

MC536

Trabalho Final

Cluster F:

Danilo Valente
Gabriel Bounous
Hélder Rocha
Lucas Souza
Thiago Santos

Proposta

Atualmente existem dois serviços conhecidos para a busca de ofertas de emprego e classificação de empresas, sendo eles o StackOverflow Careers e o Glassdoor. A aplicação desenvolvida como resultado deste projeto tem como objetivo servir de ponte entre estes dois serviços, de forma que a partir das ofertas publicadas no SO Careers e das avaliações de cada empresa cadastrada no GlassDoor seja possível cruzar ambas informações e oferecer ao usuário as ofertas de emprego que sejam mais adequadas ao seu perfil.

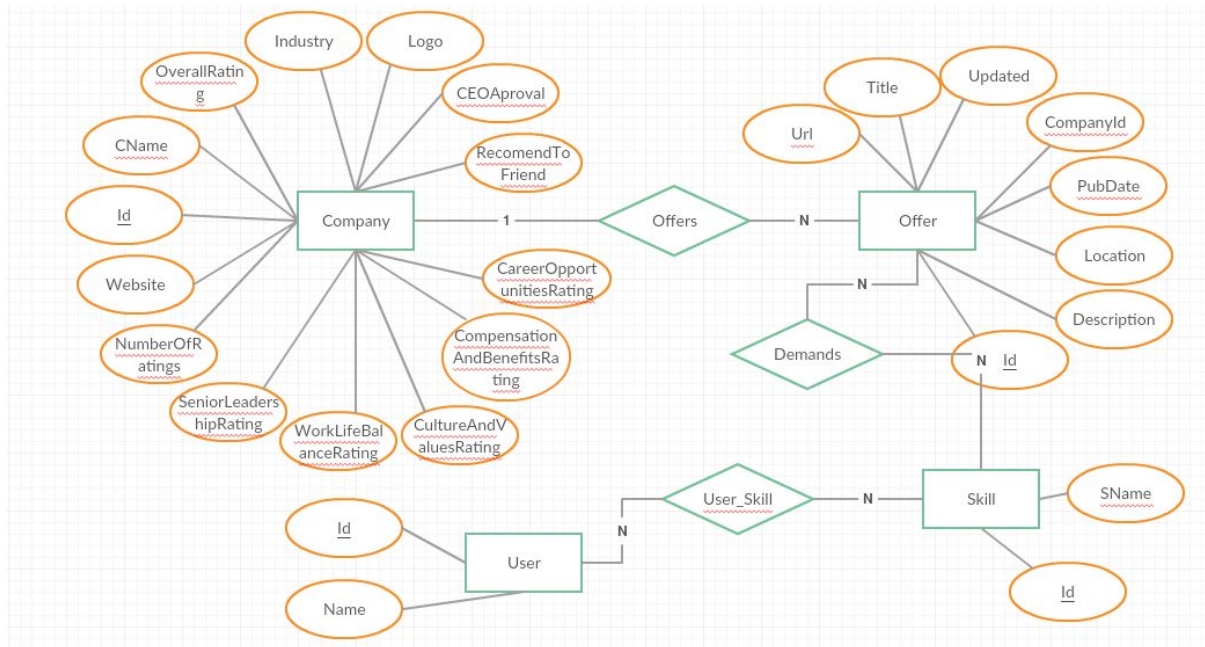


Logo do StackOverflow Careers, base de dados de onde são extraídas as ofertas de emprego.



Logo do Glassdoor, base de dados de onde são extraídas as informações de cada empresa, bem como sua classificação em diversos requisitos.

Modelo Conceitual



Modelo Relacional

Company(Id, CName, Website, Industry, NumberOfRatings, Logo, OverallRating, CultureAndValuesRating, SeniorLeadershipRating, CompensationAndBenefitsRating, CareerOpportunitiesRating, WorkLifeBalanceRating, RecomendToFriend, CEOAproval)

Offer(Id, Title, Description, Location, Url, PubDate, Updated, CompanyId)
 Offer[CompanyId] -> Company[Id]

Skill(Id, SName)

User(Id, Name)

Demands(OfferId, SkillId)
 Demands[OfferId] -> Offer[Id]
 Demands[SkillId] -> Skill[Id]

User_Skill(UserId, SkillId)
 User_Skill[UserId] -> User[Id]
 User_Skill[SkillId] -> Skill[Id]

Arquitetura

A versão final do trabalho é uma aplicação web desenvolvida dividida em duas camadas principais:

