AZORID

Guia prático de integração OpenID

Versão 1.0, 2026-09-11

GOVERNO DOS AÇORES
SECRETARIA REGIONAL DAS FINANÇAS, PLANEAMENTO E ADMINISTRAÇÃO PÚBLICA

datapoint .software

PRR Plano de Recuperação e Resiliência
GOVERNO DOS AÇORES
REPÚBLICA PORTUGUESA
Financiado pela União Europeia NextGenerationEU

MODERNIZAÇÃO E DIGITALIZAÇÃO DA ADMINISTRAÇÃO PÚBLICA-RAA C19

Índice

1. Revisões . . . 1
2. Introdução . . . 2
    2.1. Âmbito . . . 2
    2.2. Pré-requisitos . . . 2
3. Visão geral do fluxo . . . 3
4. Descoberta de configuração . . . 4
    4.1. Pedido de exemplo . . . 4
    4.2. Corpo da resposta de exemplo . . . 4
    4.3. Utilização prática . . . 4
5. Pedido de autorização . . . 5
    5.1. Parâmetros do pedido . . . 5
    5.2. Pedido de exemplo . . . 5
    5.3. Resposta de sucesso . . . 5
    5.4. Resposta de erro . . . 6
6. Obtenção de tokens . . . 7
    6.1. Autenticação do cliente . . . 7
    6.2. Pedido de exemplo . . . 7
    6.3. Corpo da resposta de exemplo . . . 7
    6.4. Estrutura do token . . . 8
    6.5. Respostas de erro . . . 9
7. Consulta de dados do utilizador . . . 11
    7.1. Pedido de exemplo . . . 11
    7.2. Corpo da resposta de exemplo . . . 11
    7.3. Respostas de erro . . . 12
    7.4. Fotografia ou logótipo (scope `picture`) . . . 12
8. Processamento de dados: contas de organização e representação . . . 14
    8.1. Quando é que isto se aplica . . . 14
    8.2. O atributo `impersonator` . . . 14
    8.3. Porque é que o sistema cliente deve processar este atributo . . . 15
    8.4. O atributo `act` . . . 16
9. Logout . . . 17
    9.1. Parâmetros do pedido . . . 17
    9.2. Pedido de exemplo . . . 17
    9.3. Resposta . . . 17
    9.4. Respostas de erro . . . 17
10. Backchannel Logout . . . 19
    10.1. Expor um endpoint de logout . . . 19
    10.2. Pedido enviado pelo AzorID . . . 19
    10.3. Processamento pelo sistema cliente . . . 20
    10.4. Resposta esperada . . . 20
11. Notificação de alterações de identidade (Shared Signals Framework) . . . 21

11.1. Expor um endpoint de receção . . . 21
11.2. Descoberta de configuração SSF . . . 21
11.3. Pedido enviado pelo AzorID . . . 22
11.4. Processamento pelo sistema cliente . . . 23
11.5. Resposta esperada . . . 23
12. Renovação de tokens . . . 24
12.1. Parâmetros do pedido . . . 24
12.2. Pedido de exemplo . . . 24
12.3. Corpo da resposta de exemplo . . . 24
12.4. Rotação do refresh_token . . . 25
12.5. Utilização prática: obter dados atualizados via userinfo . . . 25
12.6. Respostas de erro . . . 25
13. Referências . . . 27

1. Revisões

Data	Versão	Descrição
2026-09-11	DRAFT	Revisão inicial do documento.
2026-09-22	1.0	Documentação do mecanismo de notificação de alterações de identidade (Shared Signals Framework).

2. Introdução

2.1. Âmbito

Este documento é um guia técnico e prático que complementa o "Protocolo e Especificação Técnica", focando-se exclusivamente na construção dos pedidos HTTP e na interpretação das respostas necessárias para implementar o fluxo de autenticação OpenID Connect com o AzorID.

Este guia *não* aborda as regras de negócio da plataforma, o significado de cada *scope* ou a estrutura dos dados devolvidos — esses temas encontram-se detalhados no "Protocolo e Especificação Técnica", que deve ser consultado em conjunto com este documento.

2.2. Pré-requisitos

- Conhecimentos básicos de HTTP (métodos, cabeçalhos, códigos de estado) e JSON;
- `Client ID` e `Client Secret` obtidos através do processo de adesão descrito no "Protocolo e Especificação Técnica";
- Um ou mais URLs de redirecionamento (`redirect_uri`) previamente registados junto da equipa técnica do AzorID.

NOTA Os exemplos apresentados ao longo deste documento utilizam o ambiente de qualidade, com o endereço base `https://azorid-qa.azores.gov.pt`. Para produção, deve ser utilizado o endereço base `https://azorid.azores.gov.pt`.

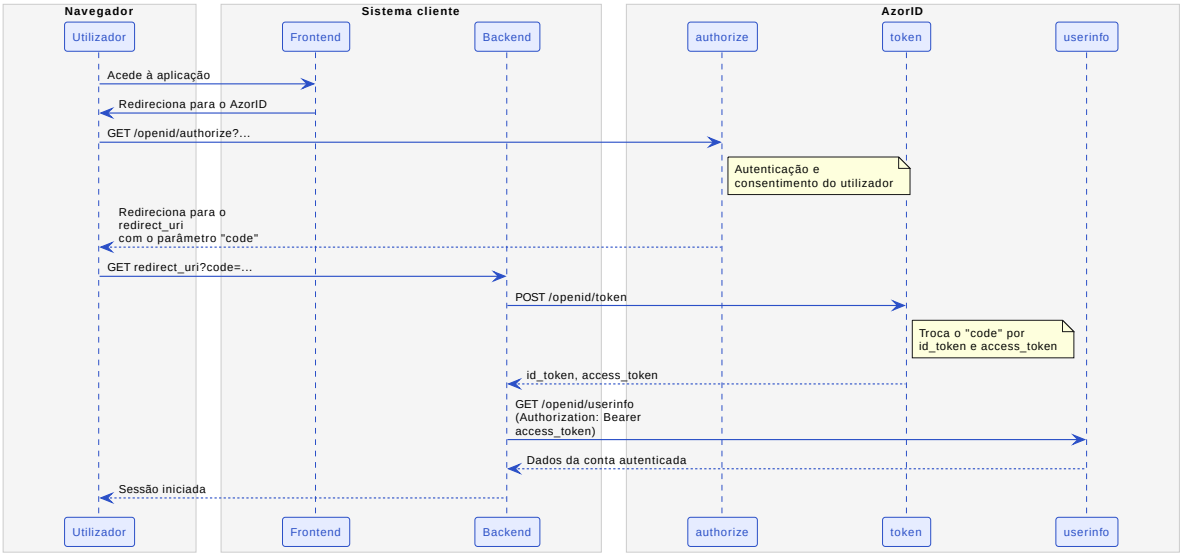
NOTA Os valores de exemplo `YOUR_CLIENT_ID`, `YOUR_CLIENT_SECRET`, `YOUR_ACCESS_TOKEN`, etc. devem ser substituídos pelos valores reais atribuídos ao sistema cliente ou obtidos em cada etapa do fluxo.

