

# Programação de Sistemas Distribuídos

Trabalho Prático

MyPhotoAlbum.com

Máximo Paulo  
1090630@isep.ipp.pt

Rúben Barros  
1100667@isep.ipp.pt

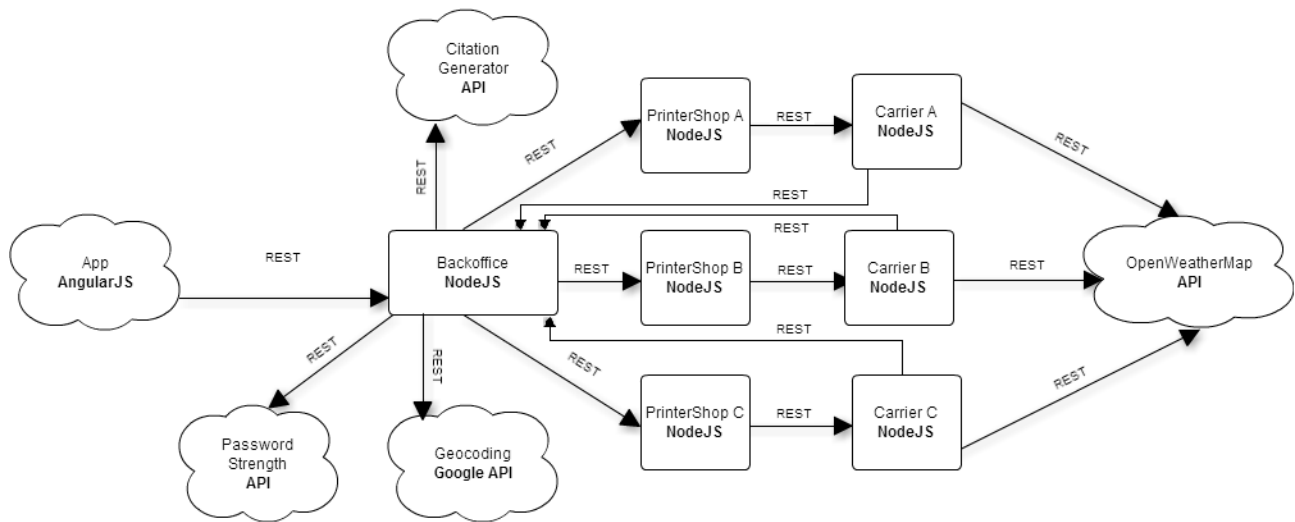
Mário Teixeira  
1090626@isep.ipp.pt



# Índice

1. Esquema do Sistema.....	3
2. Casos de Uso.....	4
2.1. Autenticar-se.....	4
2.2. Registrar-se.....	4
2.3. Criar álbum simples de fotografias.....	5
2.4. Carregar foto para um álbum.....	5
2.5. Ver os seus álbuns.....	5
2.6. Criar álbum para impressão.....	5
2.7. Consultar estado da encomenda.....	5
2.8. Encomendar impressão.....	6
2.9. Ver versão PDF dos álbuns.....	6
3. Diagramas de Sequência.....	7
3.1. Carregar foto para um álbum.....	7
3.2. Consultar estado da encomenda.....	8
3.3. Criar álbum para impressão.....	9
3.4. Fazer encomenda.....	10
3.5. Registo.....	10
4. Definição da API REST.....	12
4.1. Backoffice API.....	12
4.2. PrinterShop A API.....	13
4.3. PrinterShop B API.....	13
4.4. PrinterShop C API.....	13
4.5. Carrier A API.....	14
4.6. Carrier B API.....	14
4.7. Carrier C API.....	14
5. Request Bodies.....	15
6. Padrões Aplicados.....	20
7. Tecnologias utilizadas.....	21
8. Avaliação do Sistema segundo as características dos Sistemas Distribuídos.....	22
9. Divisão de processos / aplicações.....	23
10. Identificação das interfaces e contratos de cada componente do sistema.....	24

# 1. Esquema do Sistema



## 2. Casos de Uso



- **2.1. Autenticar-se**

O utilizador autentica-se perante o serviço.

- ***Pré-condições***

O utilizador tem de estar registado no sistema.

- ***Pós-condições***

Em caso de autenticação bem sucedida, é atribuída ao utilizador uma sessão.

- **2.2. Registar-se**

O utilizador regista-se no sistema.

- ***Pós-condições***

Em caso de sucesso, o utilizador fica registado no sistema e pode efetuar login.

- **2.3. Criar álbum simples de fotografias**

O utilizador pode criar um álbum simples para depois adicionar fotografias.

- ***Pré-condições***

- O utilizador tem de estar autenticado.

- **2.4. Carregar foto para um álbum**

O utilizador faz *upload* de uma nova foto e associa a um álbum criado.

- ***Pré-condições***

- O utilizador tem de estar autenticado;

- O álbum escolhido tem de existir.

