

# Projecto ARQSI 2017

---

## Geral:

Esta WIKI contém informação geral do projecto proposto na disciplina de Arquitetura de Sistema, relativo ao ano lectivo 2017/ 2018 (3º Ano, 1º Semestre), do ISEP.

[Glossario](#)

[Requisitos](#)

[Modelo Dominio](#)

[Modelo Relacional](#)

[Testes](#)

---

## Vistas:

[Cenarios](#)

[Logica](#)

[Implementacao](#)

[Implantação](#)

[Processo](#)

---

ARQSI 2017 - Pedro Fernandes ([1060503](#)), Diana Silva ([1151088](#))

# Glossário

[Voltar](#)

## A

**Apresentação:** Forma de apresentação do fármaco que o médico quer prescrever (modo de uso).

## B

## C

**Código de acesso (Receita)** : código que permite aceder à dispensa de medicamentos na farmácia.

## D

**Dosagem:** quantidade de medicamento ou fármaco ativo que se administra de cada vez.

## E

## F

**Fármaco:** substância activa a ser prescrita

**Forma farmacéutica:** tipo de medicamento (p.ex.: xarope, comprimido).

## G

## H

## I

**Intervalo de tempo:** de quanto em quanto tempo a administração do medicamento deverá ser repetida (p.ex.: a cada 12 horas).

## J

## K

## L

**Laboratório:** local onde são criados os medicamentos.

## M

## N

**NIF:** número de identificação fiscal (número de contribuinte).

**Número de beneficiário:** número que identifica o utente no sistema nacional de saúde.

O

P

**Paciente:** Pessoa a quem será prescrito um ou mais medicamentos

**Período de tempo (Posologia) :** durante quantos dias o medicamento vai/deve ser tomado (p.ex.: durante 7 dias).

**Posologia:** a quantidade (dos) de um medicamento que deve ser fornecida a um paciente por determinada via de administração, com intervalo de tempo constante, durante um período de tempo para alívio de um sintoma ou tratamento de uma doença.

**Prescrição:** indicação de medicamento e medida médica que um paciente deve tomar ou realizar

**Princípio Activo:** substância que deverá exercer efeito farmacológico.

Q

R

**Receita:** conjunto de prescrições

S

T

**Tamanho :** dimensoes da embalagem do medicamento (área em cm3).

U

V

**Validade (Medicamento):** data máxima em que o medicamento pode ser tomado.

**Validade (Receita):** data máxima em que o utente pode levantar dado medicamento associado a uma/mais prescrição da receita.

**Via de administração:** forma como o medicamento é aplicado (ex: via oral)

W

X

Y

Z

## Requisitos Funcionais

Pretende-se implementar um sistema que permita a prescrição de medicamentos de forma eletrónica e que permita suportar as funcionalidades de gestão associadas a esse requisito. Será necessário desenvolver um sistema de informação que permita:

1. Às entidades competentes (e.g. Infarmed) manter a informação sobre medicamentos existentes (nomes, princípios ativos, forma farmacêutica (comprimidos, xarope, ...), tamanho da embalagem, etc.).
2. Ao médico aceder à informação (mantida pelo sistema descrito em 1) de forma a selecionar os medicamentos que pretende prescrever.
3. Ao médico criar/armazenar receita médica (prescrição), associando-a ao médico e ao paciente.
4. Às farmácias/farmacêuticos, disponibilizarem total ou parcialmente as receitas dos pacientes.
5. Alertar os pacientes para a proximidade de data de expiração de alguma prescrição não totalmente aviada.
6. Aos pacientes acederem às prescrições que lhe foram atribuídas de forma a conseguirem identificar os medicamentos associados a cada prescrição com discriminação das quantidades prescritas e adquiridas. As prescrições totalmente aviadas devem manter-se em histórico.
7. Ao paciente obter informação sobre posologia dos medicamentos prescritos.

- Iteração 1

Desenvolver um conjunto de funcionalidades acessíveis através de serviços web de gestão da informação sobre medicamentos.

Algumas das funcionalidades requeridas:

URL	Método	Semântica
Medicamento/{id}	GET	Medicamento com id={id}
Medicamento/	POST	Novo medicamento
Medicamento/?nome={nome}	GET	Medicamento com nome={nome}
Medicamento/{id}/Apresentacoes	GET	Apresentações de medicamento {id}
Medicamento/{id}/Posologias	GET	Posologias de medicamento {id}
Farmaco/{id}	GET	Fármaco com id={id}
Farmaco/?nome={nome}	GET	Fármaco com nome={nome}
Farmaco/{id}/Medicamentos	GET	Medicamentos com fármaco {id}
Farmaco/{id}/Posologias	GET	Posologias de fármaco {id}
Farmaco/{id}/Apresentações	GET	Apresentações de fármaco {id}
Apresentacao/{id}	GET	Apresentação com {id}
Apresentacao/	POST	Nova apresentação

Tecnologias: Net Core 2.0 WebAPI, Microsoft SQL Server, Postman e SOAPUI.

## Requisitos Não Funcionais

- Devem ser usadas classificações, nomenclaturas e outros formatos de representação e partilha de dados sobre medicamentos.
- O sistema de ser acedido apenas mediante autenticação e autorização válidas.
- Serão usadas funcionalidades de sistemas terceiros (e.g. SNS, ADSE, seguradoras).
- O sistema deve ser implantado na Nuvem, em máquinas virtuais ou em Docker.

- O sistema deve ser escalável, embora decorram ainda avaliações de necessidades reais.
- A usabilidade do sistema está neste momento em análise por uma equipa multidisciplinar.
- O desenvolvimento do software deve adotar:
  - Adotar um processo iterativo e incremental;
  - Adotar boas práticas, estilos e padrões adequados;
  - Desenvolver e aplicar testes de regressão em todas as disciplinas do processo;
  - Adotar controlo de versões.

# Modelo Dominio

[Voltar](#)

