

Curso Mongo, Express, Angular e Node - Primeira Aplicação do ZERO!

Leonardo Leitão

Versão 1.0, 10/05/2017

Índice

1. Introdução	2
1.1. Bem-vindo	2
1.2. Visão Geral do Curso	2
1.3. Conhecendo o Projeto	2
1.4. Assine o Nosso Canal	2
1.5. Repositório do Curso	2
1.6. Instalação MongoDB (Windows)	2
1.7. Instalação Node (Windows)	2
2. Mongo	3
2.1. Mongo - Visão Geral	3
2.2. Mongo - Exercício 01: Comandos Básicos	3
2.3. Mongo - Exercício 02: Inserções	5
2.4. Mongo - Exercício 03: Consultas	7
2.5. Mongo - Exercício 04: Agregação	8
2.6. Mongo - Exercício 05: Atualização	9
2.7. Mongo - Exercício 06: Contador E Remoções	9
3. Node	11
3.1. Node - Exercício 01: Javascript Básico	11
3.2. Node - Exercício 02: Sistema de Módulos	12
3.3. Node - Exercício 03: Singleton	12
3.4. Node - Exercício 04: Objeto Global	13
3.5. Node - Exercício 05: This	14
3.6. Node - Exercício 06: Módulo Externo (Lodash)	14
3.7. Node - Exercício 07: Passagem de Parâmetros	16
3.8. Node - Exercício 08: Process (ARGV)	16
3.9. Node - Exercício 09: Process (STDIN/STDOUT)	17
3.10. Node - Exercício 10: Módulo FS	17
3.11. Node - Exercício 11: Módulo HTTP	17
4. Express	19
4.1. Express - Visão Geral	19
4.2. Express - Exercício 01: Configuração e Mapeando uma Rota	19
4.3. Express - Exercício 02: Cadeia de Middlewares	20
4.4. Express - Exercício 03: Método USE	21
4.5. Express - Exercício 04: Método Route	22
4.6. Express - Exercício 05: Express Router	22
4.7. Express - Exercício 06: Express e Router são Singletons?	23
5. Angular	24
5.1. Angular - Instalando via NPM	24

5.2. Angular - Exercício 01: Configurando uma Página com Angular	25
5.3. Angular - Exercício 02: Binding	25
5.4. Angular - Exercício 03: Controller	26
5.5. Angular - Exercício 04: Controller As	26
5.6. Angular - Exercício 05: Filter	27
5.7. Angular - Exercício 06: Factory	28
5.8. Angular - Exercício 07: Service	29
5.9. Angular - Exercício 08: Directive	30
5.10. Angular - Exercício 09: Component	31
6. Backend - Configurações Iniciais	33
6.1. Versão Inicial do Projeto Backend	33
6.2. Adicionando o .gitignore ao Projeto	34
6.3. Implementando o Servidor com Express	35
6.4. Criando a Conexão com MongoDB	35
7. Backend - Ciclo de Pagamento API	37
7.1. Mapeamento ODM do Objeto Ciclo de Pagamentos	37
7.2. Serviço de Ciclo de Pagamentos	37
7.3. Criando o Arquivo de Rotas	38
7.4. Registrando as Rotas do Serviço de Ciclo de Pagamentos	39
7.5. Testando a API de Ciclo de Pagamentos (Parte 1)	39
7.6. Testando a API de Ciclo de Pagamentos (Parte 2)	40
7.7. Testando a API de Ciclo de Pagamentos (Parte 3)	41
7.8. Serviço Contador (count) de Ciclo de Pagamentos	41
8. Backend - Sumário de Pagamento API	43
8.1. Serviço de Sumário de Pagamentos	43
8.2. Registrando a Rota do Serviço de Sumário de Pagamentos	43
9. Backend - Ajustes Finais	45
9.1. Uniformizando as Mensagens de Erro	45
10. Frontend - Configuração do Build do Projeto (Gulp)	46
10.1. Versão Inicial do Projeto Frontend	46
10.2. Adicionando o .gitignore ao Projeto	47
10.3. Estrutura Inicial do Build	48
10.4. Build dos Arquivos das Dependências	50
10.5. Build dos Arquivos do Projeto	52
10.6. Configurando o Servidor com Reload Automático	54
11. Frontend - Estrutura Básica do Projeto (Template)	55
11.1. Implementando a Página Index	55
11.2. Adicionando o Angular ao Projeto	56
11.3. Implementação o Header	57
11.4. Implementação o Footer	58
11.5. Implementação o Menu	58

11.6. Configurando as Rotas com UI Router	59
12. Frontend - Dashboard	62
12.1. Versão Inicial do Dashboard	62
12.2. Componente de Cabeçalho do Conteúdo	63
12.3. Factory para Ajudar no Layout (Bootstrap GridSystem)	64
12.4. Componente ValueBox	65
12.5. Inicializando o Backend com PM2	67
12.6. Dashboard Controller	68
12.7. Atualizando o DashboardController para Angular 1.6	69
12.8. Habilitando o CORS na Aplicação Backend	69
12.9. Aplicando dois Padrões do Angular Style Guide	71
13. Frontend - Cadastro de Ciclo de Pagamentos	73
13.1. Estrutura de Abas do Cadastro	73
13.2. Estrutura Inicial do Formulário	74
13.3. Componente Campo de Formulário (Field)	74
13.4. Versão Inicial do Controller de Ciclo de Pagamentos	76
13.5. Atualizando o BillingCycleController para Angular 1.6	80
13.6. Factory de Exibição de Mensagens	81
13.7. Lista de Ciclo de Pagamentos (Parte 1)	83
13.8. Lista de Ciclo de Pagamentos (Parte 2)	84
13.9. Factory para Controlar a Exibição das Abas	85
13.10. Exibição das Abas de Alterar e Excluir	89
13.11. Exclusão do Ciclo de Pagamentos	93
13.12. Alteração do Ciclo de Pagamentos	96
13.13. Lista de Créditos	98
13.14. Lista de Débitos	102
13.15. Créditos e Débitos: Adicionar, Clonar e Excluir	104
13.16. Sumário dos Créditos e Débitos (Parte 1)	109
13.17. Sumário dos Créditos e Débitos (Parte 2)	111
13.18. Desabilitar Ações Durante a Exclusão	118
13.19. Implementando a Paginação	120
13.20. Componente de Paginação	124
14. Refactories e Correções	130
14.1. One-Time Binding e AngularJS Batarang	130
14.2. Atualizando o Projeto para Angular 1.6	130
15. PrimeiraAPP: Autenticação	137
15.1. Backend: Novas Dependências	137
15.2. Backend: Implementar Autenticação	137
15.3. Frontend: Implementar Autenticação	144
Appendix A: Tabela de Códigos	158
Glossário	160

Sumário

Apostila do curso de Mongo, Express, Angular e Node - Primeira Aplicação do ZERO! da Cod3r.

<https://www.cod3r.com.br>

1. Introdução

1.1. Bem-vindo

1.2. Visão Geral do Curso

1.3. Conhecendo o Projeto

1.4. Assine o Nosso Canal



Olá amigos, convido vocês a conhecer o canal da COD3R no Youtube.

Visite o canal: <https://www.youtube.com/aulasdeprogramacao>

1.5. Repositório do Curso



Os arquivos do curso estão disponíveis no GitHub.

Repositório do curso: <https://github.com/CursosCod3r/mean-primeira-app>

1.6. Instalação MongoDB (Windows)

1.7. Instalação Node (Windows)

2. Mongo

2.1. Mongo - Visão Geral

2.2. Mongo - Exercício 01: Comandos Básicos

Abra o terminal e execute o seguinte comando:

Listagem 1 - Inicializar o servidor do Mongo

```
mongod
```



Para iniciar o **console do Mongo** abra um novo terminal.

Listagem 2 - Iniciar o console do Mongo

```
mongo
```

Listagem 3 - Listar todas as databases

```
show dbs
```

Listagem 4 - Usar database db_finance

```
use db_finance
```

Listagem 5 - Verificar o database atual

```
db
```

Listagem 6 - *Listar todas as databases*

```
show dbs
```

Listagem 7 - *Criar Collection billingCycles*

```
db.createCollection('billingCycles')
```

Listagem 8 - *Listar todas as databases*

```
show dbs
```

Listagem 9 - *Listar as Collections*

```
show collections
```

Listagem 10 - *Criar Collection billingcycles*

```
db.createCollection('billingcycles')
```

Listagem 11 - *Listar as Collections*

```
show collections
```

Listagem 12 - *Remover a Collection billingCycles*

```
db.billingCycles.drop()
```


Listagem 13 - *Listar as Collections*

```
show collections
```

Listagem 14 - *Remover a Collection billingcycles*

```
db.billingcycles.drop()
```

Listagem 15 - *Listar as Collections*

```
show collections
```

Listagem 16 - *Listar todas as databases*

```
show dbs
```

2.3. Mongo - Exercício 02: Inserções

Listagem 17 - *Limpar o console*

```
cls
```

Listagem 18 - *Listar todas as databases*

```
show dbs
```

Listagem 19 - *Usar database db_finance*

```
use db_finance
```

Listagem 20 - Listar as Collections

```
show collections
```

Listagem 21 - Inserção na Collection billingcycles

```
db.billingcycles.insert({name:"Janeiro/17", month: 1, year: 2017})
```

Listagem 22 - Listar as Collections

```
show collections
```

Listagem 23 - Listar todas as databases

```
show dbs
```

Listagem 24 - Salvar na Collection billingcycles

```
db.billingcycles.save({name:"Fevereiro/17", month: 2, year: 2017})
```

Listagem 25 - Inserção na Collection billingcycles

```
db.billingcycles.insert({
  name: "Março/17",
  month: 3,
  year: 2017,
  credits: [
    { name: "Salário", value: 5000 }
  ],
  debts: [
    { name: "Luz", value: 100, status: "PAGO" },
    { name: "Telefone", value: 100, status: "PENDENTE" }
  ]
})
```

2.4. Mongo - Exercício 03: Consultas

Listagem 26 - Limpar o console

```
cls
```

Listagem 27 - Usar database db_finance

```
use db_finance
```

Listagem 28 - Listar todos os registros da Collection

```
db.billingcycles.find()
```

Listagem 29 - Listar os registros da Collection com uma melhor leitura

```
db.billingcycles.find().pretty()
```

Listagem 30 - Listar apenas um registro da Collection

```
db.billingcycles.findOne()
```

Listagem 31 - Listar apenas um registro da Collection que contém mês 2

```
db.billingcycles.findOne({month: 2})
```

Listagem 32 - Listar apenas um registro da Collection que contém mês 1 ou mês 2 com uma melhor leitura

```
db.billingcycles.findOne({$or: [{month: 1}, {month: 2}]}).pretty()
```

Listagem 33 - *Listar os registros da Collection com o atributo credits com uma melhor leitura*

```
db.billingcycles.find({credits:{$exists:true}}).pretty()
```

Listagem 34 - *Listar os registros da Collection que contém ano 2017*

```
db.billingcycles.findOne({year: 2017})
```

Listagem 35 - *Listar os registros da Collection menos o primeiro registro*

```
db.billingcycles.findOne({year: 2017}).skip(1)
```

Listagem 36 - *Listar apenas um registros da Collection menos o primeiro registro*

```
db.billingcycles.findOne({year: 2017}).skip(1).limit(1)
```

2.5. Mongo - Exercício 04: Agregação

Listagem 37 - *Criar fluxo do pipeline para somatória de credits e debts*

```
db.billingcycles.aggregate([{\n  $project:{credit:{$sum: "$credits.value"},debt:{$sum: "$debts.value"}\n},{\n  $group: {_id:null,credit:{$sum: "$credit"},debt:{$sum: "$debt"}\n}])
```

Listagem 38 - *Listar apenas um registro da Collection que contém mês 3*

```
db.billingcycles.findOne({month: 3})
```

2.6. Mongo - Exercício 05: Atualização

Listagem 39 - Limpar o console

```
cls
```

Listagem 40 - Atualizar o registro da collection

```
db.billingcycles.update(  
  {$and:[{month: 1},{year:2017}]},  
  {$set:{credits:[{name:"Salário",value:5000}]}}  
)
```

Listagem 41 - Listar apenas um registro da Collection

```
db.billingcycles.findOne()
```

Listagem 42 - Listar os registros da Collection com atributo credits e retornar o atributo name com uma melhor leitura

```
db.billingcycles.find({credits:{$exists:true}}, {_id:0, name: 1}).pretty()
```

2.7. Mongo - Exercício 06: Contador E Remoções

Listagem 43 - Limpar o console

```
cls
```

Listagem 44 - Contar quantos registros tem a Collection

```
db.billingcycles.count()
```

Listagem 45 - Remover os registros da Collection que contém mês 2

```
db.billingcycles.remove({month: 2})
```

Listagem 46 - Contar quantos registros tem a Collection

```
db.billingcycles.count()
```

Listagem 47 - Remover apenas um registros da Collection que contém ano 2017

```
db.billingcycles.remove({year: 2017}, 1)
```

Listagem 48 - Contar quantos registros tem a Collection

```
db.billingcycles.count()
```

Listagem 49 - Remover a database db_finance

```
db.dropDatabase()
```

Listagem 50 - Listar todas as databases

```
show dbs  
-----
```

3. Node

3.1. Node - Exercício 01: Javascript Básico

Abra o terminal e dentro da pasta **Desktop** execute o seguinte comando:

Listagem 51 - Criar pasta FundamentosMEAN

Desktop/FundamentosMEAN

```
mkdir FundamentosMEAN && cd FundamentosMEAN
```

Listagem 52 - Criar pasta node

FundamentosMEAN/node

```
mkdir node && cd node
```

Listagem 53 - Abrir o Atom

```
atom .
```

Listagem 54 - Criar arquivo ex01.js

FundamentosMEAN/node/ex01.js

```
const ola = () => console.log('Ola node!')  
setInterval(ola, 1000)
```



Abra o terminal e dentro da pasta **FundamentosMEAN/node** execute os seguintes comandos:

Listagem 55 - Executar arquivo ex01.js

```
node ex01
```

Listagem 56 - Parar de executar o arquivo ex01.js

```
ctrl + c
```



Existe um plugin do Atom chamado `atom-runner` para executar seus arquivos.

3.2. Node - Exercício 02: Sistema de Módulos

Listagem 57 - Criar arquivo ex02_utils.js

FundamentosMEAN/node/ex02_utils.js

```
function upper(text) {  
  return text.toUpperCase()  
}  
  
module.exports = { upper }
```

Listagem 58 - Criar arquivo ex02_teste.js

FundamentosMEAN/node/ex02_teste.js

```
const utils = require('./ex02_utils')  
console.log(utils.upper('show de bola!'))
```



Se você instalou o plugin `atom-runner` no Atom, basta está no documento e apertar `alt + r` para executar o arquivo `ex02_teste.js`.

3.3. Node - Exercício 03: Singleton

Listagem 59 - Criar arquivo ex03_singleton.js

FundamentosMEAN/node/ex03_singleton.js

```
let numero = 1

function exibirProximo() {
  console.log(numero++)
}

module.exports = { exibirProximo }
```

Listagem 60 - Criar arquivo ex03_teste.js

FundamentosMEAN/node/ex03_teste.js

```
const s1 = require('./ex03_singleton')
const s2 = require('./ex03_singleton')

s1.exibirProximo()
s2.exibirProximo()
s1.exibirProximo()
s2.exibirProximo()
```



No Atom aperte **alt + r** para executar o arquivo **ex03_teste.js**.

3.4. Node - Exercício 04: Objeto Global

Listagem 61 - Criar arquivo ex04_global.js

FundamentosMEAN/node/ex04_global.js

```
const PI = 3.14
console.log(global.PI)

global.obj = { name: 'Estou no global!' }
```



No Atom aperte **alt + r** para executar o arquivo **ex04_global.js**.

Listagem 62 - Criar arquivo ex04_teste.js

FundamentosMEAN/node/ex04_teste.js

```
require('./ex04_global')  
  
console.log(global.obj.name)  
console.log(obj.name)
```



No Atom aperte **alt + r** para executar o arquivo **ex04_teste.js**.

3.5. Node - Exercício 05: This

Listagem 63 - Criar arquivo ex05_module.js

FundamentosMEAN/node/ex05_module.js

```
console.log(global === this)  
console.log(module === this)  
console.log(module.exports === this)  
  
this.digaOi = function () {  
  console.log('Oi!!!')  
}
```

Listagem 64 - Criar arquivo ex05_teste.js

FundamentosMEAN/node/ex05_teste.js

```
const modulo = require('./ex05_module')  
  
modulo.digaOi()
```



No Atom aperte **alt + r** para executar o arquivo **ex05_teste.js**.

3.6. Node - Exercício 06: Módulo Externo (Lodash)

Listagem 65 - Criar arquivo ex06_lodash.js

FundamentosMEAN/node/ex06_lodash.js

```
const _ = require('lodash')

const alunos = [{
  nome: 'Joao',
  nota: 7.6
}, {
  nome: 'Maria',
  nota: 8.6
}, {
  nome: 'Pedro',
  nota: 8.1
}]

const media = _.sumBy(alunos, 'nota') / alunos.length
console.log(media)
```



Abra o terminal e dentro da pasta **FundamentosMEAN/node** execute os seguintes comandos:

Listagem 66 - Criar arquivo package.json

FundamentosMEAN/node

```
npm init -y
```

Listagem 67 - Instalando a dependência lodash

```
npm i lodash --save
```

Listagem 68 - Criar arquivo .gitignore

FundamentosMEAN/node

```
node_modules
```



No Atom aperte **alt + r** para executar o arquivo **ex06_lodash.js**.