- **2.5. Ver os seus álbuns**

O utilizador pode escolher um dos seus álbuns para ver as fotografias a ele associadas.

- ***Pré-condições***

- O utilizador tem de estar autenticado.

- **2.6. Criar álbum para impressão**

O utilizador pode criar um novo álbum para impressão.

- ***Pré-condições***

- O utilizador tem de estar autenticado;

- **2.7. Consultar estado da encomenda**

O utilizador pode saber o estado da sua encomenda.

- ***Pré-condições***

- O utilizador tem de estar autenticado;

- O utilizador teve de ter criado um álbum para impressão e ter associado fotografias.

- **2.8. Encomendar impressão**

O utilizador pode escolher um álbum para impressão e mandá-lo imprimir.

- ***Pré-condições***

- O utilizador tem de estar autenticado;

- O utilizador teve de ter criado um álbum para impressão e ter associado fotografias.

- **2.9. Ver versão PDF dos álbuns**

O utilizador pode visualizar a versão PDF de um álbum que mandou imprimir.

- ***Pré-condições***

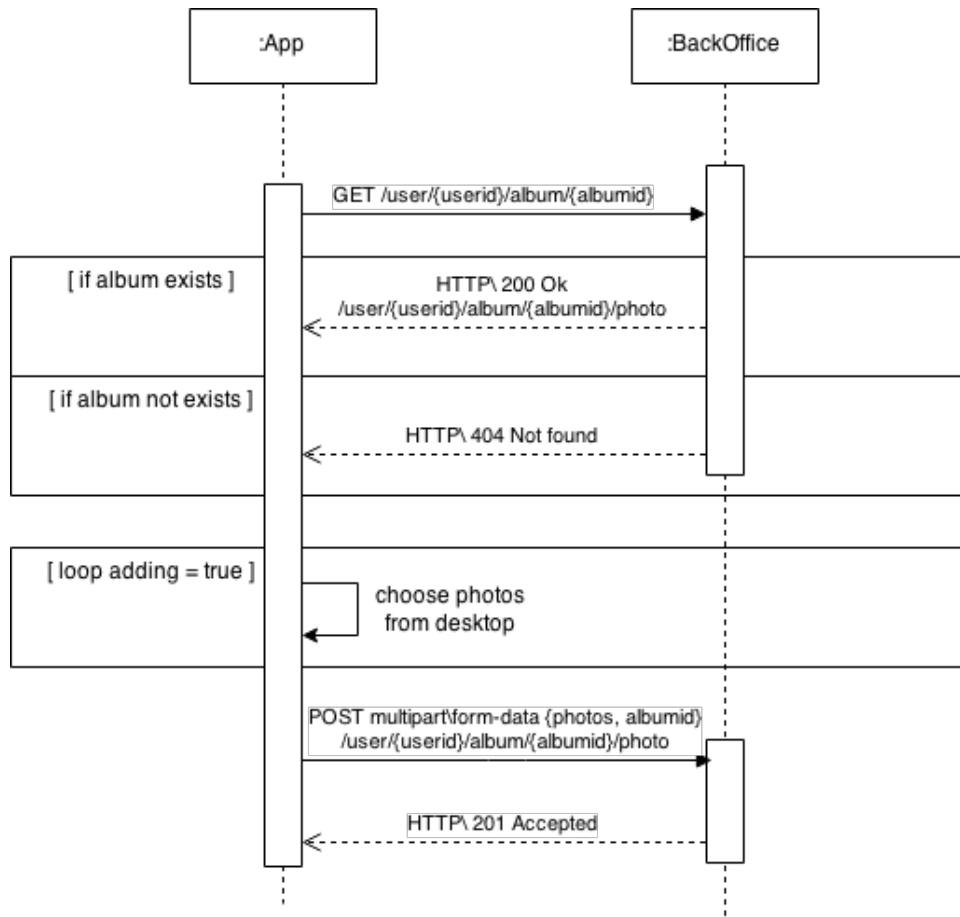
- O utilizador tem de estar autenticado;

- O utilizador teve de ter encomendado um álbum para impressão.

## 3. Diagramas de Sequência

### 3.1. Carregar foto para um álbum

O seguinte diagrama demonstra como é que um utilizador pode carregar novas fotos para um álbum previamente criado.



