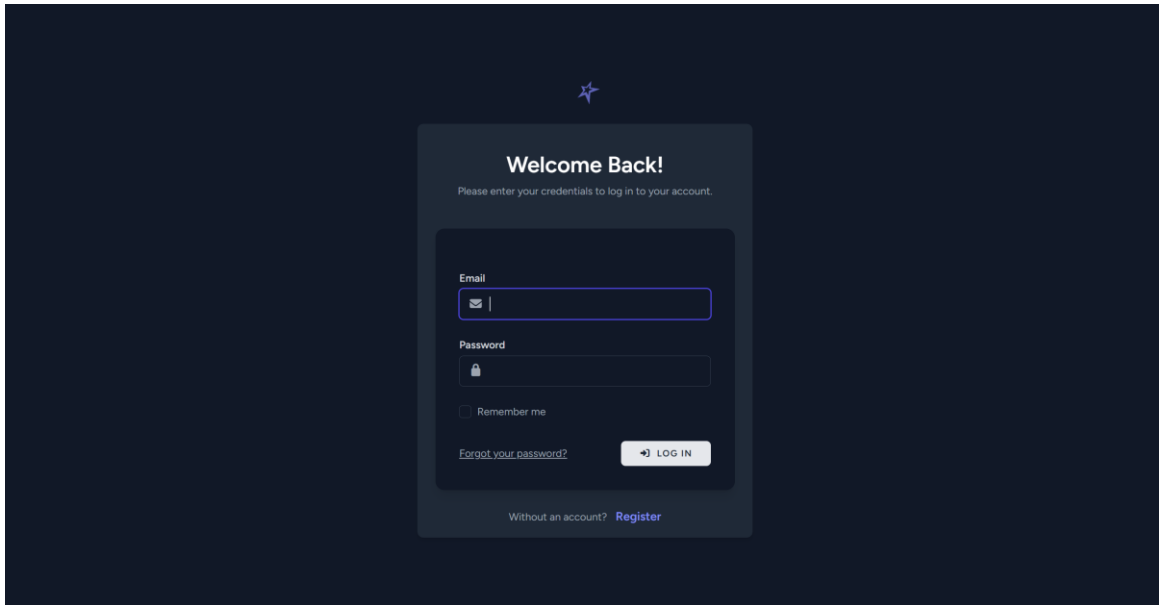
Listagem de Conferências:



Visualização de uma Conferência:



Login:



A login form titled "Welcome Back!" with a subtitle "Please enter your credentials to log in to your account." The form is centered on a dark blue background with a small star icon at the top. It contains fields for "Email" and "Password", a "Remember me" checkbox, a "Forgot your password?" link, and a "LOG IN" button. At the bottom, there is a link to "Register" for users without an account.

Welcome Back!
Please enter your credentials to log in to your account.