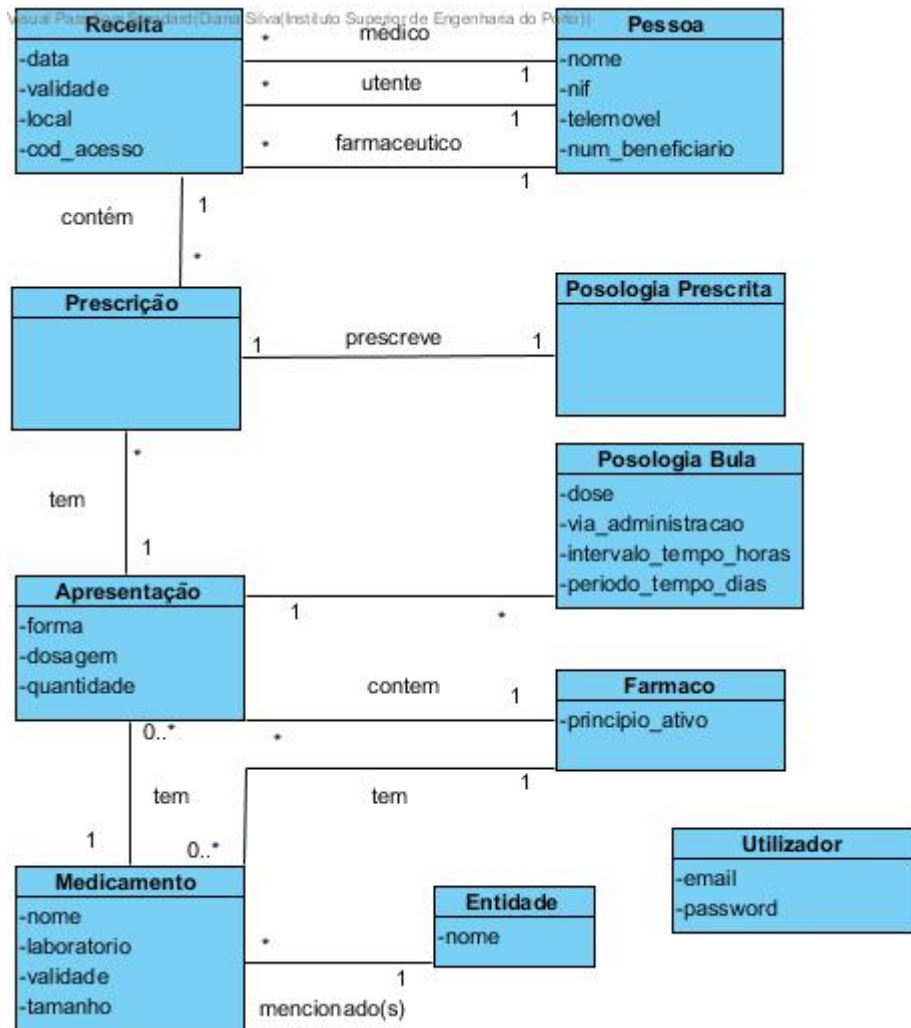
## Identificação de Classes de Domínio

Medicamento, Fármaco, PosologiaPrescrita, Apresentação, Prescrição, Pessoa, Receita, Entidade.

## Identificação de Associações

Informações do cliente: Podem existir medicamentos sem prescrições e vice-versa? Sim.

Uma prescrição tem sempre apenas o princípio ativo apresentado ou poderá ter o medicamento? Pode ter as duas coisas.



Pressupostos: - deverão haver restrições na criação de uma apresentação em que se adiciona ou medicamento ou fármaco

- deverão haver restrições na criação de uma apresentação em que ao inserir posologia,obtem-se de fármaco escolhido.

## Wiki

ARQSI2017-3DA-1060503-1151088 / modelo dominio

View

History

Edit

Delete

## Modelo Dominio

[Voltar](#)

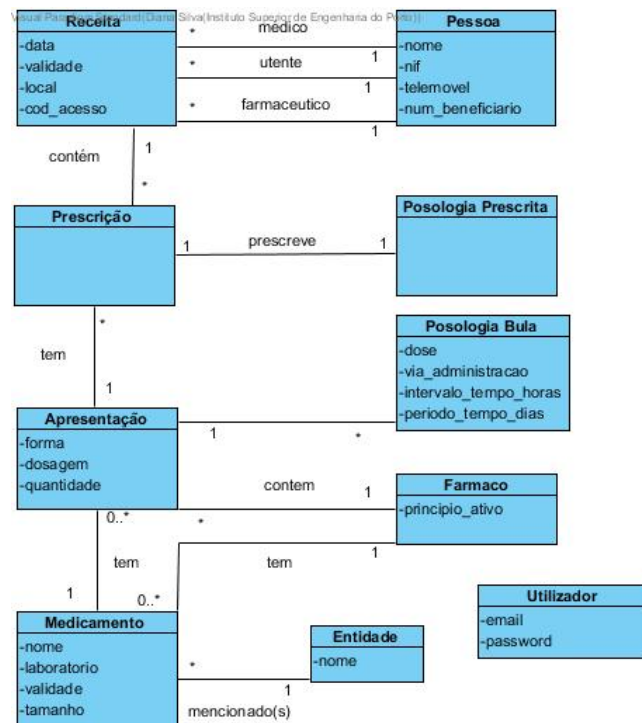
## Identificação de Classes de Domínio

Medicamento, Fármaco, PosologiaPrescrita, Apresentação, Prescrição, Pessoa, Receita, Entidade.

## Identificação de Associações

Informações do cliente: Podem existir medicamentos sem prescrições e vice-versa? Sim.

Uma prescrição tem sempre apenas o princípio ativo apresentado ou poderá ter o medicamento? Pode ter as duas coisas.



Pressupostos: - deverão haver restrições na criação de uma apresentação em que se adiciona ou medicamento ou fármaco

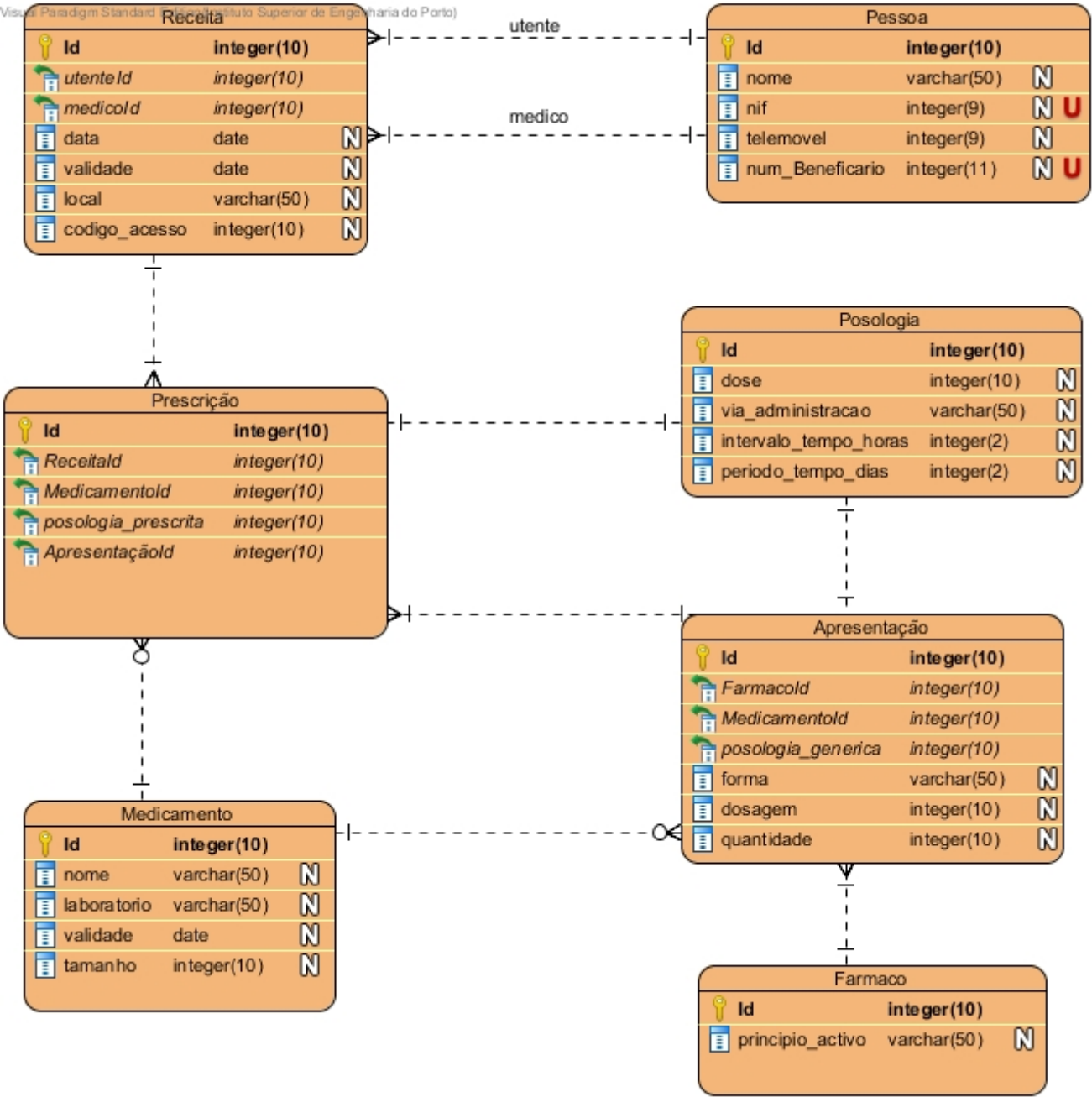
- deverão haver restrições na criação de uma apresentação em que ao inserir posologia, obtem-se de fármaco escolhido.