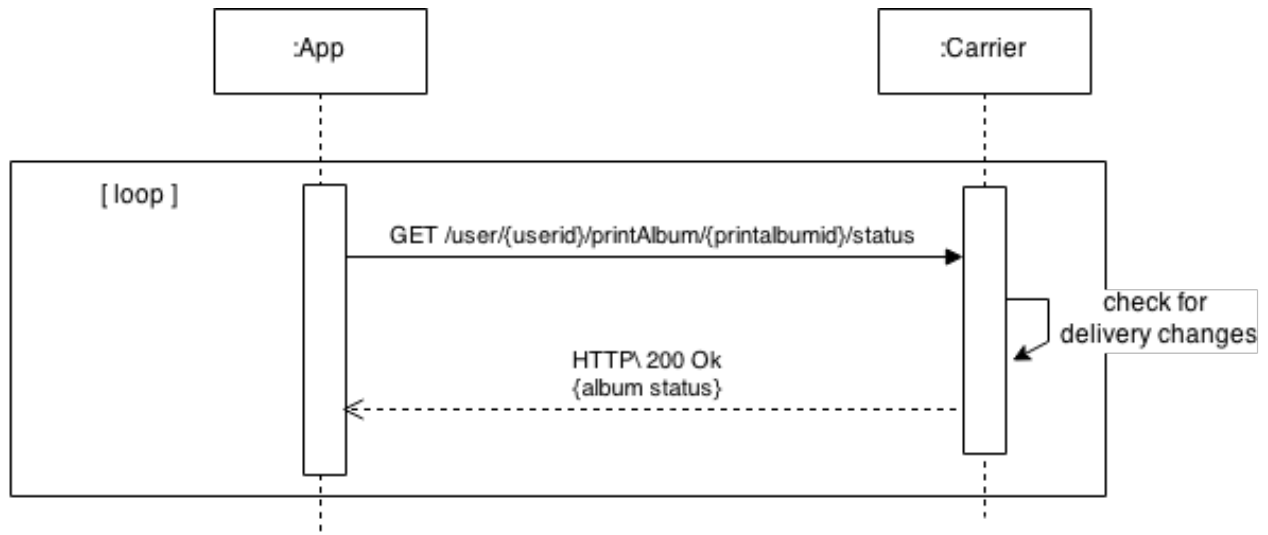
- **Pré-condições**

O utilizador tem de estar autenticado;

O utilizador tem de ter um álbum previamente criado para lhe adicionar fotos.

### 3.2. Consultar estado da encomenda

O seguinte diagrama representa o cenário de verificação de mudanças na encomenda



- **Pré-condições**

O utilizador tem de estar autenticado;

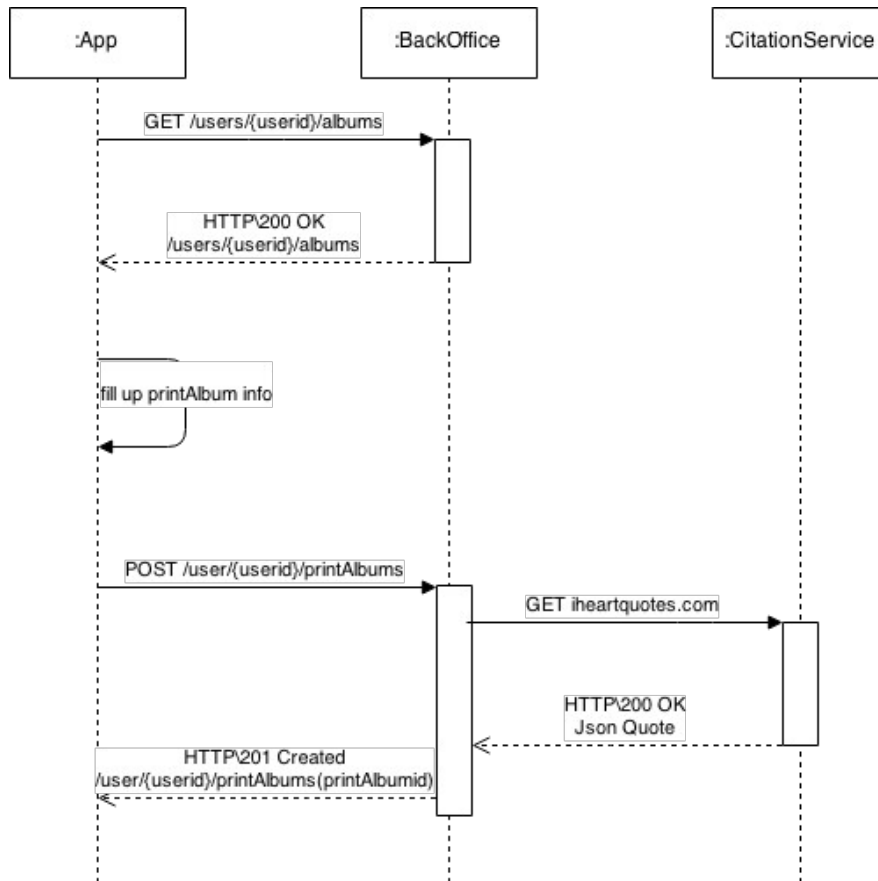
O utilizador teve de ter criado um álbum para impressão;

O utilizador teve de ter feito uma encomenda.



### 3.3. Criar álbum para impressão

O seguinte diagrama representa a sequência de passos que são necessários para descrever o processo de criação de um álbum para impressão.



