IDLAC Broker Modelo Regional

DOCUMENTACIÓN TÉCNICA
Parte II.a: Arquitectura

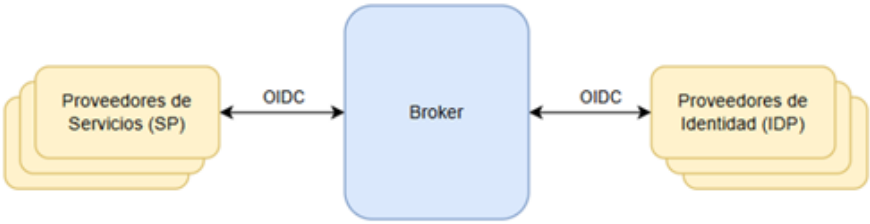
Versión	Modificación	Fecha	Modificado por:
1.0	Versión inicial	2025-09-04	Mariana Silvera
1.1	Incluye URL well-known de Testing y logout uri. Mapeo de Claims	2025-10-06	Mariana Silvera
1.2	Incluye documentación final Fase 1	2025-12-04	Mariana Silvera
1.3	Separa documentación técnica del broker del Modelo	2025-12-10	JP García

Documento en permanente evolución elaborado por el Grupo de trabajo de Red Gealc con apoyo del Consorcio Ciudadano Digital Regional (BID, Banco Mundial, CoDevelop y OEA)

8. Arquitectura de la solución

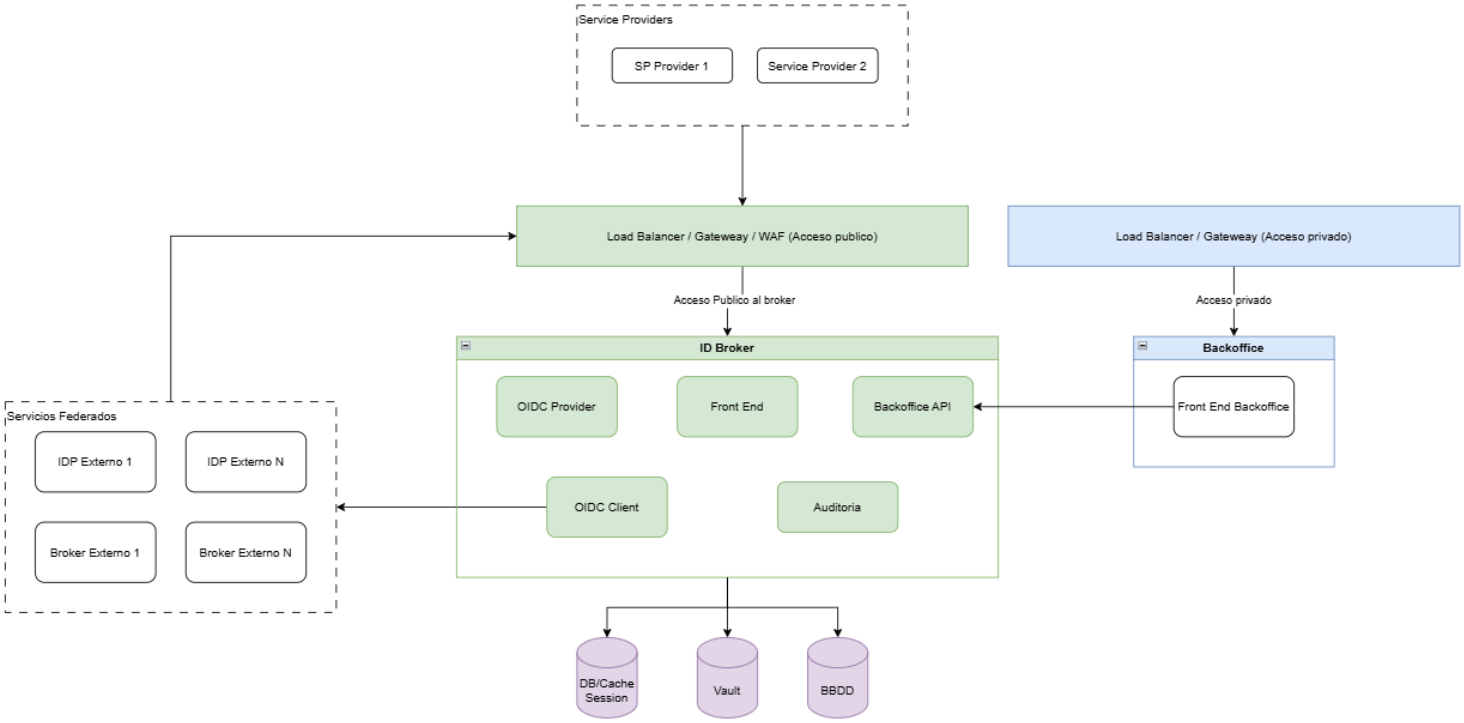
8.1. Introducción

Desde el punto de vista del usuario, el Broker consiste en una aplicación web que permite iniciar sesión en múltiples proveedores de identidad para acceder a los recursos de múltiples proveedores de servicios.



La idea del Broker como producto es poder modularizar este vínculo con múltiples IdPs y hacer que los SPs implementen una única integración. De esta forma, los SPs reciben del Broker siempre respuestas con la misma estructura y no deben preocuparse por las múltiples integraciones con IdPs.

8.2. Arquitectura General



A continuación, se incluye una descripción resumida de los componentes:

- **Frontend**: Sitio web al que el usuario es redirigido para poder iniciar sesión. Cuenta con una pantalla que lista las opciones de IDPs disponibles y otra que muestra los datos del usuario que tiene la sesión abierta.

- **OIDC Provider**: Modulo del backend que implementa el protocolo de autenticación.

- **OIDC Client**: Módulo que instancia las integraciones con los IDPs.

- **Base de datos**: Almacenamiento de información de conexión con SPs e IdPs. También almacena las autenticaciones entre el broker y los IDPs, y las autenticaciones entre el broker y los SPs

- **Auditoría**: Módulo encargado de generar las trazas de las autenticaciones y cambios en los IDPs, SPs y configuración global

- **DB/Cache**: Gestión de sesiones

- **Bakcoffice APIs**: Modulo que se encarga de exponer las APIs para la gestion del backoffice. Para la autenticacion es como un service provider más

- **Backoffice**: Front end de administración de SPs e IdPs, donde se puede realizar el alta, baja, modificación y demás configuraciones vía API Rest del ID Broker.

- **Load Balancer**: Componente de infraestructura encargado de gestionar las conexiones en escenarios de HA, no incluido como entregable del producto.

- **Servicios Federados**: Todos los IDPs y Brokers externos con los que se integra el ID Broker.

- **Service Providers**: Todos los SPs externos con los que se integra el ID Broker.

8.3. Componentes del sistema

8.3.1. Visión general

El Broker se implementa como una solución basada en **Java 21** y **Spring Boot 3.x**, organizada en tres grandes módulos:
el **Proveedor OIDC del Broker**, el **Cliente OIDC del Broker** y las **APIs administrativas del Backoffice**.
Sobre estos módulos se construyen dos frontends: el sitio web de interacción del usuario final y la consola administrativa.

La arquitectura se apoya en **Spring Authorization Server**, que constituye la base técnica del modelo de autorización, persistencia de consentimientos, emisión de tokens y soporte para OAuth2 y OpenID Connect.
El proyecto extiende esta implementación estándar con modelos, endpoints y reglas propias del dominio del Broker.