Updated 18 hours ago

# Modelo Relacional

[Voltar](#)

Visual Paradigm Standard (© 2019 Instituto Superior de Engenharia do Porto)





Wiki

ARQSI2017-3DA-1060503-1151088 / testes

View History Edit Delete

Testes

Testes de Integração (Restfull - API)

Ferramenta utilizada: POSTMAN (cliente REST)

Testes efetuados: todos os testes mencionados abaixo sendo que para efeitos de documentação demonstramos um por acção rest implementada (get, post, put, delete).

- GET

api/Medicamento

GET

http://localhost:50647/api/Medicamento

Params

Send

Authorization

Headers (2)

Body

Pre-request Script

Tests

1 //----- GET TEST ----

2

3 tests["Codigo com estado 200"]=responseCode.code===200;

4 tests["Codigo para falhar"]= !(responseCode.code===400 || responseCode.code===404);

5

6 var obj= '{"id":1,"nome":"Aspirina Prevent"}'; //presente na base de dados

7 var obj2= '{"id":2,"nome":"Aspirina Prevent"}'; //não presente na base de dados

8 var jsonObj=JSON.parse(obj);

9 var jsonObj2=JSON.parse(obj2);

10

11 tests["Lista Medicamentos (json contem medicamento com id)"]= responseBody.has(obj);

12 tests["Lista Medicamentos (nao contem medicamento com id)"]=!responseBody.has(obj2);

13

14

15

Test scripts are written in run after the response is

Learn more about tests

SNIPPETS

Clear a global variable

Clear an environment var

Response body: Contains

Response body: Convert: Object

Response body: Is equal I

Response body: JSON vali

Response headers: Conte

Body

Cookies

Headers (6)

Test Results (4/4)

Status: 200

All

Passed

Failed

PASS

Codigo com estado 200

PASS

Codigo para falhar

PASS

Lista Medicamentos (json contem medicamento com id)

PASS

Lista Medicamentos (nao contem medicamento com id)

api/Farmaco

api/Posologia

api/Apresentacao

api/Medicamento/{id}

api/Posologia/{id}

api/Apresentacao/{id}

?



GET

http://localhost:50647/api/Apresentacao/1

Params

AuthorizationHeaders (2)BodyPre-request ScriptTests

1

//----- GET TEST ({id}) -----

2

3

tests["Codigo com estado 200"]=responseCode.code===200;

4

tests["Codigo para falhar"]=!(responseCode.code===400 || responseCode.code===404);

5

6

var obj= '{"id":1,"forma\_adm":"Xarope","concentracao":"25mg","qtd":"1ml"}'; //presente na base de dados

7

var obj2= '{"id":2,"nome":"Aspirina Prevent"}'; //não presente na base de dados

8

var jsonObj=JSON.parse(obj);

9

var jsonObj=JSON.parse(obj2);

10

11

tests["Lista Apresentacoes (json contem id 1)"]= responseBody.has(obj);

12

tests["Lista Apresentacoes (nao contem id 1)"]=!responseBody.has(obj2);

13

14

15

Test scrip

run after

Learn m

SNIPPETS

Clear a g

Clear an

Respons

Respons

Object

Respons

Respons

Respons

BodyCookiesHeaders (6)Test Results (4/4)

AllPassedFailed

PASS

Codigo com estado 200

PASS

Codigo para falhar

PASS

Lista Apresentacoes (json contem id 1)

PASS

Lista Apresentacoes (nao contem id 1)

- api/Medicamento/{id}/Apresentacoes
- api/Medicamentos/{id}/Posologias
- api/Farmaco/{id}/Medicamentos
- api/Farmaco/{id}/Posologias
- api/Farmaco/{id}/Apresentacoes
- api/Medicamento/nome={nome}
  - POST
- api/Medicamento



GET

http://localhost:50647/api/Medicamento

Params

Send

Authorization

Headers (2)

Body

Pre-request Script

Tests

1 //----- POST TEST -----  
2  
3 var obj= '{"id":8,"nome":"TESTE"}'; //não presente na base de dados  
4 var jsonObj=JSON.parse(obj);  
5  
6 tests["Lista Medicamentos ainda nao contem medicamento teste"]=!responseBody.has(obj);  
7  
8  
9  
10

Test scripts are written in JavaScript and run after the response is received. Learn more about tests

SNIPPETS  
Clear a global variable  
Clear an environment variable  
Response body: Contains  
Response body: Convert to JSON  
Response body: Is equal  
Response body: JSON value  
Response headers: Contains

Body

Cookies

Headers (6)

Test Results (1/1)

Status: 200

All

Passed

Failed

PASS

Lista Medicamentos ainda nao contem medicamento teste

POST

http://localhost:50647/api/Medicamento

Params

Send

Authorization

Headers (2)

Body

Pre-request Script

Tests

1 //----- POST TEST -----  
2  
3 tests["POST com sucesso"]= responseCode.code===201 || responseCode.code===202;  
4  
5 var obj= '{"id":8,"nome":"TESTE"}'; // presente na base de dados  
6 var jsonObj=JSON.parse(obj);  
7  
8  
9  
10 tests["Lista Medicamentos contem medicamento teste"]=responseBody.has(obj);  
11  
12  
13

Test scripts are written in JavaScript and run after the response is received. Learn more about tests

SNIPPETS  
Clear a global variable  
Clear an environment variable  
Response body: Contains  
Response body: Convert to JSON  
Response body: Is equal  
Response body: JSON value  
Response headers: Contains

Body

Cookies

Headers (7)

Test Results (1/2)

Status: 200

All

Passed

Failed

PASS

POST com sucesso



http://localhost:50647

http://localhost:50647

+

...

GET

http://localhost:50647/api/Medicamento/

Authorization

Headers (2)

Body

Pre-request Script

Tests

1

2

3

4

5

6

7

8

9

10

//----- POST TEST ----

tests["GET com sucesso"]= responseCode.code===200;

var obj= '{"id":8,"nome":"TESTE"}'; // presente na base de dados

var jsonObj=JSON.parse(obj);

tests["Lista Medicamentos contem medicamento teste"]=responseBody.has(obj);

Body

Cookies

Headers (6)

Test Results (2/2)

All

Passed

Failed

PASS

GET com sucesso

PASS

Lista Medicamentos contem medicamento teste

api/Farmaco

api/Posologia

api/Apresentacao

api/Account/

api/Account/Login

api/Account/Token





- api/Medicamento/{id}
- api/Farmaco/{id}
- api/Posologia/{id}
- api/Apresentacao/{id}

Put



http://localhost:50647

http://localhost:50647

+

...

