



APLICAÇÃO DE MONITORIZAÇÃO

Desenvolvimento de Aplicações Empresariais

SÍNTESE

Este projeto tem como objetivo implementar e testar uma aplicação empresarial de monitorização de embalagens inteligentes.

Miguel Venâncio Crespo - 2222046

Bernardo José Mendes Lopes - 2222048

José Miguel Delgado - 2222049

Índice

Introdução	3
Mapa Lógico dos Sistemas	5
Especificação da API REST	6
Sistema de Apoio ao Cliente.....	6
Sistema de Logística	10
Sistema Operacional.....	16
Sistema de Sensores.....	21

Introdução

Aplicação de Monitorização

A Aplicação de Monitorização permite recolher informação de sensores de monitorização integrados nas embalagens de alguns produtos ou encomendas. Para além de receber informação dos sensores a Aplicação de Monitorização, faz a comunicação, e dá resposta a todas as funcionalidades dos diversos sistemas, exceto ao Sistema de E-Commerce, que para fins académicos será simulado internamente no nosso ambiente de desenvolvimento back-end.

Sistema de E-Commerce

Refere-se à plataforma de comércio eletrónico, onde o cliente realiza a compra.

Está ligado à Logística, indicando quando o cliente faz uma encomenda, as informações dos produtos e do utilizador são enviadas para a logística, para serem tratadas posteriormente pela Aplicação de Monitorização.

Sistema de Apoio ao Cliente

Representa o Serviço de Apoio ao Cliente, que se conecta diretamente à Aplicação de Monitorização. Aqui, o cliente pode consultar o estado, a localização entre outros detalhes das suas encomendas, bem como, fazer o cancelamento das encomendas que ainda estão numa fase de processamento.

O Cliente poderá consultar o histórico de todas as encomendas, as encomendas por entregar e em processamento, as encomendas já entregues e receber alertas quando existe um inconveniente com as suas encomendas.

O Cliente terá acesso a um histórico de leituras feitas pelos sensores de uma determinada categoria, que abrange todas as suas encomendas.

Sistema de Logística

Responsável por fazer a associação de um sensor a um determinado volume de uma encomenda existente, fazer a criação de uma nova encomenda o que implica a escolha de todos os produtos da encomenda e a seleção do utilizador, dono dessa mesma encomenda.

Neste sistema é onde é feita a leitura do sensor que ficará associado a um volume, pode consultar-se os detalhes de um determinado volume e de uma encomenda específica. Como extra decidimos que cada volume de uma encomenda terá sempre associado um sensor de GPS, pois os clientes e os gestores conseguirão acompanhar de forma visual a geolocalização dos volumes da mesma.

O Sistema de Logística permite recolher a lista de encomendas com o estado “por entregar”, para que quando uma determinada encomenda chega ao seu ponto de destino, possa alterar o seu estado para “entregue” e informar a Aplicação de Monitorização para fazer a atualização dessa informação, bem como, a desativação dos sensores associados ao(s) volume(s) dessa mesma encomenda.

Sistema Operacional

Interage com a Aplicação de Monitorização para gerir e supervisionar detalhadamente as operações do sistema, como a monitorização de volumes, sensores e encomendas.

Será possível ver alertas rigorosos das diversas encomendas em distribuição, com a possibilidade de verificar a última leitura registada pelos sensores de uma categoria específica e ainda ativos nas suas encomendas.

Sistema de Sensores

O Sistema de Sensores monitoriza aspetos como temperatura, localização, entre outros dados, e enviam essa informação diretamente para a Aplicação de Monitorização, assegurando que as encomendas estão nas condições corretas durante o transporte. Com a constante monitorização dos dados dos sensores, quando o sistema deteta um sensor com 1% de bateria, informa a Aplicação de Monitorização para que o mesmo seja desativado.

