

IMAP Worker - Documentación

Descripción General

El IMAP Worker es un servicio interno del backend que ingiere correos electrónicos de forma idempotente desde un servidor IMAP. Solo procesa mensajes cuando hay un turno activo y ejecuta un backlog al iniciar cada turno.

Configuración

Variables de Entorno

Agregar en `/apps/api/.env` :

<code># IMAP Configuration</code>	
<code>IMAP_HOST=imap.example.com</code>	<code># Servidor IMAP</code>
<code>IMAP_PORT=993</code>	<code># Puerto (993 para SSL/TLS)</code>
<code>IMAP_USER=pedidos@example.com</code>	<code># Usuario/email</code>
<code>IMAP_PASSWORD=your_password_here</code>	<code># Contraseña</code>
<code>IMAP_FOLDER=INBOX</code>	<code># Carpeta a monitorear</code>
<code>IMAP_POLL_SECONDS=15</code>	<code># Intervalo de polling en segundos</code>
<code>IMAP_SECURE=true</code>	<code># Usar conexión segura (SSL/TLS)</code>

Funcionamiento

Ciclo de Vida del Worker

1. **Inicio:** El worker se inicia automáticamente cuando arranca el servidor API
2. **Conexión:** Se conecta al servidor IMAP configurado
3. **Polling:** Cada `IMAP_POLL_SECONDS` segundos verifica si hay turno activo
4. **Ingesta:** Si hay turno activo, procesa mensajes nuevos
5. **Shutdown:** Se detiene gracefully al recibir SIGTERM/SIGINT

Estrategia de Idempotencia

La idempotencia se garantiza mediante:

1. **Unique Index:** La tabla `lote` tiene un índice único en `(imap_uidvalidity, imap_uid)`
2. **Verificación previa:** Antes de crear un lote, se verifica si ya existe
3. **Eventos de duplicado:** Se registra `DUPPLICADO_IMAP_IGNORADO` cuando se detecta un duplicado

Manejo de UIDValidity

El `uidValidity` es un valor que cambia cuando el buzón se recrea o se modifica estructuralmente:

1. El worker guarda el `uidValidity` actual junto con el cursor
2. Si cambia el `uidValidity`, se resetea el cursor a 0
3. Esto evita confusión con UIDs que podrían haberse reasignado

Cursor IMAP

El cursor se almacena en la tabla `imap_state` :

```
CREATE TABLE imap_state (
    id uuid PRIMARY KEY,
    mailbox text DEFAULT 'INBOX',
    uidvalidity bigint NULL,
    last_uid bigint DEFAULT 0,
    last_poll_at timestampz NULL
);
```

Backlog al Iniciar Turno

Cuando se inicia un nuevo turno:

1. Se crea el turno con estado ACTIVO
2. Se ejecuta `imapWorker.executeBacklog(turnoId)`
3. Procesa todos los mensajes con UID > lastUid
4. No bloquea el inicio de turno si hay errores

Manejo de Errores

Errores de Conexión

- Se aplica backoff exponencial (1s inicial, máx 60s)
- Se reintenta la conexión automáticamente
- El error se registra en `lastError`

Errores al Procesar Mensajes

- Se crea un lote con `parse_status = 'ERROR_PARSE'`
- Se registra el evento `ERROR_EN_LECTURA_DE_CORREO`
- Se continúa con el siguiente mensaje

Backoff Exponencial

```
Delay inicial: 1 segundo
Multiplicador: 2x en cada error
Delay máximo: 60 segundos
Reset: Al conectar exitosamente
```

API Endpoints

GET /api/imap/status

Permisos: Solo DIOS

Respuesta:

```
{
  "isRunning": true,
  "isConnected": true,
  "lastError": null,
  "lastPollTime": "2024-01-15T10:30:00Z",
  "cursor": {
    "lastUid": 1234,
    "uidValidity": 567890
  }
}
```

POST /api/imap/force-poll

Permisos: Solo DIOS

Fuerza un ciclo de polling inmediato.

Respuesta:

```
{
  "success": true,
  "message": "Polling forzado ejecutado",
  "timestamp": "2024-01-15T10:30:05Z"
}
```

POST /api/imap/restart

Permisos: Solo DIOS

Reinicia el worker IMAP (stop + start).

Respuesta:

```
{
  "success": true,
  "message": "IMAP worker reiniciado",
  "timestamp": "2024-01-15T10:30:10Z"
}
```

Eventos Registrados

Evento	Descripción
NUEVO_CORREO_RECIBIDO	Nuevo lote creado exitosamente
ERROR_EN_LECTURA_DE_CORREO	Error al procesar un mensaje
DUPLICADO_IMAP_IGNORADO	Mensaje duplicado ignorado por idempotencia

Troubleshooting

El worker no se conecta

1. Verificar credenciales IMAP en `.env`

2. Verificar que el servidor IMAP permite conexiones desde la IP del servidor
3. Verificar puerto y configuración de seguridad (TLS/SSL)
4. Revisar `lastError` en `/api/imap/status`

No se procesan mensajes nuevos

1. Verificar que hay un turno ACTIVO
2. Verificar el cursor con `/api/imap/status`
3. Forzar un polling con `/api/imap/force-poll`
4. Revisar logs del servidor

Mensajes duplicados

Los mensajes no se duplican gracias al índice único. Si un mensaje aparece dos veces:

1. Verificar que no cambió el `uidValidity`
2. El sistema registra `DUPPLICADO_IMAP_IGNORADO` automáticamente

Reiniciar el worker

1. Usar el botón “Reiniciar Worker” en la UI de DIOS
2. O llamar a `POST /api/imap/restart`
3. O reiniciar todo el servidor API

Archivos Relacionados

- `/apps/api/src/services/imap-worker.ts` - Implementación del worker
- `/apps/api/src/lib/imap-cursor.ts` - Gestión del cursor IMAP
- `/apps/api/src/routes/imap.ts` - Endpoints de API
- `/apps/web/src/pages/Dios.tsx` - Monitor IMAP en UI

Notas Importantes

1. **No se hace parsing de correos en TASK 04** - Solo se guarda `subject_raw` y `body_raw`. El parsing real se implementará en TASK 05.
2. **parse_status = ‘OK’ por defecto** - A menos que falle la lectura del mensaje, en cuyo caso será `ERROR_PARSE`.
3. **El worker corre dentro del proceso del API** - No es un servicio separado.
4. **Graceful Shutdown** - El worker se detiene correctamente al recibir SIGTERM/SIGINT.