- **Front-end:** É a interface gráfica da aplicação, desenvolvida utilizando JavaScript, jQuery, Bootstrap e HTML, principalmente. Todo o conteúdo dinâmico é consumido do webservice através de chamadas Ajax.
- **Back-end:** É a camada de baixo nível da aplicação, responsável pela lógica de negócio e pela conexão com o banco de dados. É um webservice MVC e foi desenvolvida em Java utilizando os frameworks Spring MVC e Hibernate, sendo este último o responsável pelo mapeamento entre objetos Java e entidades relacionais (ORM).
- **Banco de dados:** A aplicação utiliza o MySQL como SGBD para o banco de dados relacional do sistema.

```
@ManyToOne(fetch = FetchType.EAGER)
@JoinColumn(name = "CompanyId", nullable = false, insertable = false, updatable = false)
private Company company;

@ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
@JoinTable(
    name = "Demands",
    joinColumns = { @JoinColumn(name = "OfferId", nullable = false) },
    inverseJoinColumns = { @JoinColumn(name = "SkillId", nullable = false)}
)
private List<Skill> skills;
```

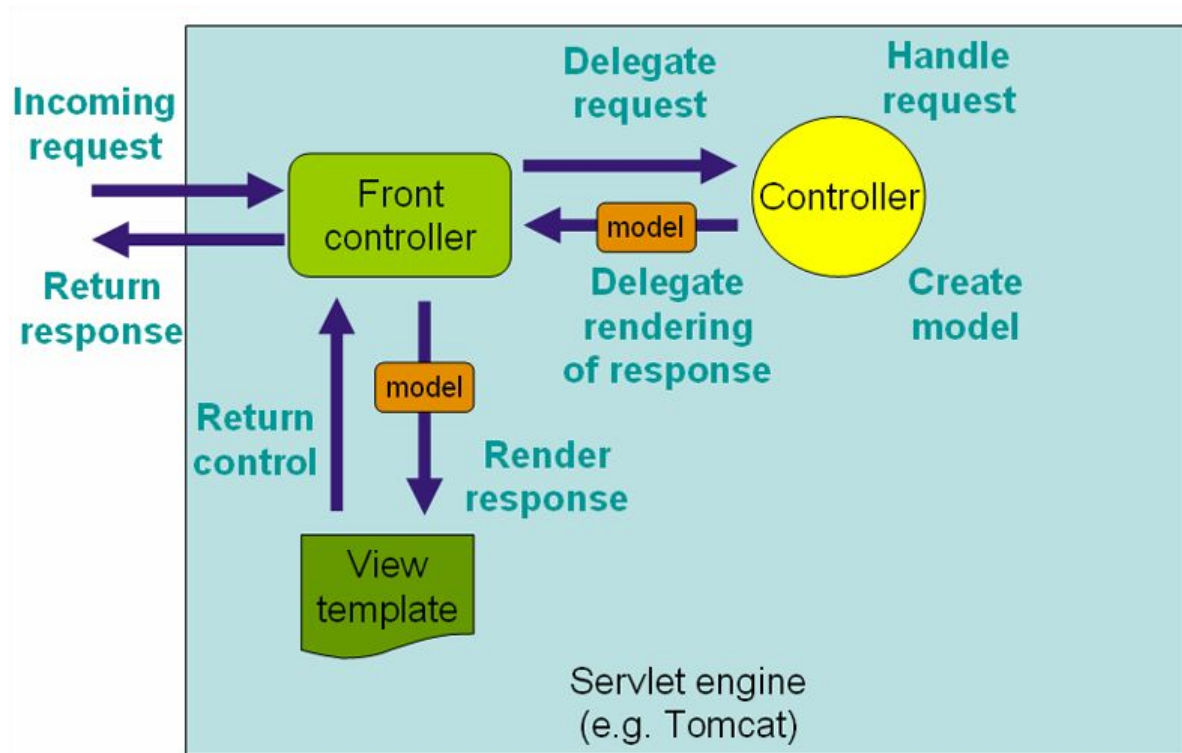
Mapeamentos ManyToMany e ManyToOne entre as entidades Offer/Skill e Offer/Company utilizando as funcionalidades do Hibernate.



Mapeamento entre entidades relacionais e objetos Java.

```
private static final String SEARCH_QUERY = "" +
    "select o from Offer as o" +
    " inner join o.skills as s" +
    " with s.name in (:skills) or :any_skill is true" +
    " inner join o.company as c" +
    " order by (" +
    " c.overallRating * :overallRatingWeigth + " +
    " c.cultureAndValuesRating * :cultureAndValuesRatingWeight +" +
    " c.seniorLeadershipRating * :seniorLeadershipRatingWeight +" +
    " c.compensationAndBenefitsRating * :compensationAndBenefitsRatingWeight +" +
    " c.careerOpportunitiesRating * :careerOpportunitiesRatingWeight +" +
    " c.workLifeBalanceRating * :workLifeBalanceRatingWeight +" +
    " c.recomendToFriend * :recomendToFriendWeight" +
    " ) desc," +
    " o.publicationDate desc";
```

Query HQL (Hibernate Query Language) de recomendação de ofertas de emprego com base nos critérios de busca desejados.



Fluxo da modelagem MVC (Model-View-Controller) providenciada pelo Spring MVC.