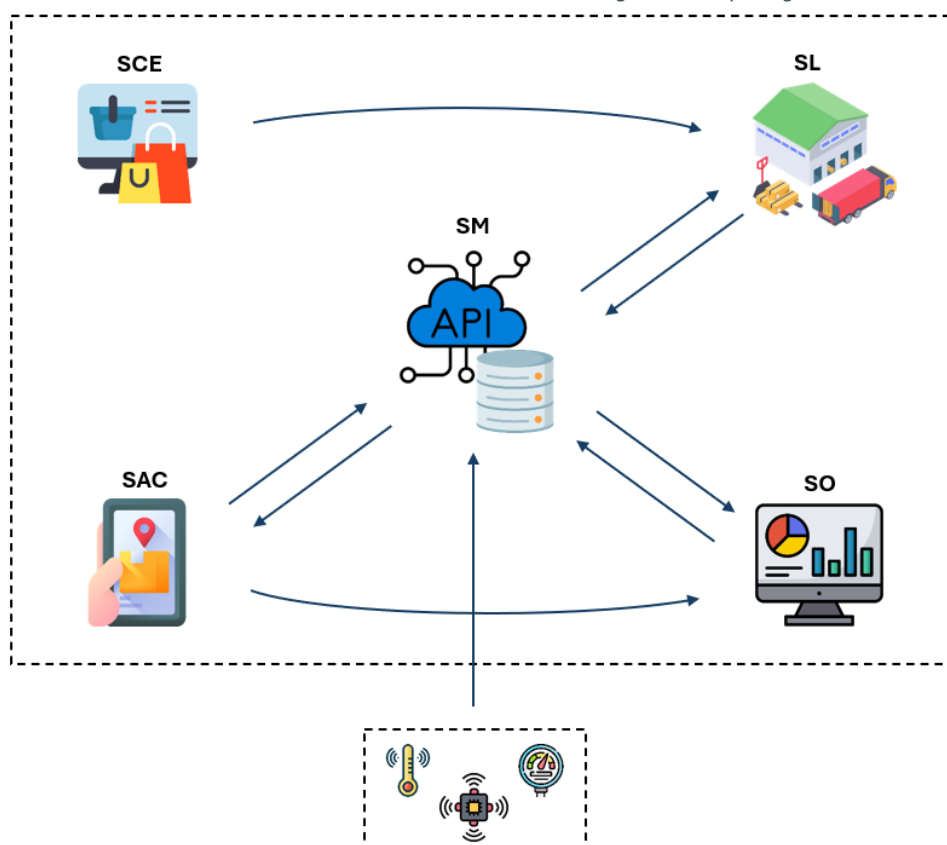
Mapa Lógico dos Sistemas

A Figura 1 - Mapa Lógico dos Sistemas ilustra todos os sistemas presentes no cenário da empresa AmazonJBM, como o Sistema de Monitorização (SM), Sistema de Comercio Eletrónico (SCE), Sistema de Logística (SL), Sistema Operacional (SO), Sistema de Apoio ao Cliente (SAC), e os sensores que estarão fora da empresa, bem como as respetivas comunicações entre eles. Na realização deste projeto apenas iremos desenvolver os sistemas que comunicam diretamente com o sistema de monitorização.

No centro podemos observar o nosso sistema de monitorização, nele estará presente a API juntamente com a base de dados da empresa. A API dará suporte à integração dos sistemas, permitindo que os mesmos comuniquem entre si de forma eficiente e fluida, utilizando o protocolo HTTP. A base de dados da empresa guardará os diversos dados da mesma, como os dados dos utilizadores, todas as encomendas, os produtos existentes, os tipos de sensores, etc.

Na figura, também podemos observar os sensores que estarão localizados fora da empresa e que terão comunicação direta com o sistema de monitorização, este, por sua vez, será responsável por armazenar os dados capturados pelos sensores e por notificar os diferentes sistemas sobre quaisquer alterações nesses dados. Na prática a simulação dos sensores será realizada através de uma página front-end, nela será possível alterar os dados dos mesmos para efeitos de teste.

