

RECURSOS HUMANOS

Especificação de Requisitos de Software

Sistemas de Informação
3º B - Osório

Victor Augusto Rodrigues
Heitor Leonardo
Fernando Henrique

Envolvidos

Nome	Papel
Victor Augusto Rodrigues	Analista
Heitor Leonardo	Líder/Dev
Fernando Henrique	Dev

Histórico de Revisões

Data	Versão	Descrição	Responsável
03/07/2018	1.0	Criação do documento para Especificação de Requisitos de Software e Diagramas de sequência	Victor Augusto
03/07/2018	1.1	Aprimoramentos e inclusão de detalhes sobre as integrações	Heitor Leonardo, Fernando Henrique

OBJETIVOS DO DOCUMENTO

Este documento tem como objetivo especificar para o leitor e para o desenvolvedor as necessidades e funcionalidades do módulo como um todo, considerando também integrações com outros módulos.

A partir das especificações técnicas demonstradas nos capítulos seguintes, o time de analistas buscou transformar e traduzir as necessidades e requisitos (funcionais e não funcionais) adquiridos e validados com o cliente, para a linguagem técnica na qual os desenvolvedores poderão obter o máximo de êxito e objetividade nas criações e construções de arquitetura do sistema e codificação.

VISÃO GERAL

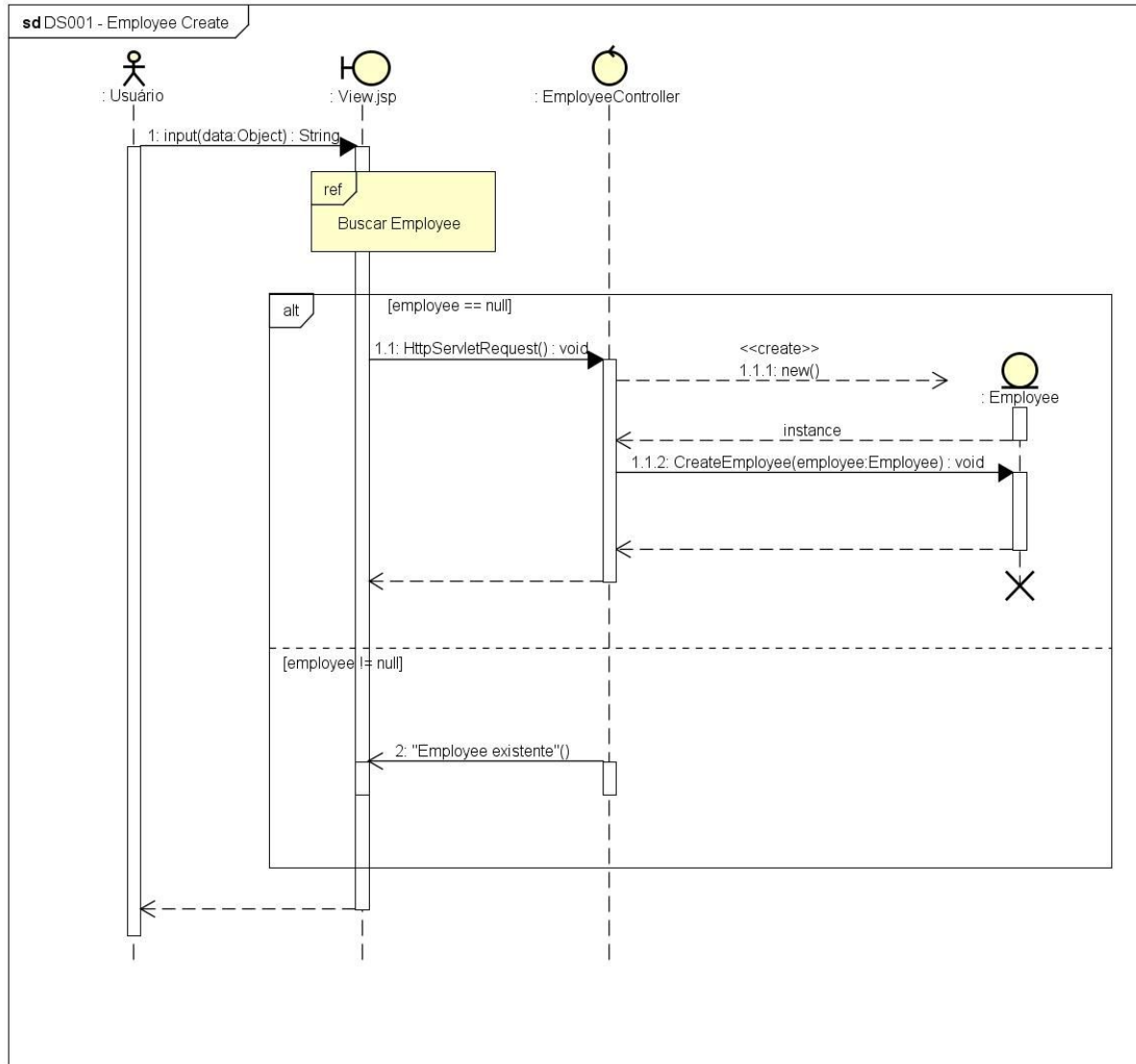
Breve descrição dos Use Cases:

- UC001 - Employee Create:
Realiza a criação de um novo objeto do tipo Employee através do método com a seguinte assinatura:
CreateEmployee(Employee employee)
- UC002 - Employee Read:
Realiza a busca de um objeto do tipo Employee através do método com a seguinte assinatura e retorna um objeto do tipo Employee ao final de sua execução:
ReadEmployee(int businessid)
- UC003 - Employee Update:
Realiza uma atualização de registro de um objeto do tipo Employee existente, através do método com a seguinte assinatura:
UpdateEmployee(Employee employee)
- UC004 - Employee Delete:
Realiza a exclusão de um registro de um objeto do tipo Employee existente através do método com a seguinte assinatura:
DeleteEmployee(int businessid)
- UC005 - JobCandidate Create:
Realiza a criação de um novo objeto do tipo JobCandidate através do método com a seguinte assinatura:
CreateJobCandidate(JobCandidate jobcandidate)
- UC006 - JobCandidate Read:
Realiza a busca de um objeto do tipo JobCandidate através do método com a seguinte assinatura e retorna um objeto do tipo JobCandidate ao final da execução::
ReadJobCandidate(int jobcandidateid)
- UC007 - JobCandidate Update:
Realiza uma atualização de registro de um objeto do tipo JobCandidate existente, através do método com a seguinte assinatura:
UpdateJobCandidate (JobCandidate jobcandidate)
- UC008 - JobCandidate Delete:
Realiza a exclusão de um registro de um objeto do tipo JobCandidate existente através do método com a seguinte assinatura:
DeleteJobCandidate(int jobcandidateid)

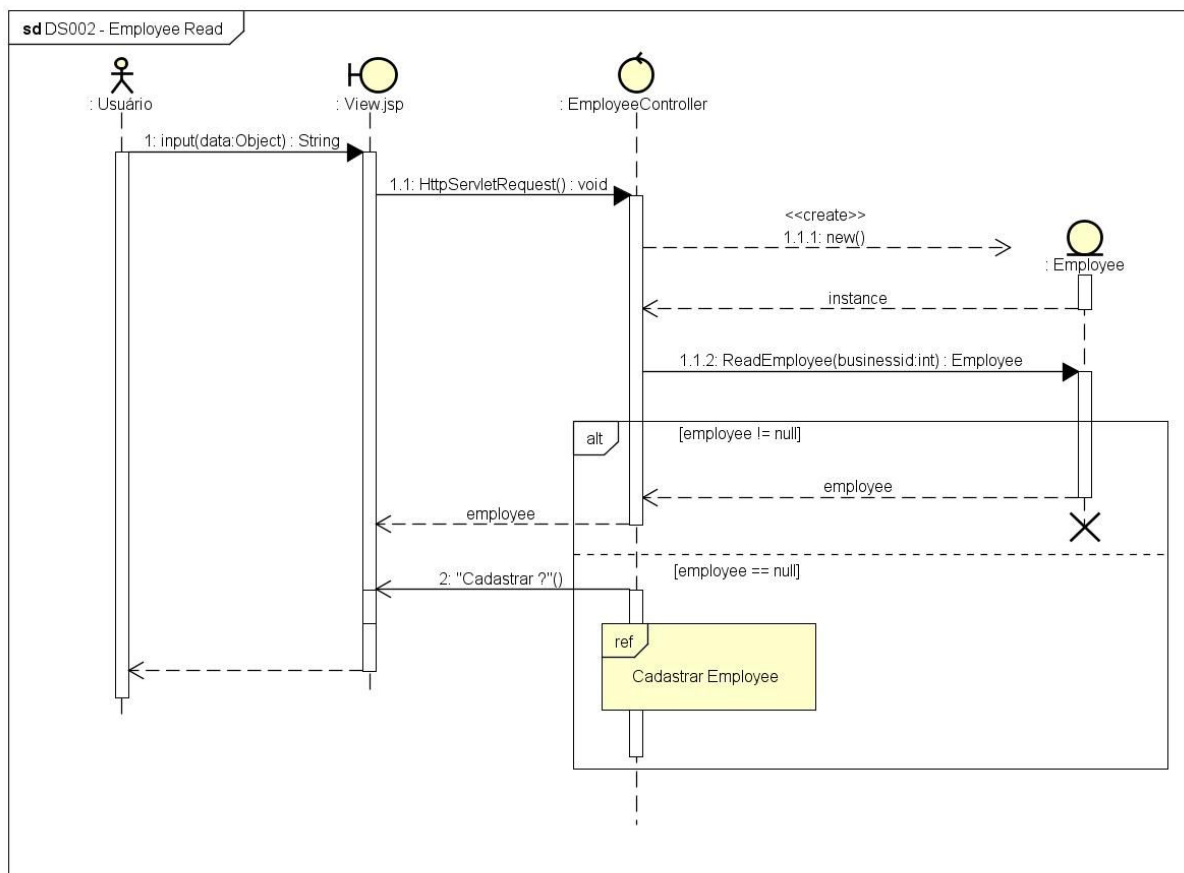
ESPECIFICAÇÕES TÉCNICAS

Diagramas de Sequência:

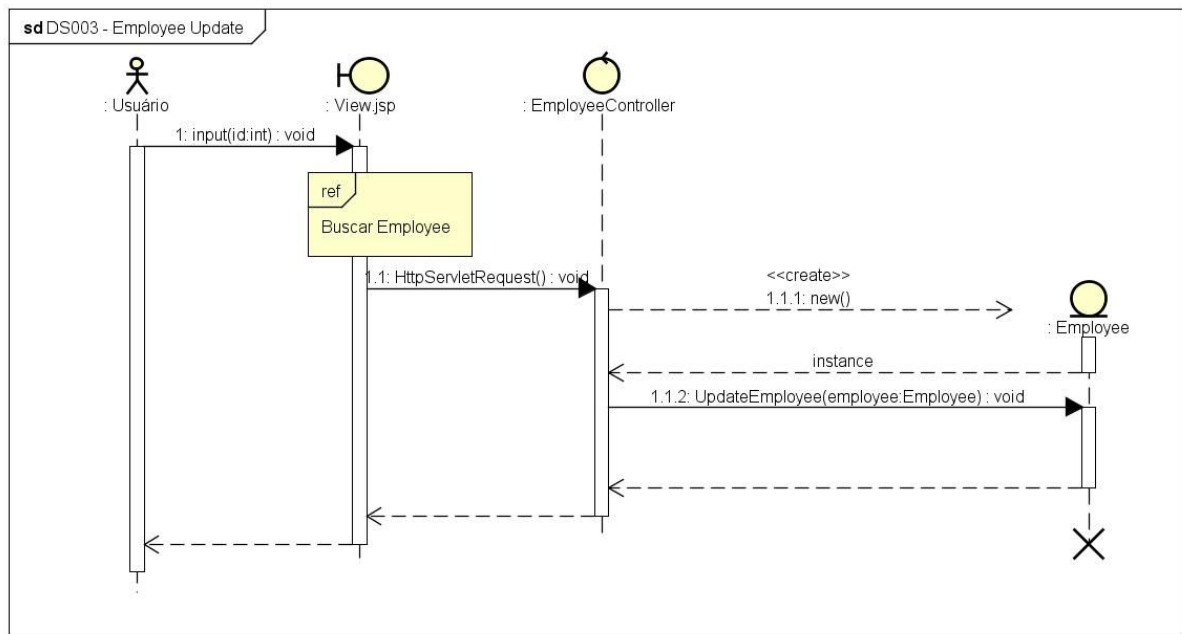
- DS001 - Employee Create



- DS002 - Employee Read

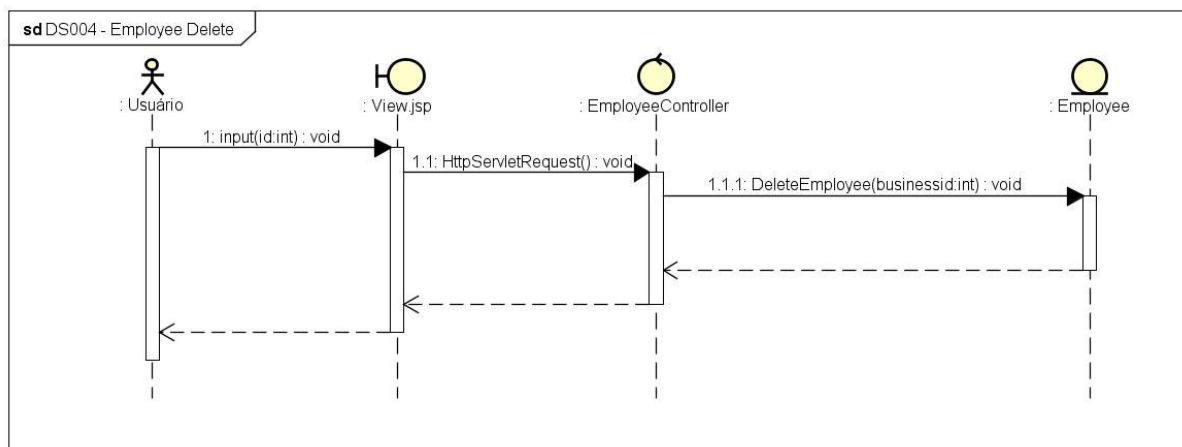


- DS003 - Employee Update



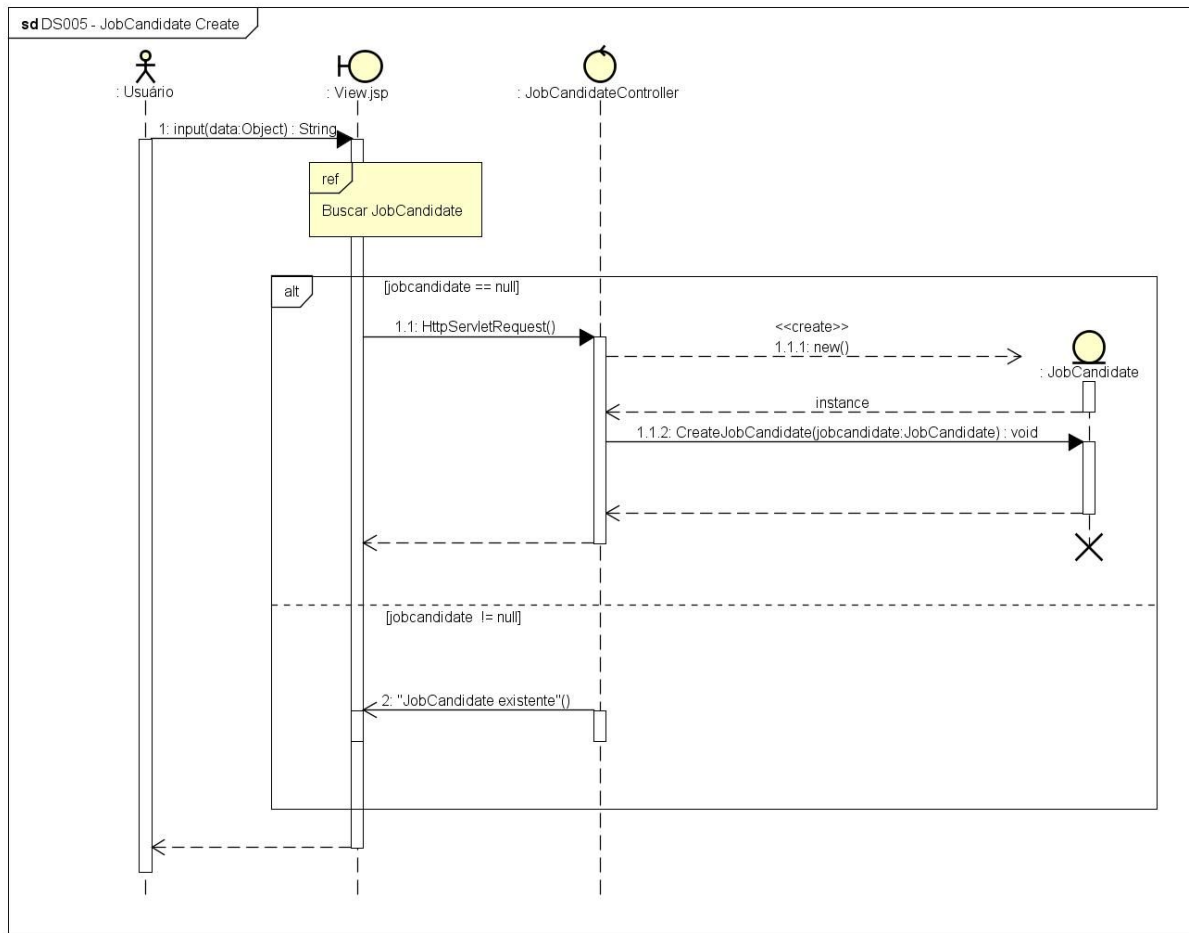
powered by Astah

- DS004 - Employee Delete

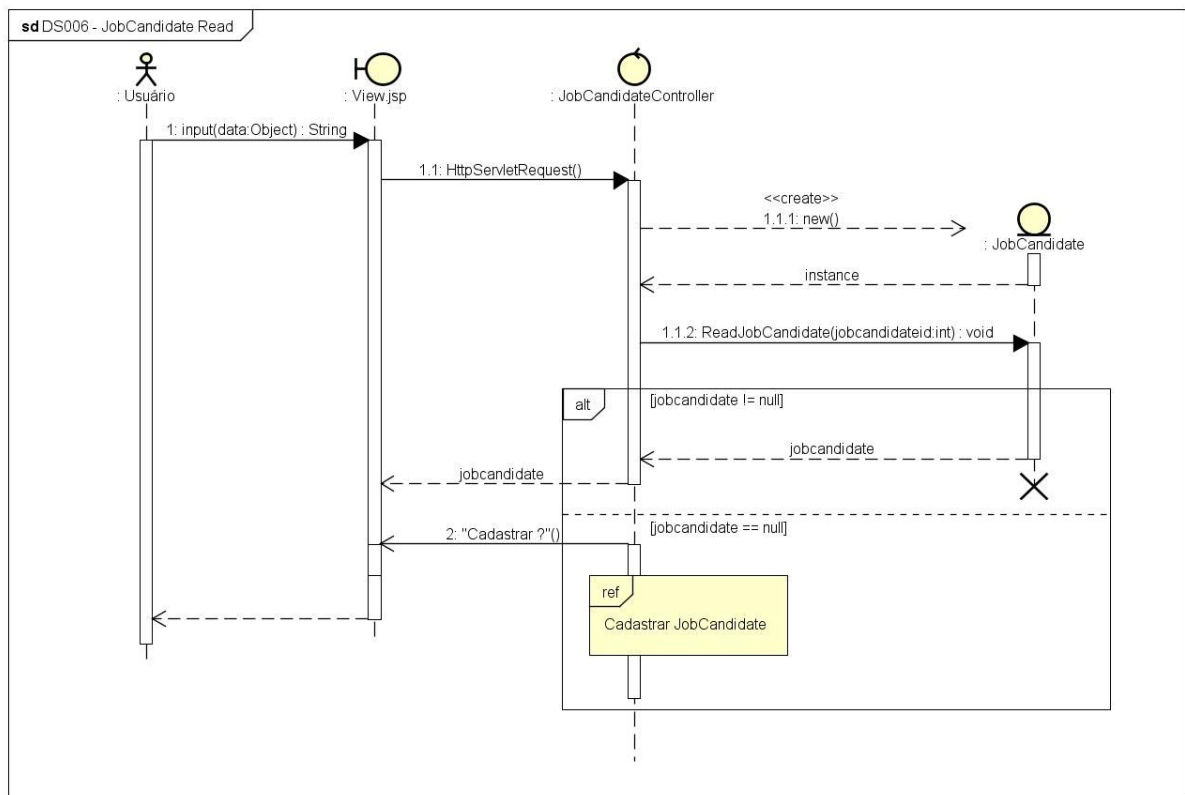


powered by Astah

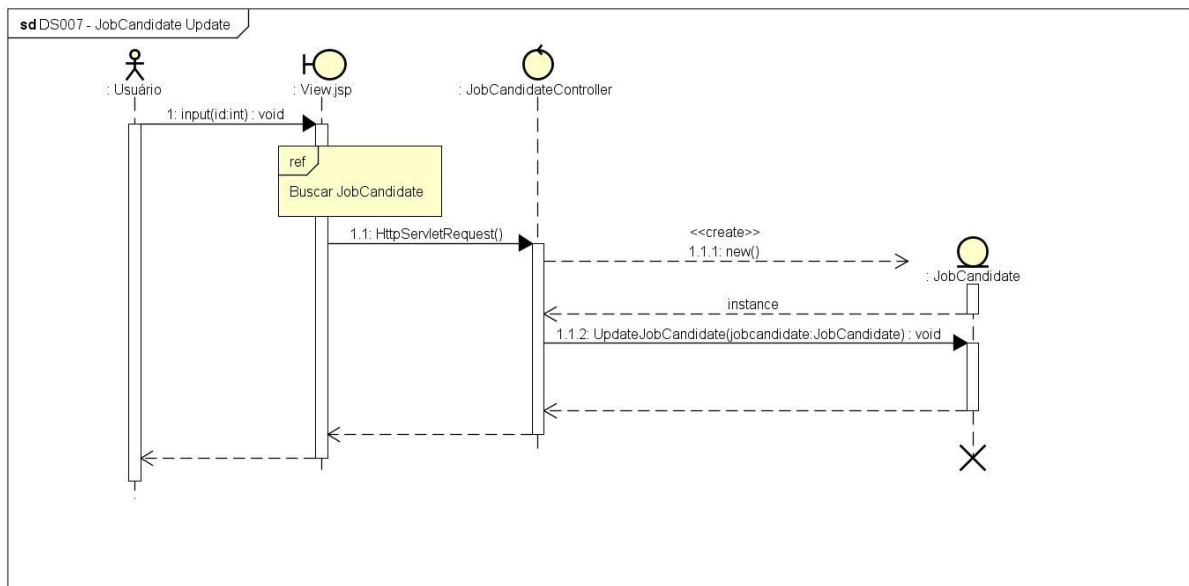