La solución corre completamente en contenedores de **AWS ECS Fargate**, utiliza **PostgreSQL** como base de datos relacional y **Redis** como soporte para sesiones de autenticación.

8.3.2. Componentes lógicos y tecnologías

1. Frontend del Broker (usuario final)

- Implementado con **Thymeleaf** integrado dentro de Spring Boot.
- Sirve las pantallas donde el usuario selecciona el Identity Provider, visualiza datos mínimos de su sesión y completa steps del flujo OIDC.
- Se entrega desde la propia aplicación backend sin necesidad de un frontend separado.

2. Frontend del Backoffice (administración)

- Implementado como una SPA en **React**.
- Consume las APIs REST del módulo de backoffice para la gestión de Service Providers, Identity Providers y parámetros globales.
- Se despliega junto con el backend dentro del esquema de ECS Fargate.

3. Proveedor OIDC del Broker

- Módulo backend basado en **Spring Authorization Server**.
- Expone los endpoints estándar de autorización, emisión de tokens e información del usuario.
- Es responsable de firmar tokens, validar flujos, gestionar sesiones, autorizar clientes y persistir datos en la base PostgreSQL.
- El proyecto se apoya directamente en la implementación de Spring Authorization Server, que actúa como **componente central y pilar arquitectónico**.

4. Cliente OIDC del Broker

- Módulo backend basado en Spring Security OAuth2 Client.
- Se utiliza para establecer flujos federados contra Identity Providers externos y otros brokers.
- Valida ID tokens recibidos, inicia flujos de autorización y normaliza claims para ser consumidos por los Service Providers.

5. APIs del Backoffice

- Endpoints REST que exponen la administración completa del Broker.
- Se autentican vía OAuth2 contra el mismo Proveedor OIDC del Broker.
- Modelan alta, baja y actualización de Service Providers, Identity Providers y configuración central.

6. Base de datos relacional

- Base **PostgreSQL** que almacena:
  - configuraciones del broker (tablas *service_provider*, *identity_providers*, *broker_config*, usuarios del backoffice)
  - tablas estándar de Spring Authorization Server (autorizaciones, tokens y clientes)
- Único origen de verdad para configuraciones y metadatos del sistema.

7. Cache y manejo de sesiones

- Utiliza **Redis** como almacenamiento en memoria.
- Se usa para:
  - caché de sesiones durante los flujos de autenticación
- Reduce carga en la base relacional y mejora latencia en flujos críticos.

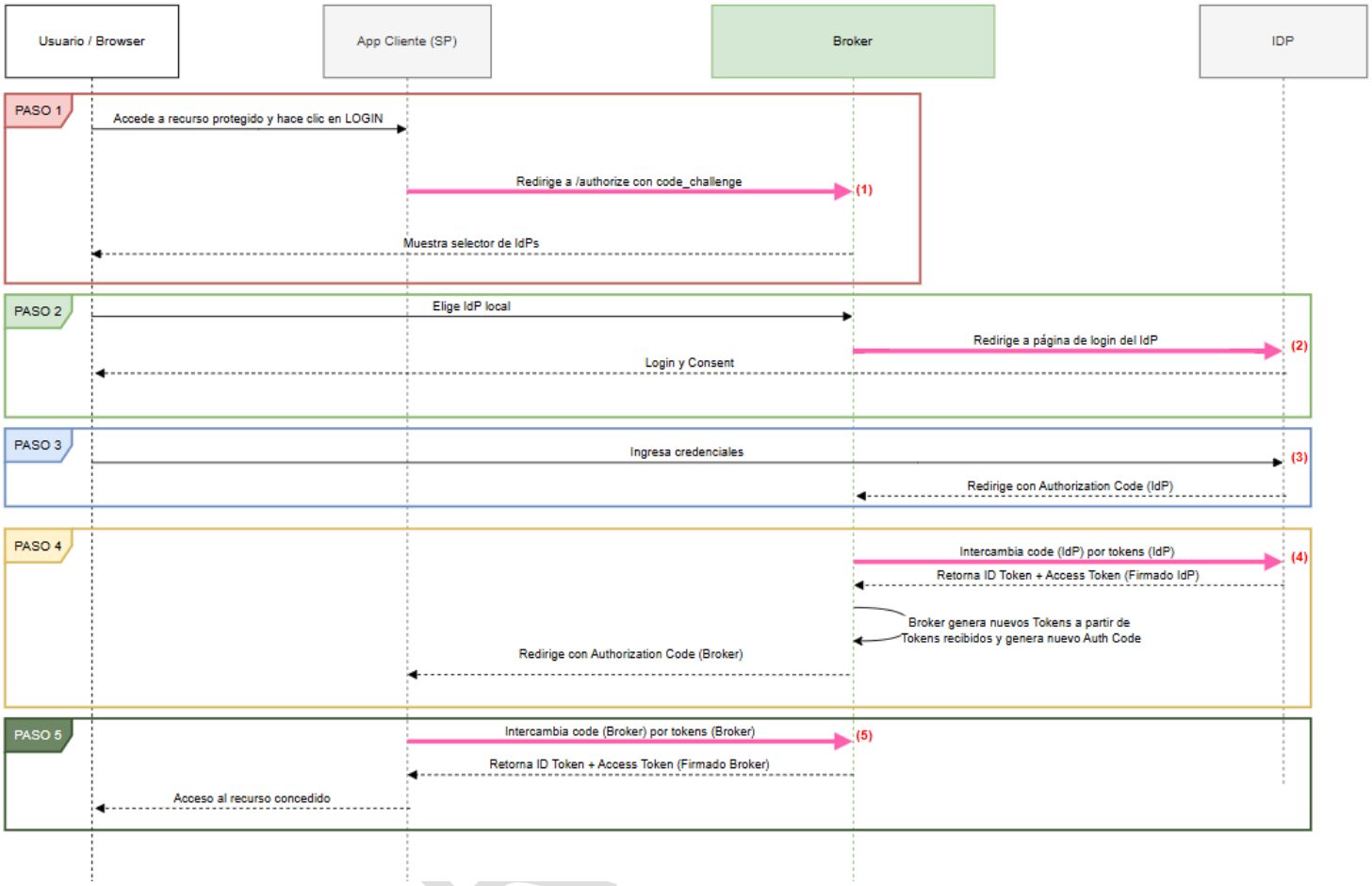
8. Infraestructura en AWS

- Contenedores desplegados en **Amazon ECS Fargate**.
- **Application Load Balancer** para exponer endpoints OIDC y Backoffice.
- **Route 53** para dominios públicos.
- **CloudWatch** y servicios asociados para logging, métricas y monitoreo.