Email

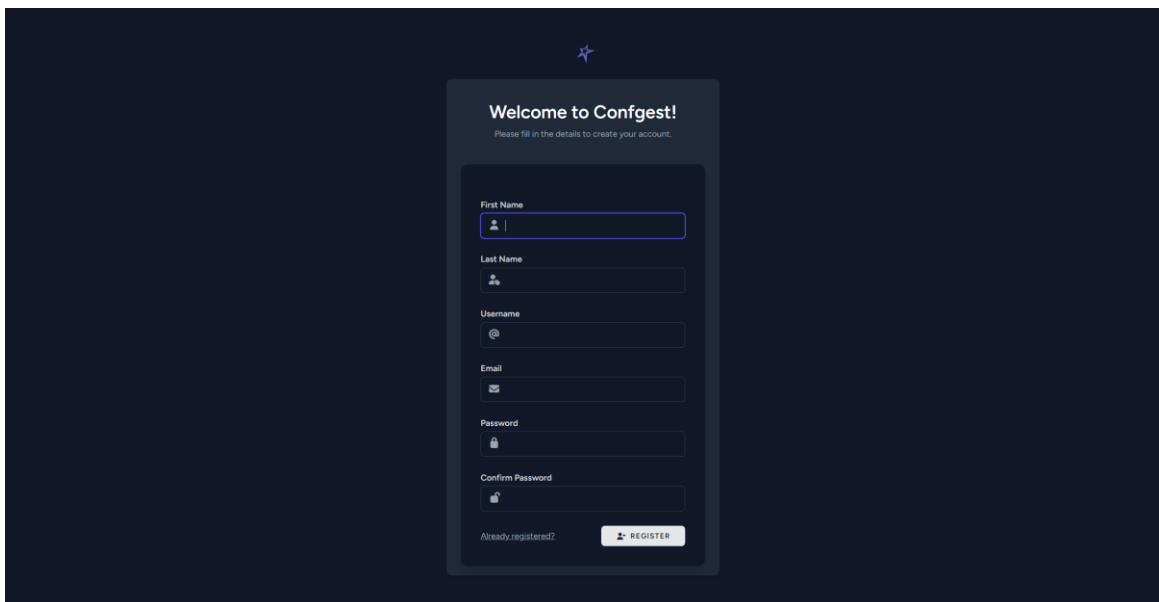
Password

☐ Remember me

[Forgot your password?](#) **LOG IN**

Without an account? [Register](#)

Registo:



A registration form titled "Welcome to Confgest!" with a subtitle "Please fill in the details to create your account." The form is centered on a dark blue background with a small star icon at the top. It contains fields for "First Name", "Last Name", "Username", "Email", "Password", and "Confirm Password", along with a "REGISTER" button. At the bottom, there is a link to "Already registered?" for existing users.

Welcome to Confgest!
Please fill in the details to create your account.

First Name

Last Name

Username


Email

Password

Confirm Password

[Already registered?](#) **REGISTER**

Meu Perfil:

 Dashboard

My Conferences

Conferences

Default ▾

Profile

Profile Information

Update your account's profile information and email address.

First Name

Default

Last Name

Participant

Username

participant

Description

Email

participant@configtest.test

SAVE

Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

New Password

Confirm Password


SAVE

Delete Account

Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain.

DELETE ACCOUNT

As Minhas Conferências:

 Dashboard

My Conferences

Conferences

Default ▾

My Conferences

Upcoming Conferences

CONF-KTL — Corrupti facere autem quo.
Lebackfort · 16/11/2025

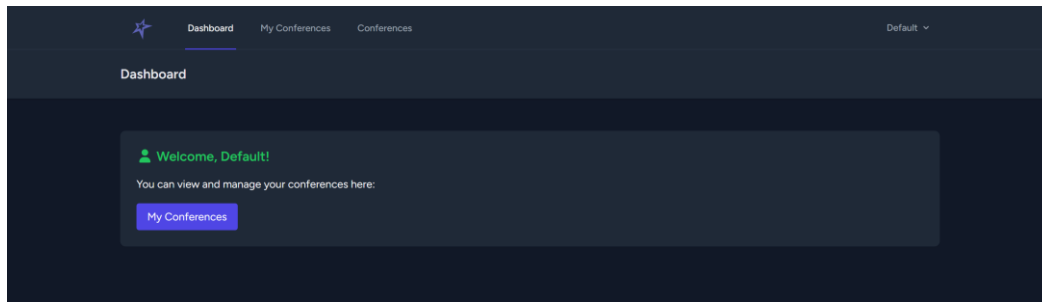
CONF-EZY — Nihil occaecati excepturi.
Lise Leonhardt · 29/11/2025

CONF-EWK — Et provident laborum.
Christiansenborough · 05/01/2026

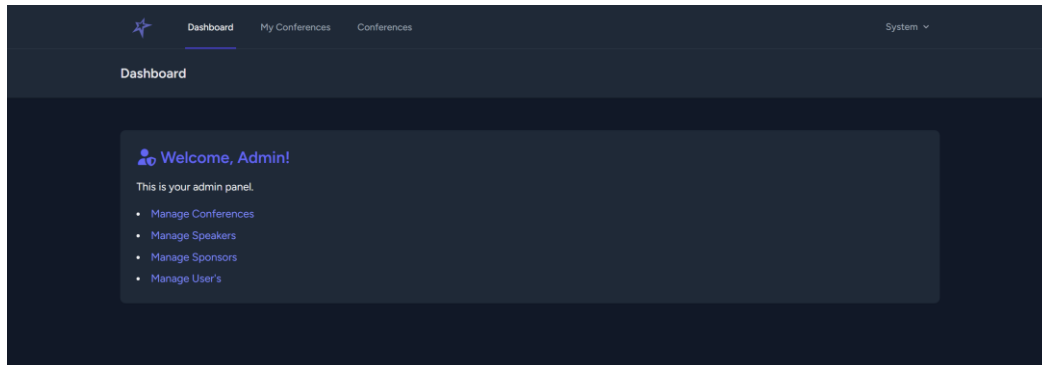
Past Conferences

Ainda não participou em nenhuma conferência.