Test

PUT

http://localhost:50647/api/Farmaco/1

Params

Send

Authorization

Headers (2)

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1 {

2     "FarmacoId": "1",

3     "principio\_ativo": "TESTE Acido acetil-salicilico"

4 }

Body Cookies Headers (4) Test Results (1/3) Status: 204 No C

All Passed Failed

PASS PUT com sucesso

http://localhost:50647

http://localhost:50647

+

...

Tes

GET

http://localhost:50647/api/Farmaco/

Pa

Authorization

Headers (2)

Body

Pre-request Script

Tests

1 // put

2

3 /\*tests["PUT com sucesso"]= responseCode.code===201 || responseCode.code===202 ||

4     responseCode.code===204;

5 \*/

6 var obj= '{"id":1,"principio\_ativo":"Acido acetil-salicilico"}';

7 var jsonObj=JSON.parse(obj);

8 tests["Posologia PUT - antigo nao encontrado "]=responseBody.has(obj);

9

10 var obj2= '{"id":1,"principio\_ativo":"TESTE Acido acetil-salicilico"}';

11 var jsonObj=JSON.parse(obj2);

12 tests["Farmaco PUT - objeto modificado encontrado "]=responseBody.has(obj2);

Body Cookies Headers (6) Test Results (1/2)

All Passed Failed

FAIL Posologia PUT - antigo nao encontrado

PASS Farmaco PUT - objeto modificado encontrado





PUT

http://localhost:50647/api/Farmaco/1

Authorization

Headers (2)

Body

Pre-request Script

Tests

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

1

2

3

4

{

"FarmacoId": "1",

"principio\_ativo": "TESTEAcido acetil-salicilico"

}

Body

Cookies

Headers (6)

Test Results (0/1)

Pretty

Raw

Preview

JSON

1

2

3

4

{

"id": 1,

"principio\_ativo": "TESTEAcido acetil-salicilico"

}

Delete

[illegible]





GET

http://localhost:50647/api/Posologia/

Params

Send

Save

AuthorizationHeaders (2)BodyPre-request ScriptTests

1//----- DELETE tests -----

2

3var obj= '{"id":2,"dose":400,"via\_administracao":"via oral","intervalo\_tempo\_horas":6

4,"periodo\_tempo\_dias":5}';

5var jsonObj=JSON.parse(obj);

6tests["Posologia DELETE - objeto existente em pre-delete"]=responseBody.has(obj);

7

8/\*

9tests["DELETE com sucesso"]= responseCode.code===201 || responseCode.code===202 ||

10responseCode.code===204;

11

12var obj= '{"id":3,"dose":40,"via\_administracao":"via oral","intervalo\_tempo\_horas":8

13,"periodo\_tempo\_dias":8}';

14var jsonObj=JSON.parse(obj);

15tests["Posologia DELETE - objeto nao existente pos-delete"]=responseBody.has(obj);

16\*/

Test scripts are written in JavaScript, and are run after the response is received.  
Learn more about tests

SNIPPETS

Clear a global variable

Clear an environment variable

Response body: Contains string

Response body: Convert XML body to a JSON Object

Response body: Is equal to a string

Response body: JSON value check

Response headers: Content-Type header check

BodyCookiesHeaders (6)Test Results (1/1)

Status: 200 OKTime: 46 ms

AllPassedFailed

PASSPosologia DELETE - objeto existente em pre-delete

DELETE

http://localhost:50647/api/Posologia/2

Params

Send

AuthorizationHeaders (2)BodyPre-request ScriptTests

1//----- DELETE tests -----

2

3/\*

4var obj= '{"id":2,"dose":400,"via\_administracao":"via oral","intervalo\_tempo\_horas":6

5,"periodo\_tempo\_dias":5}';

6var jsonObj=JSON.parse(obj);

7tests["Posologia DELETE - objeto existente em pre-delete"]=responseBody.has(obj);

8

9/\*

10tests["DELETE com sucesso"]= responseCode.code===200;

11

12var obj= '{"id":2,"dose":400,"via\_administracao":"via oral","intervalo\_tempo\_horas":6

13,"periodo\_tempo\_dias":5}';

14var jsonObj=JSON.parse(obj);

15tests["Posologia DELETE FAIL - objeto nao existente pos-delete"]=!responseBody.has

16(obj);

17

Test scripts are written in JavaScript, and are run after the response is received.  
Learn more about tests

SNIPPETS

Clear a global variable

Clear an environment variable

Response body: Contains string

Response body: Convert XML body to a JSON Object

Response body: Is equal to a string

Response body: JSON value check

Response headers: Content-Type header check

BodyCookiesHeaders (6)Test Results (2/2)

Status: 200 OK

AllPassedFailed

PASSEDELETE com sucesso

PASSPosologia DELETE FAIL - objeto nao existente pos-delete

- Commits
- ARQSI2017-3DA-1060...
- Branches
- Pull requests
- Pipelines
- Wiki
- Downloads
- Boards
- Settings

GET
http://localhost:50647/api/Apresentacao
Params
Send

Authorization
Headers (2)
Body
Pre-request Script
Tests

Type
No Auth

Body
Cookies
Headers (6)
Test Results (2/2)
Status

Pretty
Raw
Preview

```
[{"id":3,"forma_admin":"Supositorio","concentracao":"10mg","qtd":"2ml"},
{"id":5,"forma_admin":"Comprimido","concentracao":"25mg","qtd":"8ml"},
{"id":6,"forma_admin":"Comprimido","concentracao":"25mg","qtd":"8ml"},
{"id":7,"forma_admin":"Liquido","concentracao":"50mg","qtd":"3ml"},
{"id":8,"forma_admin":"Comprimido","concentracao":"25mg","qtd":"8ml"}]
```

SQL Server Object Explorer
dbo.Apresentacao [Data]

SQL Server
(localdb)\MSSQLLocalDB (SQL Server 13.0.4001)
Databases
System Databases
MedicamentosAPIContext-ca95e933-595
Tables
System Tables
External Tables
dbo\_EFMigrationsHistory
dbo.Apresentacao
dbo.AspNetRoleClaims
dbo.AspNetRoles
dbo.AspNetUserClaims
dbo.AspNetUserLogins
dbo.AspNetUserRoles
dbo.AspNetUsers
dbo.AspNetUserTokens
dbo.Farmaco
dbo.Medicamento
dbo.Posologia
Views
System Views
Synonyms
Programmability
External Resources
Service Broker
Storage
Security
SementeContext-7e0f97ac-ea8a-48b6-8f
Security
Server Objects
Projects

Max Rows: 1000

Apresentac...	Farmacoid	Medicame...	Posologia...	dosagem	forma_admin	quantidade
3	3	5	5	10	Supositorio	2
5	3	2	1	25	Comprimido	8
6	3	3	3	25	Comprimido	8
7	2	8	4	50	Liquido	3
8	2	6	3	25	Comprimido	8
* NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Testes Unitários