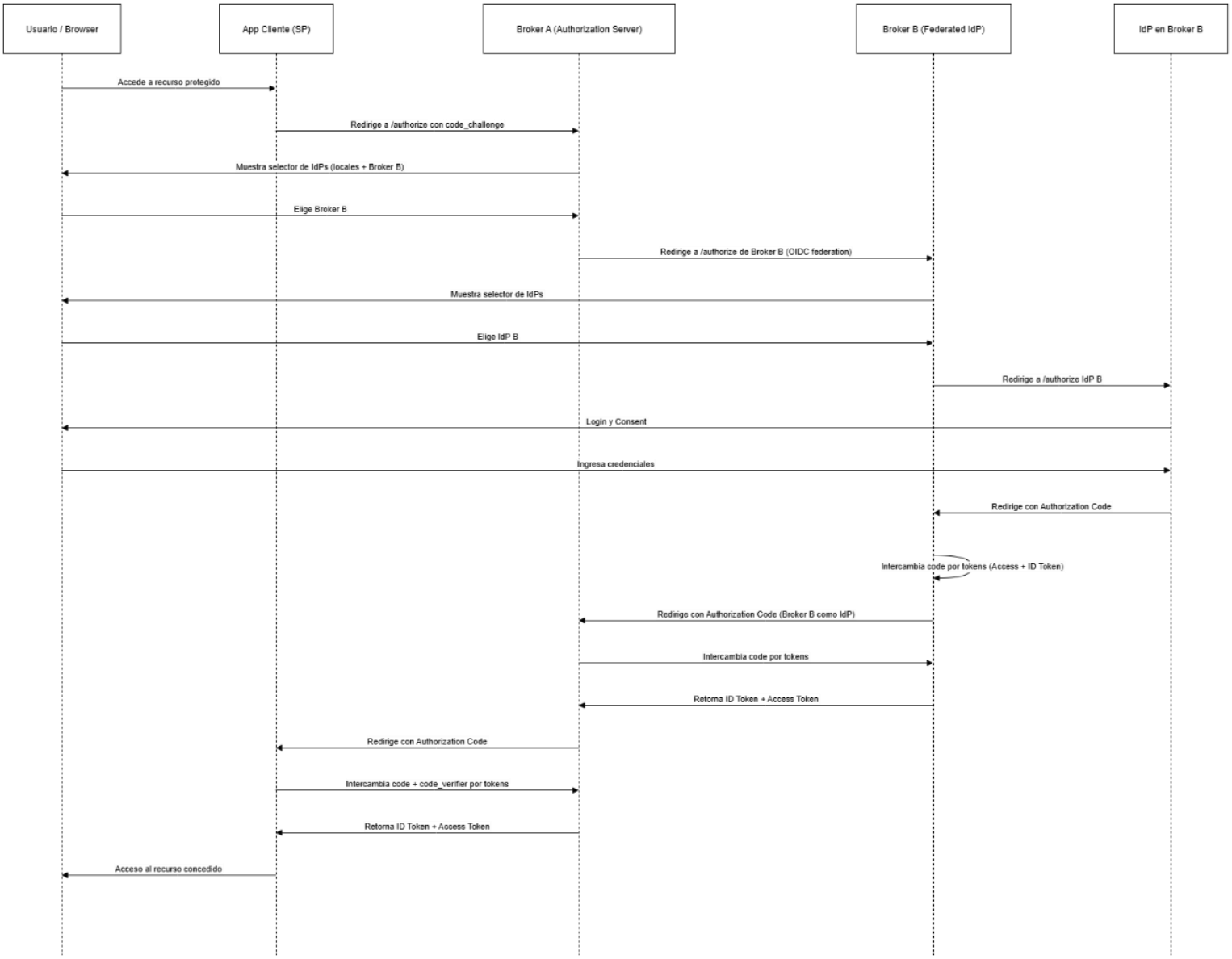
8.3.3. Flujos

Flujo Simple - IDP Local

El siguiente diagrama detalla la comunicación entre los distintos actores de la solución, para el caso de un usuario autenticándose contra un IDP integrado a ID Broker:



Flujo entre broker federados



8.3.4. Modelo de datos

Visión general

El broker utiliza una base de datos relacional PostgreSQL con un esquema compacto, pensado para delegar la mayor parte de la gestión de OAuth2 y OpenID Connect en Spring Authorization Server y extenderla solo donde es necesario para el dominio de RedGealc.

El modelo de datos se organiza en las siguientes áreas lógicas:

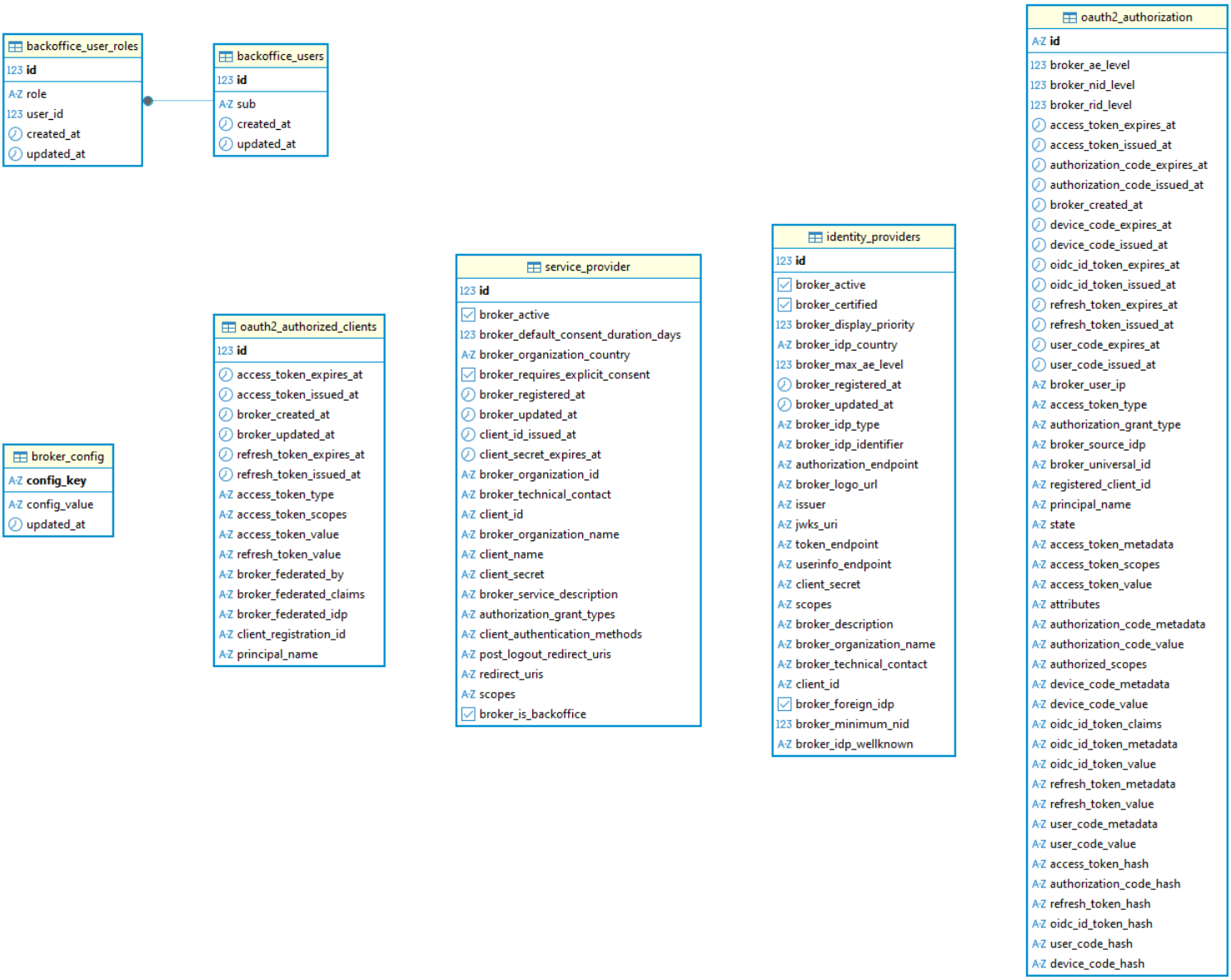
1. Gestión de administración
    - backoffice_users
    - backoffice_user_roles
2. Configuración global del broker

- broker_config

3. Proveedores de servicio (Service Providers) registrados
    - service_provider
4. Proveedores de identidad federados
    - identity_providers
5. Infraestructura de autorizaciones y clientes OAuth2 OIDC
    - oauth2_authorized_clients
    - oauth2_authorization

Estas siete tablas representan el conjunto completo del modelo persistente del sistema.