- **Pré-condições**

O utilizador tem de estar autenticado;

O utilizador tem de ter álbuns normais criados.

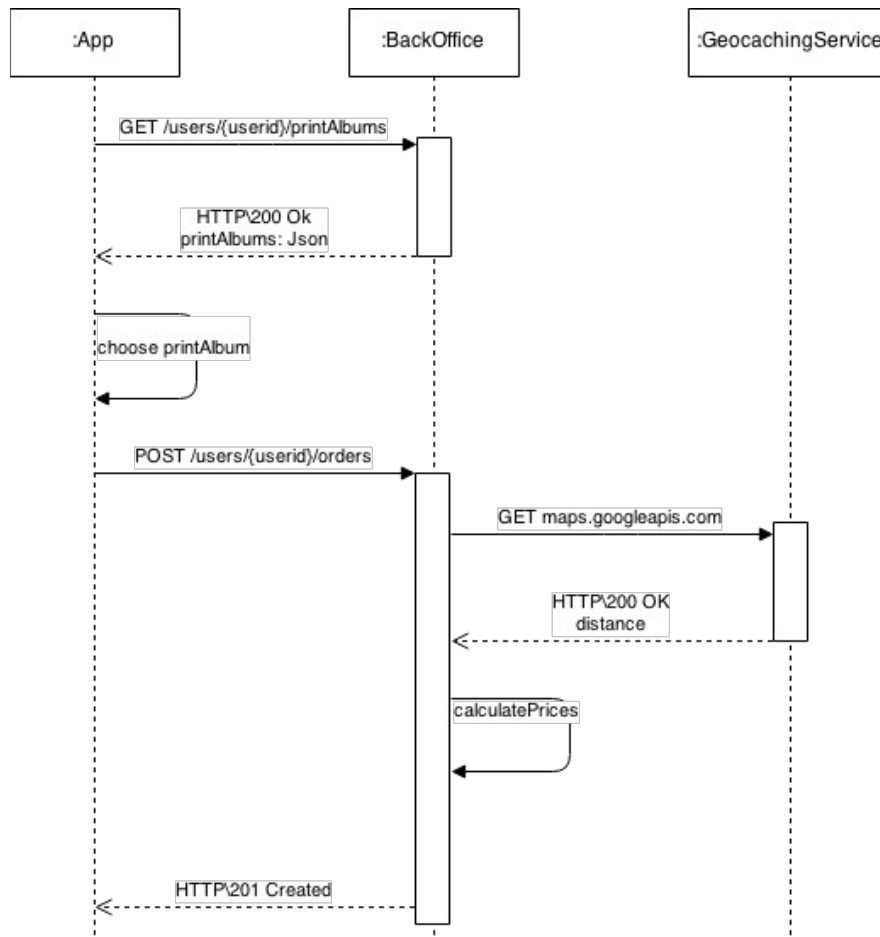
- **Pós-condições**

O álbum criado fica com o custo do álbum (sem custo de transporte);

O álbum criado fica pronto para impressão e para *download*.

### 3.4. Fazer encomenda

O seguinte diagrama demonstra os passos para a realização de uma encomenda.



- **Pré-condições**

O utilizador tem de estar autenticado;

O utilizador tem de ter álbum criados para impressão.