- testarGETMedicamentoOK()
- testarGETIdMedicamentoCorretoDevolvido()
- testarGETMedicamentoDevolveLista()
- testarPUTMedicamentoOk()
- testarPUTMedicamentoQuandoNaoExiste()
- testarDELETEQuandoExiste()
- testarDELETEQuandoNaoExiste()
- testarDefinicaoBD()
- testarMedicamentosCarregadosBD()

Updated 18 hours ago

# Cenários de Uso:

[Modelo Casos Uso](#)

[Diagrama Casos Uso](#)

[Casos Uso](#)

## Actores:

- Infarmed
- Médico
- Paciente
- Farmácia
- Sistema Alertas

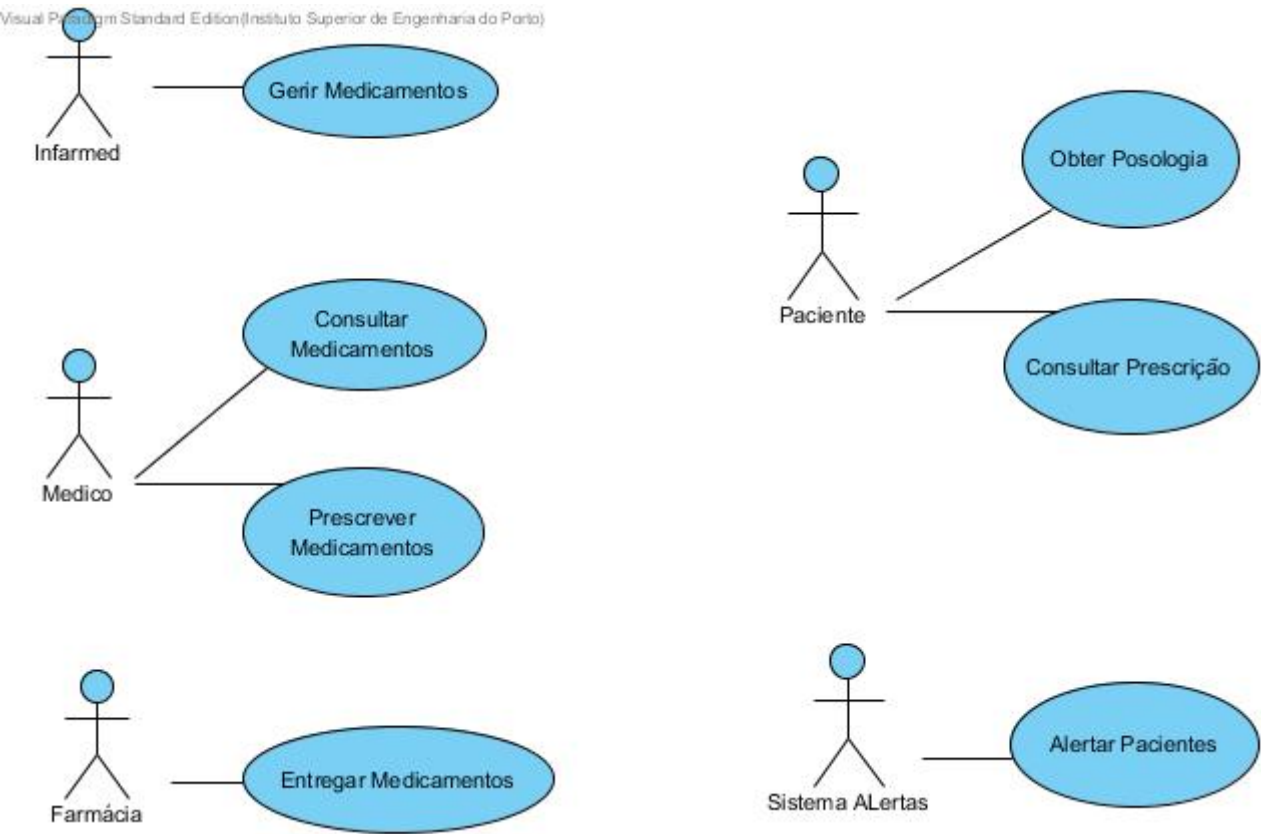
## Casos de Uso:

- Gerir Medicamentos
- Consultar Medicamentos
- Prescrever Medicamentos
- Obter Posologia
- Consultar Prescrição
- Entregar Medicamentos
- Alertar Pacientes

# Diagrama Casos Uso

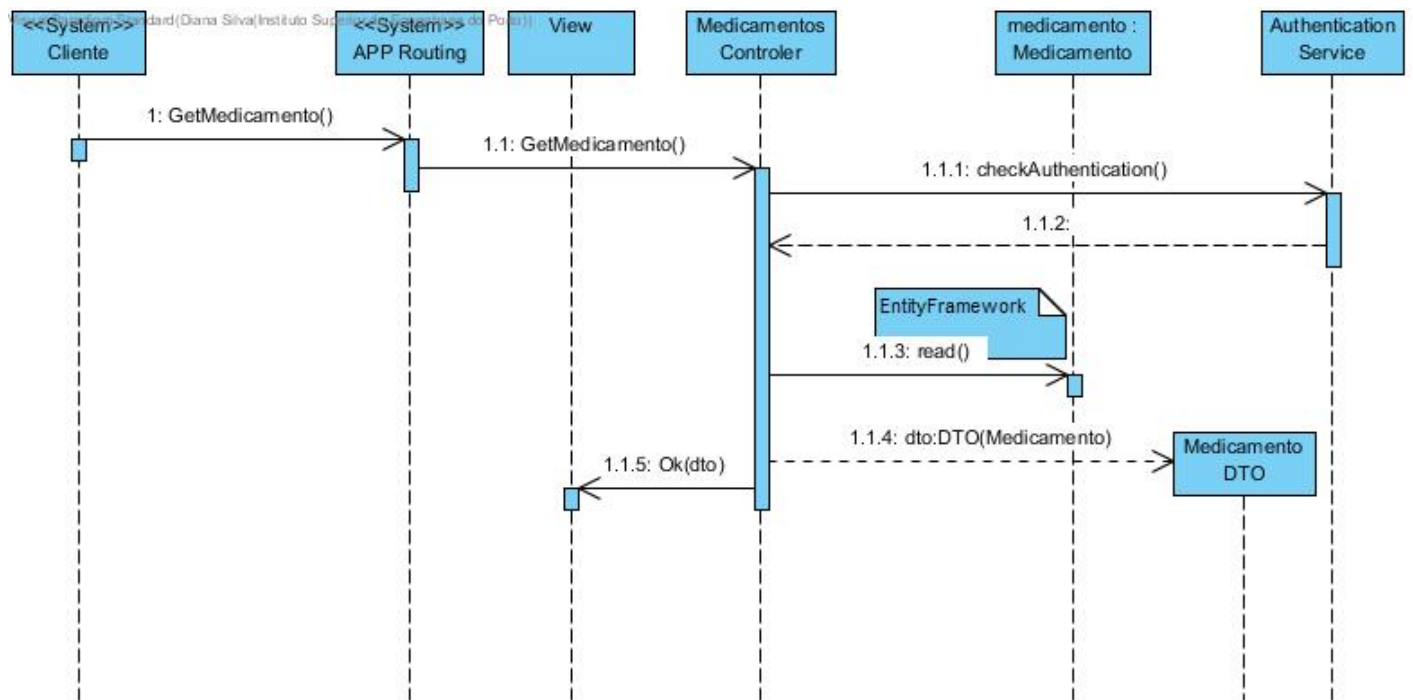
[Voltar](#)