3.7. Node - Exercício 07: Passagem de Parâmetros

Listagem 69 - Criar arquivo ex07_param.js

FundamentosMEAN/node/ex07_param.js

```
module.exports = function(param) {  
  console.log(`0 param informado foi ${param}`)  
}
```

Listagem 70 - Criar arquivo ex07_teste.js

FundamentosMEAN/node/ex07_teste.js

```
const moduloComoParam = require('./ex07_param')  
moduloComoParam('param1')
```



No Atom aperte **alt + r** para executar o arquivo `ex07_teste.js`.

3.8. Node - Exercício 08: Process (ARGV)



No Atom aperte **alt + r** para executar o arquivo **ex08_process.js**.

Listagem 71 - Criar arquivo ex08_process.js

FundamentosMEAN/node/ex08_process.js

```
function temParam(param) {  
  return process.argv.indexOf(param) !== -1  
}  
  
if(temParam('--producao')){  
  console.log('Atenção total!')  
} else {  
  console.log('Tranquilo!!!')  
}
```



Abra o terminal e dentro da pasta **FundamentosMEAN/node** execute os seguintes comandos:

Listagem 72 - Executar arquivo ex08_process.js

```
node ex08_process
```

Listagem 73 - Executar arquivo ex08_process.js

```
node ex08_process --prducao
```

3.9. Node - Exercício 09: Process (STDIN/STDOUT)

Listagem 74 - Criar arquivo ex09_process.js

FundamentosMEAN/node/ex09_process.js

```
process.stdout.write('Está gostando do curso? ')
process.stdin.on('data', function(data) {
  process.stdout.write(`Sua resposta foi ${data.toString()}Obrigado!\n`)
  process.exit()
})
```



No Atom aperte **alt + r** para executar o arquivo `ex09_process.js`.

3.10. Node - Exercício 10: Módulo FS

Listagem 75 - Criar arquivo ex10_fs.js

FundamentosMEAN/node/ex10_fs.js

```
const fs = require('fs')
const files = fs.readdirSync(__dirname)

files.forEach(f => console.log(f))
```



No Atom aperte **alt + r** para executar o arquivo `ex10_fs.js`.

3.11. Node - Exercício 11: Módulo HTTP

Listagem 76 - Criar arquivo ex11_http.js

FundamentosMEAN/node/ex11_http.js

```
const http = require('http')
const server = http.createServer(function(req, res) {
  res.writeHead(200, {"Content-Type": "text/html"})
  res.end('<h1>Acho que é melhor usar o Express, não?</h1>')
})

const porta = 3456
server.listen(porta, function() {
  console.log(`Escutando a ${porta}`)
})
```



No Atom aperte **alt + r** para executar o arquivo **ex11_http.js**.

4. Express

4.1. Express - Visão Geral

4.2. Express - Exercício 01: Configuração e Mapeando uma Rota

Abra o terminal e dentro da pasta do projeto FundamentosMEAN execute o seguinte comando:

Listagem 77 - Criar pasta express

FundamentosMEAN/express

```
mkdir express && cd express
```

Listagem 78 - Criar arquivo package.json

FundamentosMEAN/express/package.json

```
npm init
```



Durante a criação do package.json mude o nome do projeto para **exercicios_express** e coloque seu nome como autor e as demais configurações deixe padrão.

Listagem 79 - Instalando a dependência

```
npm i --save express
```

Listagem 80 - Abrir o Atom

```
atom .
```

Listagem 81 - Criar arquivo ex01.js

FundamentosMEAN/express/ex01.js

```
const express = require('express')
const server = express()

server.get('/', function(req, res){
  res.send('<h1>Index</h1>')
})

server.all('/teste', function(req, res){
  res.send('<h1>Teste!</h1>')
})

server.get(/api/, function(req, res){
  res.send('<h1>API!</h1>')
})

server.listen(3000, () => console.log('Executando...'))
```



No Atom aperte **alt + r** para executar o arquivo **ex01.js**.



Outro plugin que foi instalado no Atom foi o **browser-plus** para visualizar as páginas e para executar o browser aperte **ctrl + shift + p** que será exibido um campo e você digita **Browser Plus Open** e clique nele. No browser digite a url **http:localhost:3000** que será exibida a página index.

4.3. Express - Exercício 02: Cadeia de Middlewares

Listagem 82 - Criar arquivo ex02.js

FundamentosMEAN/express/ex02.js

```
const express = require('express')
const server = express()

server.get('/', function(req, res, next){
  console.log('Inicio...')
  next()
  console.log('Fim...')
})

server.get('/', function(req, res){
  console.log('Resposta...')
  res.send('<h1>Olá Express</h1>')
})

server.listen(3000, () => console.log('Executando...'))
```



No Atom aperte **alt + r** para executar o arquivo **ex02.js**.

4.4. Express - Exercício 03: Método USE

Listagem 83 - Criar arquivo ex03.js

FundamentosMEAN/express/ex03.js

```
const express = require('express')
const server = express()

server.use(function(req, res, next){
  console.log('Inicio...')
  next()
  console.log('Fim...')
})

server.use(function(req, res){
  console.log('Resposta...')
  res.send('<h1>API!</h1>')
})

server.listen(3000, () => console.log('Executando...'))
```



No Atom aperte **alt + r** para executar o arquivo **ex03.js**.

4.5. Express - Exercício 04: Método Route

Listagem 84 - Criar arquivo ex04.js

FundamentosMEAN/express/ex04.js

```
const express = require('express')
const server = express()

server.route('/clientes')
  .get((req, res) => res.send('Lista de Clientes'))
  .post((req, res) => res.send('Novo Cliente'))
  .put((req, res) => res.send('Altera Cliente'))
  .delete((req, res) => res.send('Remove Cliente'))

server.listen(3000, () => console.log('Executando...'))
```

4.6. Express - Exercício 05: Express Router

Listagem 85 - Criar arquivo ex05_routes.js

FundamentosMEAN/express/ex05_routes.js

```
const express = require('express')
const router = express.Router()

router.use((req, res, next) => {
  const init = Date.now()
  next()
  console.log(`Tempo = ${Date.now() - init} ms.`)
})

router.get('/produtos/:id', (req, res) => {
  res.json({id: req.params.id, name: 'Caneta'})
})

router.get('/clientes/:id/:name', (req, res) => {
  res.json({id: req.params.id, name: req.params.name})
})

module.exports = router
```

Listagem 86 - Criar arquivo ex05.js

FundamentosMEAN/express/ex05.js

```
const express = require('express')
const server = express()
const router = require('./ex05_routes')

server.use('/api', router)

server.listen(3000, () => console.log('Executando...'))
```



No Atom aperte **alt + r** para executar o arquivo **ex05.js**.

4.7. Express - Exercício 06: Express e Router são Singletons?

Listagem 87 - Criar arquivo ex06.js

FundamentosMEAN/express/ex06.js

```
const express1 = require('express')
const express2 = require('express')
console.log(express1 === express2)

const server1 = express1()
const server2 = express1()
console.log(server1 === server2)

const router1 = express1.Router()
const router2 = express1.Router()
console.log(router1 === router2)
```



No Atom aperte **alt + r** para executar o arquivo **ex06.js**.

5. Angular

5.1. Angular - Instalando via NPM

Abra o terminal e dentro da pasta do projeto FundamentosMEAN execute o seguinte comando:

Listagem 88 - Criar pasta angular1

FundamentosMEAN/angular1

```
mkdir angular1 && cd angular1
```

Listagem 89 - Criar arquivo package.json

FundamentosMEAN/angular1/package.json

```
npm init
```



Durante a criação do package.json mude o nome do projeto para **exercicios angular** e coloque seu nome como autor e as demais configurações deixe padrão.

Listagem 90 - Instalando a dependência

```
npm i angular --save
```

Listagem 91 - Abrir o Atom

```
atom .
```

Listagem 92 - Criar arquivo .gitignore

FundamentosMEAN/angular1/.gitignore

```
node_modules
```

5.2. Angular - Exercício 01: Configurando uma Página com Angular

Listagem 93 - Criar arquivo ex01.html

FundamentosMEAN/angular1/ex01.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
    </script>
  </head>
  <body>
    {{ 1 + 1 }}
  </body>
</html>
```

5.3. Angular - Exercício 02: Binding

Listagem 94 - Criar arquivo ex02.html

FundamentosMEAN/angular1/ex02.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
    </script>
  </head>
  <body>
    <h1>{{ value }}</h1>
    <input ng-model="value">
  </body>
</html>
```

5.4. Angular - Exercício 03: Controller

Listagem 95 - Criar arquivo ex03.html

FundamentosMEAN/angular1/ex03.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').controller('MeuController', [
        '$scope',
        function($scope){
          $scope.value = 10
          $scope.inc = function() {
            $scope.value++
          }
        }
      ])
    </script>
  </head>
  <body>
    <div ng-controller="MeuController">
      <h1>{{ value }}</h1>
      <input ng-model="value"/>
      <button ng-click="inc()">Inc</button>
    </div>
  </body>
</html>
```

5.5. Angular - Exercício 04: Controller As

Listagem 96 - Criar arquivo ex04.html

FundamentosMEAN/angular1/ex04.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').controller('MeuController', [
        function() {
          const self = this
          self.value = 10
          self.inc = function () {
            self.value++
          }
        }
      ])
    </script>
  </head>
  <body>
    <div ng-controller="MeuController as ctrl">
      <h1>{{ ctrl.value }}</h1>
      <input ng-model="ctrl.value"/>
      <button ng-click="ctrl.inc()">Inc</button>
    </div>
  </body>
</html>
```

5.6. Angular - Exercício 05: Filter

Listagem 97 - Criar arquivo ex05.html

FundamentosMEAN/angular1/ex05.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').filter('upper', function() {
        return function (input) {
          return input.toUpperCase()
        }
      })
      angular.module('app').controller('MeuController', [
        function(){
          const self = this
          self.value = "texto"
        }
      ])
    </script>
  </head>
  <body>
    <div ng-controller="MeuController as ctrl">
      <h1>{{ ctrl.value | upper }}</h1>
      <input ng-model="ctrl.value"/>
    </div>
  </body>
</html>
```

5.7. Angular - Exercício 06: Factory

Listagem 98 - Criar arquivo ex06.html

FundamentosMEAN/angular1/ex06.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').factory('utils', function() {
        function upper(input) {
          return input.toUpperCase()
        }
        return{upper}
      })
      angular.module('app').controller('MeuController', [
        'utils',
        function (utils) {
          const self = this
          self.value = utils.upper("texto")
        }
      ])
    </script>
  </head>
  <body>
    <div ng-controller="MeuController as ctrl">
      <h1>{{ ctrl.value }}</h1>
    </div>
  </body>
</html>
```

5.8. Angular - Exercício 07: Service

Listagem 99 - Criar arquivo ex07.html

FundamentosMEAN/angular1/ex07.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').service('utils', function() {
        this.upper = function(input) {
          return input.toUpperCase()
        }
      })
      angular.module('app').controller('MeuController', [
        'utils',
        function (utils) {
          const self = this
          self.value = utils.upper("texto")
        }
      ])
    </script>
  </head>
  <body>
    <div ng-controller="MeuController as ctrl">
      <h1>{{ ctrl.value }}</h1>
    </div>
  </body>
</html>
```

5.9. Angular - Exercício 08: Directive

Listagem 100 - Criar arquivo ex08.html

FundamentosMEAN/angular1/ex08.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').directive('meuRelogio', function() {
        return {
          restrict: 'AE',
          template: '<h2>${new Date}</h2>'
        }
      })
    </script>
  </head>
  <body>
    <h1>Atributo</h1>
    <div meu-relogio></div>
    <hr>
    <h1>Elemento</h1>
    <meu-relogio></meu-relogio>
  </body>
</html>
```

5.10. Angular - Exercício 09: Component

Listagem 101 - Criar arquivo ex09.html

FundamentosMEAN/angular1/ex09.html

```
<!DOCTYPE html>
<html ng-app='app'>
  <head>
    <meta charset="utf-8">
    <title>Fundamentos de Angular</title>
    <script type="text/javascript" src="node_modules/angular/angular.min.js"
  ></script>
    <script type="text/javascript">
      angular.module('app', [])
      angular.module('app').component('field', {
        bindings: {
          id: '@',
          label: '@',
          model: '='
        },
        template: `
          <label for="{{ $ctrl.id }}">{{ $ctrl.label }}</label>
          <input id="{{ $ctrl.id }}" ng-model='$ctrl.model'>
        `
      })
      angular.module('app').controller('MeuController', function() {
        const self = this
        self.name = 'Anônimo'
      })
    </script>
  </head>
  <body>
    <div ng-controller="MeuController as ctrl">
      <h1>{{ ctrl.name }}</h1>
      <field id="nome" label="Nome" model="ctrl.name"></field>
    </div>
  </body>
</html>
```

6. Backend - Configurações Iniciais

6.1. Versão Inicial do Projeto Backend

Abra o terminal e dentro da pasta Desktop execute o seguinte comando:

Listagem 102 - Criar pasta CursoFramesWeb

Desktop/CursoFramesWeb

```
mkdir CursoFramesWeb && cd CursoFramesWeb
```

Listagem 103 - Criar pasta backend

CursoFramesWeb/backend

```
mkdir backend && cd backend
```

Listagem 104 - Criar arquivo package.json

CursoFramesWeb/backend/package.json

```
npm init
```



Durante a criação do package.json mude o entry point para `loader.js` e coloque seu nome como autor e as demais configurações deixe padrão.

Listagem 105 - Instalando as dependências

```
npm i express body-parser mongoose node-restful mongoose-paginate lodash express-query-int pm2 --save
```

Listagem 106 - Instalando a dependência

```
npm i nodemon --save-dev
```

Listagem 107 - Abrir o Atom

```
atom .
```

Listagem 108 - Alterar arquivo package.json

CursoFramesWeb/backend/package.json (aprox. linha 7)

```
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "loader.js",
  "scripts": {
    "dev": "nodemon",
    "production": "pm2 start loader.js --name backend"
  },
  "author": "Leonardo Leitão",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.17.1",
    "express": "^4.15.2",
    "express-query-int": "^1.0.1",
    "lodash": "^4.17.4",
    "mongoose": "^4.9.2",
    "mongoose-paginate": "^5.0.3",
    "node-restful": "^0.2.6",
    "pm2": "^2.4.2"
  },
  "devDependencies": {
    "nodemon": "^1.11.0"
  }
}
```

6.2. Adicionando o .gitignore ao Projeto

Listagem 109 - Criar arquivo .gitignore

CursoFramesWeb/backend/.gitignore

```
node_modules
*.log
```

6.3. Implementando o Servidor com Express

Listagem 110 - Criar arquivo loader.js

CursoFramesWeb/backend/loader.js

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log(`BACKEND is running on port ${port}.`);
})
```

Listagem 111 - Executar a aplicação

```
npm run dev
```

6.4. Criando a Conexão com MongoDB



Criar uma pasta chamada **config** em **backend/config**.



O conteúdo do arquivo **loader.js** foi movido para o arquivo **server.js**.

Listagem 112 - Criar arquivo server.js

CursoFramesWeb/backend/config/server.js

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log(`BACKEND is running on port ${port}.`);
})
```

Listagem 113 - Criar arquivo database.js

CursoFramesWeb/backend/config/database.js

```
const mongoose = require('mongoose')
module.exports = mongoose.connect('mongodb://localhost/db_finance')
```

Listagem 114 - Alterar arquivo loader.js

CursoFramesWeb/backend/loader.js (na linha 1)

```
require('./config/server')
require('./config/database')
```



Abra um novo terminal para executar o servidor do Mongo.