- DS005 - JobCandidate Create



- DS006 - JobCandidate Read

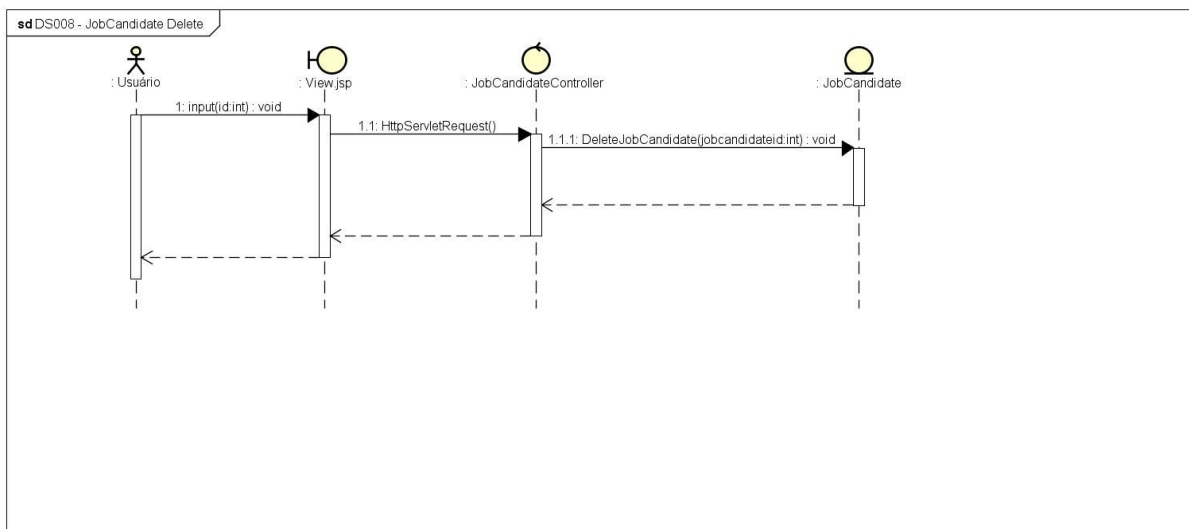


- DS007 - JobCandidate Update



powered by Astah

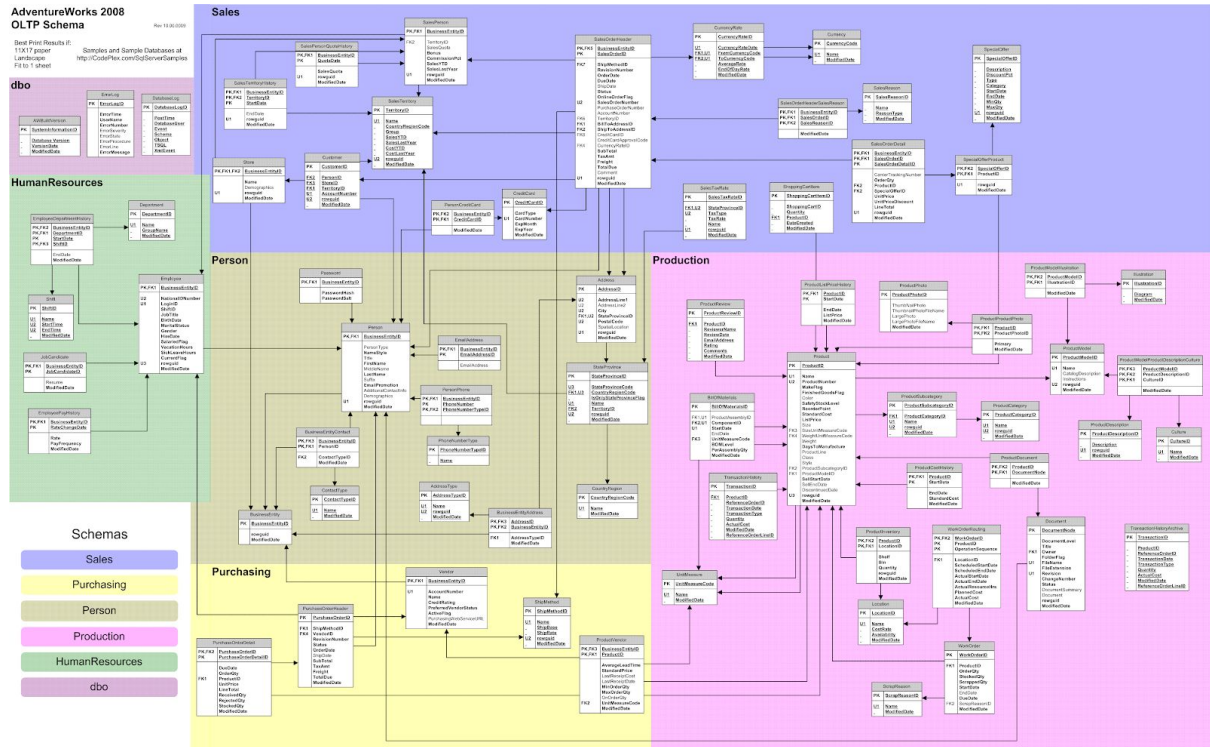
- DS008 - JobCandidate Delete



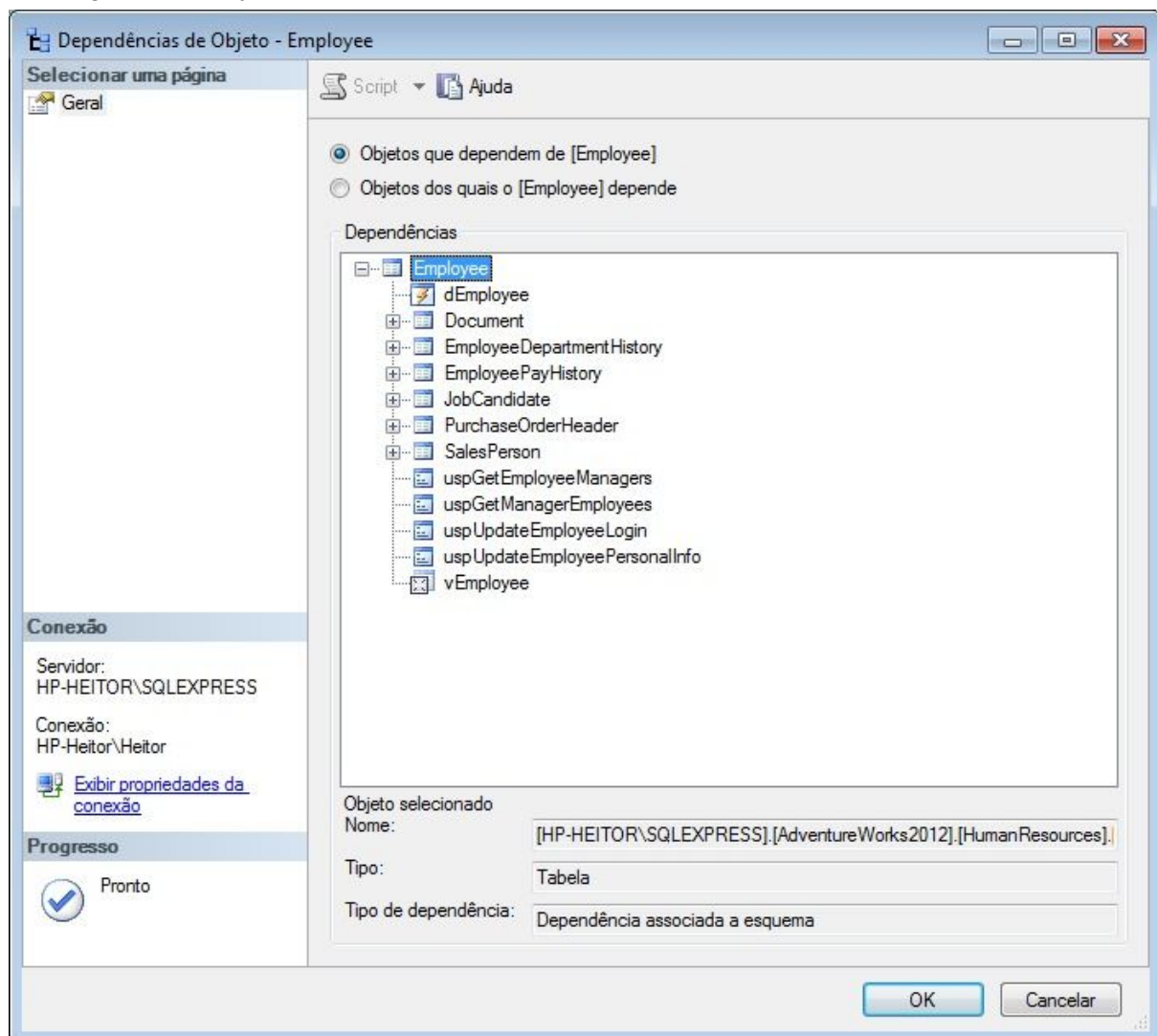
powered by Astah

INTEGRAÇÕES

