



INSTITUTO FEDERAL

Bahia

Campus Vitória da Conquista

IMPLEMENTAÇÃO DE APLICAÇÃO SPA COM A *STACK* MERN

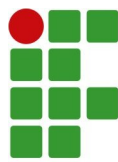
Gabriel Almeida Dias

Josafá Santos dos Reis

Maurício de Abreu Cordeiro

Dezembro, 2020

Vitória da Conquista, Bahia



INSTITUTO FEDERAL

Bahia

Campus Vitória da Conquista

IMPLEMENTAÇÃO DE APLICAÇÃO SPA COM A *STACK* MERN

Relatório apresentado como requisito parcial para obtenção de aprovação nas disciplinas de Frameworks de Persistência de Dados e Programação Web II, do Curso de Pós-Graduação *Lato Sensu* em Desenvolvimento Web, do Instituto Federal de Educação, Ciência e Tecnologia da Bahia, Campus Vitória da Conquista.

Prof. Alexandro dos Santos Silva

Gabriel Almeida Dias

Josafá Santos dos Reis

Maurício de Abreu Cordeiro

Dezembro, 2020

Vitória da Conquista, Bahia

SUMÁRIO

1. INTRODUÇÃO.....	3
2. MODELAGEM DE DADOS.....	3
3. ARQUITETURA DA APLICAÇÃO.....	3
4. CONCLUSÕES.....	3

1 INTRODUÇÃO

O presente trabalho tem por objetivo relatar o desenvolvimento de uma aplicação *web*, baseando-se na arquitetura MVC (do inglês *Model View Controller*) e utilizando a *stack* MERN (**M**ongo **E**xpress.js **R**eact **N**ode.js).

A referida aplicação tem como foco o gerenciamento de contratos de compras e de prestação de serviços, incluindo o controle de contas e pagamentos desses contratos e a manutenção dos cadastros das empresas contratadas.

2 MODELAGEM DE DADOS

Apesar do banco de dados da aplicação ser não-relacional, para fins de entendimento das relações entre as partes do seu modelo de dados, segue uma representação Entidade-Relacionamento:

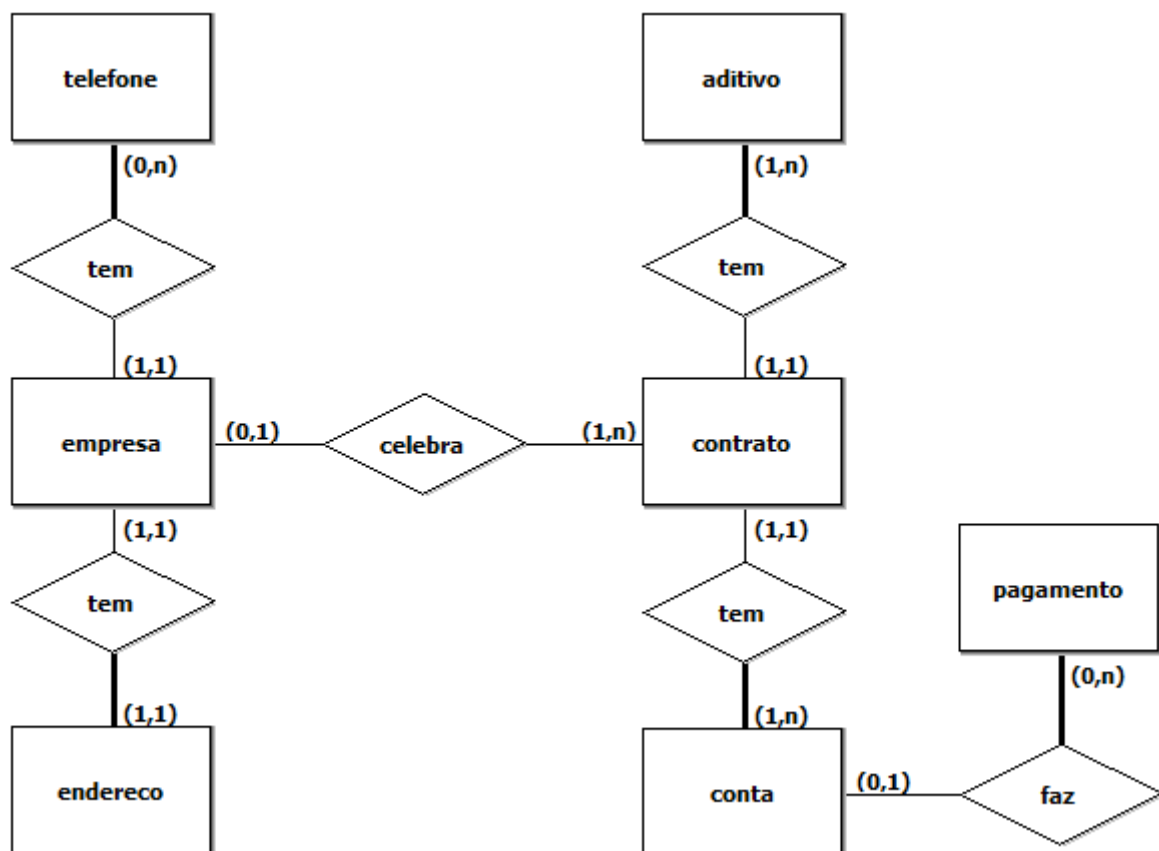


Figura 1: Diagrama E-R da aplicação

A partir de análise sobre o diagrama acima, é possível perceber que existem apenas duas entidades fortes em todo o modelo de dados. Esse fato foi determinante para a definição do modelo de dados para o MongoDB, orientado a documentos.

