

Projeto Recruta-IF

Alison da Silva Bueno

Sistemas da Informação - Faculdades da Indústria -
Av. Rui Barbosa, 5.881, Afonso Pena I 83.045-350
alissondasilvabueno@gmail.com

Everton Luiz Sausen

Sistemas da Informação - Faculdades da Indústria -
Av. Rui Barbosa, 5.881, Afonso Pena I 83.045-350
everton.sausen@hotmail.com

Joziel Alves de Souza

Sistemas da Informação - Faculdades da Indústria -
Av. Rui Barbosa, 5.881, Afonso Pena I 83.045-350
jozielalvess@hotmail.com

Anderson José de Souza Inácio

Sistemas da Informação - Faculdades da Indústria -
Av. Rui Barbosa, 5.881, Afonso Pena I 83.045-350
nosrednawall@gmail.com

Fernando André de Lima

Sistemas da Informação - Faculdades da Indústria -
Av. Rui Barbosa, 5.881, Afonso Pena I 83.045-350
fernando_andre23@hotmail.com

Resumo *Este artigo tem como objetivo descrever os principais pontos do projeto apresentado pelo grupo, citado acima, neste semestre. O Recruta-If apresenta-se como uma solução simples e objetiva para auxiliar no processo de recrutamento interno de uma instituição. O software tende a facilitar a divulgação e a seleção de candidatos para vagas internas, assim otimizando o trabalho e o tempo da equipe de recrutamento da empresa.*

Palavras-Chave: Recrutamento interno, Software, Recruta-IF

Abstract *This article aims to describe the main points of the project presented by the group, mentioned above, in this semester. Recruit-If presents itself as a simple and objective solution to assist in the internal recruitment process of an institution. The software tends to facilitate the dissemination and selection of candidates for internal positions, thus optimizing the work and time of the company's recruiting team.*

Keywords: Internal Recruitment, Software, Recruit-IF

1 Introdução

A globalização somada ao surgimento de novas tecnologias tornou o mercado de trabalho um ambiente predatório, onde empresas de todos os portes se digladiam em busca dos melhores profissionais disponíveis. Contratar um profissional através de seleção externa é sempre um risco, pois a empresa deve ter em mente que irá gastar tempo e, como consequência, recursos financeiros para treinar e adequar o novo integrante a instituição. O risco está justamente no fato de que o mesmo

profissional possa vir a não se adaptar a cultura organizacional da empresa ou ainda aceitar outra oferta de emprego.

Muitas empresas têm procurado saídas para diminuir o risco de perdas com contratações malsucedidas, e muitas tem encontrado como solução recrutar internamente funcionários que procuram evoluir dentro da mesma. O recrutamento interno possibilita que profissionais capacitados possam suprir vagas e ascender dentro da instituição. As principais vantagens deste tipo de recrutamento é que são menos suscetíveis a riscos, visto que uma vez o

funcionário já está adaptado as políticas da empresa e que os gestores e recrutadores conhecem o perfil de trabalho deste profissional e, portanto, é mais barata para a empresa. Outra vantagem deste tipo de contratação é que ela gera motivação para o funcionário recrutado e também motiva outros funcionários a continuar em constante aperfeiçoamento.

É nesse contexto que surge o projeto Recruta-if, cujo o objetivo é atender empresas, que possuam o desejo de criar um processo de recrutamento interno ou querem otimizar o processo já existente, tenham uma alternativa simples e funcional.

1.1 Objetivo Geral

Este projeto tem como objetivo a criação de uma ferramenta web capaz de otimizar o processo de recrutamento interno de uma instituição.

1.2 Objetivos Específicos

O desenvolvimento da ferramenta Recruta-If tem como objetivos específicos:

- Levantar os requisitos base para o desenvolvimento do projeto;
- Desenvolver e aplicar as competências com programação Java para aplicações web;
- Desenvolver e ampliar as habilidades com linguagem UML;
- Conhecer a metodologia de desenvolvimento rápido RUP;
- Adquirir conhecimento com APIs e Frameworks;
- Desenvolver e ampliar o conhecimento com orientação a objetos.

1.3 Metodologia

Por orientação do professor do projeto integrador, foi estipulado o uso da metodologia RUP para o desenvolvimento da aplicação.