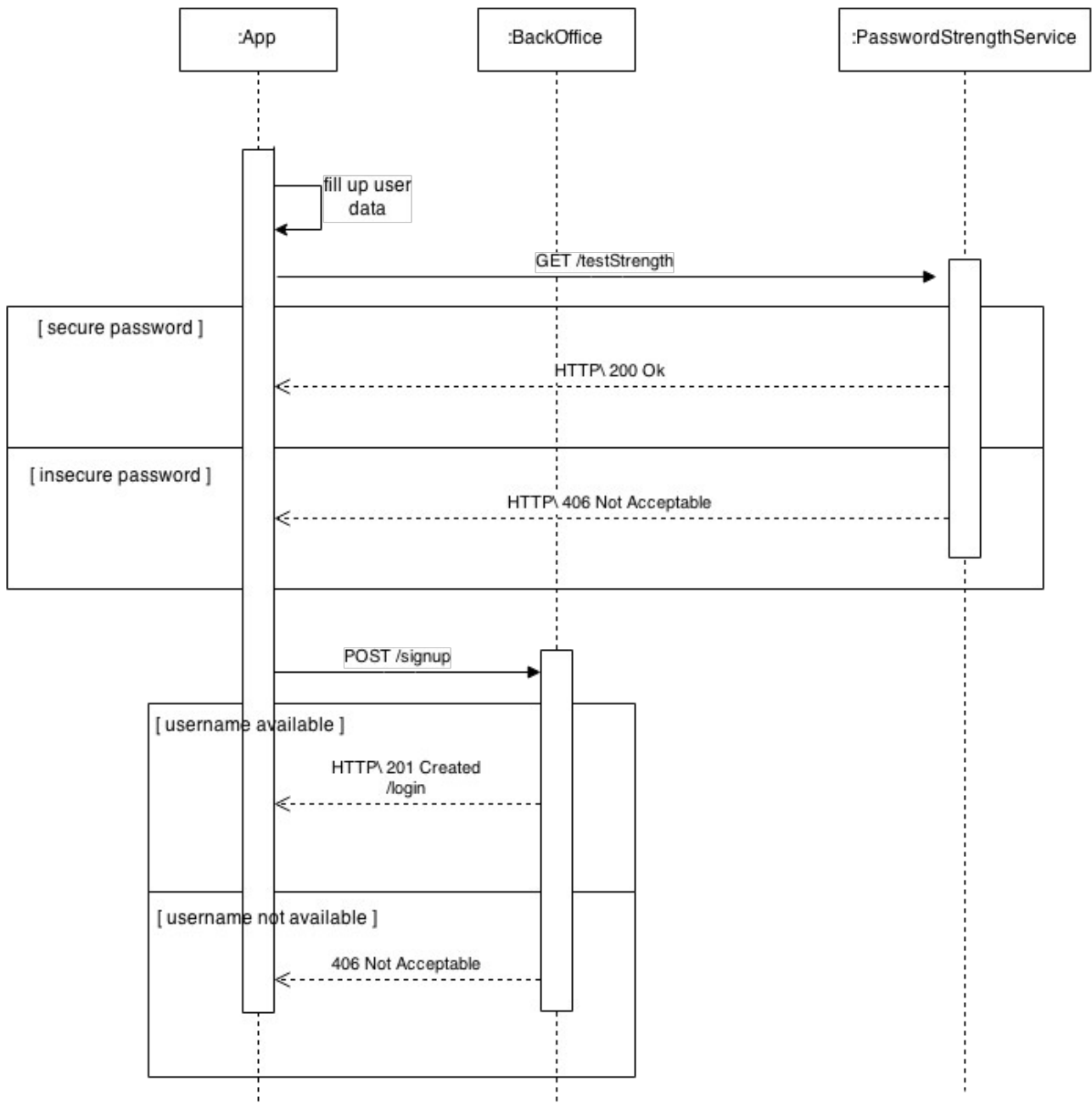
- **Pós-condições**

O álbum para impressão fica com o preço final (custo de impressão + transporte);

O álbum pode ser encomendado.

### 3.5. Registo

O seguinte diagrama demonstra como um utilizador se pode registar



- ***Pós-condições***

O utilizador fica registado no sistema.

## 4. Definição da API REST

### 4.1. Backoffice API

URI	Descrição da URI	Verbos	Descrição dos verbos	Códigos de estado
/signup	Registo de novos utilizadores	GET	-	405 – Não permitido
		POST	Registar utilizador	201 – Criado 406 – Não aceite
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/login	Autenticação de utilizadores	GET	-	405 – Não permitido
		POST	Autenticar utilizador	202 – Aceite 400 – Falha na autenticação
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users	Coleção de utilizadores	GET	Apresentar todos utilizadores	200 - OK
		POST	-	405 – Não permitido
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users/:id	Utilizador individual	GET	Obter informações de um utilizador	200 - OK
		POST	Atualizar senha do utilizador	200 – OK 204 – Utilizador não encontrado
		PUT	-	405 – Não permitido
		DELETE	Eliminar um utilizador	200 – OK 204 – Utilizador não encontrado
/users/:id/albums	Coleção de álbuns de um utilizador	GET	Retornar todos os álbuns	200 - OK
		POST	Criar um novo álbum	201 - Criado
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users/:uID/albums/:aID	Álbum individual	GET	Obter informações de um álbum	200 - OK
		POST	Atualizar um álbum	200 – OK 204 – Álbum não encontrado
		PUT	-	405 – Não permitido
		DELETE	Eliminar um álbum	200 – OK 204 – Álbum não encontrado
/users/:uID/albums/:aID/photos	Coleção de fotos de um álbum	GET	Retornar todas as fotos	200 - OK
		POST	Upload de uma foto	201 – Criado 500 – Erro no upload
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users/:uID/albums/:aID/photos/:pID	Foto individual	GET	Obter informações de um foto	200 - OK
		POST	Atualizar uma fotografia	200 – OK 204 – Foto não encontrada
		PUT	-	405 – Não permitido
		DELETE	Eliminar foto	200 – OK 204 – Foto não encontrada