NOTA Neste guia são mencionados termos como o "cliente" ou "sistema cliente". Considera-se cliente o sistema que está a integrar com o AzorID para consumir as informações de autenticação, também designado como plataforma aderente.

3. Visão geral do fluxo

O fluxo "Authorization Code Flow" — recomendado para a generalidade das integrações — envolve as seguintes etapas, cada uma detalhada num capítulo próprio deste guia:

1. **Descoberta de configuração**: o sistema cliente obtém os endereços dos restantes endpoints a partir do endpoint de configuração;
2. **Pedido de autorização**: o navegador do utilizador é redirecionado para o AzorID, onde este se autentica e autoriza o acesso;
3. **Obtenção de tokens**: o servidor do sistema cliente troca o código de autorização recebido por tokens de acesso;
4. **Consulta de dados do utilizador**: opcionalmente, o sistema cliente consulta o endpoint `userinfo` para obter dados;
5. **Logout**: o sistema cliente termina a sessão do utilizador.



4. Descoberta de configuração

O primeiro passo de qualquer integração é obter a configuração do AzorID através do endpoint de descoberta, em conformidade com a especificação [OpenID]. Esta configuração indica os restantes endpoints a utilizar, pelo que deve ser o ponto de partida da integração, em vez de assumir diretamente os endereços dos restantes endpoints.

4.1. Pedido de exemplo

```
GET /.well-known/openid-configuration HTTP/1.1
Host: azorid-qa.azores.gov.pt
```

4.2. Corpo da resposta de exemplo

```
{
  "issuer": "https://azorid-qa.azores.gov.pt",
  "authorization_endpoint": "https://azorid-qa.azores.gov.pt/openid/authorize",
  "jwks_uri": "https://azorid-qa.azores.gov.pt/openid/jwks",
  "token_endpoint": "https://azorid-qa.azores.gov.pt/openid/token",
  "userinfo_endpoint": "https://azorid-qa.azores.gov.pt/openid/userinfo",
  "end_session_endpoint": "https://azorid-qa.azores.gov.pt/openid/logout",
  "scopes_supported": [ "openid", "profile", "email", "..." ], ①
  (...)
}
```

① A lista completa de *scopes* suportados está descrita no "Protocolo e Especificação Técnica".

4.3. Utilização prática

Numa integração real, este pedido deve ser feito uma única vez (por exemplo, no arranque da aplicação, ou em cache com invalidação periódica), e os valores de `authorization_endpoint`, `token_endpoint`, `userinfo_endpoint`, `end_session_endpoint` e `jwks_uri` devem ser guardados para uso nos passos seguintes, em vez de serem escritos diretamente no código do sistema cliente.

NOTA

A maioria das bibliotecas cliente de OpenID Connect trata este passo automaticamente, bastando indicar o `issuer` (também designado por `Authority`) como configuração.

5. Pedido de autorização

Para iniciar a autenticação, o navegador do utilizador deve ser redirecionado (por exemplo, através de um `HTTP 302` ou de um `<a href="…">`) para o `authorization_endpoint`, junto com os parâmetros do pedido, incluídos na *query string*.

5.1. Parâmetros do pedido

Parâmetro	Obrigatório	Descrição
`client_id`	Sim	Identificador do sistema cliente, atribuído durante o processo de adesão.
`redirect_uri`	Sim	URL para o qual o utilizador é reencaminhado após a autenticação. Deve corresponder exatamente a um dos URLs registados para o cliente.
`response_type`	Sim	Fixo em `code`, indicando que se pretende utilizar o fluxo "Authorization Code Flow".
`scope`	Sim	Lista de *scopes* pedidos, separados por espaço (por exemplo, `openid profile name email`). O *scope* `openid` é sempre obrigatório.
`state`	Recomendado	Valor arbitrário gerado pelo cliente, devolvido inalterado no reencaminhamento final. Utilizado para associar a resposta ao pedido original e mitigar ataques CSRF.

5.2. Pedido de exemplo

Redirecionar o navegador do utilizador para `https://azorid-qa.azores.gov.pt/openid/authorize?client_id=YOUR_CLIENT_ID&redirect_uri=https%3A%2F%2Fclient.example.com%2Fcallback&response_type=code&scope=openid%20profile%20name%20email&state=xyzABC123`, com as devidas substituições dos valores `client_id`, `redirect_uri`, `scope` e `state` pelos valores reais.

5.3. Resposta de sucesso

Após o utilizador se autenticar e autorizar o pedido, o AzorID reencaminha o navegador de volta para o `redirect_uri` indicado, incluindo o código de autorização e o valor de `state` original:

Na prática, se o redirect URL definido é `https://client.example.com/callback`, então o navegador do utilizador será redirecionado para `https://client.example.com/callback?code=AUTHORIZATION_CODE&state=xyzABC123`.