Diagrama MER



Origen de los campos

El esquema combina columnas estándar de Spring Authorization Server con columnas específicas del dominio del broker.

- Todas las columnas cuyos nombres **no** comienzan con `broker_` corresponden al modelo de datos provisto por Spring Authorization Server.

- Todas las columnas que **sí** comienzan con el prefijo `broker_` fueron definidas por el equipo del broker y representan metadatos propios, reglas adicionales de negocio o información necesaria para la federación entre países y sistemas.

Esta convención facilita distinguir qué partes del modelo son estándar de plataforma y cuáles forman parte de la extensión específica del proyecto.

Descripción funcional de las tablas

backoffice_users y backoffice_user_roles

Representan a los usuarios administrativos del backoffice y sus roles asociados.
Permiten modelar qué personas pueden gestionar Identity Providers, Service Providers y configuraciones del broker.

broker_config

Tabla clave valor utilizada para parámetros de configuración global del broker, por ejemplo tiempos de vida de tokens o banderas de comportamiento. Permite ajustar el funcionamiento sin necesidad de cambios de código.

service_provider

Modela cada aplicación cliente que confía en el broker. Incluye tanto la información estándar de un cliente OAuth2 OIDC como metadatos específicos del dominio, por ejemplo datos de organización, contacto técnico, etc.
Los secretos de cliente de los Service Providers se almacenan utilizando hashing **bcrypt**.

identity_providers

Representa cada proveedor de identidad federado con el broker. Contiene la información necesaria para construir dinámicamente la configuración de cliente frente al IDP, así como metadatos de certificación, tipo de IDP, país y parámetros de integración.
Los secretos de cliente utilizados para autenticarse frente a los IDP se almacenan cifrados con **AES GCM**, es decir, de forma reversible bajo una clave controlada por el broker.

oauth2_authorized_clients

Tabla estándar de Spring Authorization Server que representa la relación entre un usuario, un cliente registrado y los tokens asociados en el contexto de un flujo OAuth2 OIDC.
El broker la utiliza para registrar autorizaciones persistentes y para asociar información federada adicional mediante columnas con prefijo `broker_`.

oauth2_authorization

Tabla central de Spring Authorization Server donde se registran autorizaciones y tokens en detalle, incluyendo códigos de autorización, tokens de acceso, tokens de actualización, ID tokens y códigos de dispositivo, junto con su metadata, scopes y atributos.
Las columnas con prefijo `broker_` amplían esta información con datos propios del dominio del broker.

Estrategia de cifrado y columnas con sufijo _hash

Los datos sensibles relacionados con tokens y claims se almacenan cifrados con **AES GCM**. Este enfoque protege el contenido en reposo, aunque implica que para realizar búsquedas directas por valor sería necesario cifrar o descifrar en cada consulta, lo que resultaría poco eficiente y aumentaría la superficie de riesgo.

Para resolver este problema se introdujo una convención adicional:

- Las columnas cuyo nombre termina en `_hash` son una **representación hasheada** del valor original correspondiente.
- El valor original se almacena cifrado con AES GCM en su columna principal.
- El hash se utiliza como clave de búsqueda para comparaciones de igualdad sin necesidad de descifrar el valor de la columna.

De esta manera, el sistema mantiene la protección de los datos sensibles en reposo y al mismo tiempo permite búsquedas eficientes en la base, usando los hashes como índice lógico.

8.4. Auditoría de registros

Introducción

El modelo de datos incorpora una auditoría básica a nivel de registro mediante columnas de tipo `updated_at` en las tablas relevantes.
Estas columnas se actualizan cuando se modifica el registro y permiten conocer la última fecha de actualización de cada entidad, lo que es suficiente para:

- entender la antigüedad de la configuración de un IDP o SP
- trazar cambios en configuraciones globales
- analizar el ciclo de vida de ciertos registros operativos

No se mantienen tablas de auditoría adicionales en base de datos. La trazabilidad más detallada se apoya en los logs de aplicación y en herramientas de observabilidad externas.

Alcance

La funcionalidad de auditoría del broker de identidades permite registrar trazas de eventos relevantes para análisis futuro. El alcance está limitado a la captura de información y su persistencia en archivos de auditoría en formato JSON con el objetivo de que sea amigable para algún concentrador de logs del estilo Kibana.

El sistema registra autenticaciones exitosas realizadas a través del broker y cambios efectuados desde el backoffice en entidades de configuración. No se incluye la

detección de saltos de autenticación (SSO) ni el registro de errores de autenticación, ya que los IDPs no reportan estos fallos.

La integración con un concentrador de logs se deja para una etapa posterior

Auditoría de Autenticaciones Exitosas

Descripción

Cada autenticación exitosa iniciada desde un Service Provider y procesada por un Identity Provider a través del broker produce un evento de auditoría. El objetivo es garantizar la trazabilidad completa de cada operación de autenticación.

Datos registrados

Los eventos incluyen:

- sub del usuario autenticado.
- Dirección IP de origen.
- Identity Provider (IDP) utilizado.
- Claims de identidad emitidos por el broker:

o rid, ae, nid,

o document

o Claims openid estándar que hayan sido emitidos (o mapeados) desde el IDP

- Service Provider (SP) que inició el flujo.
- Timestamp exacto del evento.

La auditoría de autenticaciones persiste información sensible del usuario que se esta autenticando. Se deja para una etapa posterior el ofuscamiento y/o selección de atributos a auditar.

Auditoría de Cambios en Backoffice

Descripción

El sistema registra todas las modificaciones realizadas desde el backoffice sobre las entidades de configuración del broker. Esto permite garantizar trazabilidad operativa y control de cambios.

Entidades auditadas

- Identity Providers: alta, modificación y eliminación.
- Service Providers: alta, modificación y eliminación.
- Configuración global del sistema: modificación.

Datos registrados

Cada evento de backoffice contiene:

- Usuario del backoffice que ejecuta la acción (sub).
- Identificador de la entidad modificada.
- Tipo de operación (alta, modificación o eliminación).
- Valores de entrada del cambio.
- Resultado de la operación.
- Timestamp del evento.

Notas Técnicas

Formato del registro

Todos los eventos se registran en un archivo dedicado a auditoría. Cada línea del archivo contiene un objeto JSON independiente con el detalle completo del evento correspondiente.

Ubicación del archivo

La ubicación del archivo se define mediante la variable de entorno BROKER_AUDIT_LOG_FILE. Si no se especifica, se utilizan las rutas configuradas por Spring Boot para logs, siguiendo el siguiente orden de prioridad: LOG_PATH, LOG_TEMP, java.io.tmpdir o /tmp.

Nombre por defecto

El archivo se denomina broker_audit.log cuando no se define explícitamente otra ubicación.

Políticas de retención y rotación del archivo de auditoría

La rotación se realiza mediante una política combinada por tamaño y por día. El archivo se rota diariamente y cada vez que supera el tamaño máximo definido por `LOGBACK_ROLLINGPOLICY_MAX_FILE_SIZE` (10MB por defecto). Los archivos rotados se generan comprimidos con el patrón `${BROKER_AUDIT_LOG_FILE}.yyyy-MM-dd.i.gz`.