Desse modo, cada entidade forte se tornou uma coleção (*empresa* e *contrato*), e suas entidades fracas se tornaram sub-documentos dentro de cada documento da coleção. Esse modelo é descrito nas listagens abaixo (em notação JSON):

```
{
  "empresa": {
    "_id": "ObjectId",
    "cnpj": "String",
    "razaoSocial": "String",
    "nomeFantasia": "String",
    "email": "String",
    "endereco": {
      "rua": "String",
      "numero": "String",
      "bairro": "String",
      "cidade": "String",
      "uf": "String"
    },
    "telefones": [
      {
        "ddd": "String",
        "telefone": "String"
      }
    ]
  }
}

{
  "contrato": {
    "_id": "ObjectId",
    "_idEmpresa": "ObjectId",
    "numero": "String",
    "copiaContrato": "String",
    "tipoContrato": "Number",
    "prestacao": "Number",
    "statusContrato": "Number",
    "valorContrato": "Number",
    "dataCelebracaoContrato": "Date",
    "dataFinalizacaoContrato": "Date",
    "observacoesContrato": "String",
    "aditivos": [
      {
        "copiaContratoAditivo": "String",
        "valorContratoAditivo": "Number",
        "dataRenovacao": "Date",
        "dataVencimento": "Date",
        "observacoes": "String"
      }
    ],
    "contas": [
      {
        "copiaConta": "String",
        "tipoConta": "Number",
        "statusConta": "Number",
        "dataVencimentoConta": "Date",
        "dataRecebimentoSetor": "Date",
        "dataEnvioDof": "Date",
        "valorConta": "Number",
        "observacoesConta": "String",
```

```
"pagamentos": [  
  {  
    "dataPagamentoConta": "Date",  
    "valorPago": "Number",  
    "taxaJuros": "Number"  
  }  
]  
}  
]  
}
```

3 ARQUITETURA DA APLICAÇÃO

A aplicação está dividida, arquiteturalmente, conforme apresentado na Figura 2:

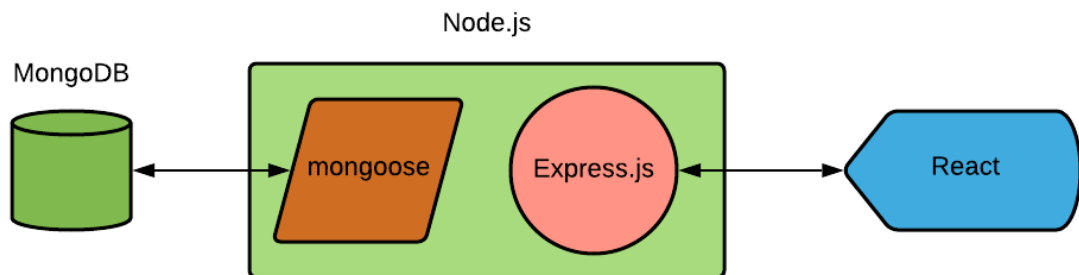


Figura 2: Arquitetura da aplicação (stack MERN)

Nessa arquitetura, o **MongoDB** atua como sistema gerenciador de banco de dados; o **Node.js** atua como serviço de *backend* da aplicação, sendo o **mongoose** um *framework* para acesso ao MongoDB e o **Express.js** um *framework* para tratamento de requisições HTTP; e o **React** atua como aplicação *frontend*.

4 CONCLUSÕES

O processo de desenvolvimento da aplicação descrita neste trabalho foi capaz de demonstrar o alto nível de produtividade que é possível alcançar utilizando a *stack* MERN, onde todas as etapas do processo puderam ser realizadas com a utilização de uma única linguagem (o JavaScript), e com *frameworks* capazes de otimizar tanto a implementação quanto a implantação da aplicação.

Como *frameworks* JavaScript (que incluem os da *stack* MERN) são amplamente utilizados e possuem uma comunidade ativa, não houve nenhum problema durante a implementação da aplicação que não tenha sido rapidamente solucionado com poucas pesquisas na internet.

Futuramente, pode-se aproveitar da infraestrutura que serviços de PaaS (*Plataform as a Service*) disponibilizam para, facilmente, implantar a aplicação aqui desenvolvida.