Listagem 115 - Executar o servidor Mongo

```
mongod
```


7. Backend - Ciclo de Pagamento API

7.1. Mapeamento ODM do Objeto Ciclo de Pagamentos



Criar uma pasta chamada **api** e dentro dela criar uma pasta chamada **billingCycle** em **backend/api/billingCycle**

Listagem 116 - Criar arquivo *billingCycle.js*

CursoFramesWeb/backend/api/billingCycle/billingCycle.js

```
const restful = require('node-restful')
const mongoose = restful.mongoose

const creditSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: true }
})

const debtSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: true },
  status: { type: String, required: false, uppercase: true,
    enum: ['PAGO', 'PENDENTE', 'AGENDADO'] }
})

const billingCycleSchema = new mongoose.Schema({
  name: { type: String, required: true },
  month: { type: Number, min: 1, max: 12, required: true },
  year: { type: Number, min: 1970, max: 2100, required: true },
  credits: [creditSchema],
  debts: [debtSchema]
})

module.exports = restful.model('BillingCycle', billingCycleSchema)
```

7.2. Serviço de Ciclo de Pagamentos

Listagem 117 - Criar arquivo *billingCycleService.js*

CursoFramesWeb/backend/api/billingCycle/billingCycleService.js

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])

module.exports = BillingCycle
```

7.3. Criando o Arquivo de Rotas

Listagem 118 - Criar arquivo *routes.js*

CursoFramesWeb/backend/config/routes.js

```
const express = require('express')

module.exports = function(server) {

  // API Routes
  const router = express.Router()
  server.use('/api', router)
}
```

Listagem 119 - Alterar arquivo *loader.js*

CursoFramesWeb/backend/loader.js (aprox. linha 1 e 3)

```
const server = require('./config/server')
require('./config/database')
require('./config/routes')(server)
```

Listagem 120 - Alterar arquivo server.js

CursoFramesWeb/backend/config/server.js (aprox. linha 14)

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())

server.listen(port, function() {
  console.log('BACKEND is running on port ${port}.');
})

module.exports = server
```

7.4. Registrando as Rotas do Serviço de Ciclo de Pagamentos

Listagem 121 - Alterar arquivo routes.js

CursoFramesWeb/backend/config/routes.js (aprox. linha 9)

```
const express = require('express')

module.exports = function (server) {

  // API Routes
  const router = express.Router()
  server.use('/api', router)

  // rotas da API
  const billingCycleService = require('../api/billingCycle/billingCycleService')
  billingCycleService.register(router, '/billingCycles')
}
```

7.5. Testando a API de Ciclo de Pagamentos (Parte 1)

Listagem 122 - Alterar arquivo database.js

CursoFramesWeb/backend/config/database.js (aprox. linha 4)

```
const mongoose = require('mongoose')
module.exports = mongoose.connect('mongodb://localhost/db_finance')

mongoose.Error.messages.general.required = "O atributo '{PATH}' é obrigatório"
```

Listagem 123 - Alterar arquivo billingCycle.js

CursoFramesWeb/backend/api/billingCycle/billingCycle.js (aprox. linha 11)

```
const restful = require('node-restful')
const mongoose = restful.mongoose

const creditSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: true }
})

const debtSchema = new mongoose.Schema({
  name: { type: String, required: true },
  value: { type: Number, min: 0, required: [true, 'Informe o valor do débito!']
},
  status: { type: String, required: false, uppercase: true,
    enum: ['PAGO', 'PENDENTE', 'AGENDADO'] }
})

const billingCycleSchema = new mongoose.Schema({
  name: { type: String, required: true },
  month: { type: Number, min: 1, max: 12, required: true },
  year: { type: Number, min: 1970, max: 2100, required: true },
  credits: [creditSchema],
  debts: [debtSchema]
})

module.exports = restful.model('BillingCycle', billingCycleSchema)
```

7.6. Testando a API de Ciclo de Pagamentos (Parte 2)

Listagem 124 - Alterar arquivo *billingCycleService.js*

CursoFramesWeb/backend/api/billingCycle/billingCycleService.js (aprox. linha 4)

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})

module.exports = BillingCycle
```

Listagem 125 - Alterar arquivo *database.js*

CursoFramesWeb/backend/config/database.js (aprox. linha 5)

```
const mongoose = require('mongoose')
module.exports = mongoose.connect('mongodb://localhost/db_finance')

mongoose.Error.messages.general.required = "O atributo '{PATH}' é obrigatório"
mongoose.Error.messages.Number.min = "O '{VALUE}' informado é menor que o limite mínimo de '{MIN}'."
mongoose.Error.messages.Number.max = "O '{VALUE}' informado é maior que o limite máximo de '{MAX}'."
mongoose.Error.messages.String.enum = "'{VALUE}' não é válido para o atributo '{PATH}'."
```

7.7. Testando a API de Ciclo de Pagamentos (Parte 3)

7.8. Serviço Contador (count) de Ciclo de Pagamentos

Listagem 126 - Alterar arquivo *billingCycleService.js*

CursoFramesWeb/backend/api/billingCycle/billingCycleService.js (aprox. linha 6)

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})

BillingCycle.route('count', function (req, res, next) {
  BillingCycle.count(function (error, value) {
    if (error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json({value})
    }
  })
})

module.exports = BillingCycle
```

8. Backend - Sumário de Pagamento API

8.1. Serviço de Sumário de Pagamentos



Criar uma pasta chamada `billingSummary` em `backend/api/billingSummary`

Listagem 127 - Criar arquivo `billingSummaryService.js`

`CursoFramesWeb/backend/api/billingSummary/billingSummaryService.js`

```
const _ = require('lodash')
const BillingCycle = require('../billingCycle/billingCycle')

// Mais uma função middleware
function getSummary(req, res) {
  BillingCycle.aggregate({
    $project: {credit: {$sum: "$credits.value"}, debt: {$sum: "$debts.value"}}
  }, {
    $group: {_id: null, credit: {$sum: "$credit"}, debt: {$sum: "$debt"}}
  }, {
    $project: {_id: 0, credit: 1, debt: 1}
  }, function (error, result) {
    if (error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json(_.defaults(result[0], {credit:0, debt: 0}))
    }
  })
}

module.exports = { getSummary }
```

8.2. Registrando a Rota do Serviço de Sumário de Pagamentos

Listagem 128 - Alterar arquivo routes.js

CursoFramesWeb/backend/config/routes.js (aprox. linha 13)

```
const express = require('express')

module.exports = function (server) {

  // API Routes
  const router = express.Router()
  server.use('/api', router)

  // rotas da API
  const billingCycleService = require('../api/billingCycle/billingCycleService')
  billingCycleService.register(router, '/billingCycles')

  const billingSummaryService =
  require('../api/billingSummary/billingSummaryService')
  router.route('/billingSummary').get(billingSummaryService.getSummary)
}
```


9. Backend - Ajustes Finais

9.1. Uniformizando as Mensagens de Erro

Listagem 129 - Alterar arquivo *billingCycleService.js*

CursoFramesWeb/backend/api/billingCycle/billingCycleService.js (aprox. linha 7, 9 e 20)

```
const BillingCycle = require('./billingCycle')

BillingCycle.methods(['get', 'post', 'put', 'delete'])
BillingCycle.updateOptions({new: true, runValidators: true})

BillingCycle.after('post', sendErrorsOrNext).after('put', sendErrorsOrNext)

function sendErrorsOrNext(req, res, next) {
  const bundle = res.locals.bundle

  if (bundle.errors) {
    const errors = parseErrors(bundle.errors)
    res.status(500).json({errors})
  } else {
    next()
  }
}

function parseErrors(nodeRestfulErrors) {
  const errors = []
  _.forIn(nodeRestfulErrors, error => errors.push(error.message))
  return errors
}

BillingCycle.route('count', function (req, res, next) {
  BillingCycle.count(function (error, value) {
    if (error) {
      res.status(500).json({errors: [error]})
    } else {
      res.json({value})
    }
  })
})

module.exports = BillingCycle
```

10. Frontend - Configuração do Build do Projeto (Gulp)

10.1. Versão Inicial do Projeto Frontend

Abra o terminal e dentro da pasta CursoFramesWeb execute o seguinte comando:

Listagem 130 - Criar pasta angular1

CursoFramesWeb/angular1

```
mkdir angular1 && cd angular1
```

Listagem 131 - Criar arquivo package.json

CursoFramesWeb/angular1/package.json

```
npm init
```



Durante a criação do package.json coloque seu nome como autor e as demais configurações deixe padrão.

Listagem 132 - Instalando as dependências

```
npm install admin-lte angular angular-animate angular-toastr angular-ui-router  
font-awesome babel-preset-es2015 gulp gulp-babel gulp-concat gulp-htmlmin gulp-  
uglify gulp-uglifycss gulp-util gulp-watch gulp-webserver --save-dev
```

Listagem 133 - Abrir o Atom

```
atom .
```

Listagem 134 - Alterar arquivo package.json

CursoFramesWeb/angular1/package.json (aprox. linha 7)

```
{
  "name": "angular1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "dev": "gulp",
    "production": "gulp --production"
  },
  "keywords": [],
  "author": "Leonardo Leitão",
  "license": "ISC",
  "devDependencies": {
    "admin-lte": "^2.3.11",
    "angular": "^1.6.4",
    "angular-animate": "^1.6.4",
    "angular-toastr": "^2.1.1",
    "angular-ui-router": "^0.4.2",
    "babel-preset-es2015": "^6.24.0",
    "font-awesome": "^4.7.0",
    "gulp": "^3.9.1",
    "gulp-babel": "^6.1.2",
    "gulp-concat": "^2.6.1",
    "gulp-htmlmin": "^3.0.0",
    "gulp-uglify": "^2.1.2",
    "gulp-uglifycss": "^1.0.8",
    "gulp-util": "^3.0.8",
    "gulp-watch": "^4.3.11",
    "gulp-webserver": "^0.9.1",
    "install": "^0.8.8",
    "npm": "^4.4.4"
  }
}
```

10.2. Adicionando o .gitignore ao Projeto

Listagem 135 - Criar arquivo .gitignore

CursoFramesWeb/angular1/.gitignore

```
node_modules
public
*.log
```

10.3. Estrutura Inicial do Build

Listagem 136 - Criar arquivo gulpfile.js

CursoFramesWeb/angular1/gulpfile.js

```
const gulp = require('gulp')
const util = require('gulp-util')

require('./gulpTasks/app')
require('./gulpTasks/deps')
require('./gulpTasks/server')

gulp.task('default', function() {
  if(util.env.production) {
    gulp.start('deps', 'app')
  } else {
    gulp.start('deps', 'app', 'server')
  }
})
```



Criar uma pasta chamada **gulpTasks** em **angular1/gulpTasks**

Listagem 137 - Criar arquivo app.js

CursoFramesWeb/angular1/gulpTasks/app.js

```
const gulp = require('gulp')

gulp.task('app', ['app.html', 'app.css', 'app.js', 'app.assets'])

gulp.task('app.html', function () {

})

gulp.task('app.css', function () {

})

gulp.task('app.js', function () {

})

gulp.task('app.assets', function () {

})
```

Listagem 138 - Criar arquivo deps.js

CursoFramesWeb/angular1/gulpTasks/deps.js

```
const gulp = require('gulp')

gulp.task('deps', ['deps.js', 'deps.css', 'deps.fonts'])

gulp.task('deps.js', function () {

})

gulp.task('deps.css', function () {

})

gulp.task('deps.fonts', function () {

})
```

Listagem 139 - Criar arquivo server.js

CursoFramesWeb/angular1/gulpTasks/server.js

```
const gulp = require('gulp')

gulp.task('server', ['watch'], function(){

})

gulp.task('watch', function () {

})
```



Abra o terminal e dentro da pasta **angular1** execute o seguinte comando:

Listagem 140 - Executar a aplicação

```
npm run dev
```

10.4. Build dos Arquivos das Dependências

Listagem 141 - Alterar arquivo *deps.js*

CursoFramesWeb/angular1/gulpTasks/deps.js (aprox. linha 2, 8, 24, e 37)

```
const gulp = require('gulp')
const uglify = require('gulp-uglify')
const concat = require('gulp-concat')
const uglifycss = require('gulp-uglifycss')

gulp.task('deps', ['deps.js', 'deps.css', 'deps.fonts'])

gulp.task('deps.js', function () {
  gulp.src([
    'node_modules/angular/angular.min.js',
    'node_modules/angular-ui-router/release/angular-ui-router.min.js',
    'node_modules/angular-animate/angular-animate.min.js',
    'node_modules/angular-toastr/dist/angular-toastr.tpls.min.js',
    'node_modules/admin-lte/plugins/jquery/jquery-2.2.3.min.js',
    'node_modules/admin-lte/bootstrap/js/bootstrap.min.js',
    'node_modules/admin-lte/plugins/slimScroll/jquery.slimscroll.min.js',
    'node_modules/admin-lte/dist/js/app.min.js'
  ])
  .pipe(uglify())
  .pipe(concat('deps.min.js'))
  .pipe(gulp.dest('public/assets/js'))
})

gulp.task('deps.css', function () {
  gulp.src([
    'node_modules/angular-toastr/dist/angular-toastr.min.css',
    'node_modules/font-awesome/css/font-awesome.min.css',
    'node_modules/admin-lte/bootstrap/css/bootstrap.min.css',
    'node_modules/admin-lte/dist/css/AdminLTE.min.css',
    'node_modules/admin-lte/dist/css/skins/_all-skins.min.css'
  ])
  .pipe(uglifycss({ "uglyComments": true }))
  .pipe(concat('deps.min.css'))
  .pipe(gulp.dest('public/assets/css'))
})

gulp.task('deps.fonts', function () {
  gulp.src([
    'node_modules/font-awesome/fonts/*.*',
    'node_modules/admin-lte/bootstrap/fonts/*.*'
  ])
  .pipe(gulp.dest('public/assets/fonts'))
})
```



Abra o terminal e dentro da pasta **angular1** execute o seguinte comando:

Listagem 142 - Executar a aplicação

```
npm run dev
```

10.5. Build dos Arquivos do Projeto



Criar uma pasta chamada **assets** em **angular1/assets**

Listagem 143 - Alterar arquivo app.js

CursoFramesWeb/angular1/gulpTasks/app.js (aprox. linha 2, 10, 16, 23 e 31)

```
const gulp = require('gulp')
const htmlmin = require('gulp-htmlmin')
const uglifycss = require('gulp-uglifycss')
const concat = require('gulp-concat')
const babel = require('gulp-babel')
const uglify = require('gulp-uglify')

gulp.task('app', ['app.html', 'app.css', 'app.js', 'app.assets'])

gulp.task('app.html', function () {
  gulp.src('app/**/*.html')
    .pipe(htmlmin({ collapseWhitespace: true }))
    .pipe(gulp.dest('public'))
})

gulp.task('app.css', function () {
  gulp.src('app/**/*.css')
    .pipe(uglifycss({ "uglyComments": true }))
    .pipe(concat('app.min.css'))
    .pipe(gulp.dest('public/assets/css'))
})

gulp.task('app.js', function () {
  gulp.src('app/**/*.js')
    .pipe(babel({ presets: ['es2015'] }))
    .pipe(uglify())
    .pipe(concat('app.min.js'))
    .pipe(gulp.dest('public/assets/js'))
})

gulp.task('app.assets', function () {
  gulp.src('assets/**/*.*)
    .pipe(gulp.dest('public/assets'))
})
```



Abra o terminal e dentro da pasta **angular1** execute o seguinte comando:

Listagem 144 - Executar a aplicação

```
npm run dev
```

10.6. Configurando o Servidor com Reload Automático

Listagem 145 - Alterar arquivo server.js

CursoFramesWeb/angular1/gulpTasks/server.js (aprox. linha 3, 5 e 13)

```
const gulp = require('gulp')
const watch = require('gulp-watch')
const webserver = require('gulp-webserver')

gulp.task('server', ['watch'], function(){
  gulp.src('public').pipe(webserver({
    livereload: true,
    port: 3000,
    open: true
  }))
})

gulp.task('watch', function () {
  watch('app/**/*.html', () => gulp.start('app.html'))
  watch('app/**/*.css', () => gulp.start('app.css'))
  watch('app/**/*.js', () => gulp.start('app.js'))
  watch('assets/**/*..*', () => gulp.start('app.assets'))
})
```



Abra o terminal e dentro da pasta **angular1** execute o seguinte comando:

Listagem 146 - Executar a aplicação

```
npm run dev
```

11. Frontend - Estrutura Básica do Projeto (Template)

11.1. Implementando a Página Index



Criar uma pasta chamada `app` em `angular1/app`

Listagem 147 - Criar arquivo `index.html`

CursoFramesWeb/angular1/app/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Primeira App</title>
    <link rel="stylesheet" href="/assets/css/deps.min.css">
    <link rel="stylesheet" href="/assets/css/app.min.css">
  </head>
  <body class="skin-blue fixed sidebar-mini">
    <div class="wrapper">
      <header class="main-header" ng-include="'/template/header.html'"></header>
      <aside class="main-sidebar" ng-include="'/template/menu.html'"></aside>
      <div class="content-wrapper" ui-view></div>
      <footer class="main-footer" ng-include="'/template/footer.html'"></footer>
    </div>
    <script src="/assets/js/deps.min.js"></script>
    <script src="/assets/js/app.min.js"></script>
  </body>
</html>
```



Criar uma pasta chamada `template` em `app/template`

Listagem 148 - Criar arquivo `menu.html`

CursoFramesWeb/angular1/app/template/menu.html

```
<h1>Menu</h1>
```

Listagem 149 - Criar arquivo header.html

CursoFramesWeb/angular1/app/template/header.html

```
<h1>Header</h1>
```

Listagem 150 - Criar arquivo footer.html

CursoFramesWeb/angular1/app/template/footer.html

```
<h1>Footer</h1>
```

11.2. Adicionando o Angular ao Projeto

Listagem 151 - Criar arquivo index.js

CursoFramesWeb/angular1/app/index.js

```
angular.module('primeiraApp', [  
  'ui.router',  
  'ngAnimate',  
  'toastr'  
)
```

Listagem 152 - Alterar arquivo index.html

CursoFramesWeb/angular1/app/index.html (aprox. linha 2)

```
<!DOCTYPE html>
<html ng-app="primeiraApp">
  <head>
    <meta charset="utf-8">
    <title>Primeira App</title>
    <link rel="stylesheet" href="/assets/css/deps.min.css">
    <link rel="stylesheet" href="/assets/css/app.min.css">
  </head>
  <body class="skin-blue fixed sidebar-mini">
    <div class="wrapper">
      <header class="main-header" ng-include="'/template/header.html'"></header>
      <aside class="main-sidebar" ng-include="'/template/menu.html'"></aside>
      <div class="content-wrapper" ui-view></div>
      <footer class="main-footer" ng-include="'/template/footer.html'"></footer>
    </div>
    <script src="/assets/js/deps.min.js"></script>
    <script src="/assets/js/app.min.js"></script>
  </body>
</html>
```



Abra o terminal e dentro da pasta **angular1** execute o seguinte comando:

Listagem 153 - Executar a aplicação

```
npm run dev
```

Listagem 154 - Criar arquivo template-custom.css

CursoFramesWeb/angular1/app/template/template-custom.css