Cabe ao sistema cliente processar os atributos entregues (`code`, `state`).

O sistema cliente deve validar que o valor de `state` recebido corresponde ao valor originalmente gerado, e extrair o valor de `code` para o passo seguinte (Obtenção de tokens).

5.4. Resposta de erro

Caso ocorra um erro que possa ser comunicado ao cliente (por exemplo, um *scope* inválido), o utilizador é igualmente reencaminhado para o `redirect_uri`, mas com os parâmetros `error` e, opcionalmente, `error_description`, em conformidade com a especificação [RFC6749]:

Exemplo: `https://client.example.com/callback?error=invalid_scope&error_description=One+or+more+requested+scopes+are+invalid&state=xyzABC123`

NOTA

Alguns erros — nomeadamente `client_id` ou `redirect_uri` inválidos — não podem ser comunicados de forma segura ao `redirect_uri` (uma vez que este ainda não foi validado), pelo que resultam numa página de erro apresentada diretamente pelo AzorID, em vez de um reencaminhamento.

6. Obtenção de tokens

Havendo um código de autorização (`code`) obtido no passo anterior, o servidor do sistema cliente deve trocá-lo por um `id_token` e um `access_token`, através de um pedido `HTTP POST` ao `token_endpoint`.

IMPORTANTE	Este pedido deve ser realizado pelo servidor (*backend*) do sistema cliente, e **nunca a partir do navegador do utilizador**, uma vez que envolve o `client_secret`, que deve permanecer **confidencial**.

6.1. Autenticação do cliente

O `client_id` e o `client_secret` podem ser transmitidos de duas formas, à escolha do sistema cliente:

- incluídos diretamente no corpo do pedido (`client_secret_post`), como no exemplo abaixo; ou
- através do cabeçalho HTTP `Authorization`, no formato `Basic base64(client_id:client_secret)` (`client_secret_basic`).

6.2. Pedido de exemplo

Iniciado pelo servidor do sistema cliente e direcionado ao AzorID.

```
POST /openid/token HTTP/1.1
Host: azorid-qa.azores.gov.pt
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&code=AUTHORIZATION_CODE
&redirect_uri=https%3A%2F%2Fclient.example.com%2Fcallback
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET
```

O `redirect_uri` deve ser igual ao enviado no pedido de autorização (Pedido de autorização); um valor diferente é rejeitado com `invalid_grant`. Para já, um pedido sem `redirect_uri` continua a ser aceite, mas o parâmetro passará a ser obrigatório numa versão futura, pelo que deve ser sempre enviado.

6.3. Corpo da resposta de exemplo

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 28800,
  "scope": "openid profile email",
  "refresh_token": "YOUR_REFRESH_TOKEN",
```

```
  "refresh_token_expires_in": 2592000
}
```

NOTA O `refresh_token` permite obter um novo conjunto de tokens sem reencaminhar novamente o utilizador — ver Renovação de tokens.

6.4. Estrutura do token

O `id_token` é um *JSON Web Token* (JWT) [RFC7519], assinado com o algoritmo RS256. Um JWT é composto por três partes separadas por `.` (cabeçalho, corpo e assinatura), cada uma codificada em Base64URL. A descodificação do cabeçalho e do corpo (sem validação da assinatura, apenas para leitura) pode ser feita com qualquer biblioteca de JWT, ou manualmente através de um descodificador Base64.

O corpo (*payload*) do token contém, entre outros, os seguintes atributos:

```
{
  "iss": "https://azorid-qa.azores.gov.pt",
  "aud": "YOUR_CLIENT_ID",
  "exp": 1735732800,
  "iat": 1735704000,
  "jti": "3f6a1e2b-2f0a-4b1d-9e3e-1a2b3c4d5e6f",
  "sid": "7c9e6d4a-8b1f-4e2a-9c3d-5f6a7b8c9d0e",
  "sub": "00000000-0000-0000-0000-000000000001",
  "display_name": "João Silva",
  "account_type": "citizen",
  "account_number": "123456",
  "identity_certified": true,
  "name": "João Silva",
  "fiscal_number": "100000001",
  "email": "joao.silva@example.com",
  "certification_state": {
    "name": { "status": "CERTIFIED" },
    "fiscal_number": { "status": "CERTIFIED" }
  }
}
```

NOTA `account_number` é o número de conta AzorID (6 dígitos), o mesmo que o utilizador vê na sua área pessoal. Está sempre presente, tanto para contas de cidadão como para contas de organização (`account_type` igual a `organization`), e nunca é partilhado entre contas.

NOTA O conjunto exato de atributos presentes no token depende dos *scopes* pedidos no passo de autorização (Pedido de autorização) — ver o "Protocolo e Especificação Técnica" para o significado de cada *scope* e atributo.

IMPORTANTE O sistema cliente deve guardar os valores de `sub` e `sid` associados à sessão local que cria a partir deste token — estes identificadores serão necessários mais adiante, nomeadamente para processar notificações de *Backchannel Logout* (Backchannel

Logout).

NOTA

`sid` identifica esta autorização especificamente (mantém-se inalterado ao longo de toda a sessão, incluindo após renovações via `refresh_token` — ver Renovação de tokens), enquanto `sub` identifica a conta. Caso o sistema cliente crie mais do que uma sessão local para a mesma conta (por exemplo, sessões concorrentes em dispositivos diferentes), `sid` é o identificador que permite distinguir qual delas corresponde a este token especificamente. `jti`, por contraste, identifica apenas esta emissão do token em concreto — muda a cada renovação — e não deve ser usado para este efeito.

IMPORTANTE

A descodificação do token, por si só, *não* constitui uma validação da sua autenticidade. O sistema cliente pode adicionalmente validar a assinatura e as declarações (*claims*) `iss`, `aud`, `exp` e, quando aplicável, `nonce`.

6.5. Respostas de erro

Todas as respostas do `token_endpoint`, de sucesso ou de erro, incluem o cabeçalho `Cache-Control: no-store`, e não devem ser guardadas em *cache*.

Em caso de erro, o endpoint responde com o código HTTP e o corpo definidos na especificação [RFC6749] (secção 5.2): um objeto JSON com o código do erro em `error` e uma descrição, em inglês, em `error_description`.

`error`	**Código HTTP**	**Quando**
`invalid_request`	`400`	Falta um parâmetro obrigatório (`grant_type`, `code` ou `refresh_token`).
`invalid_client`	`401`	O `client_id` não existe, o sistema cliente está inativo, ou o `client_secret` está errado. Se as credenciais foram enviadas no cabeçalho `Authorization` (`client_secret_basic`), a resposta inclui também o cabeçalho `WWW-Authenticate: Basic`.
`invalid_grant`	`400`	O `code` é inválido, já foi utilizado ou pertence a uma autorização revogada; o `redirect_uri` é diferente do enviado no pedido de autorização; ou o `refresh_token` é inválido, expirou, já foi utilizado ou pertence a uma autorização revogada (Renovação de tokens).
`unsupported_grant_type`	`400`	O `grant_type` não é `authorization_code`, `refresh_token` nem `client_credentials`.