La retención está gobernada por `LOGBACK_ROLLINGPOLICY_MAX_HISTORY`, que conserva archivos por hasta 7 días, eliminando automáticamente los más antiguos. Si se define `LOGBACK_ROLLINGPOLICY_TOTAL_SIZE_CAP`, se limita además el tamaño total del historial; por defecto no hay límite. `LOGBACK_ROLLINGPOLICY_CLEAN_HISTORY_ON_START` controla si se limpia el historial al iniciar la aplicación (por defecto no).

Ejemplos

Autenticación


```
{
  "@timestamp": "2025-11-20T12:14:50.384580302-03:00",
  "logger_name": "com.redgealc.identitybroker.audit.events",
  "thread_name": "https-jsse-nio-8443-exec-4",
```

```
"level": "INFO",
"event": {
    "type": "broker:authentication:success",
    "generatedAt": "2025-11-20T15:14:50.379035475Z",
    "key": "CR-ID-CRI330016",
    "payload": {
    "userSub": "CR-ID-CRI330016",
    "idpClientId": "idp-nacional-uy-dev",
    "spClientId": "broker-backoffice",
    "userClaims": {
    "website": null,
    "zoneinfo": null,
    "birthdate": "1990-01-16",
    "email_verified": null,
    "gender": null,
    "ae": 3,
    "profile": null,
    "document": {
    "document_country": "CR",
    "document_id": "CRI330016",
    "document_type": "ID"
    },
    "nid": 3,
    "phone_number_verified": null,
    "preferred_username": null,
    "given_name": "Given3",
    "middle_name": "Sample",
    "locale": null,
    "rid": 3,
    "picture": null,
    "updated_at": null,
    "name": "Given3 Family3",
```

```
"nickname": "Given3",
"phone_number": "+506-223300",
"family_name": "Family3",
"email": "user.ae3.rid3@example.cr"
},
"ipAddress": "127.0.0.1"
}
}
}
```


Backoffice

Crear usuario


```
{
"@timestamp": "2025-11-20T12:15:36.606131945-03:00",
"logger_name": "com.redgealc.identitybroker.audit.events",
"thread_name": "https-jsse-nio-8443-exec-3",
"level": "INFO",
"event": {
"type": "backoffice:user:create",
"generatedAt": "2025-11-20T15:15:36.602433060Z",
"key": 2,
"payload": {
"userSub": "CR-ID-CRI330016",
"arguments": {
"body": {
"id": null,
"sub": "CR-ID-CRI0000001",
"createdAt": null,
"updatedAt": null
}
},
"output": {
"headers": {
"Location": [
```

```
"/backoffice/apis/users/admin/2"
]
},
"body": {
"id": 2,
"sub": "CR-ID-CRI0000001",
"createdAt": "2025-11-20T15:15:36.592048966Z",
"updatedAt": "2025-11-20T15:15:36.592048966Z"
},
"statusCode": "CREATED",
"statusCodeValue": 201
}
}
}
}
```


Borrar usuario


```
{
  "@timestamp": "2025-11-20T12:15:57.191254382-03:00",
  "logger_name": "com.redgealc.identitybroker.audit.events",
  "thread_name": "https-jsse-nio-8443-exec-1",
  "level": "INFO",
  "event": {
"type": "backoffice:user:delete",
"generatedAt": "2025-11-20T15:15:57.191098711Z",
"key": 2,
"payload": {
"userSub": "CR-ID-CRI330016",
"arguments": {
"id": 2
},
"output": {
"headers": {},
```

```
                "body": null,
                "statusCode": "NO_CONTENT",
                "statusCodeValue": 204
                }
                }
        }
    }
```


Crear IDP


```
    {
     "@timestamp": "2025-11-20T12:16:12.686331719-03:00",
     "logger_name": "com.redgealc.identitybroker.audit.events",
     "thread_name": "https-jsse-nio-8443-exec-7",
     "level": "INFO",
     "event": {
                "type": "backoffice:identity-provider:create",
                "generatedAt": "2025-11-20T15:16:12.685403585Z",
                "key": 3,
                "payload": {
                "userSub": "CR-ID-CRI330016",
                "arguments": {
                "body": {
                "id": 3,
                "issuer": "https://accounts.example-idp.com/openid",
                "clientId": "example-client-id",
                "authorizationEndpoint": "https://accounts.example-
idp.com/openid/authorize",
                "tokenEndpoint": "https://accounts.example-idp.com/openid/token",
                "userinfoEndpoint": "https://accounts.example-
idp.com/openid/userinfo",
                "jwksUri": "https://accounts.example-idp.com/openid/jwks",
                "scopes": "openid profile email",
                "brokerOrganizationName": "ExampleGov - Test2222",
                "brokerIdpIdentifier": "example-idp",
```

"brokerIdpCountry": "EX",

"brokerMaxAeLevel": 2,

"brokerActive": true,

"brokerTechnicalContact": "techcontact@example.gov",

"brokerDescription": "Example IDP for testing purposes",

"brokerRegisteredAt": "2025-11-20T15:16:12.681531978Z",

"brokerUpdatedAt": "2025-11-20T15:16:12.681813827Z",

"brokerDisplayPriority": 10,

"brokerLogoUrl": "https://cdn.example.com/logos/example-idp.png",

"brokerForeignIdP": false,

"brokerMinimumNid": 1

}

},

"output": {

"headers": {

"Location": [

"/backoffice/apis/idps/3"

]

},

"body": {

"id": 3,

"issuer": "https://accounts.example-idp.com/openid",

"clientId": "example-client-id",

"authorizationEndpoint": "https://accounts.example-idp.com/openid/authorize",

"tokenEndpoint": "https://accounts.example-idp.com/openid/token",

"userinfoEndpoint": "https://accounts.example-idp.com/openid/userinfo",

"jwksUri": "https://accounts.example-idp.com/openid/jwks",

"scopes": "openid profile email",

"brokerOrganizationName": "ExampleGov - Test2222",

"brokerIdpIdentifier": "example-idp",

"brokerIdpCountry": "EX",

"brokerMaxAeLevel": 2,
"brokerActive": true,
"brokerTechnicalContact": "techcontact@example.gov",
"brokerDescription": "Example IDP for testing purposes",
"brokerRegisteredAt": "2025-11-20T15:16:12.681531978Z",
"brokerUpdatedAt": "2025-11-20T15:16:12.681813827Z",
"brokerDisplayPriority": 10,
"brokerLogoUrl": "https://cdn.example.com/logos/example-idp.png",
"brokerForeignIdP": false,
"brokerMinimumNid": 1
},
"statusCode": "CREATED",
"statusCodeValue": 201
}
}
}
}