```
.main-footer{position: fixed; bottom: 0; width: 100%;}
```

11.3. Implementação o Header

Listagem 155 - Alterar arquivo header.html

CursoFramesWeb/angular1/app/template/header.html (na linha 1)

```
<a href="/" class="logo">
  <span class="logo-mini"><b>P</b>AP</span>
  <span class="logo-lg"><i class="fa fa-leaf"></i> <b>Primeira</b>APP</span>
</a>

<nav class="navbar navbar-static-top">
  <a href="#" class="sidebar-toggle" data-toggle="offcanvas"></a>
</nav>
```

11.4. Implementação o Footer

Listagem 156 - Alterar arquivo footer.html

CursoFramesWeb/angular1/app/template/footer.html (na linha 1)

```
<strong>
  copyright &copy; 2017
  <a href="http://www.cod3r.com.br" target="_blank">Cod3r</a>.
</strong>
```

11.5. Implementação o Menu

Listagem 157 - Alterar arquivo menu.html

CursoFramesWeb/angular1/app/template/menu.html (na linha 1)

```
<section class="sidebar">
  <ul class="sidebar-menu">
    <li>
      <a>
        <i class="fa fa-dashboard"></i> <span>Dashboard</span>
      </a>
    </li>

    <li class="treeview">
      <a href>
        <i class="fa fa-edit"></i> <span>Cadastros</span>
        <i class="fa fa-angle-left pull-right"></i>
      </a>

      <ul class="treeview-menu">
        <li>
          <a><i class="fa fa-usd"></i> Ciclo de Pagamentos</a>
        </li>
      </ul>
    </li>
  </ul>
</section>
```

11.6. Configurando as Rotas com UI Router



Criar uma pasta chamada **dashboard** em **app/dashboard**

Listagem 158 - Criar arquivo dashboard.html

CursoFramesWeb/angular1/app/dashboard/dashboard.html

```
<h1>Dashboard</h1>
```



Criar uma pasta chamada **billingCycle** em **app/billingCycle**

Listagem 159 - Criar arquivo tabs.html

CursoFramesWeb/angular1/app/billingCycle/tabs.html

```
<h1>Ciclo de Pagamentos</h1>
```



Criar uma pasta chamada **config** em **app/config**

Listagem 160 - Criar arquivo routes.js

CursoFramesWeb/angular1/app/config/routes.js

```
angular.module('primeiraApp').config([
  '$stateProvider',
  '$urlRouterProvider',
  function ($stateProvider, $urlRouterProvider) {
    $stateProvider.state('dashboard', {
      url: "/dashboard",
      templateUrl: "dashboard/dashboard.html"
    }).state('billingCycle', {
      url: "/billingCycles",
      templateUrl: "billingCycle/tabs.html"
    })

    $urlRouterProvider.otherwise('/dashboard')
  }
])
```


Listagem 161 - Alterar arquivo menu.html

CursoFramesWeb/angular1/app/template/menu.html (aprox. linha 4 e 17)

```
<section class="sidebar">
  <ul class="sidebar-menu">
    <li>
      <a ui-sref="dashboard">
        <i class="fa fa-dashboard"></i> <span>Dashboard</span>
      </a>
    </li>

    <li class="treeview">
      <a href>
        <i class="fa fa-edit"></i> <span>Cadastros</span>
        <i class="fa fa-angle-left pull-right"></i>
      </a>

      <ul class="treeview-menu">
        <li>
          <a ui-sref="billingCycle"><i class="fa fa-usd"></i> Ciclo de
Pagamentos</a>
        </li>
      </ul>
    </li>
  </ul>
</section>
```

12. Frontend - Dashboard

12.1. Versão Inicial do Dashboard

Listagem 162 - Alterar arquivo dashboard.html

CursoFramesWeb/angular1/app/dashboard/dashboard.html (na linha 1)

```
<section class="content-header">
  <h1>Dashboard <small>Versão 1.0</small></h1>
</section>

<section class="content">
  <div class="col-xs-4">
    <div class="small-box bg-green">
      <div class="inner">
        <h3>R$ 10</h3>
        <p>Total de Créditos</p>
      </div>
      <div class="icon">
        <i class="fa fa-bank"></i>
      </div>
    </div>
  </div>
  <div class="col-xs-4">
    <div class="small-box bg-red">
      <div class="inner">
        <h3>R$ 10</h3>
        <p>Total de Débitos</p>
      </div>
      <div class="icon">
        <i class="fa fa-credit-car"></i>
      </div>
    </div>
  </div>
  <div class="col-xs-4">
    <div class="small-box bg-blue">
      <div class="inner">
        <h3>R$ 10</h3>
        <p>Valor Consolidado</p>
      </div>
      <div class="icon">
        <i class="fa fa-money"></i>
      </div>
    </div>
  </div>
</section>
```

Listagem 163 - Alterar arquivo *template-custom.css*

CursoFramesWeb/angular1/app/template/template-custom.css (na linha 1)

```
html { background-color: #ecf0f5; }  
.main-footer{position: fixed; bottom: 0; width: 100%;}
```

12.2. Componente de Cabeçalho do Conteúdo



Criar uma pasta chamada **common** e dentro dela criar uma pasta chamada **components** em **app/common/components**

Listagem 164 - Criar arquivo *contentHeader.js*

CursoFramesWeb/angular1/app/common/components/contentHeader.js

```
angular.module('primeiraApp').component('contentHeader',{  
  bindings: {  
    name: '@',  
    small: '@',  
  },  
  template: `  
    <section class="content-header">  
      <h1>{{ $ctrl.name }} <small>{{ $ctrl.small }}</small></h1>  
    </section>  
  `,  
})
```

Listagem 165 - Alterar arquivo *dashboard.html*

CursoFramesWeb/angular1/app/dashboard/dashboard.html (na linha 1)

```
<content-header name="Dashboard" small="Versão 1.0"></content-header>

<section class="content">
  <div class="col-xs-4">
    <div class="small-box bg-green">
      <div class="inner">
        <h3>R$ 10</h3>
        <p>Total de Créditos</p>
      </div>
      <div class="icon">
        <i class="fa fa-bank"></i>
      </div>
    </div>
  </div>
  <div class="col-xs-4">
    <div class="small-box bg-red">
      <div class="inner">
        <h3>R$ 10</h3>
        <p>Total de Débitos</p>
      </div>
      <div class="icon">
        <i class="fa fa-credit-car"></i>
      </div>
    </div>
  </div>
  <div class="col-xs-4">
    <div class="small-box bg-blue">
      <div class="inner">
        <h3>R$ 10</h3>
        <p>Valor Consolidado</p>
      </div>
      <div class="icon">
        <i class="fa fa-money"></i>
      </div>
    </div>
  </div>
</section>
```

12.3. Factory para Ajudar no Layout (Bootstrap GridSystem)



Criar uma pasta chamada **factories** em **app/common/factories**

Listagem 166 - Criar arquivo gridSystemFactory.js

CursoFramesWeb/angular1/app/common/factories/gridSystemFactory.js

```
angular.module('primeiraApp').factory('gridSystem', [ function () {  
  
    function toCssClasses (numbers) {  
        const cols = numbers ? numbers.split(' ') : []  
        let classes = ''  
  
        if(cols[0]) classes += `col-xs-${cols[0]}`  
        if(cols[1]) classes += ` col-sm-${cols[1]}`  
        if(cols[2]) classes += ` col-md-${cols[2]}`  
        if(cols[3]) classes += ` col-lg-${cols[3]}`  
  
        return classes  
    }  
  
    return { toCssClasses }  
}])
```

12.4. Componente ValueBox

Listagem 167 - Criar arquivo valueBox.js

CursoFramesWeb/angular1/app/common/components/valueBox.js

```
angular.module('primeiraApp').component('valueBox', {
  bindings: {
    grid: '@',
    colorClass: '@',
    value: '@',
    text: '@',
    iconClass: '@',
  },
  controller: [
    'gridSystem',
    function(gridSystem){
      this.$onInit = () => this.gridClasses = gridSystem.toCssClasses(
this.grid)
    }
  ],
  template: `
<div class="{{ $ctrl.gridClasses }}">
  <div class="small-box {{ $ctrl.colorClass }}">
    <div class="inner">
      <h3>{{ $ctrl.value }}</h3>
      <p>{{ $ctrl.text }}</p>
    </div>
    <div class="icon">
      <i class="{{ $ctrl.iconClass }}"></i>
    </div>
  </div>
</div>
`
})
```

Listagem 168 - Alterar arquivo dashboard.html

CursoFramesWeb/angular1/app/dashboard/dashboard.html (aprox. linha 4)

```
<content-header name="Dashboard" small="Versão 1.0"></content-header>

<section class="content">
  <value-box grid="4" color-class="bg-green" icon-class="fa fa-bank"
    value="R$ 10" text="Total de Créditos"></value-box>
  <value-box grid="4" color-class="bg-red" icon-class="fa fa-credit-card"
    value="R$ 10" text="Total de Débitos"></value-box>
  <value-box grid="4" color-class="bg-blue" icon-class="fa fa-bank"
    value="R$ 10" text="Valor Consolidado"></value-box>
</section>
```

12.5. Inicializando o Backend com PM2



Abra o terminal e dentro da pasta `CursoFramesWeb/backend` execute o seguinte comando:

Listagem 169 - Executar a aplicação com PM2

```
npm run production
```

Listagem 170 - Mostrar o Status do PM2

```
pm2 status backend
```



Abra um novo terminal para executar o servidor do Mongo.

Listagem 171 - Executar o servidor Mongo

```
mongod
```

Listagem 172 - Instalando o PM2 global

```
npm install pm2 -g
```

Listagem 173 - Remover uma aplicação do registro do PM2

```
pm2 delete backend
```

Listagem 174 - Executar a aplicação com PM2

```
npm run production
```

Listagem 175 - Mostrar o Status da aplicação

```
pm2 status backend
```

12.6. Dashboard Controller

Listagem 176 - Criar arquivo dashboardController.js

CursoFramesWeb/angular1/app/dashboard/dashboardController.js

```
angular.module('primeiraApp').controller('DashboardCtrl', [
    '$scope',
    '$http',
    DashboardController
])

function DashboardController($scope, $http) {

    $scope.getSummary = function() {
        const url = 'http://localhost:3003/api/billingSummary'
        $http.get(url).success(function({credit = 0, debt = 0}) {
            $scope.credit = credit
            $scope.debt = debt
            $scope.total = credit - debt
        })
    }

    $scope.getSummary()
}
```

Listagem 177 - Alterar arquivo dashboard.html

CursoFramesWeb/angular1/app/dashboard/dashboard.html (aprox. linha 3, 5, 7 e 9)

```
<content-header name="Dashboard" small="Versão 1.0"></content-header>

<section class="content" ng-controller="DashboardCtrl">
    <value-box grid="4" color-class="bg-green" icon-class="fa fa-bank"
        value="R$ {{ credit | number: 2 }}" text="Total de Créditos"></value-box>
    <value-box grid="4" color-class="bg-red" icon-class="fa fa-credit-card"
        value="R$ {{ debt | number: 2 }}" text="Total de Débitos"></value-box>
    <value-box grid="4" color-class="bg-blue" icon-class="fa fa-bank"
        value="R$ {{ total | number: 2 }}" text="Valor Consolidado"></value-box>
</section>
```

12.7. Atualizando o DashboardController para Angular 1.6

12.8. Habilitando o CORS na Aplicação Backend

Listagem 178 - Criar arquivo cors.js

CursoFramesWeb/backend/config/cors.js (aprox. linha 9)

```
module.exports = function (req, res, next) {  
  res.header('Access-Control-Allow-Origin', '*')  
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, PATCH,  
DELETE')  
  res.header('Access-Control-Allow-Headers', 'Origin, X-Requested-with, Content-  
Type, Accept')  
  next()  
}
```

Listagem 179 - Alterar arquivo server.js

CursoFramesWeb/backend/config/server.js (aprox. linha 6 e 10)

```
const port = 3003  
  
const bodyParser = require('body-parser')  
const express = require('express')  
const server = express()  
const allowCors = require('./cors')  
  
server.use(bodyParser.urlencoded({ extended: true })))  
server.use(bodyParser.json())  
server.use(allowCors)  
  
server.listen(port, function() {  
  console.log('BACKEND is running on port ${port}.');  
})
```



Se o terminal do backend estiver executando a aplicação pare e depois execute os seguintes comandos:

Listagem 180 - Remover uma aplicação do registro do PM2

```
pm2 delete backend
```

Listagem 181 - Executar a aplicação com PM2

```
npm run production
```

Listagem 182 - Executar o PM2 monit

```
pm2 monit
```

12.9. Aplicando dois Padrões do Angular Style Guide

Listagem 183 - Alterar arquivo `dashboardController.js`

`CursoFramesWeb/angular1/app/dashboard/dashboardController.js` (aprox. linha 1, 3, 7, 8, 9, 12, 13, 14, 18 e 20)

```
(function(){
    angular.module('primeiraApp').controller('DashboardCtrl', [
        '$http',
        DashboardController
    ])

    function DashboardController($http) {
        const vm = this
        vm.getSummary = function() {
            const url = 'http://localhost:3003/api/billingSummary'
            $http.get(url).success(function ({credit = 0, debt = 0}) {
                vm.credit = credit
                vm.debt = debt
                vm.total = credit - debt
            })
        }

        vm.getSummary()
    }
})();
```

Listagem 184 - Alterar arquivo *dashboard.html*

CursoFramesWeb/angular1/app/dashboard/dashboard.html (aprox. linha 3, 5, 7 e 9)

```
<content-header name="Dashboard" small="Versão 1.0"></content-header>

<section class="content" ng-controller="DashboardCtrl as dashboard">
  <value-box grid="4" color-class="bg-green" icon-class="fa fa-bank"
    value="R$ {{ dashboard.credit | number: 2 }}" text="Total de Créditos"
  ></value-box>
  <value-box grid="4" color-class="bg-red" icon-class="fa fa-credit-card"
    value="R$ {{ dashboard.debt | number: 2 }}" text="Total de Débitos"></value-
box>
  <value-box grid="4" color-class="bg-blue" icon-class="fa fa-bank"
    value="R$ {{ dashboard.total | number: 2 }}" text="Valor Consolidado"
  ></value-box>
</section>
```

13. Frontend - Cadastro de Ciclo de Pagamentos

13.1. Estrutura de Abas do Cadastro

Listagem 185 - Alterar arquivo tabs.html

CursoFramesWeb/angular1/app/billingCycle/tabs.html (na linha 1)

```
<content-header name="Ciclo de Pagamentos" small="Cadastro"></content-header>
<section class="content">
  <div class="nav-tabs-custom">
    <ul class="nav nav-tabs">
      <li>
        <a href data-target="#tabList" data-toggle="tab">
          <i class="fa fa-bars"></i> Lista
        </a>
      </li>
      <li>
        <a href data-target="#tabCreate" data-toggle="tab">
          <i class="fa fa-plus"></i> Incluir
        </a>
      </li>
      <li>
        <a href data-target="#tabUpdate" data-toggle="tab">
          <i class="fa fa-pencil"></i> Alterar
        </a>
      </li>
      <li>
        <a href data-target="#tabDelete" data-toggle="tab">
          <i class="fa fa-trash-o"></i> Excluir
        </a>
      </li>
    </ul>
    <div class="tab-content">
      <div class="tab-pane" id="tabList" ng-include="'billingCycle/list.html'"
    ></div>
      <div class="tab-pane" id="tabCreate" ng-include=
"'billingCycle/form.html'"></div>
      <div class="tab-pane" id="tabUpdate" ng-include=
"'billingCycle/form.html'"></div>
      <div class="tab-pane" id="tabDelete" ng-include=
"'billingCycle/form.html'"></div>
    </div>
  </div>
</section>
```

13.2. Estrutura Inicial do Formulário

Listagem 186 - Criar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html

```
<div class="form">
  <div class="box-body">
    <div class="col-xs-12 col-sm-4">
      <div class="form-group">
        <label for="name">Nome</label>
        <input id="name" class="form-control" placeholder="Informe o nome" />
      </div>
    </div>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary">Incluir</button>
    <button class="btn btn-info">Alterar</button>
    <button class="btn btn-danger">Excluir</button>
    <button class="btn btn-default">Cancelar</button>
  </div>
</div>
```

13.3. Componente Campo de Formulário (Field)

Listagem 187 - Criar arquivo field.js

CursoFramesWeb/angular1/app/common/components/field.js

```
(function() {
  angular.module('primeiraApp').component('field', {
    bindings: {
      id: '@',
      label: '@',
      grid: '@',
      placeholder: '@',
      type: '@',
    },
    controller: [
      'gridSystem',
      function(gridSystem){
        this.gridClasses = gridSystem.toCssClasses(this.grid)
      }
    ],
    template: `
    <div class="{{ $ctrl.gridClasses }}">
      <div class="form-group">
        <label for="{{ $ctrl.id }}">{{ $ctrl.label }}</label>
        <input id="{{ $ctrl.id }}" class="form-control" placeholder="{{
$ctrl.placeholder }}"
          type="{{ $ctrl.type }}" />
      </div>
    </div>
    `
  })
})();
```

Listagem 188 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 3)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
  ></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês" type
="number"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano" type
="number"></field>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary">Incluir</button>
    <button class="btn btn-info">Alterar</button>
    <button class="btn btn-danger">Excluir</button>
    <button class="btn btn-default">Cancelar</button>
  </div>
</div>
```