Figura 1 - Mapa Lógico dos Sistemas



Especificação da API REST

Sistema de Apoio ao Cliente

1. Um utilizador não autenticado no Sistema de Apoio ao Cliente **efetua o login** através do protocolo HTTP, verbo **POST**, para o sítio:

/backend/api/sac/login

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "username" : "Bernardo"
  "password" : "12#45"
}
```

Após o login, em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

2. Um utilizador autenticado no Sistema de Apoio ao Cliente **visualiza todas as suas encomendas** através do protocolo HTTP, verbo GET, para o sítio:

/backend/api/sac/encomendas

A resposta **devolvida** retorna todas as encomendas e segue o seguinte formato JSON:

```
[
  {
    "data_entrega": "2024-10-29T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "PorEntregar",
    "id": 1,
    "username": "Bernardo"
  },
  {
    "data_entrega": "2024-10-29T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "EmProcessamento",
    "id": 4,
    "username": "Bernardo"
  }
  . . .
]
```

3. Um utilizador autenticado no Sistema de Apoio ao Cliente, **faz um pedido para ver os detalhes de uma encomenda**, através do protocolo HTTP, verbo **GET**, para o sítio: **/backend/api/sac/encomendas/{id}**

A resposta **devolvida** retorna uma determinada encomenda de um cliente específico e segue o seguinte formato JSON:

```
{
  "data_entrega": "2024-10-29T15:30:00",
  "data_expedicao": "2024-10-29T15:30:00",
  "estado": "PorEntregar",
  "id": 1,
  "username": "Bernardo",
  "volumes": [
    {
      "id": 1,
      "nome_produto": "Gelados",
      "quantidade": 5,
      "sensores": [
        {
          "timestamp": "2024-11-03T19:13:37",
          "tipoNome": "Temperatura",
          "valor": "29.0"
        },
        {
          "timestamp": "2024-11-03T19:13:37",
          "tipoNome": "GPS",
          "valor": "39.7344200469475, -8.821063143811228"
        }
      ]
    },
    {
      "id": 2,
      "nome_produto": "Perfume",
      "quantidade": 1,
      "sensores": [
        {
          "timestamp": "2024-11-03T19:13:37",
          "tipoNome": "GPS",
          "valor": "39.73965392397057, -8.818502730615972"
        },
        {
          "timestamp": "2024-11-03T19:13:37",
          "tipoNome": "Pressão Atmosférica",
          "valor": "1000"
        }
      ]
    }
  ]
}
```

4. Um utilizador autenticado no Sistema de Apoio ao Cliente **pesquisa as encomendas por um determinado estado** através do protocolo HTTP, verbo **GET**, para o sítio: **/backend/api/sac/encomendas/estado/{estado}**

A resposta **devolvida** por este recurso retorna o seguinte formato JSON:

```
[
  {
    "data_entrega": "2024-11-01T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "PorEntregar",
    "id": 1,
    "username": "Bernardo"
  },
  {
    "data_entrega": "2024-11-03T15:30:00",
    "data_expedicao": "2024-11-01T15:30:00",
    "estado": "PorEntregar",
    "id": 2,
    "username": "Bernardo"
  }
  . . .
]
```

5. Um utilizador autenticado no Sistema de Apoio ao Cliente **vê os alertas de uma determinada encomenda** com um inconveniente, através do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/sac/encomendas/{id}/alertas

A resposta devolvida por este recurso seguinte formato JSON:

```
{
  "sensores": [
    {
      "alertas": [
        {
          "id": 1,
          "mensagem": "Valor acima do limite máximo (35) para o sensor Temperatura",
          "timeStamp": "2024-11-02T21:51:58.271305",
          "valor": "35"
        },
        {
          "id": 2,
          "mensagem": "Valor abaixo do limite mínimo (15) para o sensor Temperatura",
          "timeStamp": "2024-11-02T21:51:58.272265",
          "valor": "10"
        }
      ],
      "id": 1,
      "tipo": "Temperatura"
    }
  ]
}
```

6. Um utilizador autenticado no Sistema de Apoio ao Cliente **cancela uma encomenda** através do protocolo HTTP, verbo **PATCH**, para o sítio:
- /backend/api/sac/encomendas/{id}**

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "estado" : "cancelada"
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

7. Um utilizador autenticado no Sistema de Apoio ao Cliente, **verifica a última leitura dos sensores de um determinado tipo, das suas encomendas**, através do protocolo HTTP, verbo **GET**, para o sítio:
/backend/api/sac/encomendas/sensor/{tipo-sensor}

A resposta devolvida por este recurso seguinte formato JSON:

```
[
  {
    "bateria": 97,
    "estado": "ativo",
    "id": 2,
    "id_encomenda": 1,
    "id_volume": 2,
    "timeStamp": "2024-11-02T21:51:58",
    "tipoNome": "GPS",
    "valor": "40.7123, -74.12312"
  },
  {
    "bateria": 65,
    "estado": "ativo",
    "id": 5,
    "id_encomenda": 3,
    "id_volume": 1,
    "timeStamp": "2024-11-03T17:22:18",
    "tipoNome": "GPS",
    "valor": "12.7123, -75.12312"
  }
  . . .
]
```

Sistema de Logística

1. Um utilizador não autenticado no Sistema de Logística **efetua o login** através do protocolo HTTP, verbo **POST**, para o sítio:

/backend/api/sl/login

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "username" : "Bernardo"
  "password" : "12#45"
}
```

Após o login, em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

2. Um utilizador autenticado no Sistema de Logística **faz um pedido para obter todas as encomendas com um determinado estado** através do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/sl/encomendas/{estado}

A resposta devolvida por este recurso seguinte formato JSON:

```
[
  {
    "data_entrega": "2024-10-29T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "EmProcessamento",
    "id": 4,
    "username": "Bernardo"
  },
  {
    "data_entrega": "2023-10-29T15:30:00",
    "data_expedicao": "2023-10-29T15:30:00",
    "estado": "EmProcessamento",
    "id": 5,
    "username": "Ferreira"
  },
  . . .
]
```

3. Um utilizador autenticado no Sistema de Logística, **faz um pedido para ver os detalhes de uma encomenda**, através do protocolo HTTP, verbo **GET**, para o sítio:
- /backend/api/sl/encomendas/{id}**

A resposta **devolvida** por este recurso segue o formato JSON:

```
{
  "data_entrega": "2024-10-29T15:30:00",
  "data_expedicao": "2024-10-29T15:30:00",
  "estado": "PorEntregar",
  "id": 1,
  "username": "Bernardo",
  "volumes": [
    {
      "id": 1,
      "nome_produto": "Gelados",
      "quantidade": 5,
      "sensores": [
        {
          "bateria": 100,
          "estado": "ativo",
          "id": 1,
          "timeStamp": "2024-11-05T16:19:01",
          "tipoNome": "Temperatura",
          "valor": "29.0"
        },
        {
          "bateria": 100,
          "estado": "ativo",
          "id": 2,
          "timeStamp": "2024-11-05T16:19:01",
          "tipoNome": "GPS",
          "valor": "39.7344200469475, -8.821063143811228"
        }
      ]
    }
  ]
}
```

4. Um utilizador autenticado no Sistema de Logística **altera os estado de uma encomenda** através do protocolo HTTP, verbo **PATCH**, para o sítio:
- /backend/api/sl/encomendas/{id}**

O corpo do pedido **recebido** por este recurso segue o seguinte formato JSON:

```
{
  "estado": "entregue",
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

5. Um utilizador autenticado no Sistema de Logística **associa um sensor a um volume** através do protocolo HTTP, verbo **POST**, para o sítio:

/backend/api/sl/volume/{id}/sensor

O corpo do pedido **recebido** por este recurso segue o seguinte formato JSON:

```
{
  "valor": 90,
  "tipoId": 4,
  "estado": "ativo",
  "bateria": 72,
  "tipoNome": "GPS"
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **201 Created**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

6. Um utilizador autenticado no Sistema de Logística vai poder **ver os detalhes de um volume**, através da utilização do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/sl/volume/{id}

A resposta **devolvida** por este recurso seguinte formato JSON:

```
{
  "id": 1,
  "nome_produto": "Gelados",
  "quantidade": 5,
  "sensores": [
    {
      "bateria": 100,
      "estado": "ativo",
      "id": 1,
      "timeStamp": "2024-11-05T16:19:01.061509",
      "tipoNome": "Temperatura",
      "valor": "29.0"
    },
    {
      "bateria": 100,
      "estado": "ativo",
      "id": 2,
      "timeStamp": "2024-11-05T16:19:01.066674",
      "tipoNome": "GPS",
      "valor": "39.7344200469475, -8.821063143811228"
    },
    . . .
  ]
}
```