Actualizar IDP

{
"@timestamp": "2025-11-20T12:16:24.219391788-03:00",
"logger_name": "com.redgealc.identitybroker.audit.events",
"thread_name": "https-jsse-nio-8443-exec-10",
"level": "INFO",
"event": {
"type": "backoffice:identity-provider:update",
"generatedAt": "2025-11-20T15:16:24.218934098Z",
"key": 3,
"payload": {
"userSub": "CR-ID-CRI330016",
"arguments": {
"id": 3,
"body": {

```
"id": 3,
"issuer": "https://auth-testing.iduruguay.gub.uy/oidc/v9",
"clientId": "857120333",
"authorizationEndpoint": "https://auth-testing.iduruguay.gub.uy/oidc/v1/authorize",
"tokenEndpoint": "https://auth-testing.iduruguay.gub.uy/oidc/v1/token",
"userinfoEndpoint": "https://auth-testing.iduruguay.gub.uy/oidc/v1/userinfo",
"jwksUri": "https://auth-testing.iduruguay.gub.uy/oidc/v2/jwks",
"scopes": "openid profile email document auth_info",
"brokerOrganizationName": "GUB UY (Test)22",
"brokerIdpIdentifier": "idp-nacional-uy-test-8",
"brokerIdpCountry": "UY",
"brokerMaxAeLevel": 3,
"brokerActive": true,
"brokerTechnicalContact": "soporte@id-desarrollo.uy",
"brokerDescription": "Broker de identidades de ID Uruguay para pruebas y desarrollo.",
"brokerRegisteredAt": "2025-10-30T15:02:22.684487Z",
"brokerUpdatedAt": "2025-10-30T15:02:22.684487Z",
"brokerDisplayPriority": 2,
"brokerLogoUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/f/fe/Flag_of_Uruguay.svg/512px-Flag_of_Uruguay.svg.png",
"brokerForeignIdP": true,
"brokerMinimumNid": 3
}
},
"output": {
"headers": {},
"body": {
"id": 3,
"issuer": "https://auth-testing.iduruguay.gub.uy/oidc/v9",
"clientId": "857120333",
```

"authorizationEndpoint": "https://auth-testing.iduruguay.gub.uy/oidc/v1/authorize",

"tokenEndpoint": "https://auth-testing.iduruguay.gub.uy/oidc/v1/token",

"userinfoEndpoint": "https://auth-testing.iduruguay.gub.uy/oidc/v1/userinfo",

"jwksUri": "https://auth-testing.iduruguay.gub.uy/oidc/v2/jwks",

"scopes": "openid profile email document auth_info",

"brokerOrganizationName": "GUB UY (Test)22",

"brokerIdpIdentifier": "idp-nacional-uy-test-8",

"brokerIdpCountry": "UY",

"brokerMaxAeLevel": 3,

"brokerActive": true,

"brokerTechnicalContact": "soporte@id-desarrollo.uy",

"brokerDescription": "Broker de identidades de ID Uruguay para pruebas y desarrollo.",

"brokerRegisteredAt": "2025-11-20T15:16:12.681532Z",

"brokerUpdatedAt": "2025-11-20T15:16:24.212431386Z",

"brokerDisplayPriority": 2,

"brokerLogoUrl": "https://upload.wikimedia.org/wikipedia/commons/thumb/f/fe/Flag_of_Uruguay.svg/512px-Flag_of_Uruguay.svg.png",

"brokerForeignIdP": true,

"brokerMinimumNid": 3

},

"statusCode": "OK",

"statusCodeValue": 200

}

}

}

}

Eliminar IDP

{

"@timestamp": "2025-11-20T12:16:30.734777366-03:00",

"logger_name": "com.redgealc.identitybroker.audit.events",
"thread_name": "https-jsse-nio-8443-exec-3",
"level": "INFO",
"event": {
"type": "backoffice:identity-provider:delete",
"generatedAt": "2025-11-20T15:16:30.734526014Z",
"key": 3,
"payload": {
"userSub": "CR-ID-CRI330016",
"arguments": {
"id": 3
},
"output": {
"headers": {},
"body": null,
"statusCode": "NO_CONTENT",
"statusCodeValue": 204
}
}
}
}

Crear SP

{
"@timestamp": "2025-11-20T12:16:55.094171688-03:00",
"logger_name": "com.redgealc.identitybroker.audit.events",
"thread_name": "https-jsse-nio-8443-exec-4",
"level": "INFO",
"event": {
"type": "backoffice:service-provider:create",
"generatedAt": "2025-11-20T15:16:55.093128972Z",
"key": 3,
"payload": {

```
"userSub": "CR-ID-CRI330016",
"arguments": {
"body": {
"id": 3,
"clientId": "sp-client-test2",
"clientIdIssuedAt": "2025-11-20T15:16:55.015563328Z",
"clientSecretExpiresAt": null,
"clientName": "Portal Ciudadano de Pruebas22222",
"clientAuthenticationMethods":
"[\"client_secret_basic\",\"client_secret_post\"]",
"authorizationGrantTypes":
"[\"authorization_code\",\"refresh_token\"]",
"redirectUris": "[\"https://portal-ciudadano.uy/callback\"]",
"postLogoutRedirectUris": "[\"https://portal-ciudadano.uy/logout-
success\"]",
"scopes":
"[\"openid\",\"profile\",\"email\",\"document\",\"address\",\"phone\",\"auth_inf
o\"]",
"brokerOrganizationName": "Ministerio de Innovación Digital",
"brokerOrganizationId": "MIN-INNOVACION-002",
"brokerOrganizationCountry": "UY",
"brokerTechnicalContact": "tecnico@innovacion.uy",
"brokerActive": true,
"brokerServiceCategory": "government",
"brokerServiceDescription": "Portal ciudadano para autenticación
centralizada de servicios digitales.",
"brokerRegisteredAt": "2025-11-20T15:16:55.015563550Z",
"brokerUpdatedAt": null,
"brokerRequiresExplicitConsent": false,
"brokerDefaultConsentDurationDays": 365,
"brokerIsBackoffice": false
}
},
"output": {
"headers": {
```

"Location": [

"/backoffice/apis/service-providers/3"

]

},

"body": {

"id": 3,

"clientId": "sp-client-test2",

"clientIdIssuedAt": "2025-11-20T15:16:55.015563328Z",

"clientSecretExpiresAt": null,

"clientName": "Portal Ciudadano de Pruebas22222",

"clientAuthenticationMethods": "[\"client_secret_basic\",\"client_secret_post\"]",

"authorizationGrantTypes": "[\"authorization_code\",\"refresh_token\"]",

"redirectUris": "[\"https://portal-ciudadano.uy/callback\"]",

"postLogoutRedirectUris": "[\"https://portal-ciudadano.uy/logout-success\"]",

"scopes": "[\"openid\",\"profile\",\"email\",\"document\",\"address\",\"phone\",\"auth_info\"]",

"brokerOrganizationName": "Ministerio de Innovación Digital",

"brokerOrganizationId": "MIN-INNOVACION-002",

"brokerOrganizationCountry": "UY",

"brokerTechnicalContact": "tecnico@innovacion.uy",

"brokerActive": true,

"brokerServiceCategory": "government",

"brokerServiceDescription": "Portal ciudadano para autenticación centralizada de servicios digitales.",

"brokerRegisteredAt": "2025-11-20T15:16:55.015563550Z",

"brokerUpdatedAt": null,

"brokerRequiresExplicitConsent": false,

"brokerDefaultConsentDurationDays": 365,

"brokerIsBackoffice": false

},

"statusCode": "CREATED",

```
            "statusCodeValue": 201
            }
            }
     }
    }
```