Visual Programming Standard Edition(Instituto Superior de Engenharia do Porto)

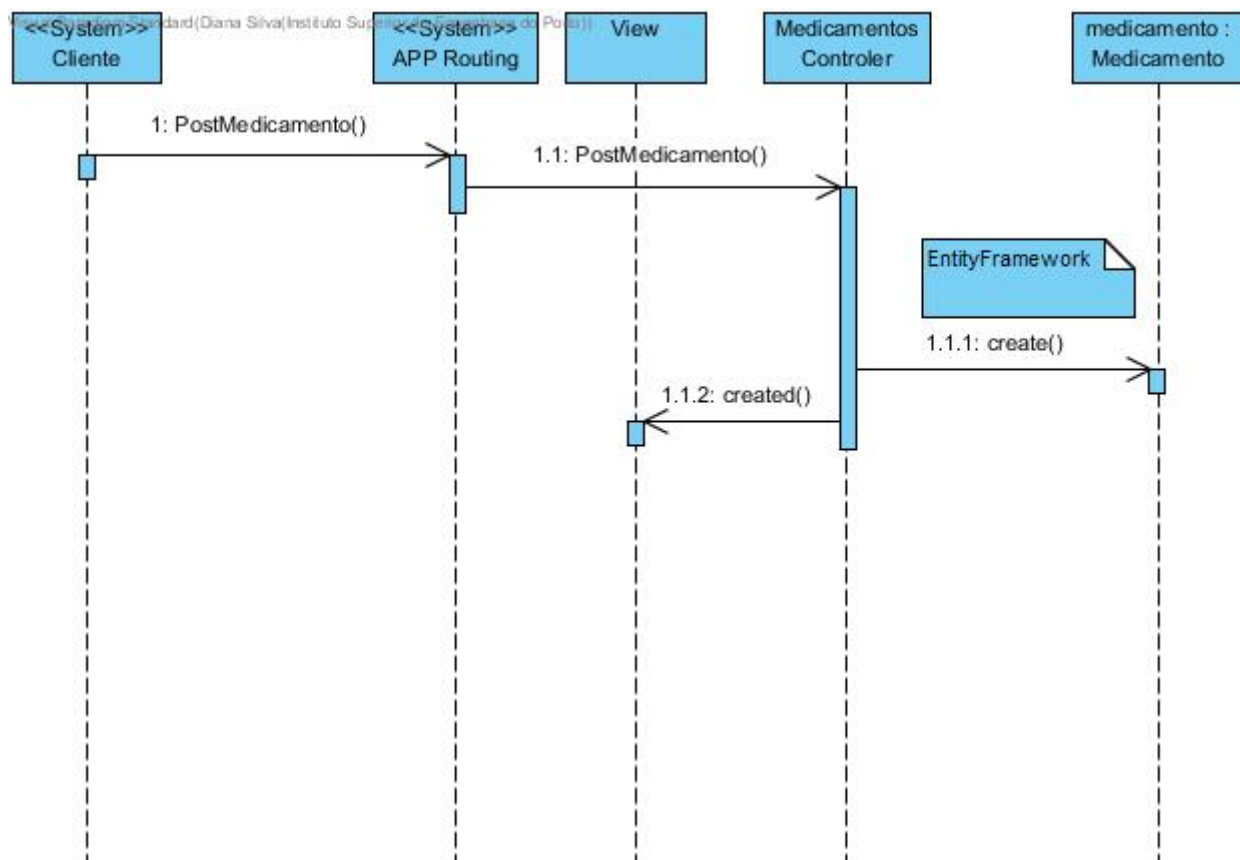


# Casos Uso

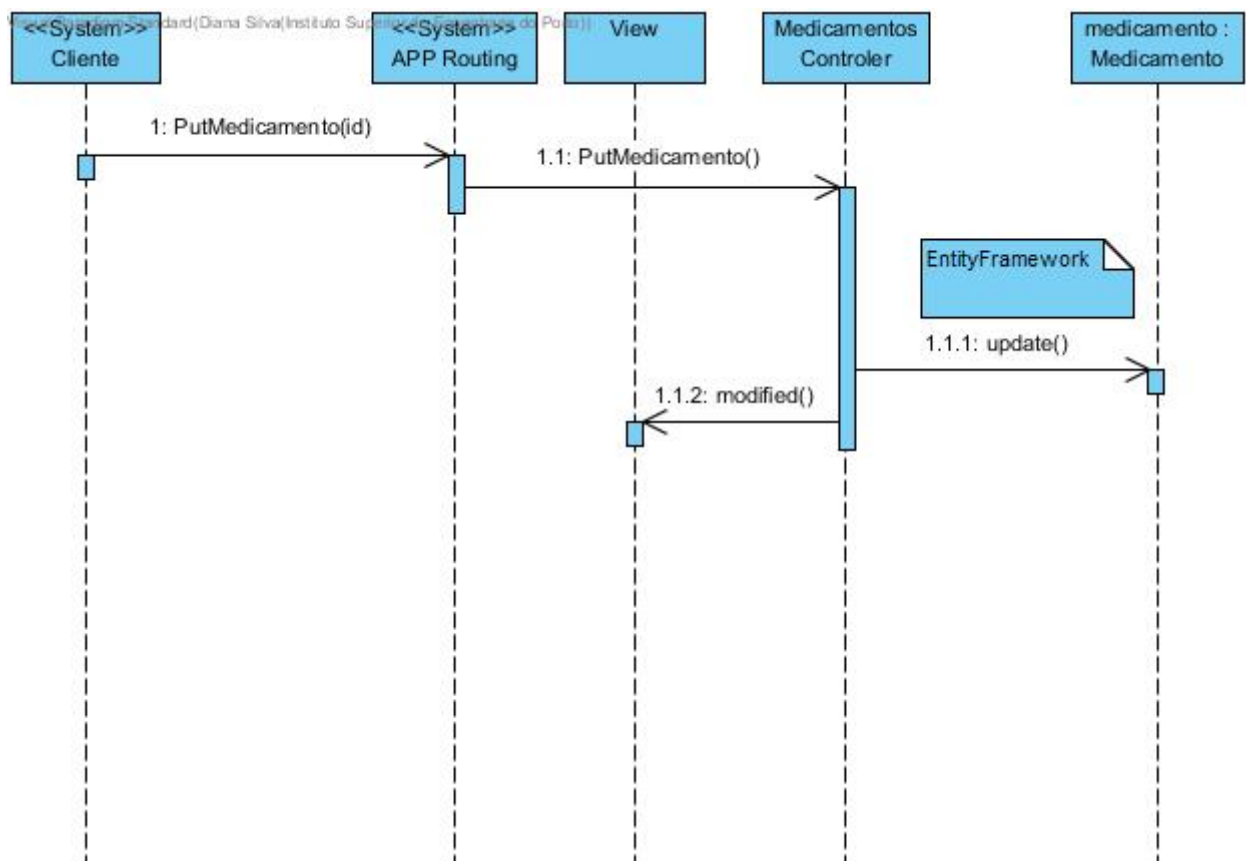
## GetMedicamento



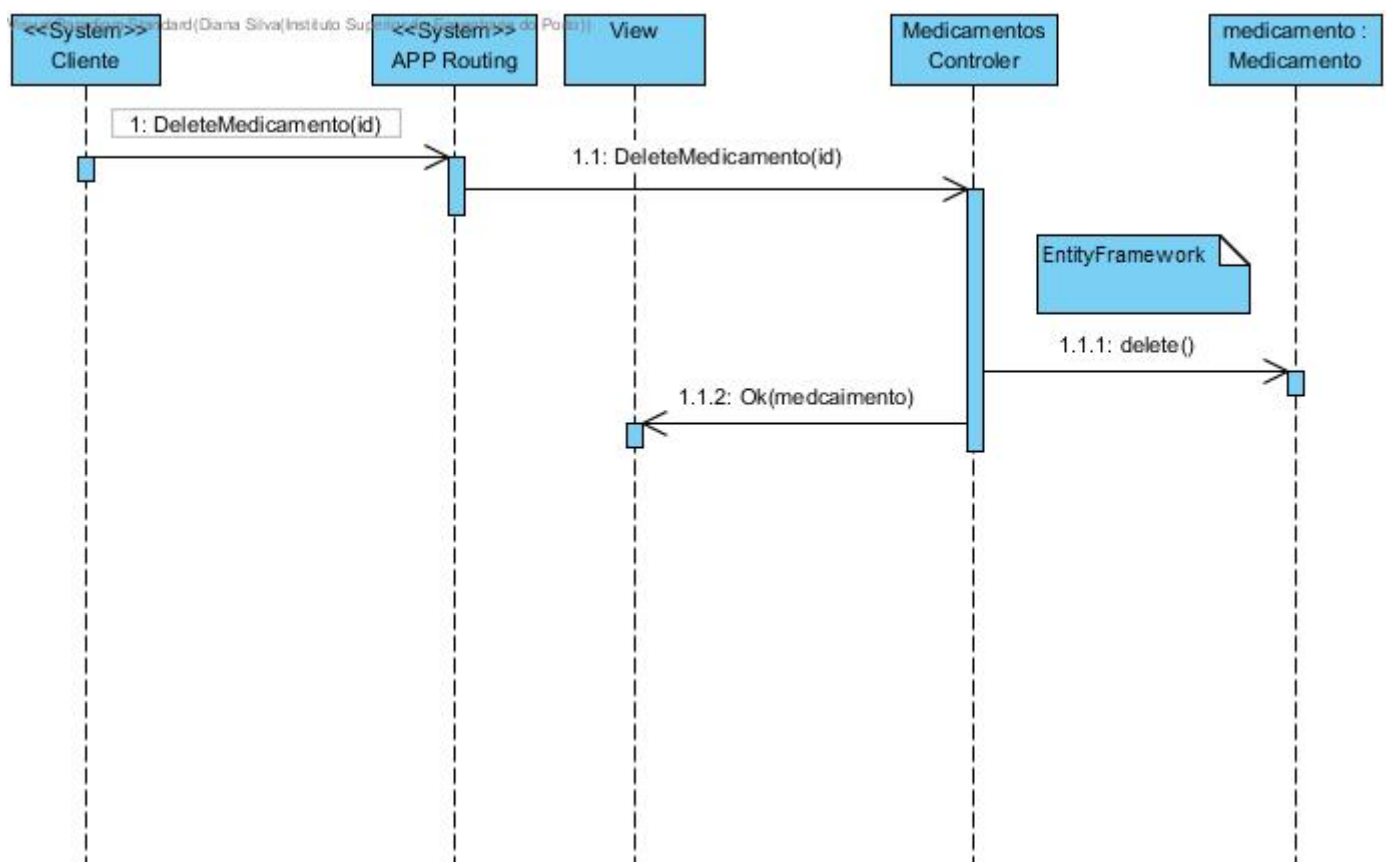
## PostMedicamento



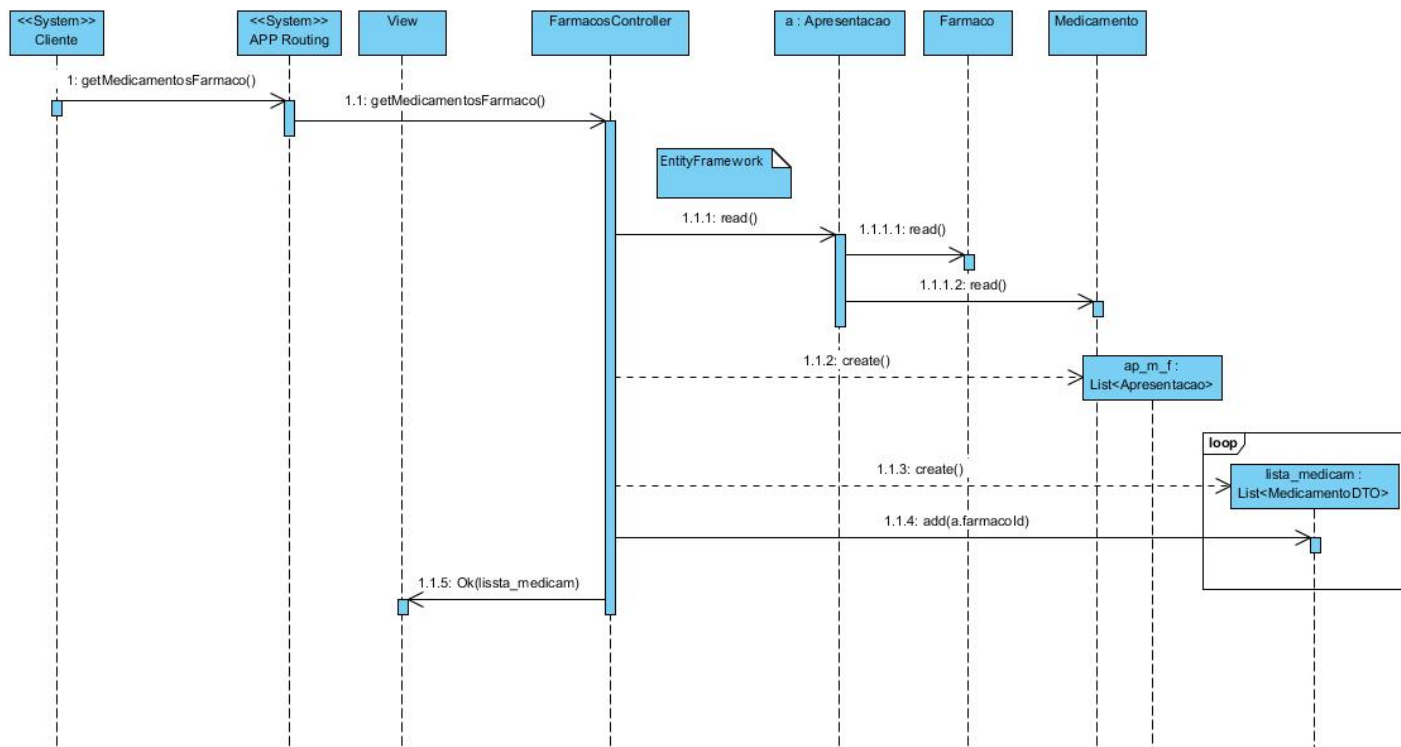