/users/:uID/printAlbums	Coleção de PrintAlbums	GET	Obter todos os printalbums	200 - OK
		POST	Criar novo printAlbum	201 - Criado
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users/:uID/printAlbums/:pID	PrintAlbum individual	GET	Obter informações de um printAlbum	200 - OK
		POST	-	405 – Não permitido
		PUT	-	405 – Não permitido
		DELETE	Eliminar printAlbum	200 – OK 204 – PrintAlbum
/users/:uID/printAlbums/:pID/pdf	Transformação do álbum de impressão para formato PDF	GET	Obter álbum em formato PDF	200 - OK
		POST	-	405 – Não permitido
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users/:uID/orders	Coleção de todas as encomendas	GET	Obter todas as encomendas	200 - OK
		POST	Criar nova encomenda	201 - Criado
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido
/users/:uID/orders/:oID	Encomenda individual	GET	Obter informações de uma encomenda	200 - OK
		POST	Atualizar estado da encomenda	200 – OK 404 – Não encontrado 409 - Conflito
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

## 4.2. PrinterShop A API

/printerShop1	Interface da printer shop 1	GET	Retorna o preço individual de cada foto	200 - OK
		POST	Enviar álbum para impressão	200 - OK
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

## 4.3. PrinterShop B API

/printerShop2?nPhotos={num}	Interface da printer shop 2	GET	Retorna o preço de acordo com a quantidade de fotos	200 – OK
		POST	Enviar álbum para impressão	200 - OK
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

## 4.4. PrinterShop C API

/printerShop3/{nPhotos}	Interface da printer shop 3	GET	Retorna preço de acordo com o número de fotos	200 – OK
		POST	Enviar álbum para impressão	200 - OK
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

## 4.5. Carrier A API

/carrier1	Interface da carrier 1	GET	Retorna estado da encomenda	200 – OK
		POST	Enviar pedido de envio	200 – OK 406 – Não Aceite
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

## 4.6. Carrier B API

/carrier2	Interface da carrier 2	GET	Retorna estado da encomenda	200 – OK
		POST	Enviar pedido de envio	200 – OK 406 – Não Aceite
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

## 4.7. Carrier C API

/carrier3	Interface da carrier 3	GET	Retorna estado da encomenda	200 – OK
		POST	Enviar pedido de envio	200 – OK 406 – Não Aceite
		PUT	-	405 – Não permitido
		DELETE	-	405 – Não permitido

### Módulos auxiliares

- users-handler → Tratamento das informações relacionadas com utilizadores.
- albums-handler → Tratamento das informações relacionadas com os álbuns.
- photos-handler → Tratamento das informações relacionadas com as fotos.
- printAlbums-handler → Tratamento das informações relacionadas com os PrintAlbums.
- orders-handler → Tratamento das informações relacionadas com as encomendas.
- geolocation-handler → obter coordenadas.
- processOrder-handler → Processamento das encomendas.

### Serviços auxiliares:

- PasswordStrengthService → Verificar a força da password inserida pelo utilizador.
- getCitation → Obter citação do website iheartquotes.com

## 5. Request Bodies

URI	Method	Request body	Response Body
/signup	POST	{ "username": "rica", "password": "1234" }	
/login	POST	{ "username": "rica", "password": "1234" }	
/users	GET		[{ "id": 1, "username": "rica", "password": "1234" }, { "id": 2, "username": "pedro", "password": "1234" }]
/users/:uID	GET		{ "id": 1, "username": "rica", "password": "123456" }
/users/:uID	PUT	{ "username": "ricx", "password": "1234" }	{ "id": 1, "username": "ricx", "password": "1234" }
/users/:uID	DELETE		
/users/:uID/albums	GET		[{ "id": 1, "title": "Vida", "description": "Exemplo de descrição", "startDate": 1418066023321, "endDate": 1418066023321, "owner": 1, "photos": [ { "id": 1, "albumId": 1, "photo": "photos/41kjb2h4bkb.jpg", "date": 1418066023321 }, { "id": 2, "albumId": 1, "photo": "photos/d89sad798sd.jpg", "date": 1418066023321 } ] }]
/users/:uID/albums	POST	{ "title": "Vida", "description": "Exemplo de descrição", }	{ "id": 1, "title": "Vida", "description": "Exemplo de descrição", }

		<pre>"startDate": 1418066023321, "endDate": 1418066023321, "owner": 1 }</pre>	<pre>"startDate": 1418066023321, "endDate": 1418066023321, "owner": 1, "photos": [] }</pre>
/users/:uID/albums/:aID	GET		<pre>{   "id": 1,   "title": "Vida",   "description": "Exemplo de descrição",   "startDate": 1418066023321,   "endDate": 1418066023321,   "owner": 1,   "photos": [     {       "id": 1,       "albumId": 1,       "photo": "photos/41kjb2h4bkb.jpg",       "date": 1418066023321     },     {       "id": 2,       "albumId": 1,       "photo": "photos/d89sad798sd.jpg",       "date": 1418066023321     }   ] }</pre>
/users/:uID/albums/:aID	PUT	<pre>{   "title": "Vida",   "description": "Outra descrição",   "startDate": 1418066023321,   "endDate": 1418066023321,   "owner": 1 }</pre>	<pre>{   "id": 1,   "title": "Vida",   "description": "Outra descrição",   "startDate": 1418066023321,   "endDate": 1418066023321,   "owner": 1,   "photos": [     {       "id": 1,       "albumId": 1,       "photo": "photos/41kjb2h4bkb.jpg",       "date": 1418066023321     },     {       "id": 2,       "albumId": 1,       "photo": "photos/d89sad798sd.jpg",       "date": 1418066023321     }   ] }</pre>
/users/:uID/albums/:aID	DELETE		
/users/:uID/albums/:aID/photo	GET		<pre>[   {     "id": 1,     "albumId": 1,     "photo": "photos/41kjb2h4bkb.jpg",     "date": 1418066023321   },   {     "id": 2,     "albumId": 1, </pre>