7. Um utilizador autenticado no Sistema de Logística, vai **associar um volume a uma encomenda**, através da utilização do protocolo HTTP, verbo **POST**, para o sítio:
/backend/api/sl/encomendas/{id}/volume

O corpo do pedido **recebido** por este recurso segue o seguinte formato JSON:

```
{
  "id_produto": 3,
  "quantidade": 10,
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **201 Created**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

8. Um utilizador autenticado no Sistema de Logística, vai **criar uma encomenda** através do protocolo HTTP, verbo **POST**, para o sítio:
/backend/api/sl/encomendas

O corpo do pedido **recebido** por este recurso segue o seguinte formato JSON:

```
{
  "username": "Bernardo",
  "estado": "PorEntregar",
  "data_expedicao": "2024-10-31T10:00:00",
  "data_entrega": "2024-11-05T10:00:00",
  "produtos": [
    {
      "id": 1,
      "nome": "Gelados",
      "quantidade_por_volume": 10
    },
    {
      "id": 2,
      "nome": "Banana",
      "quantidade_por_volume": 10
    },
    . . .
  ]
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **201 Created**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

9. No Sistema de Logística para a criação de uma encomenda, é necessário **recolher a informação de todos os cliente**, através da utilização do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/sl/clientes

A resposta **devolvida** por este recurso seguinte formato JSON:

```
[
  {
    "username": "Bernardo"
  },
  {
    "username": "Carvalho"
  },
  {
    "username": "Ferreira"
  },
  {
    "username": "Sousa"
  },
  {
    "username": "Tendeiro"
  },
  . . .
]
```

10. No Sistema de Logística para a criação de uma encomenda, é necessário **recolher a informação de todos os produtos**, através da utilização do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/sl/produtos

A resposta **devolvida** por este recurso seguinte formato JSON:

```
[
  {
    "categoria": "Alimentos",
    "id": 1,
    "nome": "Maçã",
    "quantidade_por_volume": 5
  },
  {
    "categoria": "Alimentos",
    "id": 3,
    "nome": "Gelados",
    "quantidade_por_volume": 15
  },
  {
    "categoria": "Tv e Som",
    "id": 6,
    "nome": "Fones",
    "quantidade_por_volume": 20
  },
  . . .
]
```

11. No Sistema de Logística para a associar um sensor a um volume, é necessário **saber os tipos de sensores**, através da utilização do protocolo HTTP, verbo **GET**, para o sítio:
/backend/api/sl/tipoSensores

A resposta **devolvida** por este recurso seguinte formato JSON:

```
[
  {
    "id": 1,
    "tipo": "Temperatura"
  },
  {
    "id": 2,
    "tipo": "Aceleração"
  },
  {
    "id": 3,
    "tipo": "Pressão Atmosférica"
  },
  {
    "id": 4,
    "tipo": "GPS"
  },
  . . .
]
```

Sistema Operacional

1. Um utilizador não autenticado no Sistema Operacional efetua o login através do protocolo HTTP, verbo **POST**, para o sítio:

/backend/api/so/login

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "username": "Bernardo",
  "password" : "ber123"
}
```

Após o login, em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

2. Um utilizador autenticado no Sistema Operacional, **faz um pedido para receber as encomendas com estado “por entregar” e “em processamento”**, através do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/so/encomendas/pendentes

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
[
  {
    "data_entrega": "2024-10-29T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "PorEntregar",
    "id": 1,
    "username": "Bernardo"
  },
  {
    "data_entrega": "2024-11-02T12:30:00",
    "data_expedicao": "2024-11-02T12:00:00",
    "estado": "PorEntregar",
    "id": 3,
    "username": "Sousa"
  },
  {
    "data_entrega": "2024-10-29T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "EmProcessamento",
    "id": 4,
    "username": "Bernardo"
  },
  . . .
]
```


3. Um utilizador autenticado no Sistema Operacional, **pesquisa as encomendas por um determinado estado** através do protocolo HTTP, verbo **GET**, para o sítio:
/backend/api/so/encomendas/estado/{estado}