Actualizar SP


```
    {
     "@timestamp": "2025-11-20T12:17:07.700917278-03:00",
     "logger_name": "com.redgealc.identitybroker.audit.events",
     "thread_name": "https-jsse-nio-8443-exec-2",
     "level": "INFO",
     "event": {
            "type": "backoffice:service-provider:update",
            "generatedAt": "2025-11-20T15:17:07.700549864Z",
            "key": 3,
            "payload": {
            "userSub": "CR-ID-CRI330016",
            "arguments": {
            "id": 3,
            "body": {
            "id": 3,
            "clientId": "sp-client-45654",
            "clientIdIssuedAt": "2025-10-30T15:02:22.677616Z",
            "clientSecretExpiresAt": null,
            "clientName": "Aplicación Gubernamental de Desarrollosss",
            "clientAuthenticationMethods":
    "[\"client_secret_basic\",\"client_secret_post\"]",
            "authorizationGrantTypes":
    "[\"authorization_code\",\"refresh_token\"]",
            "redirectUris": "[\"http://localhost:5173/success\"]",
            "postLogoutRedirectUris": "[\"http://localhost:5173/success\"]",
```

"scopes": "[\"openid\",\"profile\",\"email\",\"document\",\"address\",\"phone\",\"auth_info\"]",

"brokerOrganizationName": "Ministerio de Pruebas Digitales Desarrollo",

"brokerOrganizationId": "MIN-PRUEBAS-001",

"brokerOrganizationCountry": "UY",

"brokerTechnicalContact": "admin@sp-desarrollo.uy",

"brokerActive": true,

"brokerServiceCategory": "government",

"brokerServiceDescription": "Aplicación de Desarrollo para validar integración OIDC del broker",

"brokerRegisteredAt": "2025-10-30T15:02:22.677616Z",

"brokerUpdatedAt": null,

"brokerRequiresExplicitConsent": false,

"brokerDefaultConsentDurationDays": 365,

"brokerIsBackoffice": false

}

},

"output": {

"headers": {},

"body": {

"id": 3,

"clientId": "sp-client-45654",

"clientIdIssuedAt": "2025-11-20T15:16:55.015563Z",

"clientSecretExpiresAt": null,

"clientName": "Aplicación Gubernamental de Desarrollosss",

"clientAuthenticationMethods": "[\"client_secret_basic\",\"client_secret_post\"]",

"authorizationGrantTypes": "[\"authorization_code\",\"refresh_token\"]",

"redirectUris": "[\"http://localhost:5173/success\"]",

"postLogoutRedirectUris": "[\"http://localhost:5173/success\"]",

"scopes": "[\"openid\",\"profile\",\"email\",\"document\",\"address\",\"phone\",\"auth_info\"]",

"brokerOrganizationName": "Ministerio de Pruebas Digitales Desarrollo",

"brokerOrganizationId": "MIN-PRUEBAS-001",

"brokerOrganizationCountry": "UY",

"brokerTechnicalContact": "admin@sp-desarrollo.uy",

"brokerActive": true,

"brokerServiceCategory": "government",

"brokerServiceDescription": "Aplicación de Desarrollo para validar integración OIDC del broker",

"brokerRegisteredAt": "2025-11-20T15:16:55.015564Z",

"brokerUpdatedAt": "2025-11-20T15:17:07.697586837Z",

"brokerRequiresExplicitConsent": false,

"brokerDefaultConsentDurationDays": 365,

"brokerIsBackoffice": false

},

"statusCode": "OK",

"statusCodeValue": 200

}

}

}

Eliminar SP

{

"@timestamp": "2025-11-20T12:17:16.261462723-03:00",

"logger_name": "com.redgealc.identitybroker.audit.events",

"thread_name": "https-jsse-nio-8443-exec-6",

"level": "INFO",

"event": {

"type": "backoffice:service-provider:delete",

"generatedAt": "2025-11-20T15:17:16.261318164Z",

"key": 3,

```
"payload": {
"userSub": "CR-ID-CRI330016",
"arguments": {
"id": 3
},
"output": {
"headers": {},
"body": null,
"statusCode": "NO_CONTENT",
"statusCodeValue": 204
}
}
}
}
```


Actualización de parámetros globales


```
{
"@timestamp": "2025-11-20T12:17:29.421188717-03:00",
"logger_name": "com.redgealc.identitybroker.audit.events",
"thread_name": "https-jsse-nio-8443-exec-9",
"level": "INFO",
"event": {
"type": "backoffice:config:update",
"generatedAt": "2025-11-20T15:17:29.420562099Z",
"key": 0,
"payload": {
"userSub": "CR-ID-CRI330016",
"arguments": {
"dto": {
"sessionMaxConcurrentSessionsPerUser": 1,
"tokenAccessTokenTtlSeconds": 1900,
"tokenIdTokenTtlSeconds": 1900,
"tokenRefreshTokenTtlSeconds": 29900,
```

```
"tokenAuthorizationCodeTtlSeconds": 700,
"tokenRefreshReuseEnabled": false
}
},
"output": {
"headers": {},
"body": {
"sessionMaxConcurrentSessionsPerUser": 1,
"tokenAccessTokenTtlSeconds": 1900,
"tokenIdTokenTtlSeconds": 1900,
"tokenRefreshTokenTtlSeconds": 29900,
"tokenAuthorizationCodeTtlSeconds": 700,
"tokenRefreshReuseEnabled": false
},
"statusCode": "OK",
"statusCodeValue": 200
}
}
}
}
```


8.5. Alta Disponibilidad (HA)

Objetivo

Este apartado describe los mecanismos de **Alta Disponibilidad (HA)** implementados en la plataforma del Broker RedGealc, tanto en su infraestructura base como en los componentes de aplicación.
El objetivo es asegurar la continuidad del servicio frente a fallos de nodo, fallos de zona, picos de tráfico o interrupciones en elementos individuales.

Arquitectura general de Alta Disponibilidad

El Broker está diseñado como un servicio **stateless** en su capa de aplicación, desplegado sobre contenedores **ECS Fargate**, con almacenamiento persistente

provisto por **PostgreSQL** y cache distribuida en **Redis**.
La arquitectura sigue el patrón habitual de un proveedor OIDC moderno: múltiples réplicas del backend detrás de un balanceador, utilización de storage externo administrado, y separación de concerns entre estado transitorio y persistencia.

Componentes con HA activa:

- Application Load Balancer (ALB)
- Múltiples tareas de ECS Fargate (réplicas del Broker)
- PostgreSQL en modo multi AZ
- Redis (MemoryDB/ElastiCache) en clúster multi AZ
- Servicios de DNS en Route 53

Componentes stateless:

- Frontend del Broker (Thymeleaf)
- Backoffice API
- Backoffice React
- Proveedor OIDC
- Cliente OIDC

Los componentes stateless permiten escalar horizontalmente sin dependencia entre instancias.

Balanceo y distribución de tráfico

Application Load Balancer (ALB)
El ALB distribuye todo el tráfico entrante entre múltiples tareas ECS Fargate activas.
Funciones clave:

- Health checks periódicos para detectar instancias no saludables.
- Redirección automática a instancias sanas.
- Sticky sessions desactivadas (el Broker es stateless).

En caso de caída de una instancia Fargate, el ALB desviará todo el tráfico hacia las restantes sin interrupción del servicio.

Capa de aplicación: ECS Fargate

Despliegue multi-réplica

El servicio ECS del Broker se configura con al menos $N \geq 2$ réplicas activas para garantizar tolerancia a fallos.

Si una tarea falla o se detiene, ECS automáticamente:

1. La marca como unhealthy

2. Levanta otra tarea en su lugar
3. El ALB deja de enviarla tráfico

Recuperación automática

ECS Fargate garantiza:

- auto-reemplazo de tareas fallidas
- reinicios automáticos
- sin dependencia entre instancias

Esto elimina riesgos clásicos de "single point of failure" por host.