			<pre>"photo": "photos/d89sad798sd.jpg", "date": 1418066023321 } ]</pre>
/users/:uID/albums/:aID/photos	POST	<pre>{   "albumId": 1,   "photo": "photos/d89sad798sd.jpg",   "date": 1418066023321 }</pre>	<pre>{   "id": 1 }</pre>
/users/:uID/albums/:aID/photos/:pID	GET		<pre>{   "id": 1,   "albumId": 1,   "photo": "photos/41kjb2h4bkb.jpg",   "date": 1418066023321 }</pre>
/users/:uID/albums/:aID/photos/:pID	DELETE		
/users/:uID/printAlbums	POST	<pre>{   "theme": "Férias",   "title": "Melhores das férias",   "message": "I will get a Triforce, like I said",   "photos": [     {       "id": 1,       "albumId": 1,       "photo": "photos/41kjb2h4bkb.jpg",       "date": 1418066023321     },     {       "id": 2,       "albumId": 1,       "photo": "photos/d89sad798sd.jpg",       "date": 1418066023321     }   ] }</pre>	<pre>{   "id": 1,   "theme": "Férias",   "title": "Melhores das férias",   "message": "I will get a Triforce, like I said",   "photos": [     {       "id": 1,       "albumId": 1,       "photo": "photos/41kjb2h4bkb.jpg",       "date": 1418066023321     },     {       "id": 2,       "albumId": 1,       "photo": "photos/d89sad798sd.jpg",       "date": 1418066023321     }   ] }</pre>
/users/:uID/orders/:oID	GET		<pre>{   "id": 1,   "printPrice": 29.75,   "transportPrice": 5.25,   "address": "Avenida da Liberdade",   "printAlbum": {     "id": 1,     "theme": "Férias",     "title": "Melhores das férias",     "message": "I will get a Triforce, like I said",     "photos": [       {         "id": 1,         "albumId": 1,         "photo": "photos/41kjb2h4bkb.jpg",         "date": 1418066023321       }, </pre>

			<pre> {   "id": 2,   "albumId": 1,   "photo": "photos/d89sad798sd.jpg",   "date": 1418066023321 } ] }, "confirmed": true, "state": "Delivered to carrier" } </pre>
/users/:uID/orders/	POST	<pre> {   "printPrice": 29.75,   "transportPrice": 5.25,   "address": "Avenida da Liberdade",   "printAlbum": 1,   "confirmed": true,   "state": "Delivered to carrier" } </pre>	<pre> {   "id": 1,   "printPrice": 29.75,   "transportPrice": 5.25,   "address": "Avenida da Liberdade",   "printAlbum": {     "id": 1,     "theme": "Férias",     "title": "Melhores das férias",     "message": "I will get a Triforce, like I said",     "photos": [       {         "id": 1,         "albumId": 1,         "photo": "photos/41kjb2h4bkb.jpg",         "date": 1418066023321       },       {         "id": 2,         "albumId": 1,         "photo": "photos/d89sad798sd.jpg",         "date": 1418066023321       }     ]   },   "confirmed": true,   "state": "Delivered to carrier" } </pre>
/printerShop1	GET		<pre> {   "individualPrice": 0.59 } </pre>
/printerShop2?nPhotos=X	GET		<pre> {   "price": X * 0.59 } </pre>
/printerShop3	GET		<pre> {   "estimatedPrice": 27.59 } </pre>
/printerShop1 /printerShop2 /printerShop3	POST	<pre> {   "id": 1,   "theme": "Férias",   "title": "Melhores das férias",   "message": "I will get a Triforce, like I said",   "photos": [     {       "id": 1,       "albumId": 1, </pre>	<pre> {   "error": false } </pre>