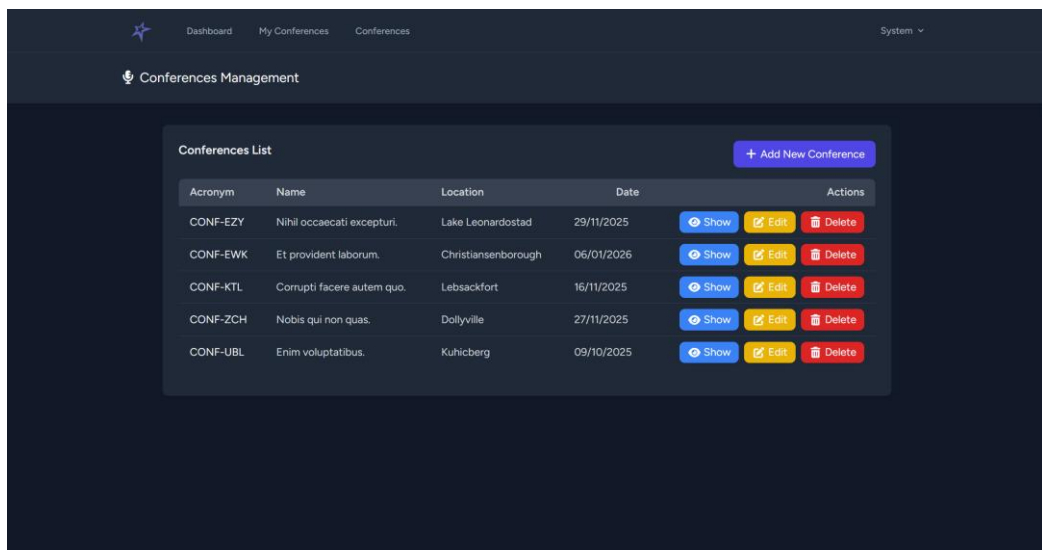
Dashboard (Utilizador Participante):



Dashboard (Administração):



Gerir Conferências:



Gerir Oradores:

Speakers Management

Speakers List

+ Add New Speaker

Name	Affiliation	Type	Action's
Missouri Abshire II	Romaguera, Wisoky and Dare	Panelist	Show Edit Delete
Dr. Hosea Reichert PhD	Bode PLC	Panelist	Show Edit Delete
Calista Johnson	Brakus Group	Panelist	Show Edit Delete
Melba Dibbert	Schamberger, Labadie and Pollich	Workshop Leader	Show Edit Delete
Toy Torphy	Romaguera Group	Panelist	Show Edit Delete

Showing 1 to 5 of 8 results

< 1 2 >

Gerir Sponsors:

Sponsors Management

Sponsors List

+ Add New Sponsor

Name	Category	Active	Action's
Orn, Rodriguez and Collier	Silver	Active	Show Edit Delete
Larson LLC	Silver	Active	Show Edit Delete
O'Kon-Rutherford	Bronze	Active	Show Edit Delete
Dicki LLC	Gold	Active	Show Edit Delete
Dickinson-Barton	Bronze	Active	Show Edit Delete

Showing 1 to 5 of 6 results

< 1 2 >

Gerir Utilizadores:

Users Management

User's List


+ Add New User

First Name	Last Name	Email	User Type	Action's
Speaker	Manager	speaker@confgest.test	Speaker	Show Edit Delete
Sponsor	Manager	sponsor@confgest.test	Sponsor	Show Edit Delete
Aurelie	Wiza	osinski.iring@example.net	Participant	Show Edit Delete
Alycia	Block	morar.kitty@example.net	Participant	Show Edit Delete
Kelsie	Grimes	reanna.pollich@example.org	Participant	Show Edit Delete

Showing 1 to 5 of 15 results

< 1 2 3 >

Exemplo de Formulário de Criação/Edição:

 [Dashboard](#) [My Conferences](#) [Conferences](#) System ▾

[➕ Create New User](#)

First Name *

Last Name *

Username *

Email *

Description *

Password *

Confirm Password *


User Type *

Admin ▾

☒ Active

Back

CREATE USER

 [Dashboard](#) [My Conferences](#) [Conferences](#) System ▾

[✎ Edit Conference: Nihil occaecati excepturi.](#)

Acronym *

CONF-EZY

Conference Name *

Nihil occaecati excepturi.