RUP (Rational Unified Process) é um processo de engenharia de software que fornece

uma abordagem disciplinada para assumir tarefas e responsabilidades dentro de uma organização de desenvolvimento, cujo objetivo é assegurar a produção de software de alta qualidade dentro de prazos e orçamentos previsíveis (Kruchten 2003, pág. 14). Derivado dos trabalhos sobre UML e do Processo Unificado de Desenvolvimento de Software, ele traz elementos de todos os modelos genéricos de processo, apoia a interação e ilustra boas práticas de especificação e projeto (Sommerville 2007, pág. 54).

2. Descrição Técnica

O desenvolvimento do software ficou dividido em 6 etapas que são:

1. Criação das classes de Modelo ou entidade, que são as classes dos objetos principal propriamente dito e também são utilizadas como base para criar as tabelas nos bancos.
2. Criação das classes DAO (Data Access Object) que são as classes responsáveis por toda a interação da aplicação com o ou os Bancos de dados.
3. Configuração do Banco de dados e a configuração dele com a aplicação, e testes de persistência dos dados no BD.
4. Desenvolvimento das classes de Rest, que são os controladores da aplicação, sendo responsáveis por receber uma requisição HTTP, seja GET, POST, PUT ou DELETE e efetuar as regras de negócio com os objetos para depois chamar as classes DAO para fazer a persistência dos dados. Incluindo também testes nessas classes.
5. Desenvolvimento das telas, utilizando primariamente apenas HTML5 e CSS3, entretanto é bem provável que essa etapa sofra muitas alterações.
6. Implementar as funcionalidades ou endereços das classes REST para as telas criadas.

2.1 Exemplo de códigos implementados

Os códigos já desenvolvidos são: Entidade ou Modelo, DAO e REST.

```
//anotações
@Entity
@Table(name = "setor")
@XmlRootElement
public class Setor implements
    Serializable {

    private static final long
    serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy =
        GenerationType.AUTO)

    @Column(name = "id", updatable =
    false, nullable = false)
    private Long id;

    private String nome;
    // getters and setters
```

Figura 1: Exemplo de código de uma classe de entidade, sem os imports, gatters e setters.

```
@Stateless
public class SetorDao {

    @PersistenceContext(unitName =
    "recrutaif-persistence-unit")
    private EntityManager em;

    public void create(Setor entity) {
        em.persist(entity);
    }

    public void deleteById(Long id) {
        Setor entity = em.
        find(Setor.class, id);
        if (entity != null) {
            em.remove(entity);
        }
    }

    public Setor findById(Long id) {
        return em.find(Setor.class, id);
    }

    public Setor update(Setor entity) {
        return em.merge(entity);
    }
}
```

Figura 2: Exemplo de código de uma classe DAO

```
@Stateless
@Path("/setores")
public class SetorEndpoint {

    @Inject
    SetorDao dao;

    @POST
    @Consumes("application/json")
    public Response create(Setor entity) {
        dao.create(entity);
        return Response.created(
            UriBuilder
            .fromResource
            (SetorEndpoint.class)
            .path(String.valueOf
            (entity.getId()))
            .build().build();
        }

    @DELETE
    @Path("/{id:[0-9][0-9]*}")
    public Response
    deleteById(@PathParam("id") Long id) {
        dao.deleteById(id);
        return Response.noContent().build();
    }

    @GET
    @Path("/{id:[0-9][0-9]*}")
    @Produces("application/json")
    public Response
    findById(@PathParam("id") Long id) {

        Setor entity = dao.findById(id);

        return Response.ok(entity).build();
    }
}
```

Figura 3: Exemplo de uma classe Rest.

2.2 Especificações e tecnologias utilizadas

Software está sendo desenvolvido utilizando as especificações abaixo:

JavaEE: O *Java Enterprise Edition* um conjunto de regras bem claras de como deve funcionar uma aplicação web Java, nesse caso utilizamos o Servidor de Aplicação Wildfly que foi construído com base no JavaEE.

JPA: O *Java Persistence Api* conjunto de regras e especificações no que tangem sobre Banco de dados Relacionais, seja o acesso, criação de tabelas e ou persistência de dados, estão todos descritos nele.

