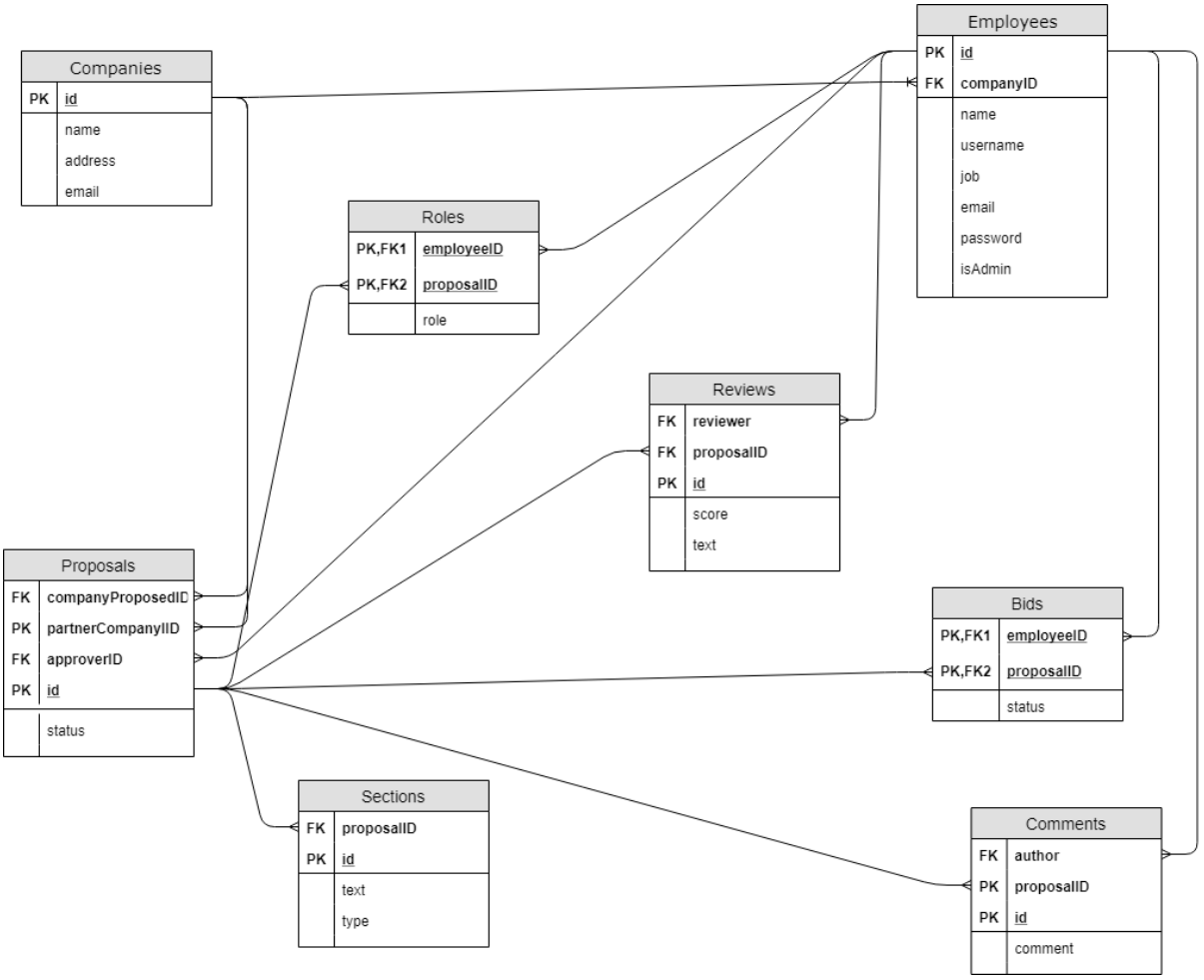
Modelo entidade-relação



André Lopes - 45617
Nelson Coquenim - 45694
Simão Dolores - 45020

Políticas de Segurança

Nesta aplicação existem os seguintes roles:

- **SuperAdmin**: Membro da Ecma que cria as empresas e adiciona ao sistema os admins das respectivas, fornecendo-lhes as credenciais (que devem ser mudadas);
- **Admin**: Membro de uma empresa responsável por gerir a informação desta;
- **Employee**: Membro de uma empresa;
- **TeamMember**: *Role* atribuído a um employee no contexto de uma proposta. Este *role* subdivide-se noutros dois:
    - **StaffMember**: Membro da empresa que fez a proposta;
    - **PartnerMember**: Membro da empresa parceira cuja proposta de destina;
- **Approver**: *Role* atribuído a um employee no contexto de uma proposta. Este é responsável por gerir o estado da proposta, tendo a última palavra em relação à aprovação da mesma, após analisar as *reviews* submetidas pelos **TeamMember**.