Description

Dolorem possimus numquam porro laudantium velit natus omnis. In animi sequi nemo et eos sed. Velit ex saepe molestiae est corrupti tempora accusamus unde.

Location *

Lake Leonardstad

Conference Date *

29/11/2025

Speakers

Missouri Abshire II
Dr. Hosea Reichert PhD
Callista Johnson
Melba Dibbert

Sponsors

Larson LLC
O'Kon-Rutherford
Dicki LLC
Dickinson-Barton
Hegmann, Weissnat and Bergnaum

Back

UPDATE CONFERENCE

1.d) Criação do projeto Laravel

O projeto foi criado usando o Composer, com o comando:

```
composer create-project laravel/laravel ConfGest
```

Após a criação, entrei na pasta do projeto:

```
cd ConfGest
```

Configurei a ligação à base de dados PostgreSQL no ficheiro .env.

Testei a instalação com:

```
php artisan serve
```

e o servidor respondeu corretamente.

1.e) Exploração dos ficheiros principais

- .env
Contém variáveis de ambiente. Aqui defini:

Nome da aplicação → APP_NAME=ConfGest

URL → APP_URL=http://confgest.test

Base de dados → DB_CONNECTION=pgsql, DB_DATABASE=confgest_dev,
DB_USERNAME=postgres, DB_PASSWORD=guga
- config/database.php
Configuração central das ligações a base de dados.
Lê os valores do .env.
No nosso caso, usamos o driver pgsql.
- routes/web.php
Define as rotas que respondem a pedidos web.
 - Inicialmente tinha só a rota para welcome.blade.php.

2. Considerando o que já definiu na etapa anterior:

a) Crie as rotas para a sua aplicação.

b) Crie as visualizações e atribua-as às rotas.

c) Teste as rotas e visualizações criadas.

Resposta: O desenvolvimento total da aplicação encontra-se presente no GitHub, através do link: <https://github.com/gustavogiao/ConfGest>

Aqui foi desenvolvido tanto a parte de *back-end*, com a criação dos models, factories, migrations, seeder's, como também os seus controladores, middlewares, form requests. Também desenvolvido as rotas para autenticação e rotas para as views criadas (index, edit, create, show).

Para correr a aplicação, basta criar um .env à sua maneira:

- `DB_CONNECTION=pgsql`
`DB_HOST=127.0.0.1`
`DB_PORT=5432`
`DB_DATABASE=confgest_dev`
`DB_USERNAME=postgres`
`DB_PASSWORD=1234`

Está preparado para correr tanto em mysql com pgSQL, não se esqueça de criar a base de dados antes, para que consiga depois fazer *php artisan migrate*. Se quiser rodar seeder's também o pode fazer com o *php artisan db:seed*. Melhor ainda é fazer desde logo, *php artisan migrate:fresh --seed* para correr as migrações e seeder's tudo de uma vez e fica com a aplicação totalmente funcional.

Boas festas e boas conferências.

Assinado: Gustavo Gião.

Worksheet II

This worksheet assumes you have completed all the exercises in the previous worksheet. The following tasks will extend your project with additional functionalities and give access to dynamic data. Tasks

1. Database Creation Use migrations to create the database that supports your application, based on the relational model you defined previously.
2. Models with Eloquent Based on your data structure: 2.a. create the models using the Eloquent ORM; 2.b. define which attributes should be fillable; 2.c. update the code associated with the routes so that they interact with the database.
3. Seeding and Factories With the models in place:
 - 3.a. populate the database automatically with initial values;
 - 3.b. use factories to generate sample data.
4. Tinker Use Tinker to interact with the database directly.

Resposta: Feito no código do projeto: lá criei as migrações, modelos com eloquent, seeders, factories, com população automática e valores iniciais, para que quando se faz "php artisan migrate:fresh --seed" ele contrua o projeto todo direitinho já com dados da BD introduzidos automaticamente.

Utilizei também o Tinker para brincar um pouco com a base de dados.