		<pre> "photo": "photos/41kjb2h4bkb.jpg", "date": 1418066023321 }, {   "id": 2,   "albumId": 1,   "photo": "photos/d89sad798sd.jpg", "date": 1418066023321 } ] } </pre>	
/carrier1 /carrier2 /carrier3	GET		<pre> {   "id": 1,   "state": "In transport" } </pre>
/carrier1 /carrier2 /carrier3	POST	<pre> {   "id": 1,   "address": "Avenida da Liberdade",   "printAlbum": {     "id": 1,     "theme": "Férias",     "title": "Melhores das férias",     "message": "I will get a Triforce, like I said",     "photos": [       {         "id": 1,         "albumId": 1,         "photo": "photos/41kjb2h4bkb.jpg", "date": 1418066023321       },       {         "id": 2,         "albumId": 1,         "photo": "photos/d89sad798sd.jpg", "date": 1418066023321       }     ]   } } </pre>	<pre> {   "error": false } </pre>
/testStrength	POST	<pre> {   "password": 123456 } </pre>	<pre> {   "strength": 1 } </pre>
/distance	GET		<pre> {   "distance": 145 } </pre>

## 6. Padrões Aplicados

Padrão	Aonde aplicável
Remote Façade	Com este padrão podemos facilitar a obtenção de, por exemplo, álbuns integrando as fotografias no documento que será enviado para a App.
Data Transport Object	Este padrão será usado em conjunto com o Remote Façade para ir buscar a informação a diferentes coleções através de apenas um pedido.
Service Layer	Este padrão vai ser implementado para definir as operações que a user interface pode aceder.
Document Processor	Tanto no backoffice, como no PrinterShop, no Carrier e na App serão utilizados contratos de documentos, em vez de contratos do tipo RPC.
Idempotent Operation	Este padrão vai ser implementado para garantir que as operações de criação, eliminação e edição de álbuns, fotografias e etc tenham sempre o mesmo resultado.
Web/Worker Roles	Em toda o backoffice, a aceitação de pedidos delegará os mesmo para um "worker" que tratará os mesmos.

## 7. Tecnologias utilizadas

<b>Sistema Operativo</b>	Distribuições Linux baseadas em Ubuntu
<b>Application Server</b>	AngularJS Framework Grunt Bower
<b>Web Server</b>	NodeJS Express HTTP Body-Parser SQLite3
<b>Linguagens de Programação</b>	AngularJS NodeJS
<b>API's Externas</b>	Google Geocoding API I <3 Quotes API

## 8. Avaliação do Sistema segundo as características dos Sistemas Distribuídos

- **Transparência:** O sistema é visto como um todo e não como uma coleção de componentes, algumas das transparências existentes são, de localização, em que os recursos são acedidos sem nunca ser dado a sua localização, transparência de falhas, em caso de ocorra alguma falha, não seja indicado como foi ultrapassada.
- **Segurança:** O sistema possui alguma segurança, sendo esta a proteção contra o acesso de conteúdo dos utilizadores.
- **Tolerância a falhas:** O sistema em face de falhas esta preparado para as ocultar de um utilizador.
- **Protocolos de interação:** Cada componentes do sistema é executado em protocolos distintos mas permitindo a comunicação entre cada uma delas.
- **Escalabilidade:** Face à não possibilidade de testar a aplicação em grande escala, não podemos garantir a constante velocidade nas respostas do sistema, mas na existência de um proxy e a base de dados esteja acessível, é possível implementar o backoffice em diversas máquinas devido ao facto que o estado não é armazenado em nenhum deles.
- **Heterogeneidade:** O sistema desenvolvido possui a características de heterogeneidade, visto que, as comunicações entre os várias interfaces (aplicação, backoffice, printerShops, carriers, etc.) é realizada entre diferentes linguagens de programação (AngularJS, NodeJS), outra justificação é o funcionamento das interfaces em Sistemas Operativos diferentes.

## **9. Divisão de processos / aplicações**

Durante o desenvolvimento do backoffice, foi feita uma divisão em diversos handlers, cada um para um componente específico (utilizadores, álbuns, encomendas, etc.) para permitir uma organização melhor e permitir a alteração dos componentes sem ser necessário alterar o núcleo do backoffice. Estes handlers contêm todas as funções necessárias para a interação com a base de dados existente. Para emular um ambiente real, foi criados diferentes API, um para cada printerShop e para cada Carrier, de forma a demonstrar a comunicação entre os vários módulos.

## 10. Identificação das interfaces e contratos de cada componente do sistema

- **App:** No caso da App existem diversos contratos com o BackOffice, um para cada entidade existente no iPhoto.
- **BackOffice:** O BackOffice para além do contrato acima referido, tem também contratos para as comunicações com as Printer shops, quando são enviados os albuns com a fotografias a serem impressas, e com os Carriers, onde será feita uma busca para recolher a informação sobre o estado da encomenda.
- **Printer Shop:** Além do contrato para a comunicação com o BackOffice, o Printer shop tem um contrato para comunicar com o seu carrier para enviar a informação sobre a encomenda.
- **Carrier:** Finalmente, o Carrier tem o contrato para receber a informação do Printer shop, e ainda outro para responder ao BackOffice sobre o estado da encomenda.