```
HTTP/1.1 400 Bad Request
Content-Type: application/json; charset=utf-8
Cache-Control: no-store

{
  "error": "invalid_grant",
  "error_description": "The authorization code is invalid, already used or revoked."
```

```
}
```

NOTA

O texto de `error_description` destina-se a quem desenvolve o sistema cliente e pode mudar; o tratamento de erros deve basear-se apenas no valor de `error`.

7. Consulta de dados do utilizador

O endpoint `userinfo` permite obter os dados da conta autenticada associada a um `access_token` válido.

É particularmente útil para obter os dados completos associados a uma conta, uma vez que não é garantido que todos os dados dos scopes pedidos serão incluídos no `id_token` por questões de dimensão/comunicações, bem como confirmar a validade do token junto do AzorID, ou para obter a versão mais atualizada dos dados, incluindo metadados de certificação mais detalhados.

NOTA Alguns atributos, como a lista de gestores de uma entidade (scope `managers`), são devolvidos **exclusivamente** por este *endpoint* e nunca incluídos no `id_token` nem no `access_token` — ver o "Protocolo e Especificação Técnica".

7.1. Pedido de exemplo

```
GET /openid/userinfo HTTP/1.1
Host: azorid-qa.azores.gov.pt
Authorization: Bearer YOUR_ACCESS_TOKEN
```

NOTA O endpoint aceita tanto `GET` como `POST`. O `access_token` deve ser sempre transmitido através do cabeçalho `Authorization`, no formato `Bearer {token}`.

7.2. Corpo da resposta de exemplo


```
{
  "sub": "00000000-0000-0000-0000-000000000001",
  "display_name": "João Silva",
  "account_type": "citizen",
  "account_number": "123456",
  "identity_certified": true,
  "name": "João Silva",
  "fiscal_number": "100000001",
  "birth_date": "1990-01-01",
  "email": "joao.silva@example.com",
  "certification_state": {
    "name": {
      "status": "CERTIFIED",
      "certified_on": "2024-01-01T12:00:00Z",
      "certified_by": "Entidade Exemplo"
    },
    "fiscal_number": {
      "status": "CERTIFIED",
      "certified_on": "2024-01-01T12:00:00Z",
      "certified_by": "Entidade Exemplo"
    },
    "birth_date": {
```

```
      "status": "UNCERTIFIED"
    }
  }
}
```

7.3. Respostas de erro

Caso o `access_token` não seja enviado, esteja mal formado, tenha expirado, ou já não seja o token de acesso mais recente para a respetiva sessão, o endpoint devolve `HTTP 401 Unauthorized`, sem corpo, com o cabeçalho `WWW-Authenticate` definido na especificação [RFC6750]: `Bearer error="invalid_token"` quando foi enviado um token que não é aceite, ou apenas `Bearer` quando não foi enviado nenhum.

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token"
```

7.4. Fotografia ou logótipo (scope `picture`)

Com o scope `picture`, o atributo `picture` contém o endereço da fotografia do cidadão ou do logótipo da entidade, por exemplo `https://azorid.azores.gov.pt/openid/pictures/00000000-0000-0000-0000-000000000002`. O atributo só é enviado quando a conta tem uma imagem aprovada.

A imagem é obtida com um pedido `GET` a esse endereço, com o `access_token` no cabeçalho `Authorization`, tal como no `userinfo`:

```
GET /openid/pictures/00000000-0000-0000-0000-000000000002 HTTP/1.1
Host: azorid-qa.azores.gov.pt
Authorization: Bearer YOUR_ACCESS_TOKEN
```

A resposta é a imagem: quadrada, com 512 × 512 píxeis, em `image/jpeg` (fotografias) ou `image/png` (logótipos).

IMPORTANTE

Como o endereço exige o `access_token`, não pode ser usado diretamente numa página (por exemplo, num elemento `<img>`), e o `access_token` nunca deve ser enviado ao navegador. O sistema cliente obtém a imagem no seu servidor e serve-a às suas páginas a partir do seu próprio endereço. O `access_token` só deve ser enviado para endereços do AzorID.

Resposta	Quando
`200 OK`	A imagem é a atual da conta e o scope `picture` foi concedido.
`404 Not Found`	A imagem já não é a atual da conta (foi substituída ou removida), pertence a outra conta, ou o scope `picture` não foi concedido.
`401 Unauthorized`	O `access_token` não foi enviado, não é aceite, ou a autorização foi revogada. O cabeçalho `WWW-Authenticate` segue as mesmas regras do `userinfo`.

O endereço muda sempre que a imagem muda, pelo que o sistema cliente pode guardar a imagem associada ao endereço (a resposta inclui `Cache-Control: private, max-age=31536000, immutable`). Quando a imagem é aprovada, substituída ou removida, o AzorID envia a notificação `attribute-changed` com `picture` em `attributes` (ver "Notificação de alterações de identidade"); o sistema cliente deve então obter de novo os dados da conta. Uma resposta `404` significa que a imagem guardada já não deve ser apresentada.

