

Curso Superior de Sistemas para Internet

Disciplina: Engenharia de Software I – 2023/1

Profa. Márcia Häfele Islabão Franco

Integrantes:
João Fernando Hanzel Demetrio
Rodrigo Pedroso Lorandi



Propósito do documento

- Especificar os requisitos dos sistemas a serem desenvolvidos
- Fornecer informações necessárias para o projeto e implementação
- Outsourcing de TI



Apresentação do projeto

- Crescimento exponencial do setor de Tecnologia da Informação (TI)
- Demanda por soluções digitais em todos os setores da economia
- Necessidade de profissionais qualificados em TI
- Consultoria de outsourcing em TI para atender necessidades específicas das empresas



Objetivos do sistema

- Gestão de RH dos profissionais da consultoria
- Acompanhamento da carreira e alocações em projetos e clientes



Stakeholders

- Clientes que contratam a consultoria para terceirizar serviços de TI
- Profissionais de TI contratados pela consultoria
- Tech recruiters responsáveis pelo recrutamento e seleção
- Business Partner que gerencia equipes junto aos clientes
- Colaboradores internos da consultoria: setores financeiro, jurídico, administrativo e recursos humanos



Requisitos funcionais

Funcionalidades conforme tipo de usuário:

- Business Partner: Login, Cadastrar/Ler Profissional, Cadastrar/Ler Cliente,
 Alocar/desalocar profissional nos clientes
- Administrativo: Login, Ler informações do Profissional e Cliente
- Diretoria: Login, Todos os privilégios de Business Partner e Administrativo,
 Acesso aos dados financeiros de profissionais e clientes

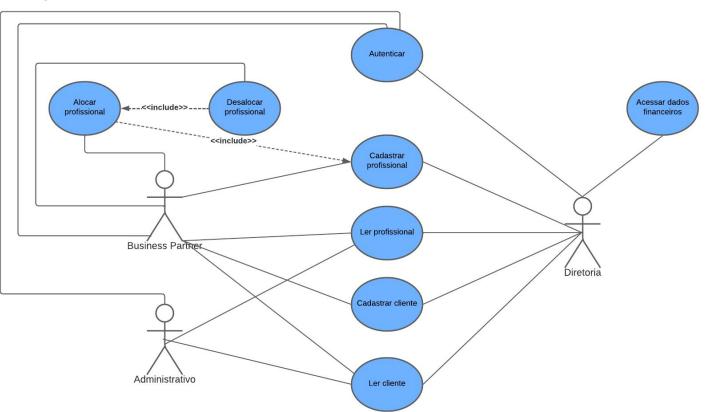


Requisitos não funcionais

- Back end feito em Java e Spring Boot
- BFF em Node
- Front end em React
- Escalabilidade: Hospedagem em nuvem pública (AWS, Microsoft Azure)
- Banco de dados PostgreSQL
- Confiabilidade: Monitoramento proativo, sistemas de backup automatizados
- Segurança: Gerenciamento de identidade (Auth0), serviço de mensageria (AWS
 SQS)



Diagrama de casos de uso



Descrição de casos de uso

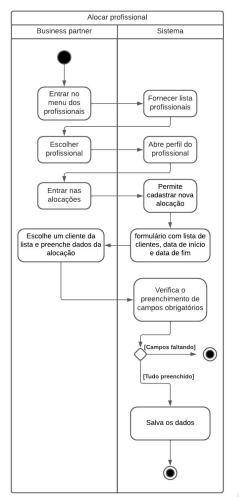
Nome do caso de uso	Alocar profissional
Descrição	Descreve como o Business Partner aloca um profissional em um cliente, considerando a disponibilidade do profissional e as necessidades do cliente.
Atores envolvidos	Business Partner e Diretoria.
Fluxo Principal	
Ações do usuário	Ações do Sistema
Na home do sistema, usuário clica no menu Profissionais.	
	2.É disponibilizada a lista de profissionais.
3.Escolhe o profissional.	
	4. Sistema abre perfil do profissional.
5. Clica na aba Alocações.	



	1.Sistema emite mensagem avisando que campo obrigatório não foi preenchido e destaca os campos não preenchidos.
Ações do usuário	Ações do sistema
Fluxo de exceção I - Campos obrigatórios não preenchidos	
	11. Apresenta novamente aba Alocações já com a nova alocação no histórico.
	10. Salva os dados.
9. Escolhe um cliente da lista e preenche dados de início e fim da alocação.	
	8. Sistema abre formulário com lista de clientes, data de início e data de fim.
7. Clica em Alocar Profissional.	
	6. Demonstra Histórico de Alocações, separadas por ativas e inativas, e apresenta botão Alocar Profissional.
	6. Demonstra Histórico de Alocações, separa-



Diagrama de atividades



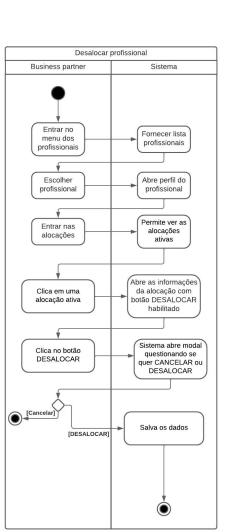




Diagrama de classes

Endereco

+ Endereco(id: Number, cidade: String,

+ atualizarEndereco(id : Number, cidade :

String, rua: String, numero: String, cep:

rua : String, numero : String, cep : String); 1

id endereco: Number

+ lerEndereco(id : Number):

+ removerEndereco(id : Number);

- cidade : String

numero : String

rua : String

cep : String

String);

Profissional id_profissional: Number nome : String corporate email: String professional status : String telefone : String endereco: Endereco salario: Float + Profissional(id: Number, nome: String, corporate email: String, professional status: String, telefone: String, endereco: Endereco, salario: Float); + lerProfissional(id : Number); + removerProfissional(id : Number): + atualizarProfissional(id : Number, nome : String, corporate_email: String, professional_status: String, telefone : String, endereco : Endereco, salario : Float); Relatorio id relatorio: Number - profissional : list < Profissional> - cliente : list <Cliente> - periodo : Date + Relatorio (id : Number, profissional : list <Profissional>, periodo : Date): + Relatorio (id : Number, cliente : list<Cliente>, periodo : Date); + Relatorio (id : Number, profissional : list <Profissional>, cliente : list <Cliente>, periodo : Date); + lerRelatorio(id : Number): + removerRelatorio(id : Number); Cliente - id_cliente : Number - nome : String endereco: Endereco telefone : String + Cliente(id : Number, nome : String, endereco : Endereco, telefone: String) +lerCliente(id : Number); + removerCliente(id : Number); + atualizarCliente(id : Number, nome : String, endereco: Endereco, telefone: String);



Alocacao

id_alocacao : Numberdata inicio : Date

- data_fim : Date

status_alocacao : Stringid_profissional : Profissional

- id_cliente : Cliente

+ Alocacao(id : Number, data_inicio : Date, data_fim : Date, status : String, id_profissional : Profissional,

id_cliente : Cliente);

+ lerAlocacao(id : Number); + removerAlocacao(id : Number);

+ atualizarAlocacao(id : Number, data_inicio : Date, data_fim : Date, status : String, id_profissional :

Profissional, id_cliente : Cliente);



Considerações finais

- Importância da modelagem e especificação do sistema
- Utilização de ferramentas como UML e requisitos funcionais e não funcionais
- Garantia da correta implementação das funcionalidades