## PutMedicamento



## DeleteMedicamento



## GetMedicamentosFarmaco

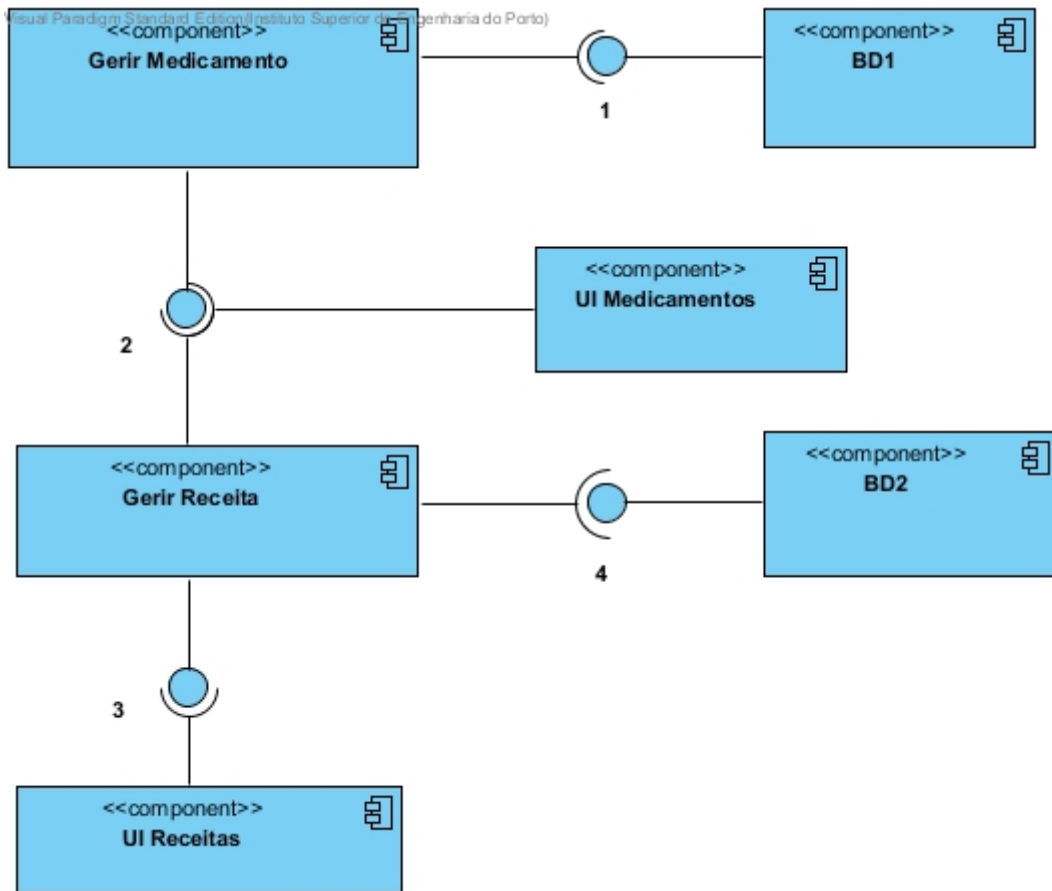


Voltar



# Lógica

[Voltar](#)

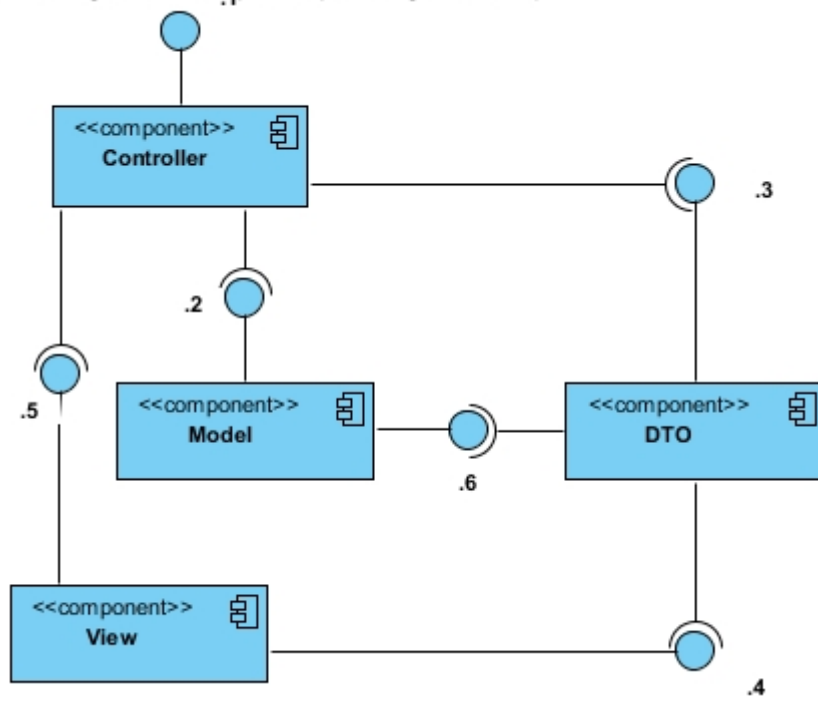


- [Logica de Gestao de Medicamentos](#)
- [Logica de Gestao de Receitas](#)

# Lógica Gestão de Medicamento

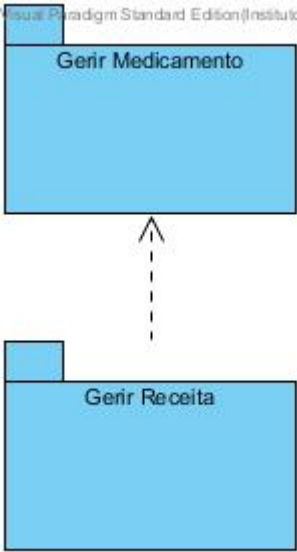