8. Processamento de dados: contas de organização e representação

Os capítulos anteriores mostraram como obter os atributos da conta autenticada, quer através do `id_token` (Obtenção de tokens), quer através do endpoint `userinfo` (Consulta de dados do utilizador). Este capítulo detalha um cuidado adicional que o sistema cliente deve ter ao processar esses atributos: distinguir a conta a que os dados pertencem (`sub`) de quem está efetivamente a operar a sessão, sempre que estas não sejam a mesma pessoa.

8.1. Quando é que isto se aplica

Isto acontece em dois cenários, ambos descritos em maior detalhe no "Protocolo e Especificação Técnica":

- **Contas de organização**: uma organização nunca se autentica diretamente — só é possível aceder a uma conta de organização através de um cidadão afiliado a essa organização, ou que a represente. Ou seja, **qualquer** sistema cliente que aceite `account_type.organization` deve estar preparado para este cenário em todos os pedidos, sem exceção.
- **Representação entre cidadãos**: um cidadão (representante) pode autenticar-se em nome de outro cidadão (representado), caso o sistema cliente permita representação e exista uma relação de representação ativa entre ambos.

Em qualquer um destes casos, os atributos principais do token (`sub`, `name`, `fiscal_number`, etc.) referem-se sempre à conta **representada** (a organização, ou o cidadão em nome de quem se está a agir) — nunca à pessoa que efetivamente iniciou sessão. Essa distinção só fica visível através do atributo `impersonator`.

8.2. O atributo `impersonator`

Sempre que a pessoa autenticada (o representante) é diferente da conta representada, o token e a resposta do `userinfo` incluem o atributo `impersonator`, com os dados do representante:

```
{
  "sub": "00000000-0000-0000-0000-000000000001",
  "name": "João Silva",
  "fiscal_number": "100000001",
  "account_type": "citizen",
  "impersonator": {
    "sub": "00000000-0000-0000-0000-000000000002",
    "name": "Maria Santos",
    "fiscal_number": "100000002",
    "services": [
      "SVC-CODE-1",
      "SVC-CODE-2"
    ]
  }
}
```

Neste exemplo, a conta autenticada (`sub`) é a de João Silva — poderia ser tanto um cidadão representado como uma organização — mas quem efetivamente iniciou sessão e está a operar a aplicação é a Maria Santos, indicada em `impersonator`.

Campo	Descrição
`impersonator.sub`	Identificador do representante — distinto do `sub` de topo, que identifica a conta representada.
`impersonator.name` / `impersonator.fiscal_number`	Identificação do representante.
`impersonator.account_number`	Número de conta AzorID do representante.
`impersonator.passport_number`	Número de passaporte do representante. Presente apenas quando o representante não possui NIF (por exemplo, cidadãos estrangeiros), em substituição de `impersonator.fiscal_number`.
`impersonator.services`	Lista de códigos de serviço (do Catálogo de Entidades e Serviços) ao abrigo dos quais o representante está autorizado a agir em nome da conta representada.
`impersonator.has_full_access`	Presente, com o valor `true`, apenas quando a conta representada é uma organização e o representante é um gestor afiliado a essa organização (e não um representante ao abrigo de uma representação). Nesse caso, o representante tem acesso total à organização. Omitido em todos os outros casos.

NOTA

A ausência do atributo `impersonator` indica que a própria conta autenticada (`sub`) iniciou sessão diretamente — o que nunca acontece para contas de organização, mas é o caso comum para contas de cidadão.

8.3. Porque é que o sistema cliente deve processar este atributo

Tratar `sub` como "quem está a usar a aplicação" quando na realidade existe um `impersonator` leva a problemas concretos:

- **Auditoria e responsabilização**: se o sistema cliente regista quem realizou uma determinada ação (por exemplo, quem submeteu um formulário), deve registar o representante (`impersonator.sub`), não apenas a conta representada — de outra forma, torna-se impossível determinar mais tarde qual a pessoa física responsável por uma ação realizada em nome de uma organização ou de outro cidadão.
- **Transparência para o utilizador**: a interface deve deixar claro que se está a atuar em nome de terceiros (por exemplo, "Maria Santos, em representação de João Silva" ou "Maria Santos, em representação de Datapoint, Lda."), em vez de apresentar a sessão como se fosse o próprio João Silva ou a própria organização a navegar na aplicação.
- **Verificações adicionais específicas do sistema cliente**: caso o sistema cliente distinga internamente entre diferentes funcionalidades ou áreas (por exemplo, mais do que um dos seus próprios serviços

partilha o mesmo ponto de atendimento), `impersonator.services` permite determinar ao abrigo de que serviço o representante está a agir, e ajustar o comportamento da aplicação em conformidade.

IMPORTANTE

A validação de que o representante tem, de facto, permissão para representar a conta indicada em `sub` é sempre feita pelo AzorID antes de emitir o token — o sistema cliente não necessita (nem deve) reimplementar essa validação. O que fica a cargo do sistema cliente é processar corretamente a distinção entre `sub` e `impersonator` nos seus próprios registos e interface, conforme descrito acima.

NOTA

É imposta a regra, a nível do sistema AzorID, de que qualquer utilizador representante tem obrigatoriamente de possuír uma conta certificada, pelo que o nome e o NIF (ou, na sua ausência, o número de passaporte) presentes nos campos `impersonator` podem ser confiados pelo sistema cliente. Uma conta é considerada certificada quando possui nome completo certificado e, pelo menos, um NIF ou um número de passaporte certificados — pelo que o sistema cliente não deve assumir que `impersonator.fiscal_number` está sempre presente.