13.4. Versão Inicial do Controller de Ciclo de Pagamentos

Listagem 189 - Criar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    BillingCycleController
  ])

  function BillingCycleController($http) {
    const vm = this

    vm.create = function(){
      const url = 'http://localhost:3003/api/billingCycles'
      $http.post(url, vm.billingCycle).success(function(response){
        vm.billingCycle = {}
        console.log('Sucesso!')
      })
    }
  }
})()
```

Listagem 190 - Alterar arquivo tabs.html

CursoFramesWeb/angular1/app/billingCycle/tabs.html (na linha 4)

```
<content-header name="Ciclo de Pagamentos" small="Cadastro"></content-header>
<section class="content">
  <div class="nav-tabs-custom" ng-controller="BillingCycleCtrl as bcCtrl">
    <ul class="nav nav-tabs">
      <li>
        <a href data-target="#tabList" data-toggle="tab">
          <i class="fa fa-bars"></i> Lista
        </a>
      </li>
      <li>
        <a href data-target="#tabCreate" data-toggle="tab">
          <i class="fa fa-plus"></i> Incluir
        </a>
      </li>
      <li>
        <a href data-target="#tabUpdate" data-toggle="tab">
          <i class="fa fa-pencil"></i> Alterar
        </a>
      </li>
      <li>
        <a href data-target="#tabDelete" data-toggle="tab">
          <i class="fa fa-trash-o"></i> Excluir
        </a>
      </li>
    </ul>
    <div class="tab-content">
      <div class="tab-pane" id="tabList" ng-include="'billingCycle/list.html'"
    ></div>
      <div class="tab-pane" id="tabCreate" ng-include=
        "'billingCycle/form.html'"></div>
      <div class="tab-pane" id="tabUpdate" ng-include=
        "'billingCycle/form.html'"></div>
      <div class="tab-pane" id="tabDelete" ng-include=
        "'billingCycle/form.html'"></div>
    </div>
  </div>
</section>
```

Listagem 191 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 4, 6, 8 e 11)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number"></field>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()">Incluir</button>
    <button class="btn btn-info">Alterar</button>
    <button class="btn btn-danger">Excluir</button>
    <button class="btn btn-default">Cancelar</button>
  </div>
</div>
```

Listagem 192 - Alterar arquivo field.js

CursoFramesWeb/angular1/app/common/components/field.js (aprox. linha 9 e 22)

```
(function() {  
  angular.module('primeiraApp').component('field', {  
    bindings: {  
      id: '@',  
      label: '@',  
      grid: '@',  
      placeholder: '@',  
      type: '@',  
      model: '=',  
    },  
    controller: [  
      'gridSystem',  
      function(gridSystem){  
        this.gridClasses = gridSystem.toCssClasses(this.grid)  
      }  
    ],  
    template: `  
    <div class="{{ $ctrl.gridClasses }}">  
      <div class="form-group">  
        <label for="{{ $ctrl.id }}">{{ $ctrl.label }}</label>  
        <input id="{{ $ctrl.id }}" class="form-control" placeholder="{{  
$ctrl.placeholder }}"  
          type="{{ $ctrl.type }}" ng-model="$ctrl.model"/>  
      </div>  
    </div>  
    `,  
  })  
})();
```

13.5. Atualizando o BillingCycleController para Angular 1.6

Listagem 193 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    BillingCycleController
  ])

  function BillingCycleController($http) {
    const vm = this

    vm.create = function(){
      const url = 'http://localhost:3003/api/billingCycles'
      $http.post(url, vm.billingCycle).then(function(response){
        vm.billingCycle = {}
        console.log('Sucesso!')
      })
    }
  }
})()
```

13.6. Factory de Exibição de Mensagens

Listagem 194 - Criar arquivo *msgsFactory.js*

CursoFramesWeb/angular1/app/common/factories/msgsFactory.js

```
(function () {  
  angular.module('primeiraApp').factory('msgs', [  
    'toastr',  
    MsgsFactory  
  ])  
  
  function MsgsFactory(toastr) {  
  
    function addMsg(msgs, title, method) {  
      if(msgs instanceof Array){  
        msgs.forEach(msg => toastr[method](msg, title))  
      } else {  
        toastr[method](msgs, title)  
      }  
    }  
  
    function addSuccess(msgs) {  
      addMsg(msgs, 'Sucesso', 'success')  
    }  
  
    function addError(msgs) {  
      addMsg(msgs, 'Erro', 'error')  
    }  
  
    return { addSuccess, addError }  
  }  
})();
```

Listagem 195 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 4, 8 e 15)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    BillingCycleController
  ])

  function BillingCycleController($http, msgs) {
    const vm = this

    vm.create = function(){
      const url = 'http://localhost:3003/api/billingCycles'
      $http.post(url, vm.billingCycle).success(function(response){
        vm.billingCycle = {}
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }
  }
})()
```

Listagem 196 - Alterar arquivo *template-custom.css*

CursoFramesWeb/angular1/app/template/template-custom.css (aprox. linha 3)

```
html { background-color: #ecf0f5; }
.main-footer{position: fixed; bottom: 0; width: 100%;}
#toast-container{top: 70px;}
```

13.7. Lista de Ciclo de Pagamentos (Parte 1)

Listagem 197 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 10, 12, 21 e 28)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    BillingCycleController
  ])

  function BillingCycleController($http, msgs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {}
        vm.billingCycles = response
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.refresh()
  }
})();
```

13.8. Lista de Ciclo de Pagamentos (Parte 2)

Listagem 198 - Criar arquivo list.html

CursoFramesWeb/angular1/app/billingCycle/list.html

```
<div class="box-body">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Mês</th>
        <th>Ano</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="billingCycle in bcCtrl.billingCycles">
        <td>{{ billingCycle.name }}</td>
        <td>{{ billingCycle.month }}</td>
        <td>{{ billingCycle.year }}</td>
        <td class="table-actions">
          <button class="btn btn-warning"><i class="fa fa-pencil"></i></button>
          <button class="btn btn-danger"><i class="fa fa-trash-o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Listagem 199 - Alterar arquivo template-custom.css

CursoFramesWeb/angular1/app/template/template-custom.css (aprox. linha 4)

```
html { background-color: #ecf0f5; }
.main-footer{position: fixed; bottom: 0; width: 100%;}
#toast-container{top: 70px;}
.table-actions{width: 140px;}
```

13.9. Factory para Controlar a Exibição das Abas

Listagem 200 - Criar arquivo tabsFactory.js

CursoFramesWeb/angular1/app/common/factories/tabsFactory.js

```
(function() {  
  angular.module('primeiraApp').factory('tabs', [ TabsFactory ])  
  
  function TabsFactory() {  
  
    function show(owner, {  
      tabList = false,  
      tabCreate = false,  
      tabUpdate = false,  
      tabDelete = false  
    }) {  
      owner.tabList = tabList  
      owner.tabCreate = tabCreate  
      owner.tabUpdate = tabUpdate  
      owner.tabDelete = tabDelete  
    }  
  
    return { show }  
  }  
})();
```

Listagem 201 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 5, 9 e 17)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
    BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {}
        vm.billingCycles = response
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.refresh()
  }
})();
```

Listagem 202 - Alterar arquivo tabs.html

CursoFramesWeb/angular1/app/billingCycle/tabs.html (na linha 6, 11, 16, 21, 29, 31, 33 e 35)

```
<content-header name="Ciclo de Pagamentos" small="Cadastro"></content-header>
<section class="content">
  <div class="nav-tabs-custom" ng-controller="BillingCycleCtrl as bcCtrl">
    <ul class="nav nav-tabs">
      <li ng-if="bcCtrl.tabList">
        <a href data-target="#tabList" data-toggle="tab">
          <i class="fa fa-bars"></i> Lista
        </a>
      </li>
      <li ng-if="bcCtrl.tabCreate">
        <a href data-target="#tabCreate" data-toggle="tab">
          <i class="fa fa-plus"></i> Incluir
        </a>
      </li>
      <li ng-if="bcCtrl.tabUpdate">
        <a href data-target="#tabUpdate" data-toggle="tab">
          <i class="fa fa-pencil"></i> Alterar
        </a>
      </li>
      <li ng-if="bcCtrl.tabDelete">
        <a href data-target="#tabDelete" data-toggle="tab">
          <i class="fa fa-trash-o"></i> Excluir
        </a>
      </li>
    </ul>
    <div class="tab-content">
      <div class="tab-pane" id="tabList" ng-include="'billingCycle/list.html'"
        ng-if="bcCtrl.tabList"></div>
      <div class="tab-pane" id="tabCreate" ng-include="'billingCycle/form.html'"
        ng-if="bcCtrl.tabCreate"></div>
      <div class="tab-pane" id="tabUpdate" ng-include="'billingCycle/form.html'"
        ng-if="bcCtrl.tabUpdate"></div>
      <div class="tab-pane" id="tabDelete" ng-include="'billingCycle/form.html'"
        ng-if="bcCtrl.tabDelete"></div>
    </div>
  </div>
</section>
```

Listagem 203 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 12, 13 e 14)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number"></field>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete">Excluir</button>
    <button class="btn btn-default">Cancelar</button>
  </div>
</div>
```

13.10. Exibição das Abas de Alterar e Excluir

Listagem 204 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 30 e 35)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {}
        vm.billingCycles = response
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabDelete: true})
    }

    vm.refresh()
  }
})();
```

Listagem 205 - Alterar arquivo list.html

CursoFramesWeb/angular1/app/billingCycle/list.html (aprox. linha 17 e 19)

```
<div class="box-body">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Mês</th>
        <th>Ano</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="billingCycle in bcCtrl.billingCycles">
        <td>{{ billingCycle.name }}</td>
        <td>{{ billingCycle.month }}</td>
        <td>{{ billingCycle.year }}</td>
        <td class="table-actions">
          <button class="btn btn-warning" ng-click
            ="bcCtrl.showTabUpdate(billingCycle)">
            <i class="fa fa-pencil"></i></button>
          <button class="btn btn-danger" ng-click
            ="bcCtrl.showTabDelete(billingCycle)">
            <i class="fa fa-trash-o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Listagem 206 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 15)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number"></field>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete">Excluir</button>
    <button class="btn btn-default" ng-click="bcCtrl.refresh()">Cancelar</button>
  </div>
</div>
```


Listagem 207 - Alterar arquivo tabs.html

CursoFramesWeb/angular1/app/billingCycle/tabs.html (aprox. linha 6, 16, 21, 29, 33 e 35)

```
<content-header name="Ciclo de Pagamentos" small="Cadastro"></content-header>
<section class="content">
  <div class="nav-tabs-custom" ng-controller="BillingCycleCtrl as bcCtrl">
    <ul class="nav nav-tabs">
      <li ng-if="bcCtrl.tabList" ng-class="{active: bcCtrl.tabList}">
        <a href data-target="#tabList" data-toggle="tab">
          <i class="fa fa-bars"></i> Lista
        </a>
      </li>
      <li ng-if="bcCtrl.tabCreate">
        <a href data-target="#tabCreate" data-toggle="tab">
          <i class="fa fa-plus"></i> Incluir
        </a>
      </li>
      <li ng-if="bcCtrl.tabUpdate" ng-class="{active: bcCtrl.tabUpdate}">
        <a href data-target="#tabUpdate" data-toggle="tab">
          <i class="fa fa-pencil"></i> Alterar
        </a>
      </li>
      <li ng-if="bcCtrl.tabDelete" ng-class="{active: bcCtrl.tabDelete}">
        <a href data-target="#tabDelete" data-toggle="tab">
          <i class="fa fa-trash-o"></i> Excluir
        </a>
      </li>
    </ul>
    <div class="tab-content">
      <div class="tab-pane" id="tabList" ng-include="'billingCycle/list.html'"
        ng-class="{active: bcCtrl.tabList}" ng-if="bcCtrl.tabList"></div>
      <div class="tab-pane" id="tabCreate" ng-include="'billingCycle/form.html'"
        ng-if="bcCtrl.tabCreate"></div>
      <div class="tab-pane" id="tabUpdate" ng-include="'billingCycle/form.html'"
        ng-class="{active: bcCtrl.tabUpdate}" ng-if="bcCtrl.tabUpdate"></div>
      <div class="tab-pane" id="tabDelete" ng-include="'billingCycle/form.html'"
        ng-class="{active: bcCtrl.tabDelete}" ng-if="bcCtrl.tabDelete"></div>
    </div>
  </div>
</section>
```

13.11. Exclusão do Ciclo de Pagamentos

Listagem 208 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 40)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {}
        vm.billingCycles = response
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabDelete: true})
    }

    vm.delete = function(){
      const deleteUrl = `${url}/${vm.billingCycle._id}`
      $http.delete(deleteUrl, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!')
      }).error(function(data){
        msgs.addError(data.errors)
      })
    }
  }
}
```

```

    })
  }

  vm.refresh()
}
})();

```

Listagem 209 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 4, 6, 8 e 15)

```

<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name" readonly="bcCtrl.tabDelete"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number" readonly
    = "bcCtrl.tabDelete"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number" readonly
    = "bcCtrl.tabDelete"></field>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete"
      ng-click="bcCtrl.delete()">Excluir</button>
    <button class="btn btn-default" ng-click="bcCtrl.refresh()">Cancelar</button>
  </div>
</div>

```

Listagem 210 - Alterar arquivo field.js

CursoFramesWeb/angular1/app/common/components/field.js (aprox. linha 9 e 23)

```
(function() {  
  angular.module('primeiraApp').component('field', {  
    bindings: {  
      id: '@',  
      label: '@',  
      grid: '@',  
      placeholder: '@',  
      type: '@',  
      model: '=',  
      readonly: '<',  
    },  
    controller: [  
      'gridSystem',  
      function(gridSystem){  
        this.gridClasses = gridSystem.toCssClasses(this.grid)  
      }  
    ],  
    template: `  
    <div class="{{ $ctrl.gridClasses }}">  
      <div class="form-group">  
        <label for="{{ $ctrl.id }}">{{ $ctrl.label }}</label>  
        <input id="{{ $ctrl.id }}" class="form-control" placeholder="{{  
$ctrl.placeholder }}"  
          type="{{ $ctrl.type }}" ng-model="$ctrl.model" ng-  
readonly="$ctrl.readonly"/>  
      </div>  
    </div>  
    `,  
  })  
})();
```

13.12. Alteração do Ciclo de Pagamentos

Listagem 211 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 40)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {}
        vm.billingCycles = response
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabDelete: true})
    }

    vm.update = function(){
      const updateUrl = `${url}/${vm.billingCycle._id}`
      $http.put(updateUrl, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!')
      }).error(function(data){
        msgs.addError(data.errors)
      })
    }
  }
}
```

```

    })
  }

  vm.delete = function(){
    const deleteUrl = `${url}/${vm.billingCycle._id}`
    $http.delete(deleteUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.refresh()
}
}())

```

Listagem 212 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 14)

```

<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name" readonly="bcCtrl.tabDelete"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number" readonly
    = "bcCtrl.tabDelete"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number" readonly
    = "bcCtrl.tabDelete"></field>
  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate"
      ng-click="bcCtrl.update()">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete"
      ng-click="bcCtrl.delete()">Excluir</button>
    <button class="btn btn-default" ng-click="bcCtrl.refresh()">Cancelar</button>
  </div>
</div>

```

13.13. Lista de Créditos

Listagem 213 - Criar arquivo creditList.html

CursoFramesWeb/angular1/app/billingCycle/creditList.html

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="credit in bcCtrl.billingCycle.credits">
        <td>
          <input ng-model="credit.name" class="form-control" placeholder
            ="Informe o Nome" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="credit.value" class="form-control" placeholder
            ="Informe o Valor" ng-readonly="bcCtrl.tabDelete" type="number" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-class="{disabled:
            bcCtrl.tabDelete}">
            <i class="fa fa-plus"></i></button>
          <button class="btn btn-warning" ng-class="{disabled:
            bcCtrl.tabDelete}">
            <i class="fa fa-clone"></i></button>
          <button class="btn btn-danger" ng-class="{disabled:
            bcCtrl.tabDelete}">
            <i class="fa fa-trash-o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Listagem 214 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 10)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name" readonly="bcCtrl.tabDelete"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number" readonly
    = "bcCtrl.tabDelete"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number" readonly
    = "bcCtrl.tabDelete"></field>

    <div class="col-xs-12 col-sm-6">
      <fieldset>
        <legend>Créditos</legend>
        <div ng-include="'billingCycle/creditList.html'"></div>
      </fieldset>
    </div>

  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate"
      ng-click="bcCtrl.update()">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete"
      ng-click="bcCtrl.delete()">Excluir</button>
    <button class="btn btn-default" ng-click="bcCtrl.refresh()">Cancelar</button>
  </div>
</div>
```


Listagem 215 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 15)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {credits: [{}], debts:[{}]}
        vm.billingCycles = response
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabDelete: true})
    }

    vm.update = function(){
      const updateUrl = `${url}/${vm.billingCycle._id}`
      $http.put(updateUrl, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!')
      }).error(function(data){
        msgs.addError(data.errors)
      })
    }
  }
}
```

```

    })
  }

  vm.delete = function(){
    const deleteUrl = `${url}/${vm.billingCycle._id}`
    $http.delete(deleteUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.refresh()
}
}())

```

13.14. Lista de Débitos

Listagem 216 - Criar arquivo debtList.html

CursoFramesWeb/angular1/app/billingCycle/debtList.html

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Status</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="debt in bcCtrl.billingCycle.debts">
        <td>
          <input ng-model="debt.name" class="form-control" placeholder="Informe
o Nome" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="debt.value" class="form-control" placeholder
="Informe o Valor" type="number" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="debt.status" class="form-control" placeholder
="Informe o Status" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-class="{disabled:
bcCtrl.tabDelete}">
            <i class="fa fa-plus"></i></button>
          <button class="btn btn-warning" ng-class="{disabled:
bcCtrl.tabDelete}">
            <i class="fa fa-clone"></i></button>
          <button class="btn btn-danger" ng-class="{disabled:
bcCtrl.tabDelete}">
            <i class="fa fa-trash-o"></i></button>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
```