REST: O *Representational State Transfer* é um estilo de arquitetura que define um conjunto de propriedades e restrições baseadas no HTTP, sendo geralmente chamado de Rest ou Restful os aplicativos que utilizam essa arquitetura. Estamos utilizando o framework Jersey como implementação do Rest.

3. Referência teórica

Com as novas tecnologias da informação e comunicação, há uma grande mudança na vida das pessoas e das organizações, não poderia ser diferente na forma de se procurar uma vaga no mercado de Trabalho e de se oferecer essas vagas, cada dia mais as pessoas recorrem a internet para procurar novas oportunidades profissionais, também os profissionais de recursos humanos nas empresas ou consultoras de recrutamento e seleção, utilizam funcionalidades de ofertas de emprego e recrutamento online.

Com base nessas informações desenvolvemos o Recruta-If voltado para o recrutamento interno com foco na facilidade e agilidade de seleção, ao longo da produção do projeto utilizamos para o desenvolvimento do sistema a metodologia RUP, que tem como seus fundamentos a disciplina, responsabilidade e a divisão de tarefas com foco no desenvolvimento de um software de alta qualidade, com as execuções de seus prazos e orçamentos. A parte documental foi feito com base na UML que dita as boas práticas para um projeto.

4. Considerações finais

A solução desenvolvida no Recruta-if, demonstra que o uso de ferramentas customizadas e elaboradas sob medida

contribuem para que empresas encontrem mais facilmente o talento que procuram. Em um mercado cada vez mais competitivo um profissional bem alocado pode ser o diferencial para uma equipe de trabalho.

Observou-se também que entre outras vantagens o sistema consiste em um método organizado de seleção de pessoas, que privilegia a formação acadêmica e torna o processo mais fluido e eficaz.

Não obstante a metodologia utilizada é a principal responsável pelo andamento das etapas e pela execução do cronograma, de fundamental importância para o projeto e seu desfecho no prazo estabelecido. Em tempo o formato adotado RUP organizou o material de trabalho e suas tarefas, mostrando graficamente um sistema tomando corpo. Houve um aumento de produtividade da equipe que compartilha os mesmos arquivos e visualiza o processo em tempo real. Dentre outras metodologias do mercado, esta que adotamos, é uma das mais tradicionais onde sua etapa de documentação é das mais demoradas, mas permite que as dúvidas da equipe sejam sanadas antes do início do desenvolvimento, tornando a fase seguinte mais clara para todo o time. Uma documentação bem elaborada evita que a programação pare por falta de definição de algo não planejado.

Além disso, a grande contribuição são as competências aprendidas durante o processo de desenvolvimento do projeto e resolução de problemas. Dentre as capacidades mais importantes nesta segunda etapa, que estamos iniciando, é sem dúvida alguma a programação em JAVA e todas as suas aplicações e funcionalidades.

Os resultados alcançados podem ser vistos dentre a documentação elaborada até o código em estágio de estruturação inicial. Desta forma, pode-se concluir que seguindo a prototipagem do projeto ele foi desenvolvido conforme Documento Visão e entregue dentro do prazo, aplicando-se todos os softwares e diretrizes previamente estabelecidas.

O projeto permitiu a equipe colocar em pratica todo conhecimento adquirido durante este e os semestres passados. Também permitiu que os membros se desenvolvessem individualmente, assumindo funções e aprendendo novas tecnologias. Com a implementação do projeto RecrutaIf, foi capaz de atingir e superar as metas estabelecidas.

Referências

Paul Deitel, Harvey Deitel, Java Como Programar 10º Edição, PEARSON São Paulo, 2017.

Ian SOMMERVILLE, Engenharia de Software, PEARSON São Paulo, 2011. Página 54.

Rafael Sakurai, Desenvolvimento Distribuído com JAVA EE, GitBook São Paulo 2010.

Wikipédia, REST
<https://pt.wikipedia.org/wiki/REST>, Abril 2018.

REDEFÁCIL, A descrição técnica, o relatório o artigo
<https://redafacil.blogspot.com.br/2010/01/7-descricao-tecnica-o-relatorio-e-o.html>, Abril 2018.

Daniel Madureira, Metodologias para desenvolver <https://usemobile.com.br/metodologias-para-desenvolver-um-aplicativo/>, Abril 2018.