Para maior entendimento sobre as integrações necessárias utilizamos o esquema de Banco de Dados do AdventureWorks, diagramas de relacionamentos e códigos de criação das tabelas. A partir do esquema podemos ver que o módulo de Recursos Humanos possui relacionamentos através da tabela Empregado(Employee) com os módulos de Vendas(Sales), Pessoa(Person), e Compras(Purchasing).

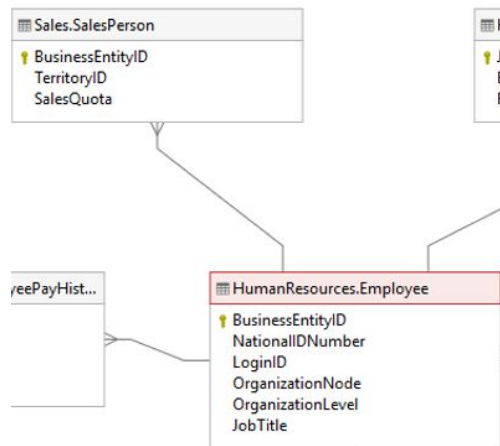


Pelo banco de dados podemos ver as tabelas que dependem diretamente da tabela Empregado(Employee).



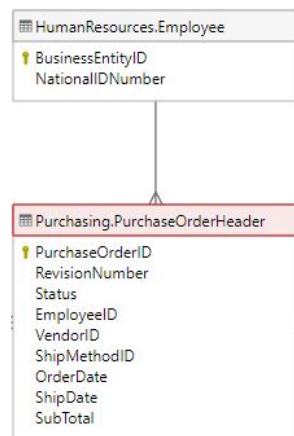
No módulo de Vendas a tabela SalesPerson(Vendedor) possui uma chave estrangeira BusinessEntityID referenciando o ID da tabela Employee como podemos ver no código de criação da tabela. E por isso o módulo de Vendas depende do recebimento de dados do módulo de Recursos Humanos.

```
ALTER TABLE [Sales].[SalesPerson]
WITH CHECK ADD CONSTRAINT [FK_SalesPerson_Employee_BusinessEntityID]
FOREIGN KEY([BusinessEntityID]) REFERENCES [HumanResources].[Employee] ([BusinessEntityID])
GO
```