Listagem 217 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 16)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name" readonly="bcCtrl.tabDelete"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number" readonly
    = "bcCtrl.tabDelete"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number" readonly
    = "bcCtrl.tabDelete"></field>

    <div class="col-xs-12 col-sm-6">
      <fieldset>
        <legend>Créditos</legend>
        <div ng-include="'billingCycle/creditList.html'"></div>
      </fieldset>
    </div>
    <div class="col-xs-12 col-sm-6">
      <fieldset>
        <legend>Débitos</legend>
        <div ng-include="'billingCycle/debtList.html'"></div>
      </fieldset>
    </div>

  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate"
      ng-click="bcCtrl.update()">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete"
      ng-click="bcCtrl.delete()">Excluir</button>
    <button class="btn btn-default" ng-click="bcCtrl.refresh()">Cancelar</button>
  </div>
</div>
```

13.15. Créditos e Débitos: Adicionar, Clonar e Excluir

Listagem 218 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 60)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {credits: [{}], debts:[{}]}
        vm.billingCycles = response
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      tabs.show(vm, {tabDelete: true})
    }

    vm.update = function(){
      const updateUrl = `${url}/${vm.billingCycle._id}`
      $http.put(updateUrl, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!')
      }).error(function(data){
        msgs.addError(data.errors)
      })
    }
  }
}
```

```

    })
  }

  vm.delete = function(){
    const deleteUrl = `${url}/${vm.billingCycle._id}`
    $http.delete(deleteUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.addCredit = function (index) {
    vm.billingCycle.credits.splice(index + 1, 0, {} )
  }

  vm.cloneCredit = function (index, {name, value}) {
    vm.billingCycle.credits.splice(index + 1, 0, {name, value} )
  }

  vm.deleteCredit = function (index) {
    if(vm.billingCycle.credits.length > 1){
      vm.billingCycle.credits.splice(index, 1 )
    }
  }

  vm.addDebt = function (index) {
    vm.billingCycle.debts.splice(index + 1, 0, {} )
  }

  vm.cloneDebt = function (index, {name, value, status}) {
    vm.billingCycle.debts.splice(index + 1, 0, {name, value, status} )
  }

  vm.deleteDebt = function (index) {
    if(vm.billingCycle.debts.length > 1){
      vm.billingCycle.debts.splice(index, 1 )
    }
  }

  vm.refresh()
}
})()

```

Listagem 219 - Alterar arquivo creditList.html

CursoFramesWeb/angular1/app/billingCycle/creditList.html (aprox. linha 22, 24 e 26)

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="credit in bcCtrl.billingCycle.credits">
        <td>
          <input ng-model="credit.name" class="form-control" placeholder
            ="Informe o Nome" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="credit.value" class="form-control" placeholder
            ="Informe o Valor" ng-readonly="bcCtrl.tabDelete" type="number" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-class="{disabled:
            bcCtrl.tabDelete}"
            ng-click="bcCtrl.addCredit($index)"><i class="fa fa-plus"></i>
          </button>
          <button class="btn btn-warning" ng-class="{disabled:
            bcCtrl.tabDelete}"
            ng-click="bcCtrl.cloneCredit($index, credit)"><i class="fa fa-
            clone"></i></button>
          <button class="btn btn-danger" ng-class="{disabled:
            bcCtrl.tabDelete}"
            ng-click="bcCtrl.deleteCredit($index)"><i class="fa fa-trash-
            o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Listagem 220 - Alterar arquivo *debtList.html*

CursoFramesWeb/angular1/app/billingCycle/debtList.html (aprox. linha 27, 29 e 31)

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Status</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="debt in bcCtrl.billingCycle.debts">
        <td>
          <input ng-model="debt.name" class="form-control" placeholder="Informe
o Nome" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="debt.value" class="form-control" placeholder
="Informe o Valor" type="number" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="debt.status" class="form-control" placeholder
="Informe o Status" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-class="{disabled:
bcCtrl.tabDelete}"
            ng-click="bcCtrl.addDebt($index)"><i class="fa fa-plus"></i>
        </button>
          <button class="btn btn-warning" ng-class="{disabled:
bcCtrl.tabDelete}"
            ng-click="bcCtrl.cloneDebt($index, debt)"><i class="fa fa-
clone"></i></button>
          <button class="btn btn-danger" ng-class="{disabled:
bcCtrl.tabDelete}"
            ng-click="bcCtrl.deleteDebt($index)"><i class="fa fa-trash-o"></
i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```


13.16. Sumário dos Créditos e Débitos (Parte 1)

Listagem 221 - Criar arquivo *summary.html*

CursoFramesWeb/angular1/app/billingCycle/summary.html

```
<div class="row">
  <value-box grid="12 4" color-class="bg-green" icon-class="fa fa-bank"
    value="R$ {{ bcCtrl.credit | number: 2 }}" text="Total de Créditos"></value-
box>
  <value-box grid="12 4" color-class="bg-red" icon-class="fa fa-credit-card"
    value="R$ {{ bcCtrl.debt | number: 2 }}" text="Total de Débitos"></value-box>
  <value-box grid="12 4" color-class="bg-blue" icon-class="fa fa-money"
    value="R$ {{ bcCtrl.total | number: 2 }}" text="Valor Consolidado"></value-
box>
</div>
```

Listagem 222 - Alterar arquivo form.html

CursoFramesWeb/angular1/app/billingCycle/form.html (aprox. linha 10)

```
<div class="form">
  <div class="box-body">
    <field id="name" label="Nome" grid="12 6 4" placeholder="Informe o nome"
      model="bcCtrl.billingCycle.name" readonly="bcCtrl.tabDelete"></field>
    <field id="month" label="Mês" grid="12 6 4" placeholder="Informe o mês"
      model="bcCtrl.billingCycle.month" type="number" readonly
    = "bcCtrl.tabDelete"></field>
    <field id="year" label="Ano" grid="12 6 4" placeholder="Informe o ano"
      model="bcCtrl.billingCycle.year" type="number" readonly
    = "bcCtrl.tabDelete"></field>

    <div class="col-xs-12">
      <fieldset>
        <legend>Resumo</legend>
        <div ng-include="'billingCycle/summary.html'"></div>
      </fieldset>
    </div>

    <div class="col-xs-12 col-sm-6">
      <fieldset>
        <legend>Créditos</legend>
        <div ng-include="'billingCycle/creditList.html'"></div>
      </fieldset>
    </div>

    <div class="col-xs-12 col-sm-6">
      <fieldset>
        <legend>Débitos</legend>
        <div ng-include="'billingCycle/debtList.html'"></div>
      </fieldset>
    </div>

  </div>
  <div class="box-footer">
    <button class="btn btn-primary" ng-click="bcCtrl.create()"
      ng-if="bcCtrl.tabCreate">Incluir</button>
    <button class="btn btn-info" ng-if="bcCtrl.tabUpdate"
      ng-click="bcCtrl.update()">Alterar</button>
    <button class="btn btn-danger" ng-if="bcCtrl.tabDelete"
      ng-click="bcCtrl.delete()">Excluir</button>
    <button class="btn btn-default" ng-click="bcCtrl.refresh()">Cancelar</button>
  </div>
</div>
```

13.17. Sumário dos Créditos e Débitos (Parte 2)

Listagem 223 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 17, 33, 39, 69, 75, 85, 91 e 95)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    'msgs',
    'tabs'
    BillingCycleController
  ])

  function BillingCycleController($http, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      $http.get(url).success(function(response){
        vm.billingCycle = {credits: [{}], debts:[{}]}
        vm.billingCycles = response
        vm.calculateValues()
        tabs.show(vm, {tabList: true, tabCreate: true})
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
      tabs.show(vm, {tabDelete: true})
    }

    vm.update = function(){
      const updateUrl = `${url}/${vm.billingCycle._id}`
      $http.put(updateUrl, vm.billingCycle).success(function(response){
```

```

        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
        msgs.addError(data.errors)
    })
}

vm.delete = function(){
    const deleteUrl = `${url}/${vm.billingCycle._id}`
    $http.delete(deleteUrl, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
        msgs.addError(data.errors)
    })
}

vm.addCredit = function (index) {
    vm.billingCycle.credits.splice(index + 1, 0, {} )
}

vm.cloneCredit = function (index, {name, value}) {
    vm.billingCycle.credits.splice(index + 1, 0, {name, value} )
    vm.calculateValues()
}

vm.deleteCredit = function (index) {
    if(vm.billingCycle.credits.length > 1){
        vm.billingCycle.credits.splice(index, 1 )
        vm.calculateValues()
    }
}

vm.addDebt = function (index) {
    vm.billingCycle.debts.splice(index + 1, 0, {} )
}

vm.cloneDebt = function (index, {name, value, status}) {
    vm.billingCycle.debts.splice(index + 1, 0, {name, value, status})
    vm.calculateValues()
}

vm.deleteDebt = function (index) {
    if(vm.billingCycle.debts.length > 1){
        vm.billingCycle.debts.splice(index, 1 )
        vm.calculateValues()
    }
}

vm.calculateValues = function () {
    vm.credit = 0

```

```
vm.debt = 0

if (vm.billingCycle) {
  vm.billingCycle.credits.forEach(function ({value}) {
    vm.credit += !value || isNaN(value) ? 0 : parseFloat(value)
  })

  vm.billingCycle.debts.forEach(function ({value}) {
    vm.debt += !value || isNaN(value) ? 0 : parseFloat(value)
  })
}
vm.total = vm.credit - vm.debt
}

vm.refresh()
})()
```

Listagem 224 - Alterar arquivo creditList.html

CursoFramesWeb/angular1/app/billingCycle/creditList.html (aprox. linha 18)

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="credit in bcCtrl.billingCycle.credits">
        <td>
          <input ng-model="credit.name" class="form-control" placeholder
            ="Informe o Nome" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="credit.value" class="form-control" placeholder
            ="Informe o Valor" ng-change="bcCtrl.calculateValues()" ng-readonly
            ="bcCtrl.tabDelete" type="number" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-class="{disabled:
            bcCtrl.tabDelete}"
            ng-click="bcCtrl.addCredit($index)"><i class="fa fa-plus"></i>
          </button>
          <button class="btn btn-warning" ng-class="{disabled:
            bcCtrl.tabDelete}"
            ng-click="bcCtrl.cloneCredit($index, credit)"><i class="fa fa-
            clone"></i></button>
          <button class="btn btn-danger" ng-class="{disabled:
            bcCtrl.tabDelete}"
            ng-click="bcCtrl.deleteCredit($index)"><i class="fa fa-trash-
            o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Listagem 225 - Alterar arquivo *debtList.html*


```

<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Status</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="debt in bcCtrl.billingCycle.debts">
        <td>
          <input ng-model="debt.name" class="form-control" placeholder="Informe
o Nome"
              ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="debt.value" class="form-control" placeholder
="Informe o Valor"
              ng-change="bcCtrl.calculateValues()" ng-readonly
="bcCtrl.tabDelete" type="number" />
        </td>
        <td>
          <input ng-model="debt.status" class="form-control" placeholder
="Informe o Status"
              ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-class="{disabled:
bcCtrl.tabDelete}"
              ng-click="bcCtrl.addDebt($index)"><i class="fa fa-plus"></i>
          </button>
          <button class="btn btn-warning" ng-class="{disabled:
bcCtrl.tabDelete}"
              ng-click="bcCtrl.cloneDebt($index, debt)"><i class="fa fa-
clone"></i></button>
          <button class="btn btn-danger" ng-class="{disabled:
bcCtrl.tabDelete}"
              ng-click="bcCtrl.deleteDebt($index)"><i class="fa fa-trash-o"></
i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>

```

13.18. Desabilitar Ações Durante a Exclusão

Listagem 226 - Alterar arquivo *creditList.html*

CursoFramesWeb/angular1/app/billingCycle/creditList.html (aprox. linha 21, 23 e 25)

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="credit in bcCtrl.billingCycle.credits">
        <td>
          <input ng-model="credit.name" class="form-control" placeholder
            ="Informe o Nome" ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="credit.value" class="form-control" placeholder
            ="Informe o Valor" ng-change="bcCtrl.calculateValues()" ng-readonly
            ="bcCtrl.tabDelete" type="number" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-disabled="bcCtrl.tabDelete"
            ng-click="bcCtrl.addCredit($index)"><i class="fa fa-plus"></i>
        </button>
          <button class="btn btn-warning" ng-disabled="bcCtrl.tabDelete"
            ng-click="bcCtrl.cloneCredit($index, credit)"><i class="fa fa-
            clone"></i></button>
          <button class="btn btn-danger" ng-disabled="bcCtrl.tabDelete"
            ng-click="bcCtrl.deleteCredit($index)"><i class="fa fa-trash-
            o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

Listagem 227 - Alterar arquivo *debtList.html*

CursoFramesWeb/angular1/app/billingCycle/debtList.html (aprox. linha 26, 28 e 30)

```
<div class="col-xs-12">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Valor</th>
        <th>Status</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="debt in bcCtrl.billingCycle.debts">
        <td>
          <input ng-model="debt.name" class="form-control" placeholder="Informe
o Nome"
              ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td>
          <input ng-model="debt.value" class="form-control" placeholder
="Informe o Valor"
              ng-change="bcCtrl.calculateValues()" ng-readonly
="bcCtrl.tabDelete" type="number" />
        </td>
        <td>
          <input ng-model="debt.status" class="form-control" placeholder
="Informe o Status"
              ng-readonly="bcCtrl.tabDelete" />
        </td>
        <td class="table-actions">
          <button class="btn btn-success" ng-disabled="bcCtrl.tabDelete"
              ng-click="bcCtrl.addDebt($index)"><i class="fa fa-plus"></i>
          </button>
          <button class="btn btn-warning" ng-disabled="bcCtrl.tabDelete"
              ng-click="bcCtrl.cloneDebt($index, debt)"><i class="fa fa-
clone"></i></button>
          <button class="btn btn-danger" ng-disabled="bcCtrl.tabDelete"
              ng-click="bcCtrl.deleteDebt($index)"><i class="fa fa-trash-o"></
i></button>
        </td>
      </tr>
    </tbody>
  </table>
</div>
```

13.19. Implementando a Paginação

Listagem 228 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 4, 10, 15 e 22)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    '$location',
    'msgs',
    'tabs'
    BillingCycleController
  ])

  function BillingCycleController($http, $location, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      const page = parseInt($location.search().page) || 1
      $http.get(`${url}?skip=${(page - 1) * 10}&limit=10}`).success(function
(response) {
        vm.billingCycle = {credits: [{}], debts:[{}]}
        vm.billingCycles = response
        vm.calculateValues()
        tabs.show(vm, {tabList: true, tabCreate: true})

        $http.get(`${url}/count`).success(function (response) {
          vm.pages = Math.ceil(response.value / 10)
        })
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
    }
  }
})
```

```

    tabs.show(vm, {tabDelete: true})
  }

  vm.update = function(){
    const updateUrl = `${url}/${vm.billingCycle._id}`
    $http.put(updateUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.delete = function(){
    const deleteUrl = `${url}/${vm.billingCycle._id}`
    $http.delete(deleteUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.addCredit = function (index) {
    vm.billingCycle.credits.splice(index + 1, 0, {} )
  }

  vm.cloneCredit = function (index, {name, value}) {
    vm.billingCycle.credits.splice(index + 1, 0, {name, value} )
    vm.calculateValues()
  }

  vm.deleteCredit = function (index) {
    if(vm.billingCycle.credits.length > 1){
      vm.billingCycle.credits.splice(index, 1 )
      vm.calculateValues()
    }
  }

  vm.addDebt = function (index) {
    vm.billingCycle.debts.splice(index + 1, 0, {} )
  }

  vm.cloneDebt = function (index, {name, value, status}) {
    vm.billingCycle.debts.splice(index + 1, 0, {name, value, status})
    vm.calculateValues()
  }

  vm.deleteDebt = function (index) {
    if(vm.billingCycle.debts.length > 1){
      vm.billingCycle.debts.splice(index, 1 )
    }
  }

```

```

        vm.calculateValues()
    }
}

vm.calculateValues = function () {
    vm.credit = 0
    vm.debt = 0

    if (vm.billingCycle) {
        vm.billingCycle.credits.forEach(function ({value}) {
            vm.credit += !value || isNaN(value) ? 0 : parseFloat(value)
        })

        vm.billingCycle.debts.forEach(function ({value}) {
            vm.debt += !value || isNaN(value) ? 0 : parseFloat(value)
        })
    }
    vm.total = vm.credit - vm.debt
}

vm.refresh()
}
})();

```

Listagem 229 - Alterar arquivo routes.js

CursoFramesWeb/angular1/app/config/routes.js (aprox. linha 9)

```

angular.module('primeiraApp').config([
    '$stateProvider',
    '$urlRouterProvider',
    function ($stateProvider, $urlRouterProvider) {
        $stateProvider.state('dashboard', {
            url: "/dashboard",
            templateUrl: "dashboard/dashboard.html"
        }).state('billingCycle', {
            url: "/billingCycles?page",
            templateUrl: "billingCycle/tabs.html"
        })
        $urlRouterProvider.otherwise('/dashboard')
    }
])