8.4. O atributo `act`

Além do atributo `impersonator` (próprio do AzorID), o `id_token`, o `access_token` e a resposta do endpoint `userinfo` incluem, nos mesmos cenários, o atributo `act`, em conformidade com a especificação [RFC8693]:

```
{
  "sub": "00000000-0000-0000-0000-000000000001",
  "act": {
    "sub": "00000000-0000-0000-0000-000000000002"
  },
  "impersonator": {
    "sub": "00000000-0000-0000-0000-000000000002",
    "name": "Maria Santos",
    "fiscal_number": "100000002",
    "services": [ "SVC-CODE-1", "SVC-CODE-2" ]
  }
}
```

`act.sub` identifica sempre o mesmo representante que `impersonator.sub`, mas segue exclusivamente o formato standard da [RFC8693] — sem os dados adicionais (`name`, `fiscal_number`, `services`) que só `impersonator` disponibiliza. Sistemas clientes que já tenham suporte genérico para "actor claims" de outros fornecedores OpenID Connect podem usar `act` diretamente; caso contrário, `impersonator` continua a ser a fonte de dados mais completa.

NOTA

Tal como `impersonator`, o atributo `act` está presente em todos os casos — `id_token`, `access_token` e resposta do endpoint `userinfo`.

9. Logout

Para terminar a sessão do utilizador, o navegador deve ser redirecionado para o `end_session_endpoint`, indicando o `id_token` obtido anteriormente e o URL para o qual o utilizador deve ser reencaminhado após o processo.

9.1. Parâmetros do pedido

Parâmetro	Obrigatório	Descrição
`id_token_hint`	Sim	O `id_token` obtido no passo de obtenção de tokens (Obtenção de tokens), utilizado para identificar a sessão e o cliente associados.
`post_logout_redirect_uri`	Sim	URL para o qual o utilizador deve ser reencaminhado após o logout. Deve corresponder a um dos URLs de pós-logout registados para o cliente.
`state`	Não	Valor arbitrário devolvido inalterado no reencaminhamento final.

9.2. Pedido de exemplo

Redirecionar o navegador do utilizador para `https://azorid-qa.azores.gov.pt/openid/logout?id_token_hint=YOUR_ID_TOKEN&post_logout_redirect_uri=https%3A%2F%2Fclient.example.com%2F&state=xyzABC123`, com as devidas substituições dos valores `id_token_hint`, `post_logout_redirect_uri` e `state` pelos valores reais.

9.3. Resposta

Caso não exista uma sessão ativa do utilizador no AzorID, o navegador é imediatamente reencaminhado para o `post_logout_redirect_uri` indicado:

Caso exista uma sessão ativa, é primeiro apresentada uma página onde o utilizador escolhe se pretende terminar sessão apenas no sistema cliente, ou também no AzorID — só depois dessa escolha ocorre o reencaminhamento final.

Na prática, se o Post-Logout Redirect URL definido é `https://client.example.com/logout_callback`, então o navegador do utilizador será redirecionado para `https://client.example.com/logout_callback?state=xyzABC123`.

9.4. Respostas de erro

Quando o pedido não pode ser aceite, o navegador não é reencaminhado e o endpoint responde, sem corpo, com:

Código HTTP	Quando
`400` Bad Request	Falta o `id_token_hint` ou o `post_logout_redirect_uri`, o `post_logout_redirect_uri` não é um URL válido, ou não é um dos URLs de pós-logout registados para o cliente.
`401` Unauthorized	O `id_token_hint` não é um `id_token` válido emitido pelo AzorID, ou a sessão que identifica já não existe.

10. Backchannel Logout

A partir do sistema AzorID o utilizador poderá pedir para terminar sessão de forma centralizada em *todos* os sistemas clientes onde tenha sessão ativa.

O AzorID concretiza esse pedido através do mecanismo de *Backchannel Logout* previsto no standard [Openid]: ao terminar a sessão do utilizador, envia diretamente, servidor-a-servidor, uma notificação a cada um dos restantes sistemas clientes com sessão ativa para esse utilizador, para que cada um possa encerrar a respetiva sessão local de forma independente.

10.1. Expor um endpoint de logout

Para suportar este mecanismo, o sistema cliente deve expor um endpoint HTTP próprio (o `backchannel_logout_uri`), que aceite pedidos `POST` do AzorID e que deve ser registado junto da equipa técnica do AzorID durante o processo de adesão, à semelhança do `redirect_uri`.

IMPORTANTE	Por se tratar de um endpoint potencialmente exposto, devem ser tomadas medidas para garantir que **apenas pedidos legítimos do AzorID são processados**. Preferencialmente, validar a assinatura do `logout_token` através das chaves públicas disponibilizadas no `jwks_uri` (ver Processamento pelo sistema cliente).

10.2. Pedido enviado pelo AzorID

Iniciado pelo AzorID e direcionado ao servidor do sistema cliente.

```
POST /backchannel-logout HTTP/1.1
Host: client.example.com
Content-Type: application/x-www-form-urlencoded

logout_token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...
```

O `logout_token` é um JWT assinado (estrutura idêntica à do `id_token`, ver Obtenção de tokens), cujo corpo (*payload*) tem o seguinte formato:

```
{
  "iss": "https://azorid-qa.azores.gov.pt",
  "aud": "YOUR_CLIENT_ID",
  "iat": 1735704000,
  "jti": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "sid": "7c9e6d4a-8b1f-4e2a-9c3d-5f6a7b8c9d0e",
  "sub": "00000000-0000-0000-0000-000000000001",
  "events": {
    "http://schemas.openid.net/event/backchannel-logout": {}
  }
}
```

Claim	Descrição
`sub`	Identificador da conta cuja sessão foi terminada.
`sid`	Identificador da autorização/sessão específica que foi terminada — o mesmo valor de `sid` presente no `id_token`/`access_token` originalmente emitidos para essa sessão (Obtenção de tokens).
`events`	Claim fixa que identifica o token como um `logout_token`, em conformidade com a especificação [OpenIDBackchannelLogout].