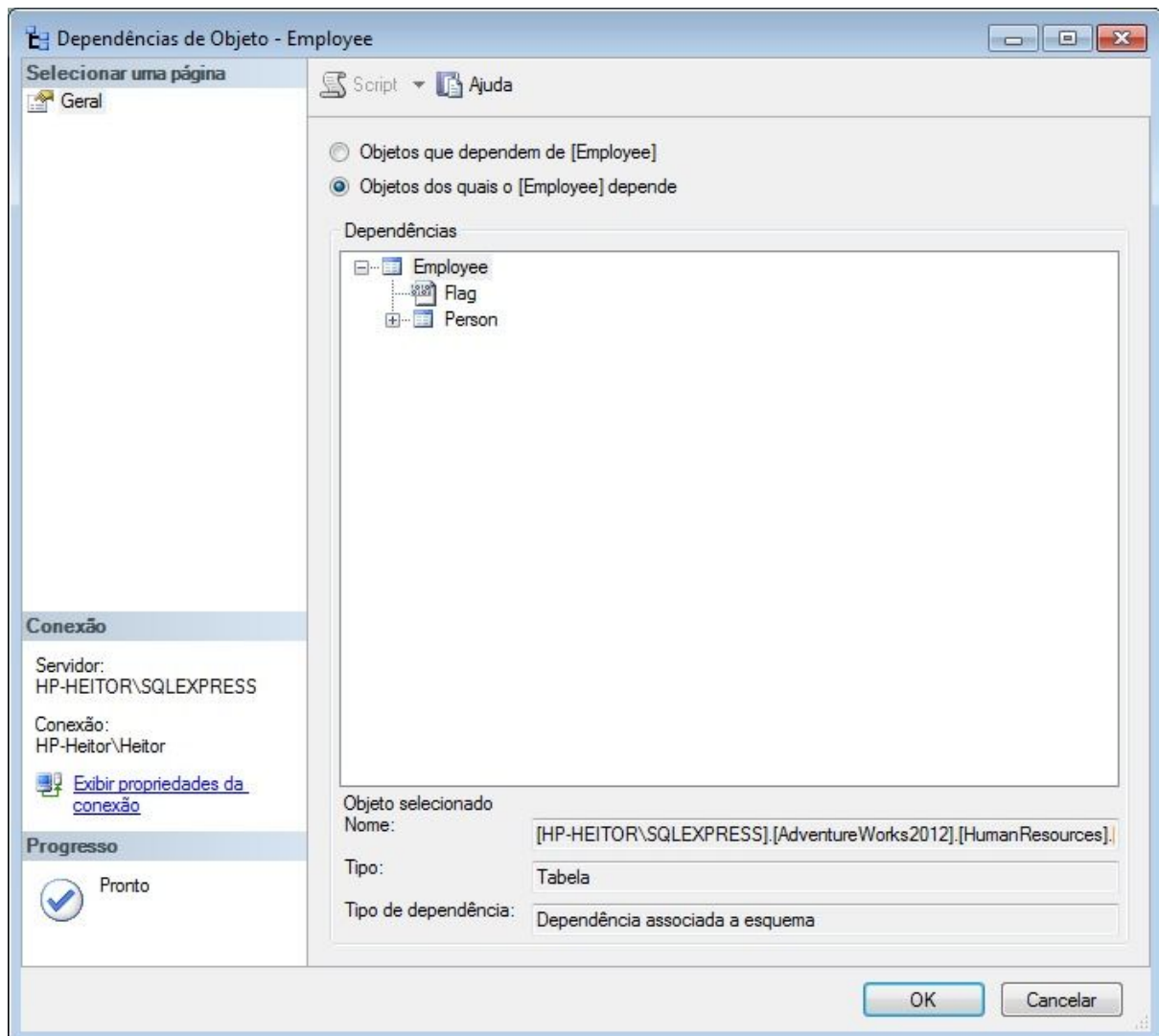


No módulo de Compras a tabela PurchaseOrderHeader (Cabeçalho da ordem de compra) possui uma chave estrangeira EmployeeID referenciando o ID da tabela Employee como podemos ver no código de criação da tabela. E por isso o módulo de Compras depende do recebimento de dados do módulo de Recursos Humanos.

```
ALTER TABLE [Purchasing].[PurchaseOrderHeader]
WITH CHECK ADD CONSTRAINT [FK_PurchaseOrderHeader_Employee_EmployeeID]
FOREIGN KEY([EmployeeID]) REFERENCES [HumanResources].[Employee] ([BusinessEntityID])
GO
```



Pelo banco de dados podemos ver também a tabela da qual Empregado(Employee) depende diretamente.



No módulo de Recursos Humanos a tabela Employee(Empregado) possui uma chave estrangeira BusinessEntityID referenciando o ID da tabela Person como podemos ver no código de criação da tabela. E por isso o módulo de Recursos Humanos depende do recebimento de dados do módulo de Pessoas.

```
ALTER TABLE [HumanResources].[Employee]
WITH CHECK ADD CONSTRAINT [FK_Employee_Person_BusinessEntityID]
FOREIGN KEY([BusinessEntityID]) REFERENCES [Person].[Person] ([BusinessEntityID])
GO
```