```

Listagem 230 - Alterar arquivo server.js_

CursoFramesWeb/backend/config/server.js (aprox. linha 7 e 12)

```
const port = 3003

const bodyParser = require('body-parser')
const express = require('express')
const server = express()
const allowCors = require('./cors')
const queryParser = require('express-query-int');

server.use(bodyParser.urlencoded({ extended: true }))
server.use(bodyParser.json())
server.use(allowCors)
server.use(queryParser())

server.listen(port, function () {
  console.log(`BACKEND is running on port ${port}.`);
})

module.exports = server
```

13.20. Componente de Paginação

Listagem 231 - Criar arquivo paginator.js

CursoFramesWeb/angular1/app/common/components/paginator.js

```
(function () {
  angular.module('primeiraApp').component('paginator', {
    bindings: {
      url: '@',
      pages: '@'
    },
    controller: [
      '$location',
      function ($location) {
        this.$onInit = function () {
          const pages = parseInt(this.pages) || 1
          this.pagesArray = Array(pages).fill(0).map((e, i) => i + 1)
          this.current = parseInt($location.search().page) || 1
          this.needPagination = this.pages > 1
          this.hasPrev = this.current > 1
          this.hasNext = this.current < this.pages

          this.isCurrent = function (i) {
            return this.current == i
          }
        }
      }
    ],
    template: `
      <ul ng-if="$ctrl.needPagination" class="pagination pagination-sm no-margin pull-right">
        <li ng-if="$ctrl.hasPrev">
          <a href="{{ $ctrl.url }}"?page={{ $ctrl.current - 1 }}">Anterior</a>
        </li>

        <li ng-class="{active: $ctrl.isCurrent(index)}" ng-repeat="index in $ctrl.pagesArray">
          <a href="{{ $ctrl.url }}"?page={{ index }}">{{ index }}</a>
        </li>

        <li ng-if="$ctrl.hasNext">
          <a href="{{ $ctrl.url }}"?page={{ $ctrl.current + 1 }}">Próximo</a>
        </li>
      </ul>
    `
  })
})()
```

Listagem 232 - Alterar arquivo list.html

CursoFramesWeb/angular1/app/billingCycle/list.html (aprox. linha 25)

```
<div class="box-body">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Mês</th>
        <th>Ano</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="billingCycle in bcCtrl.billingCycles">
        <td>{{ billingCycle.name }}</td>
        <td>{{ billingCycle.month }}</td>
        <td>{{ billingCycle.year }}</td>
        <td class="table-actions">
          <button class="btn btn-warning" ng-click
            ="bcCtrl.showTabUpdate(billingCycle)"><i class="fa fa-pencil"></i></button>
          <button class="btn btn-danger" ng-click
            ="bcCtrl.showTabDelete(billingCycle)"><i class="fa fa-trash-o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
  <div class="box-footer clearfix">
    <paginator url="/#/billingCycles" pages="{{ bcCtrl.pages }}"></paginator>
  </div>
</div>
```

Listagem 233 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 23)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    '$location',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  )

  function BillingCycleController($http, $location, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      const page = parseInt($location.search().page) || 1
      $http.get(`${url}?skip=${(page - 1) * 10}&limit=10}`).success(function
(response) {
        vm.billingCycle = {credits: [{}], debts:[{}]}
        vm.billingCycles = response
        vm.calculateValues()

        $http.get(`${url}/count`).success(function (response) {
          vm.pages = Math.ceil(response.value / 10)
          tabs.show(vm, {tabList: true, tabCreate: true})
        })
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).success(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).error(function(data) {
        msgs.addError(data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
    }
  }
})
```

```

    tabs.show(vm, {tabDelete: true})
  }

  vm.update = function(){
    const updateUrl = `${url}/${vm.billingCycle._id}`
    $http.put(updateUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.delete = function(){
    const deleteUrl = `${url}/${vm.billingCycle._id}`
    $http.delete(deleteUrl, vm.billingCycle).success(function(response){
      vm.refresh()
      msgs.addSuccess('Operação realizada com sucesso!')
    }).error(function(data){
      msgs.addError(data.errors)
    })
  }

  vm.addCredit = function (index) {
    vm.billingCycle.credits.splice(index + 1, 0, {} )
  }

  vm.cloneCredit = function (index, {name, value}) {
    vm.billingCycle.credits.splice(index + 1, 0, {name, value} )
    vm.calculateValues()
  }

  vm.deleteCredit = function (index) {
    if(vm.billingCycle.credits.length > 1){
      vm.billingCycle.credits.splice(index, 1 )
      vm.calculateValues()
    }
  }

  vm.addDebt = function (index) {
    vm.billingCycle.debts.splice(index + 1, 0, {} )
  }

  vm.cloneDebt = function (index, {name, value, status}) {
    vm.billingCycle.debts.splice(index + 1, 0, {name, value, status})
    vm.calculateValues()
  }

  vm.deleteDebt = function (index) {
    if(vm.billingCycle.debts.length > 1){
      vm.billingCycle.debts.splice(index, 1 )
    }
  }

```

```

        vm.calculateValues()
    }
}

vm.calculateValues = function () {
    vm.credit = 0
    vm.debt = 0

    if (vm.billingCycle) {
        vm.billingCycle.credits.forEach(function ({value}) {
            vm.credit += !value || isNaN(value) ? 0 : parseFloat(value)
        })

        vm.billingCycle.debts.forEach(function ({value}) {
            vm.debt += !value || isNaN(value) ? 0 : parseFloat(value)
        })
    }
    vm.total = vm.credit - vm.debt
}

vm.refresh()
}
})();

```

14. Refactories e Correções

14.1. One-Time Binding e AngularJS Batarang



Foi instalado uma extensão do Chrome chamada **AngularJS Batarang** e para acessar basta ir em ferramentas do desenvolvedor na aba **AngularJS** e ativar.

Link para instalar o AngularJS Batarang: <https://chrome.google.com/webstore/detail/angularjs-batarang/ighdmehidhipcmcojjiloacoafjmpfk>

Listagem 234 - Alterar arquivo *list.html*

CursoFramesWeb/angular1/app/billingCycle/list.html (aprox. linha 13, 14 e 15)

```
<div class="box-body">
  <table class="table">
    <thead>
      <tr>
        <th>Nome</th>
        <th>Mês</th>
        <th>Ano</th>
        <th>Ações</th>
      </tr>
    </thead>
    <tbody>
      <tr ng-repeat="billingCycle in bcCtrl.billingCycles">
        <td>{{:: billingCycle.name }}</td>
        <td>{{:: billingCycle.month }}</td>
        <td>{{:: billingCycle.year }}</td>
        <td class="table-actions">
          <button class="btn btn-warning" ng-click
            ="bcCtrl.showTabUpdate(billingCycle)"><i class="fa fa-pencil"></i></button>
          <button class="btn btn-danger" ng-click
            ="bcCtrl.showTabDelete(billingCycle)"><i class="fa fa-trash-o"></i></button>
        </td>
      </tr>
    </tbody>
  </table>
  <div class="box-footer clearfix">
    <paginator url="/#!/billingCycles" pages="{{ bcCtrl.pages }}"></paginator>
  </div>
</div>
```

14.2. Atualizando o Projeto para Angular 1.6



Abra o terminal e dentro da pasta **angular1** execute o seguinte comando:

Listagem 235 - *Atualizar as dependências*

```
npm update
```

Listagem 236 - *Executar a aplicação*

```
npm run dev
```

Listagem 237 - Alterar arquivo *billingCycleController.js*

CursoFramesWeb/angular1/app/billingCycle/billingCycleController.js (aprox. linha 16, 18, 21, 29, 32, 33, 51, 54, 55, 61, 64 e 65)

```
(function(){
  angular.module('primeiraApp').controller('BillingCycleCtrl', [
    '$http',
    '$location',
    'msgs',
    'tabs'
  ],
  BillingCycleController
  )

  function BillingCycleController($http, $location, msgs, tabs) {
    const vm = this
    const url = 'http://localhost:3003/api/billingCycles'

    vm.refresh = function () {
      const page = parseInt($location.search().page) || 1
      $http.get(`${url}?skip=${(page - 1) * 10}&limit=10}`).then(function
(response) {
        vm.billingCycle = {credits: [{}], debts:[{}]}
        vm.billingCycles = response.data
        vm.calculateValues()

        $http.get(`${url}/count`).then(function (response) {
          vm.pages = Math.ceil(response.value / 10)
          tabs.show(vm, {tabList: true, tabCreate: true})
        })
      })
    }

    vm.create = function(){
      $http.post(url, vm.billingCycle).then(function(response){
        vm.refresh()
        msgs.addSuccess('Operação realizada com sucesso!!')
      }).catch(function(response) {
        msgs.addError(response.data.errors)
      })
    }

    vm.showTabUpdate = function (billingCycle) {
      vm.billingCycle = billingCycle
      vm.calculateValues()
      tabs.show(vm, {tabUpdate: true})
    }

    vm.showTabDelete = function (billingCycle) {
      vm.billingCycle = billingCycle
    }
  }
})
```



```

        vm.calculateValues()
        tabs.show(vm, {tabDelete: true})
    }

    vm.update = function(){
        const updateUrl = `${url}/${vm.billingCycle._id}`
        $http.put(updateUrl, vm.billingCycle).then(function(response){
            vm.refresh()
            msgs.addSuccess('Operação realizada com sucesso!')
        }).catch(function(response){
            msgs.addError(response.data.errors)
        })
    }

    vm.delete = function(){
        const deleteUrl = `${url}/${vm.billingCycle._id}`
        $http.delete(deleteUrl, vm.billingCycle).then(function(response){
            vm.refresh()
            msgs.addSuccess('Operação realizada com sucesso!')
        }).catch(function(response){
            msgs.addError(response.data.errors)
        })
    }

    vm.addCredit = function (index) {
        vm.billingCycle.credits.splice(index + 1, 0, {} )
    }

    vm.cloneCredit = function (index, {name, value}) {
        vm.billingCycle.credits.splice(index + 1, 0, {name, value} )
        vm.calculateValues()
    }

    vm.deleteCredit = function (index) {
        if(vm.billingCycle.credits.length > 1){
            vm.billingCycle.credits.splice(index, 1 )
            vm.calculateValues()
        }
    }

    vm.addDebt = function (index) {
        vm.billingCycle.debts.splice(index + 1, 0, {} )
    }

    vm.cloneDebt = function (index, {name, value, status}) {
        vm.billingCycle.debts.splice(index + 1, 0, {name, value, status})
        vm.calculateValues()
    }

    vm.deleteDebt = function (index) {
        if(vm.billingCycle.debts.length > 1){

```

```

        vm.billingCycle.debts.splice(index, 1 )
        vm.calculateValues()
    }
}

vm.calculateValues = function () {
    vm.credit = 0
    vm.debt = 0

    if (vm.billingCycle) {
        vm.billingCycle.credits.forEach(function ({value}) {
            vm.credit += !value || isNaN(value) ? 0 : parseFloat(value)
        })

        vm.billingCycle.debts.forEach(function ({value}) {
            vm.debt += !value || isNaN(value) ? 0 : parseFloat(value)
        })
    }
    vm.total = vm.credit - vm.debt
}

vm.refresh()
}
})();

```

Listagem 238 - Alterar arquivo field.js

CursoFramesWeb/angular1/app/common/components/field.js (aprox. linha 15)

```
(function() {  
  angular.module('primeiraApp').component('field', {  
    bindings: {  
      id: '@',  
      label: '@',  
      grid: '@',  
      placeholder: '@',  
      type: '@',  
      model: '=',  
      readonly: '<',  
    },  
    controller: [  
      'gridSystem',  
      function(gridSystem){  
        this.$onInit = () => this.gridClasses = gridSystem.toCssClasses(  
this.grid)  
      }  
    ],  
    template: `  
    <div class="{{ $ctrl.gridClasses }}">  
      <div class="form-group">  
        <label for="{{ $ctrl.id }}">{{ $ctrl.label }}</label>  
        <input id="{{ $ctrl.id }}" class="form-control" placeholder="{{  
$ctrl.placeholder }}"  
          type="{{ $ctrl.type }}" ng-model="$ctrl.model" ng-  
readonly="$ctrl.readonly"/>  
      </div>  
    </div>  
    `,  
  })  
})();
```

Listagem 239 - Alterar arquivo *dashboardController.js*

CursoFramesWeb/angular1/app/dashboard/dashboardController.js (aprox. linha 11 e 12)

```
(function(){
  angular.module('primeiraApp').controller('DashboardCtrl', [
    '$http',
    DashboardController
  ])

  function DashboardController($http) {
    const vm = this
    vm.getSummary = function () {
      const url = 'http://localhost:3003/api/billingSummary'
      $http.get(url).then(function (response) {
        const {credit = 0, debt = 0} = response.data
        vm.credit = credit
        vm.debt = debt
        vm.total = credit - debt
      })
    }

    vm.getSummary()
  }
})();
```

15. PrimeiraAPP: Autenticação

15.1. Backend: Novas Dependências



Iremos adicionar novas dependências ao projeto usando **versões específicas** para garantir compatibilidade. As mudanças importantes nas versões dos módulos adicionados serão tratadas em vídeos adicionais no final do curso.

Novas dependências são:

- bcrypt@1.0.2
- jsonwebtoken@7.3.0

15.1.1. Instalação

Abra o terminal e dentro da pasta do projeto backend execute o seguinte comando:

Listagem 1 - Instalando as dependências

```
npm i --save bcrypt@1.0.2 jsonwebtoken@7.3.0
```

15.2. Backend: Implementar Autenticação

15.2.1. Criar arquivo .env



O objetivo principal do **.env** é armazenar variáveis de configuração da sua aplicação *backend*, por enquanto será armazenando apenas a chave usada para gerar o token **JWT**.

Listagem 2 - Criar arquivo .env

backend/.env

```
module.exports = {  
  // Você pode alterar essa chave!  
  authSecret: 'skjdhf6$$%dojkhf^(sdkjhf'  
}
```



É muito importante que o arquivo **.env** não seja commitado no repositório, pois nele está a chave secreta para geração do token.

Listagem 3 - Adicionar .env ao .gitignore

backend/.gitignore

```
node_modules
*.log
.env
```

15.2.2. Criar arquivo *user.js*



Criar uma nova pasta ao projeto *backend* chamada *user* dentro de *api*.



O objetivo principal do *user.js* é fazer o **ODM** (Mapeamento Objeto-Documento), ou seja, mapear o objeto javascript *user* para o documento que será armazenado no MongoDB.

Listagem 4 - Criar arquivo *user.js*

backend/api/user/user.js

```
const restful = require('node-restful')
const mongoose = restful.mongoose

const userSchema = new mongoose.Schema({
  name: { type: String, required: true },
  email: { type: String, required: true },
  password: { type: String, min: 6, max: 12, required: true }
})

module.exports = restful.model('User', userSchema)
```

15.2.3. Criar arquivo *authService.js*

Listagem 5 - Estrutura básica de authService.js

backend/api/user/authService.js

```
const _ = require('lodash')
const jwt = require('jsonwebtoken')
const bcrypt = require('bcrypt')
const User = require('./user')
const env = require('../../.env')

const emailRegex = /\S+@\S+\.\S+/
const passwordRegex = /((?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*@#\$%)).{6,12})/

const sendErrorsFromDB = (res, dbErrors) => {
  const errors = []
  _.forIn(dbErrors.errors, error => errors.push(error.message))
  return res.status(400).json({errors})
}
```

Listagem 6 - Criar método login de authService.js

backend/api/user/authService.js

```
const login = (req, res, next) => {
  const email = req.body.email || ''
  const password = req.body.password || ''

  User.findOne({email}, (err, user) => {
    if(err) {
      return sendErrorsFromDB(res, err)
    } else if (user && bcrypt.compareSync(password, user.password)) {
      const token = jwt.sign(user, env.authSecret, {
        expiresIn: "1 day"
      })
      const { name, email } = user
      res.json({ name, email, token })
    } else {
      return res.status(400).send({errors: ['Usuário/Senha inválidos']})
    }
  })
}
```

Listagem 7 - Criar método `validateToken` de `authService.js`

`backend/api/user/authService.js`

```
const validateToken = (req, res, next) => {  
  const token = req.body.token || ''  
  jwt.verify(token, env.authSecret, function(err, decoded) {  
    return res.status(200).send({valid: !err})  
  })  
}
```


Listagem 8 - Criar método signup de authService.js

backend/api/user/authService.js

```
const signup = (req, res, next) => {
  const name = req.body.name || ''
  const email = req.body.email || ''
  const password = req.body.password || ''
  const confirmPassword = req.body.confirm_password || ''

  if(!email.match(emailRegex)) {
    return res.status(400).send({errors: ['O e-mail informado está inválido']})
  }

  if(!password.match(passwordRegex)) {
    return res.status(400).send({errors: [
      "Senha precisar ter: uma letra maiúscula, uma letra minúscula, um número, uma caractere especial(@#$%) e tamanho entre 6-12."
    ]})
  }

  const salt = bcrypt.genSaltSync()
  const passwordHash = bcrypt.hashSync(password, salt)
  if(!bcrypt.compareSync(confirmPassword, passwordHash)) {
    return res.status(400).send({errors: ['Senhas não conferem.']})
  }