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
[
  {
    "data_entrega": "2024-10-29T15:30:00",
    "data_expedicao": "2024-10-29T15:30:00",
    "estado": "PorEntregar",
    "id": 1,
    "username": "Bernardo"
  },
  {
    "data_entrega": "2024-11-02T12:30:00",
    "data_expedicao": "2024-11-02T12:00:00",
    "estado": "PorEntregar",
    "id": 3,
    "username": "Sousa"
  },
  . . .
]
```

4. Um utilizador autenticado no Sistema Operacional, **faz um pedido para cancelar uma encomenda**, através do protocolo HTTP, verbo **PATCH**, para o sítio:
/backend/api/so/encomendas/{id}

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "estado": "Cancelada"
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

5. Um utilizador autenticado no Sistema Operacional, **faz um pedido para obter os detalhes de uma encomenda específica**, através do protocolo HTTP, verbo **GET**, para o sítio:
/backend/api /so/encomendas/{id}/detalhes

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
{
  "data_entrega": "2024-10-29T15:30:00",
  "data_expedicao": "2024-10-29T15:30:00",
  "estado": "PorEntregar",
  "id": 1,
  "username": "Bernardo",
  "volumes": [
    {
      "id": 1,
      "nome_produto": "Gelados",
      "quantidade": 5,
      "sensores": [
        {
          "bateria": 100,
          "estado": "ativo",
          "id": 1,
          "timeStamp": "2024-11-07T11:46:52.705553",
          "tipoNome": "Temperatura",
          "valor": "29.0"
        },
        {
          "bateria": 100,
          "estado": "ativo",
          "id": 2,
          "timeStamp": "2024-11-07T11:46:52.710524",
          "tipoNome": "GPS",
          "valor": "39.7344200469475, -8.821063143811228"
        },
        . . .
      ]
    },
    . . .
  ]
}
```

6. Um utilizador autenticado no Sistema Operacional, **faz um pedido para visualizar o histórico de alertas de um sensor específico**, através do protocolo HTTP, verbo **GET**, para o sítio:
- `/backend/api/so/sensor/{id}/alertas`**

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
[
  {
    "data": "2024-11-03T13:59:55.296689",
    "id": 2,
    "id_encomenda": 1,
    "mensagem": "Valor abaixo do limite mínimo (15) para o sensor Temperatura",
    "valor": "10"
  },
  {
    "data": "2024-11-03T13:59:55.293978",
    "id": 1,
    "id_encomenda": 1,
    "mensagem": "Valor acima do limite máximo (35) para o sensor Temperatura",
    "valor": "35"
  },
  . . .
]
```

7. Um utilizador autenticado no Sistema Operacional, **faz um pedido para visualizar o histórico de alertas de uma encomenda**, através do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/so/encomendas/{id}/alertas

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
{
  "sensores": [
    {
      "alertas": [
        {
          "id": 2,
          "mensagem": "Valor abaixo do limite mínimo (15) para o sensor Temperatura",
          "timeStamp": "2024-11-03T14:53:03.706538",
          "valor": "10"
        },
        {
          "id": 1,
          "mensagem": "Valor acima do limite máximo (35) para o sensor Temperatura",
          "timeStamp": "2024-11-03T14:53:03.704523",
          "valor": "35"
        }
      ],
      "id": 1,
      "tipo": "Temperatura"
    },
    . . .
  ]
}
```

8. Um utilizador autenticado no Sistema Operacional, **faz um pedido para receber os alertas de todas as encomendas por entregar**, através do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/so/encomendas/alertas

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
[
  {
    "id": 5,
    "id_encomenda": 3,
    "id_sensor": 10,
    "mensagem": "Valor abaixo do limite mínimo (5) para o sensor Aceleração",
    "time_stamp": "2024-11-03T14:53:03.708794",
    "username": "Sousa",
    "valor": "3"
  },
  {
    "id": 2,
    "id_encomenda": 1,
    "id_sensor": 1,
    "mensagem": "Valor abaixo do limite mínimo (15) para o sensor Temperatura",
    "time_stamp": "2024-11-03T14:53:03.706538",
    "username": "Bernardo",
    "valor": "10"
  },
  . . .
]
```

9. Um utilizador autenticado no Sistema Operacional, **verifica a última leitura dos sensores de um determinado tipo de todas as encomendas**, através do protocolo HTTP, verbo **GET**, para o sítio:
- /backend/api/so/sensor/{tipo-sensor}**

A resposta devolvida por este recurso segue o seguinte formato JSON:

```
[
  {
    "bateria": 100,
    "estado": "ativo",
    "id": 1,
    "id_encomenda": 1,
    "id_volume": 1,
    "timeStamp": "2024-11-03T14:53:03.683891",
    "tipoNome": "Temperatura",
    "valor": "29.0"
  },
  {
    "bateria": 99,
    "estado": "ativo",
    "id": 9,
    "id_encomenda": 3,
    "id_volume": 5,
    "timeStamp": "2024-11-03T14:53:03.698726",
    "tipoNome": "Temperatura",
    "valor": "21.0"
  },
  . . .
]
```

10. Um utilizador autenticado no Sistema Operacional, **faz um pedido para receber as coordenadas de cada volume de uma encomenda**, através do protocolo HTTP, verbo **GET**, para o sítio:
- /backend/api/so/encomendas/{id}/coordenadas**

A resposta devolvida por este recurso segue o seguinte formato JSON

```
[
  {
    "coordenadas": "39.7344200469475, -8.821063143811228",
    "produtoNome": "Gelados",
    "volumeId": 1
  },
  {
    "coordenadas": "39.73965392397057, -8.818502730615972",
    "produtoNome": "Perfume",
    "volumeId": 2
  },
  . . .
]
```

Sistema de Sensores

1. O Sistema de Sensores terá acesso a **todos os sensores ativos**, através do protocolo HTTP, verbo **GET**, para o sítio:

/backend/api/sensor

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
[
  {
    "bateria": 100,
    "estado": "ativo",
    "id": 1,
    "idEncomenda": 1,
    "idVolume": 1,
    "timeStamp": "2024-11-03T11:46:27.032481",
    "tipoNome": "Temperatura",
    "valMax": 30,
    "valMin": 10,
    "valor": "29.0"
  },
  {
    "bateria": 100,
    "estado": "ativo",
    "id": 2,
    "idEncomenda": 1,
    "idVolume": 2,
    "timeStamp": "2024-11-03T11:46:27.039485",
    "tipoNome": "GPS",
    "valor": "40.7123, -74.12312"
  },
  . . .
]
```

2. O Sistema de Sensores poderá **alterar o valor de um sensor específico** através do protocolo HTTP, verbo **PATCH**, para o sítio:

/backend/api/sensor/{id}

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "valor": "26.5",
  "bateria": 72
}
```

A resposta **devolvida** por este recurso segue o seguinte formato JSON:

```
{
  "bateria": 72,
  "estado": "ativo",
  "id": 1,
  "timeStamp": "2024-11-03T11:55:58.54821523",
  "tipoNome": "Temperatura",
  "valor": "26.5"
}
```

3. O Sistema de Sensores, quando deteta um baixo nível de bateria num determinado sensor, **altera o seu estado para desativado** através do protocolo HTTP, verbo **PATCH** para o sítio:

/backend/api/sensor/{id}/desativa

O corpo do **pedido** recebido tem o seguinte formato JSON:

```
{
  "estado": "desativado",
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.

4. Um sensor quando deteta um inconveniente com uma encomenda, **cria um alerta** através do protocolo HTTP, verbo **POST** para o sítio:

/backend/api/sensor

O corpo o **pedido** recebido tem o seguinte formato JSON:

```
{
  "mensagem": "Temperatura elevada, comida a estragar-se",
  "id_sensor": 5,
  "valor": 40,
  "bateria": 72,
  "id_volume": 4
}
```

Em caso de **sucesso**, a resposta **devolvida** por este recurso retorna um código de status utilizado pelo HTTP:

Response: status **200 OK**

Em caso de **insucesso**, a resposta **devolvida** por este recurso retorna um código de status HTTP adequado, acompanhado por uma mensagem descritiva do erro ocorrido.