Privilégios por *Role*:

Employee:

- Obter uma lista das Bids efetuadas;
- Obter a sua informação e de outros empregados;
- Obter uma lista de todos os empregados;
- Obter as propostas onde o mesmo pertence à equipa;
- Atualizar a sua informação;
- Obter uma lista dos admins pertencentes a uma empresa;
- Obter uma lista de empresas;
- Obter uma empresa em específico;
- Obter uma lista de empregados de uma empresa;
- Adicionar uma proposta;

TeamMember:

- Tudo o que os empregados já possuem permissão;
- Adicionar/Remover uma bid (somente a bid que lhe pertence);
- Adicionar/Remover/Actualizar um comentário de uma proposta;
- Obter a lista de revisões de uma proposta;
- Adicionar/Remover/Atualizar uma review de uma proposta;
- Obter todas as secções de uma proposta;
- Obter a lista de revisões de uma proposta;
- Obter os utilizadores que pertencem à equipa da proposta;
- Obter a proposta a que pertence;

André Lopes - 45617
Nelson Coquenim - 45694
Simão Dolores - 45020

StaffMember:

- Adicionar/Remover/Atualizar secções de uma proposta;
- Adicionar um Partner ou Staff à Team;
- Remover uma proposta;
- Remover-se como Staff de uma proposta (somente o próprio utilizador)

PartnerMember: (Não tem mais permissões para além das herdadas em **TeamMember**)

Approver:

- Alterar o estado de uma bid;
- Obter uma lista de todas as bids pertencentes a uma proposta (somente à proposta na qual é approver);
- Obter a lista de revisões de uma proposta;
- Alterar uma proposta;
- Obter a lista de revisões de uma proposta;
- Obter os utilizadores que pertencem à equipa da proposta;
- Obter a proposta a que pertence;

Admin:

- Atualizar/Remover uma companhia;
- Adicionar um admin a uma empresa;
- Adicionar/Remover um empregado a uma empresa;

Super Admin:

- Criar/Remover uma companhia;
- Adicionar/Remover um admin a uma empresa;

André Lopes - 45617
Nelson Coquenim - 45694
Simão Dolores - 45020

Distribuição do trabalho

Nesta seção segue-se uma breve descrição do trabalho efectuado por cada membro.

Em primeiro lugar efectuou-se a especificação, em Swagger e o ER, sendo que todos os elementos do grupo contribuíram activamente para a sua realização. O que permitiu agilizar o desenvolvimento da lógica da aplicação.

Na implementação da lógica o contributo inicial e importante foi realizado pelo Simão, o que permitiu a definição de um “esqueleto” da construção do código, sendo que o restante trabalho nesta fase foi distribuído equitativamente.

Na fase de definição das políticas de segurança o grupo primeiramente definiu o conjunto de anotações necessários e de seguida a implementação destas foi distribuído de forma, aproximadamente, igual pelos os elementos.

Existiu uma fase após a implementação da lógica da aplicação em que foi necessário um *refactor*. Este foi realizado pelo André, e daqui surgiu uma camada de abstração adicional na organização do código: os *brokers*. Esta permite um maior desacoplamento dos serviços e evita repetição de código nos diversos serviços.

Na fase de testes realizou-se também de forma equitativa os testes JUnit. Adicionalmente foram efectuados, pelo André e Nelson, testes com a web api do Swagger (disponível em localhost:8080/swagger-ui.html) para testar o fluxo geral da aplicação

André Lopes - 45617
Nelson Coquenim - 45694
Simão Dolores - 45020

swagger

Select a spec default

Ecma Events

API TOS

[Base URL: localhost:8080/]

http://localhost:8080/v2/api-docs

Application that allows Ecma to manage events envolving other partner companies

Terms of service

Ecma - Website

Send email to Ecma

License of API

companies

GET	/companies	Get the list of all companies
POST	/companies	Add a new partner company to the collection
GET	/companies/{companyId}	Find company by ID
PUT	/companies/{companyId}	Update an existing company