10.3. Processamento pelo sistema cliente

Ao receber este pedido, o sistema cliente deve:

1. Validar a assinatura do `logout_token`;
2. Confirmar a presença da claim `events` com o valor indicado acima;
3. Terminar, no seu próprio sistema, a sessão local previamente associada ao `sid` recebido — **sem reencaminhar nem depender do utilizador**.

IMPORTANTE

`sid` está sempre presente e identifica **uma única** sessão local — é o identificador que deve ser utilizado para localizar qual sessão terminar, precisamente porque o sistema cliente pode ter mais do que uma sessão local ativa para a mesma conta (`sub`), nomeadamente quando essa conta é acedida através de representação por mais do que um representante, ou a partir de mais do que um dispositivo/sessão em simultâneo. Terminar todas as sessões associadas a `sub`, ignorando `sid`, pode encerrar sessões que não deveriam ser afetadas por esta notificação.

10.4. Resposta esperada

Em caso de sucesso, o sistema cliente deve responder com `HTTP 200`, sem corpo:

```
HTTP/1.1 200 OK
```

Em caso de erro (por exemplo, `logout_token` inválido), deve ser devolvido um código de erro HTTP apropriado (`400` para um token mal formado ou inválido), em conformidade com a especificação [OpenIDBackchannelLogout].

11. Notificação de alterações de identidade (Shared Signals Framework)

O AzorID permite ainda notificar, de forma proativa, os sistemas clientes sempre que ocorra uma alteração aos dados de identidade de uma conta — nomeadamente ao nome (`name`), ao identificador fiscal (`fiscal_number`) ou ao estado de certificação global (`identity_certified`).

Este mecanismo é concretizado através do perfil de entrega por *push* do standard [OpenIDSSF]: ao ocorrer uma das alterações mencionadas, é enviado diretamente, servidor-a-servidor, um *Security Event Token* (SET, ver [RFC8417]) a cada sistema cliente subscrito.

NOTA	Apesar de arquitetura semelhante à do *Backchannel Logout* (Backchannel Logout) — um pedido `HTTP POST` direto ao sistema cliente — este é um mecanismo independente e distinto, com endpoint, formato de mensagem e finalidade próprios. Não devem ser confundidos.

11.1. Expor um endpoint de receção

Para suportar este mecanismo, o sistema cliente deve expor um endpoint HTTP próprio (o endpoint de receção SSF), que aceite pedidos `POST` do AzorID e que deve ser registado junto da equipa técnica do AzorID durante o processo de adesão, à semelhança do `backchannel_logout_uri`.

IMPORTANTE	Por se tratar de um endpoint potencialmente exposto, devem ser tomadas medidas para garantir que **apenas pedidos legítimos do AzorID são processados**. Preferencialmente, validar a assinatura do SET através das chaves públicas disponibilizadas no `jwks_uri` — as mesmas utilizadas para validar `id_token`/`access_token`/`logout_token` (ver Processamento pelo sistema cliente).

11.2. Descoberta de configuração SSF

De forma independente do endpoint de configuração OpenID Connect (Descoberta de configuração), o AzorID expõe um endpoint de descoberta próprio para este mecanismo.

```
GET /.well-known/ssf-configuration HTTP/1.1
Host: azorid-qa.azores.gov.pt
```

```
{
  "spec_version": "1.0-ID2",
  "issuer": "https://azorid-qa.azores.gov.pt",
  "jwks_uri": "https://azorid-qa.azores.gov.pt/openid/jwks",
  "delivery_methods_supported": [ "urn:ietf:rfc:8935" ]
```

```
}
```

NOTA

`jwks_uri` aponta para o mesmo conjunto de chaves já utilizado para validar os restantes tokens emitidos pelo AzorID — não existe um conjunto de chaves separado para este mecanismo.

11.3. Pedido enviado pelo AzorID

Iniciado pelo AzorID e direcionado ao servidor do sistema cliente. Ao contrário do `logout_token` (que é enviado como um campo de formulário `application/x-www-form-urlencoded`), o corpo do pedido é o próprio SET, assinado, enviado em bruto:

```
POST /ssf-receiver HTTP/1.1
Host: client.example.com
Content-Type: application/secevent+jwt

eyJhbGciOiJSUzI1NiIsInR5cCI6InNlY2V2ZW50K2p3dCJ9...
```

O corpo (*payload*) do SET tem o seguinte formato:

```
{
  "iss": "https://azorid-qa.azores.gov.pt",
  "aud": "YOUR_CLIENT_ID",
  "iat": 1735707600,
  "jti": "b2c3d4e5-f6a7-8901-bcde-f12345678901",
  "sub_id": {
    "format": "iss_sub",
    "iss": "https://azorid-qa.azores.gov.pt",
    "sub": "00000000-0000-0000-0000-000000000001"
  },
  "events": {
    "https://azorid.azores.gov.pt/schemas/ssf/attribute-changed": {
      "subject_type": "citizen",
      "attributes": [ "fiscal_number" ]
    }
  }
}
```

Claim	Descrição
`sub_id`	Identifica a conta a que a alteração diz respeito, num formato estruturado (`format: "iss_sub"`) em vez de uma claim `sub` simples — `sub_id.sub` corresponde ao mesmo identificador de conta presente nos restantes tokens.
`events`	Estrutura cuja única chave identifica o tipo de evento ocorrido (ver tabela abaixo) — o valor associado contém os detalhes específicos desse tipo de evento.

Existem dois tipos de evento, identificados pela chave presente em `events` — a descrição de negócio de cada um (quando ocorre e o que significa) está detalhada no "Protocolo e Especificação Técnica". Ao nível do formato da mensagem, cada um transporta o seguinte detalhe:

Tipo de evento (chave de `events`)	Detalhe incluído
`https://azorid.azores.gov.pt/schemas/ssf/attribute-changed`	`subject_type` (`citizen` ou `organization`) e `attributes` — lista dos atributos alterados (`name`, `fiscal_number`, `picture`).
`https://azorid.azores.gov.pt/schemas/ssf/identity-certification-status-changed`	`subject_type`, `previous_status` e `current_status` (`true`/`false`).