  User.findOne({email}, (err, user) => {
    if(err) {
      return sendErrorsFromDB(res, err)
    } else if (user) {
      return res.status(400).send({errors: ['Usuário já cadastrado.']})
    } else {
      const newUser = new User({ name, email, password: passwordHash })
      newUser.save(err => {
        if(err) {
          return sendErrorsFromDB(res, err)
        } else {
          login(req, res, next)
        }
      })
    }
  })
}
```

Listagem 9 - Exportar os métodos de authService.js

backend/api/user/authService.js

```
module.exports = { login, signup, validateToken }
```

15.2.4. Criar middleware auth.js



Esse [\[middleware\]](#) será o responsável por validar o token [JWT](#) para as rotas protegidas e garantir que a API esteja protegida.

Listagem 10 - Criar middleware auth.js

backend/config/auth.js

```
const jwt = require('jsonwebtoken')
const env = require('.../.env')

module.exports = (req, res, next) => {

  // CORS preflight request
  if(req.method === 'OPTIONS') {
    next()
  } else {
    const token = req.body.token || req.query.token ||
req.headers['authorization']

    if(!token) {
      return res.status(403).send({errors: ['No token provided.']})
    }

    jwt.verify(token, env.authSecret, function(err, decoded) {
      if(err) {
        return res.status(403).send({
          errors: ['Failed to authenticate token.']
        })
      } else {
        // req.decoded = decoded
        next()
      }
    })
  }
}
```

Listagem 11 - Adicionar novo header em cors.js

backend/config/cors.js

```
module.exports = (req, res, next) => {  
  res.header('Access-Control-Allow-Origin', '*')  
  res.header('Access-Control-Allow-Methods',  
    'GET, POST, OPTIONS, PUT, PATCH, DELETE')  
  res.header('Access-Control-Allow-Headers',  
    'Origin, X-Requested-With, Content-Type, Accept, Authorization')  
  next()  
}
```

15.2.5. Alterar arquivo *routes.js*



Além de cadastrar as novas rotas *login*, *signup* e *validateToken*, essa alteração tem por objetivo separar as rotas públicas das rotas privadas (acesso com autenticação).

Listagem 12 - Alterar arquivo routes.js

my-money-app/backend/src/config/routes.js

```
const express = require('express')
const auth = require('./auth')

module.exports = function (server) {

  /*
   * Rotas abertas
   */
  const openApi = express.Router()
  server.use('/oapi', openApi)

  const AuthService = require('../api/user/AuthService')
  openApi.post('/login', AuthService.login)
  openApi.post('/signup', AuthService.signup)
  openApi.post('/validateToken', AuthService.validateToken)

  /*
   * Rotas protegidas por Token JWT
   */
  const protectedApi = express.Router()
  server.use('/api', protectedApi)

  protectedApi.use(auth)

  const billingCycleService = require(
    '../api/billingCycle/billingCycleService')
  billingCycleService.register(protectedApi, '/billingCycles')

  const billingSummaryService =
    require('../api/billingSummary/billingSummaryService')
  protectedApi.route('/billingSummary').get(billingSummaryService.getSummary)
}
```

15.3. Frontend: Implementar Autenticação

15.3.1. Widget do Usuário

Listagem 13 - Descomentar o item user-menu

angular1/app/template/header/navbar/navbar.html (aprox. linha 9)

```
<li class="dropdown user user-menu"  
  ng-include="'/template/header/navbar/user.html'"></li>
```

Listagem 14 - Remover Menu Body

angular1/app/template/header/navbar/user.html (aprox. linhas 14-27)

```
<!-- Menu Body -->  
<li class="user-body">  
  <div class="row">  
    <div class="col-xs-4 text-center">  
      <a href>Seguidores</a>  
    </div>  
    <div class="col-xs-4 text-center">  
      <a href>Vendas</a>  
    </div>  
    <div class="col-xs-4 text-center">  
      <a href>Amigos</a>  
    </div>  
  </div>  
</li>
```

Listagem 15 - Remover Botão Perfil

angular1/app/template/header/navbar/user.html (aprox. linhas 16-18)

```
<div class="pull-left">  
  <a href class="btn btn-default btn-flat">Perfil</a>  
</div>
```

Listagem 16 - Criar AuthController

angular1/app/auth/authController.js (aprox. linhas 16-18)

```
angular.module('primeiraApp').controller('AuthCtrl', [
  '$location',
  'msgs',
  AuthController
])

function AuthController($location, msgs) {
  const vm = this

  vm.getUser = () => ({ name: 'Usuário MOCK', email: 'mock@cod3r.com.br' })

  vm.logout = () => {
    console.log('Logout...')
  }
}
```

Listagem 17 - Adicionar ng-controller

angular1/app/template/header/navbar/navbar.html (aprox. linha 9)

```
ng-controller="AuthCtrl as auth"
```

Listagem 18 - Alterar user.html para acessar controller

angular1/app/template/header/navbar/user.html (aprox. linha 3)

```
<span class="hidden-xs">{{:: auth.getUser().name }}</span>
```

angular1/app/template/header/navbar/user.html (aprox. linhas 10-11)

```
{{:: auth.getUser().name }}
<small>{{:: auth.getUser().email }}</small>
```

angular1/app/template/header/navbar/user.html (aprox. linha 16)

```
<a href class="btn btn-default btn-flat" ng-click="auth.logout()">Sair</a>
```

15.3.2. Estrutura Básica do Login/Cadastro

Listagem 19 - Funções para suportar a tela Login/Cadastro

angular1/app/auth/authController.js

```
vm.loginMode = true

vm.changeMode = () => vm.loginMode = !vm.loginMode

vm.login = () => {
  console.log(`Login... ${vm.user.email}`)
}

vm.signup = () => {
  console.log(`Signup... ${vm.user.email}`)
}
```

Listagem 20 - Componente *authField*

angular1/app/common/components/authField.js

```
angular.module('primeiraApp').component('authField', {
  bindings: {
    id: '@',
    label: '@',
    type: '@',
    grid: '@',
    icon: '@',
    model: '=',
    placeholder: '@',
    hide: '<'
  },
  controller: function() {
    this.$onInit = () => {
      this.iconClasses = `glyphicon glyphicon-${this.icon} form-control-
feedback`
    }
  },
  template: `
    <div class="form-group has-feedback">
      <input ng-model="$ctrl.model" id="{{ $ctrl.id }}" class="form-
control"
        type="{{ $ctrl.type }}" placeholder="{{ $ctrl.placeholder }}"
        ng-hide="$ctrl.hide" />
      <span class="{{ $ctrl.iconClasses }}"></span>
    </div>
  `
})
```


Listagem 21 - Formulário Login/Cadastro

angular1/app/auth/form.html

```
<div class="login-box" ng-controller="AuthCtrl as auth">
  <div class="login-logo"><b> Primeira</b> APP</div>
  <div class="login-box-body">
    <p class="login-box-msg">Bem vindo!</p>
    <form>
      <auth-field id="authName" model="auth.user.name" type="input"
        placeholder="Informe o Nome" hide="auth.loginMode"></auth-field>
      <auth-field id="authEmail" model="auth.user.email" type="email"
        placeholder="Informe o E-mail"></auth-field>
      <auth-field id="authPass" model="auth.user.password" type="password"
        placeholder="Informe a Senha"></auth-field>
      <auth-field id="authConfirmPass" model="auth.user.confirm_password"
type="password"
        placeholder="Confirme a Senha" hide="auth.loginMode"></auth-
field>
      <div class="row">
        <div class="col-sm-4">
          <button class="btn btn-primary btn-block btn-flat"
ng-click="auth.login()" ng-show="auth.loginMode"
>Entrar</button>
          <button class="btn btn-primary btn-block btn-flat"
ng-click="auth.signup()" ng-hide=
"auth.loginMode">Registrar</button>
        </div>
      </div>
    </form>
    <br />
    <a href="javascript:;" ng-click="auth.changeMode()">
      {{ auth.loginMode ?
        'Novo usuário? Registrar aqui!' :
        'Já é cadastrado? Entrar aqui!' }}
    </a>
  </div>
</div>

<style>
  html, .wrapper { background-color: #fff!important; }
  .login-box-body { background-color: #eee; }
</style>
```

Listagem 22 - Página de Login/Cadastro

angular1/app/auth.html

```
<!DOCTYPE html>
<html ng-app="primeiraApp">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>{{ consts.appName }}</title>
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no" name="viewport">
  <link rel="stylesheet" href="/assets/css/deps.min.css">
  <link rel="stylesheet" href="/assets/css/app.min.css">
</head>
<body class="skin-blue fixed sidebar-mini">
  <div class="wrapper" ng-include="'/auth/form.html'"></div>
  <script src="/assets/js/deps.min.js"></script>
  <script src="/assets/js/app.min.js"></script>
</body>
</html>
```

15.3.3. O coração do Login/Cadastro (AuthFactory)

Listagem 23 - Novas Constantes

angular1/app/auth/authController.js

```
oapiUrl: 'http://localhost:3003/oapi',
userKey: '_primeira_app_user'
```

Listagem 24 - Estrutura inicial

angular1/app/common/factories/authFactory.js

```
(function() {  
  angular.module('primeiraApp').factory('auth', [  
    '$http',  
    'consts',  
    AuthFactory  
  ])  
  
  function AuthFactory($http, consts) {  
  
    return {}  
  }  
})();
```

Listagem 25 - Funções de Login/Cadastro

angular1/app/common/factories/authFactory.js

```
function signup(user, callback) {  
  submit('signup', user, callback)  
}  
  
function login(user, callback) {  
  submit('login', user, callback)  
}  
  
function submit(url, user, callback) {  
  $http.post(`${consts.oapiUrl}/${url}`, user)  
    .then(resp => {  
      localStorage.setItem(consts.userKey, JSON.stringify(resp.data))  
      $http.defaults.headers.common.Authorization = resp.data.token  
      if (callback) callback(null, resp.data)  
    }).catch(function (resp) {  
      if (callback) callback(resp.data.errors, null)  
    })  
}
```

angular1/app/common/factories/authFactory.js (exportar)

signup, login

Listagem 26 - Adicionando AuthFactory no AuthController

angular1/app/auth/authController.js (Lista de Dependências)

```
'auth'
```

angular1/app/auth/authController.js (Parâmetro da Função Controller)

```
auth
```

Listagem 27 - Alterando AuthController (login)

angular1/app/auth/authController.js

```
auth.login(vm.user, err => err ? msgs.addError(err) : msgs.addSuccess(
  'Sucesso!'))
```

Listagem 28 - Alterando AuthController (signup)

angular1/app/auth/authController.js

```
auth.signup(vm.user, err => err ? msgs.addError(err) : msgs.addSuccess(
  'Sucesso!'))
```

Listagem 29 - Função de Logout

angular1/app/common/factories/authFactory.js

```
function logout(callback) {
  localStorage.removeItem(consts.userKey)
  $http.defaults.headers.common.Authorization = ''
  if (callback) callback(null)
}
```

angular1/app/common/factories/authFactory.js (exportar)

```
logout
```

Listagem 30 - Alterando AuthController (logout)

angular1/app/auth/authController.js

```
auth.logout(() => msgs.addSuccess('Sucesso!'))
```

Listagem 31 - Função de getUser

angular1/app/common/factories/authFactory.js

```
let user = null

function getUser() {
  if(!user) {
    user = JSON.parse(localStorage.getItem(consts.userKey))
  }
  return user
}
```

angular1/app/common/factories/authFactory.js (exportar)

```
getUser
```

Listagem 32 - Alterando AuthController (getUser)

angular1/app/auth/authController.js

```
auth.getUser()
```

15.3.4. Protegendo as Rotas

Listagem 33 - Será executado na inicialização da aplicação

angular1/app/config/routes.js

```
.run([
  '$rootScope',
  '$http',
  '$location',
  '$window',
  'auth',
  function ($rootScope, $http, $location, $window, auth) {
    console.log('Executando...')
  }
])
```

Listagem 34 - Definindo função para validar usuário

angular1/app/config/routes.js

```
validateUser()
$rootScope.$on('$locationChangeStart', () => validateUser())

function validateUser() {
  console.log('Executando...')
}
```

Listagem 35 - Validar Usuário (Versão 1)

angular1/app/config/routes.js

```
const user = auth.getUser()
const authPage = '/auth.html'
const isAuthPage = $window.location.href.includes(authPage)

if (!user && !isAuthPage) {
  $window.location.href = authPage
} else if (user && !user.isValid) {
  user.isValid = true
  $http.defaults.headers.common.Authorization = user.token
  isAuthPage ? $window.location.href = '/' : $location.path('/dashboard')
}
```

Listagem 36 - Alterar URL após Login/Signup/Logout

angular1/app/auth/authController.js (função login)

```
auth.login(vm.user, err => err ? msgs.addError(err) : $location.path('/'))
```

angular1/app/auth/authController.js (função signup)

```
auth.signup(vm.user, err => err ? msgs.addError(err) : $location.path('/'))
```

angular1/app/auth/authController.js (função logout)

```
auth.logout(() => $location.path('/'))
```

angular1/app/common/factories/authFactory.js (função logout)

```
user = null
```

==== Validar Token

backend/api/user/authService.js (aprox. linha 25)

```
expiresIn: "10 seconds"
```

angular1/app/common/factories/authFactory.js

```
function validateToken(token, callback) {
  if (token) {
    $http.post(`${consts.oapiUrl}/validateToken`, { token })
      .then(resp => {
        if (!resp.data.valid) {
          logout()
        } else {
          $http.defaults.headers.common.Authorization = getUser().token
        }
        if (callback) callback(null, resp.data.valid)
      }).catch(function (resp) {
        if (callback) callback(resp.data.errors)
      })
  } else {
    if (callback) callback('Token inválido.')
  }
}
```

angular1/app/common/factories/authFactory.js (exportar)

validateToken

Listagem 37 - Chamar validateToken_

angular1/app/config/routes.js (aprox. linha 34)

```
auth.validateToken(user.token, (err, valid) => {
  if (!valid) {
    $window.location.href = authPage
  } else {
    user.isValid = true
    $http.defaults.headers.common.Authorization = user.token
    isAuthPage ? $window.location.href = '/' : $location.path('/dashboard')
  }
})
```

Listagem 38 - Remover rota padrão_

angular1/app/config/routes.js (Remover linha)

```
$urlRouterProvider.otherwise('/dashboard')
```


15.3.5. Tratar Resposta com Erro

Listagem 39 - Factory para tratar respostas com erro

angular1/app/common/factories/handleResponseErrorFactory.js

```
(function () {  
    angular.module('primeiraApp').factory('handleResponseError', [  
        '$q',  
        '$window',  
        'consts',  
        HandleResponseErrorFactory  
    ])  
  
    function HandleResponseErrorFactory($q, $window, consts) {  
        function responseError(errorResponse) {  
            if (errorResponse.status === 403) {  
                localStorage.removeItem(consts.userKey)  
                $window.location.href = '/'  
            }  
            return $q.reject(errorResponse)  
        }  
  
        return { responseError }  
    }  
})();
```

Listagem 40 - Chamar HandleResponseErrorFactory

angular1/app/config/routes.js (lista de dependências)

```
'$httpProvider',
```

angular1/app/config/routes.js (parâmetro da função)

```
$httpProvider
```

angular1/app/config/routes.js (Adicionar aos interceptors)

```
$httpProvider.interceptors.push('handleResponseError')
```

Appendix A: Tabela de Códigos

- Listagem 1 - *Instalando as dependências*
- Listagem 2 - *Criar arquivo .env*
- Listagem 3 - *Adicionar .env ao .gitignore*
- Listagem 4 - *Criar arquivo user.js*
- Listagem 5 - *Estrutura básica de authService.js*
- Listagem 6 - *Criar método login de authService.js*
- Listagem 7 - *Criar método validateToken de authService.js*
- Listagem 8 - *Criar método signup de authService.js*
- Listagem 9 - *Exportar os métodos de authService.js*
- Listagem 10 - *Criar middleware auth.js*
- Listagem 11 - *Adicionar novo header em cors.js*
- Listagem 12 - *Alterar arquivo routes.js*
- Listagem 13 - *Descomentar o item user-menu*
- Listagem 14 - *Remover Menu Body*
- Listagem 15 - *Remover Botão Perfil*
- Listagem 16 - *Criar AuthController*
- Listagem 17 - *Adicionar ng-controller*
- Listagem 18 - *Alterar user.html para acessar controller*
- Listagem 19 - *Funções para suportar a tela Login/Cadastro*
- Listagem 20 - *Componente authField*
- Listagem 21 - *Formulário Login/Cadastro*
- Listagem 22 - *Página de Login/Cadastro*
- Listagem 23 - *Novas Constantes*
- Listagem 24 - *Estrutura inicial*
- Listagem 25 - *Funções de Login/Cadastro*
- Listagem 26 - *Adicionando AuthFactory no AuthController*
- Listagem 27 - *Alterando AuthController (login)*
- Listagem 28 - *Alterando AuthController (signup)*
- Listagem 29 - *Função de Logout*
- Listagem 30 - *Alterando AuthController (logout)*
- Listagem 31 - *Função de getUser*
- Listagem 32 - *Alterando AuthController (getUser)*
- Listagem 33 - *Será executado na inicialização da aplicação*

- Listagem 34 - *Definindo função para validar usuário*
- Listagem 35 - *Validar Usuário (Versão 1)*
- Listagem 36 - *Alterar URL após Login/Signup/Logout*
- [ex-auth-routes-5]
- [ex-validate-token-1]
- [ex-validate-token-2]
- Listagem 37 - Chamar validateToken_
- Listagem 38 - Remover rota padrão_
- Listagem 39 - *Factory para tratar respostas com erro*
- Listagem 40 - *Chamar HandleResponseErrorFactory*

Glossário

JWT

...