[Voltar](#)

Visual Paradigm Standard Edition (Instituto Superior de Engenharia do Porto)



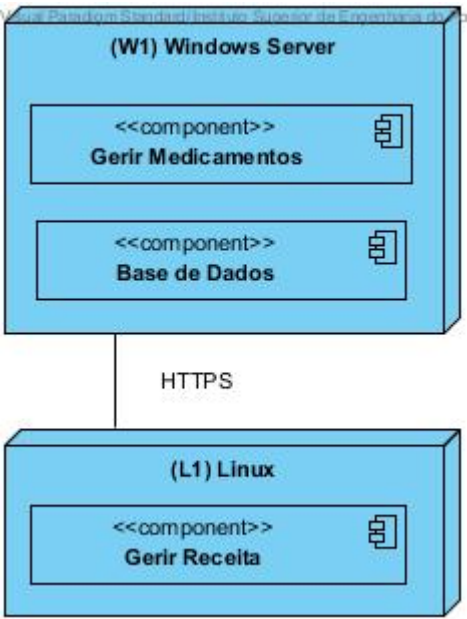
# Implementação

[Voltar](#)



# Implantação

[Voltar](#)



# Processo

[Voltar](#)

Visual Paradigm Standard Edition (Instituto Superior de Engenharia do Porto)