IMPORTANTE

Estas notificações **não incluem os novos valores** dos dados alterados — apenas identificam que algo mudou. O sistema cliente deve obter os dados atualizados através dos mecanismos já existentes (por exemplo, um novo pedido ao endpoint `userinfo`, caso o utilizador tenha uma sessão ativa).

11.4. Processamento pelo sistema cliente

Ao receber este pedido, o sistema cliente deve:

1. Validar a assinatura do SET;
2. Identificar a conta afetada através de `sub_id.sub`;
3. Identificar o(s) tipo(s) de evento presente(s) em `events` e atuar em conformidade — tipicamente, marcar os dados locais dessa conta como desatualizados e obtê-los novamente através dos mecanismos já existentes.

11.5. Resposta esperada

Em caso de sucesso, o sistema cliente deve responder com `HTTP 202`, sem corpo, em conformidade com [RFC8935]:

```
HTTP/1.1 202 Accepted
```

Em caso de erro (por exemplo, SET inválido), deve ser devolvido um código de erro HTTP apropriado (`400` para um token mal formado ou inválido), em conformidade com a mesma especificação.

12. Renovação de tokens

Sempre que um `access_token` expira, o sistema cliente pode obter um novo conjunto de tokens sem reencaminhar o utilizador de novo pelo passo de autorização (Pedido de autorização), desde que possua um `refresh_token` válido.

O `refresh_token` é devolvido juntamente com o `access_token` e o `id_token` na resposta do passo de obtenção de tokens (Obtenção de tokens) — tanto no pedido inicial (`grant_type=authorization_code`) como em cada renovação subsequente.

IMPORTANTE	Tal como o `client_secret`, o `refresh_token` deve ser armazenado de forma segura pelo servidor (*backend*) do sistema cliente, e **nunca** exposto ao navegador do utilizador.

12.1. Parâmetros do pedido

Parâmetro	Obrigatório	Descrição
`grant_type`	Sim	Fixo em `refresh_token`.
`refresh_token`	Sim	O `refresh_token` obtido no pedido de tokens anterior.
`client_id`	Sim	Identificador do sistema cliente.
`client_secret`	Sim	Segredo do sistema cliente.

12.2. Pedido de exemplo

Iniciado pelo servidor do sistema cliente e direcionado ao AzorID.

```
POST /openid/token HTTP/1.1
Host: azorid-qa.azores.gov.pt
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
&refresh_token=YOUR_REFRESH_TOKEN
&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET
```

12.3. Corpo da resposta de exemplo

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "Bearer",
  "expires_in": 28800,
```

```
  "scope": "openid profile email",
  "refresh_token": "NEW_REFRESH_TOKEN",
  "refresh_token_expires_in": 2592000
}
```

12.4. Rotação do refresh_token

Cada utilização de um `refresh_token` devolve um novo `access_token`, um novo `id_token` e um **novo** `refresh_token` — o `refresh_token` utilizado no pedido deixa de ser válido e não pode ser reutilizado.

IMPORTANTE	O sistema cliente deve substituir de imediato o `refresh_token` armazenado pelo novo valor devolvido em cada renovação. A reutilização de um `refresh_token` já consumido é tratada como um indício de comprometimento do token, e resulta na revogação de toda a autorização associada — incluindo o `refresh_token` mais recente entretanto emitido — obrigando o utilizador a autenticar-se novamente desde o início (Pedido de autorização).

NOTA	O `sub` e o `sid` do novo `id_token`/`access_token` mantêm-se sempre inalterados em relação aos tokens anteriores da mesma sessão — apenas o `refresh_token` e o `jti` mudam a cada renovação. O sistema cliente não precisa de atualizar o `sub`/`sid` que guardou (Obtenção de tokens) após uma renovação.

12.5. Utilização prática: obter dados atualizados via userinfo

Para além de evitar novos reencaminhamentos do utilizador, o `refresh_token` é também a forma recomendada de obter uma versão atualizada dos dados da conta através do endpoint `userinfo` (Consulta de dados do utilizador), sem necessitar de uma nova autenticação: basta trocar o `refresh_token` por um novo `access_token` e, de seguida, consultar o `userinfo_endpoint` com esse token.

Isto é particularmente relevante para dados que podem ter sido alterados ou certificados após o `id_token` /`access_token` originais terem sido emitidos (por exemplo, uma mudança de morada, ou uma certificação de identidade concluída posteriormente) — o AzorID recalcula estes dados no momento do pedido a `userinfo`, em vez de devolver os valores do momento da autenticação original.

12.6. Respostas de erro

Caso o `refresh_token` seja inválido, tenha expirado, já tenha sido utilizado (ver Rotação do refresh_token), ou pertença a uma autorização entretanto revogada, o pedido é rejeitado com `HTTP 400` e o erro `invalid_grant`, em conformidade com a especificação [RFC6749]. Os restantes erros do endpoint (por exemplo, `invalid_client` para credenciais erradas) estão descritos em Obtenção de tokens.

```
{
  "error": "invalid_grant",
```

```
  "error_description": "The refresh token is invalid, expired, already used or
revoked."
}
```

Quando o erro se deve à reutilização de um `refresh_token` já consumido, a autorização é revogada nesse mesmo pedido e, caso o sistema cliente tenha um URL de *Backchannel Logout* configurado, é-lhe enviada a respetiva notificação (Backchannel Logout).

13. Referências

- OAuth 2.0 Authorization Framework
- OAuth 2.0 Bearer Token Usage
- OpenID Connect Core 1.0
- JSON Web Token (JWT)
- JSON Web Key (JWK)
- OpenID Connect Back-Channel Logout 1.0
- OAuth 2.0 Token Exchange
- OpenID Shared Signals Framework 1.0
- Security Event Token (SET)
- Push-Based Security Event Token (SET) Delivery Using HTTP