DELETE /companies/{companyId} Delete company with the ID provided

GET /companies/{companyId}/admins Get the list of all employees of a company

POST /companies/{companyId}/admins Add a new admin to the company

DELETE /companies/{companyId}/admins/{adminId} Delete Admin with the ID provided

GET /companies/{companyId}/employees Get the list of all employees of a company

POST /companies/{companyId}/employees Add a new employee to the collection

DELETE /companies/{companyId}/employees/{employeeId} Delete employee with the ID provided

employees

GET /employees Get the list of all employees

GET /employees/{employeeId} Find employee by ID

PUT /employees/{employeeId} Update employee with ID provided

GET /employees/{employeeId}/bids Get the list of all bids where this employee bid

GET `/employees/{employeeId}/partnerproposals` Get the list of all proposals where this employee is partner

GET `/employees/{employeeId}/staffproposals` Get the list of all proposals where this employee is staff

proposals

POST `/proposals` Add a new proposal to the system

GET `/proposals/{proposalId}` Find proposal by ID

PUT `/proposals/{proposalId}` Updates proposal

DELETE `/proposals/{proposalId}` Delete proposal with the ID provided

GET `/proposals/{proposalId}/partnermembers/` Get all partner members by the proposal ID

POST `/proposals/{proposalId}/partnermembers/` Add a new partner member to the proposal

DELETE `/proposals/{proposalId}/partnermembers/{partnerId}` Delete partner with the ID provided

GET `/proposals/{proposalId}/staff/` Get all staff members by the proposal ID

POST `/proposals/{proposalId}/staff/` Add a new staff member to the proposal

DELETE /proposals/{proposalId}/staff/{staffId} Delete Staff with the ID provided

bids

GET /proposals/{proposalId}/bids Get all bids by the proposal ID

POST /proposals/{proposalId}/bids Add a new Bid to the proposal

PUT /proposals/{proposalId}/bids/{employeeId} Update bid with ID provided

DELETE /proposals/{proposalId}/bids/{employeeId} Delete Bid with the ID provided

comments

GET /proposals/{proposalId}/comments/ Get all comments by the proposal ID

POST /proposals/{proposalId}/comments/ Add a new comment to the proposal

PUT /proposals/{proposalId}/comments/{commentId} Update coments with ID provided

DELETE /proposals/{proposalId}/comments/{commentId} Delete comment with the ID provided

reviews

GET /proposals/{proposalId}/reviews Get all reviews by the proposal ID

POST /proposals/{proposalId}/reviews/ Add a new review to the proposal

PUT /proposals/{proposalId}/reviews/{reviewId} Update reviews with ID provided

DELETE /proposals/{proposalId}/reviews/{reviewId} Delete review with the ID provided

sections

GET /proposals/{proposalId}/sections/ Get all sections by the proposal ID

POST /proposals/{proposalId}/sections/ Add a new section to the proposal

PUT /proposals/{proposalId}/sections/{sectionId} Update sections with ID provided

DELETE /proposals/{proposalId}/sections/{sectionId} Delete section with the ID provided

Models

Bid {
- pk — **BidKey** {
  - bidder — **Employee** {...}
  - proposal — **Proposal** {...}

  }
- status — string

}

BidKey {
- bidder — **Employee** {...}
- proposal — **Proposal** {...}

}

Comment {
- author — **Employee** {...}
- comment — string
- id — integer($int64)
- proposal — **Proposal** {...}

}

Company {
- address — string
- email — string
- id — integer($int64)
- name — string

}

Employee {

admin	boolean
email	string
id	integer($int64)
job	string
name	string
username	string

}

EmployeeWithPw {

admin	boolean
company	Company {...}
email	string
job	string
name	string
password	string
username	string

}

Proposal {

approver	Employee {...}
companyProposed	Company {...}
id	integer($int64)
partnerCompany	Company {...}
status	string

}

Review {

author	**Employee** {...}
id	integer($int64)
proposal	**Proposal** {...}
score	string Enum: [BAD, POOR, OK, GREAT, EXCELENT]
text	string

}

Section {

id	integer($int64)
proposal	**Proposal** {...}
text	string
type	string

}

SimpleEmployee {

admin	boolean
email	string
id	integer($int64)
job	string
name	string
username	string

}