Persistencia: PostgreSQL

El Broker utiliza una base **PostgreSQL** administrada (RDS o equivalente), configurada en **multi-AZ**.

Beneficios HA:

- Un nodo principal en AZ1
- Un nodo en espera sincronizado en AZ2
- Failover automático en segundos ante caída del primario
- Replicación síncrona evitando pérdida de datos
- Backups automatizados

Todas las transacciones críticas (usuarios, SP, IDP, autorizaciones, tokens, auditoría mínima) dependen de la persistencia garantizada por Postgres.

Cache y estado transitorio: Redis

Redis (MemoryDB/ElastiCache) se utiliza para:

- sesiones de autenticación
- estado temporal durante flujos OIDC
- metadata cacheada
- validaciones

El clúster está configurado en modo multi-AZ con failover automático:

- Nodo primario en AZ1
- Réplicas en AZ2
- Promoción automática en caso de falla

- Latencia estable entre nodos

Esto garantiza que la capa de estado temporal del Broker no sea un punto único de falla.

DNS y resiliencia global

La plataforma utiliza Amazon Route 53:

- Health checks opcionales
- Failover routing si se configura un ambiente secundario
- Tiempo de vida de registros optimizado para cambios rápidos
- DNS altamente disponible a nivel global

Esto aporta resiliencia incluso frente a fallos de zona o red.

Patrones de HA aplicados a la aplicación

El diseño interno del Broker sigue patrones explícitos de HA:

Stateless Application Layer

Todas las instancias del Broker son intercambiables.
No almacenan estado en memoria local y no dependen de sesiones pegadas (session affinity).

Externalización de estado

- Estado duradero → PostgreSQL
- Estado transitorio → Redis
- Configuración → Base de datos + variables de entorno
- Sesiones OIDC → Redis

Idempotencia en flujos críticos

- Validación de código OIDC
- Canje de tokens
- Llamadas a IDP externos
- Endpoints de logout

Permite repetir llamadas sin producir inconsistencias.

Failover transparente al SP

Desde el punto de vista del Service Provider:

- no necesita lógica extra
- no percibe cortes si falla una instancia del Broker
- obtiene siempre los endpoints correctos gracias al ALB

Escalabilidad (complementa la HA)

La HA se refuerza con escalabilidad elástica:

ECS Fargate soporta:

- Auto scaling por CPU
- Auto scaling por memoria
- Auto scaling por cantidad de requests al ALB
- Escalado manual para eventos especiales

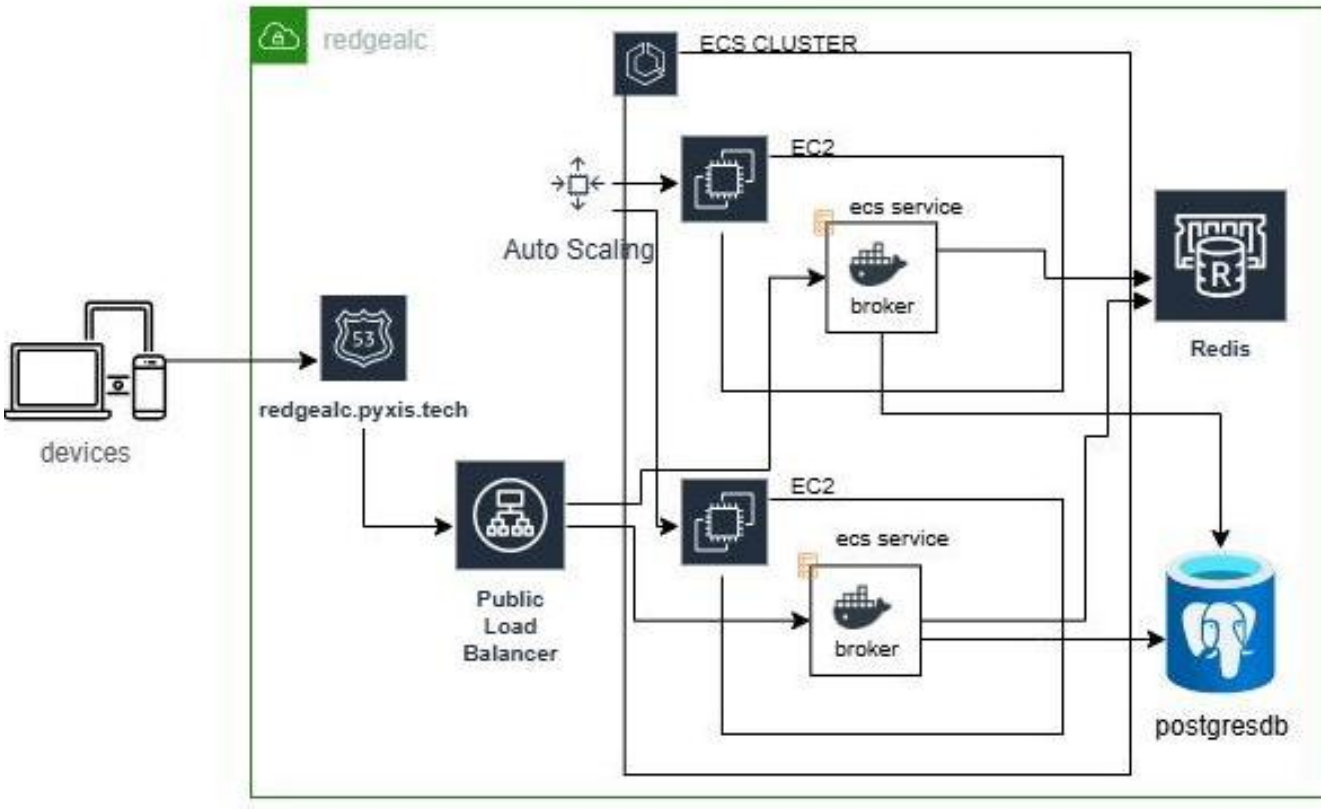
Redis y Postgres permiten:

- aumentar tamaño del clúster sin downtime
- aumentar capacidad de I/O
- añadir réplicas

Esto permite absorber picos de tráfico sin afectar disponibilidad.

8.6. Referencia en AWS

El proveedor de desarrollo (Pyxis) para pruebas con los países de la región utilizó este esquema de arquitectura en AWS para la pruebas de integraciones con otros países



Componentes desplegados:

- ECS Cluster: cluster de contenedores Docker
- EC2: nodos de cluster ECS
- Ecs_service: contenedores de cluster ECS
- AutoScaling Group: componente para escalado de instancias ID Broker
- Redis: base de datos de cache
- PostgreDB: base de datos de configuración de ID Broker
- ALB: Load Balancer para alta disponibilidad de la solución
- Route53: DNS para resolución de nombres.

8.7. Recursos de pruebas y validación

Colecciones de Postman

Como recurso complementario para desarrolladores, integradores y equipos de QA, se incluye un archivo que contiene las **colecciones de Postman** utilizadas para probar todos los endpoints expuestos por el broker.

Estas colecciones permiten:

- ejecutar los flujos principales de OAuth2 y OpenID Connect (autorización, emisión de tokens, revocación, etcétera)
- validar la configuración y el comportamiento de cada Service Provider e Identity Provider

- probar los endpoints administrativos del backoffice

El archivo puede importarse directamente en Postman o en herramientas compatibles con el formato estándar de colecciones.

ver /Adjuntos/redgelac.postman_collection.json