

Flujo P2P y ejemplos de sincronización

Flujo general de P2P

El flujo central de sincronización P2P en RutaPAE es:

1. Inicializar `DomainManager`.
2. Iniciar `P2PManager.start()`.
3. Detectar peers y hotspots.
4. Conectar con la máquina deseada.
5. Descargar entregas pendientes.
6. Confirmar entregas sincronizadas al dispositivo remoto.

Diagrama de conexión P2P

```
flowchart TD
    A[RutaPAE inicia] --> B[DomainManager.init()]
    B --> C[P2PManager.start()]
    C --> D[Descubrir peers y hotspots]
    D --> E{Máquina conocida?}
    E -->|Sí| F[connectMachine]
    E -->|No| G[connectDiscoveredMachine]
    G --> H[resolveConnectedPeerAddress]
    F --> H
    H --> I[SyncDeliveriesFromMachine]
    I --> J[GET /p2p/machine/deliveries/<page>/<size>]
    I --> K[GET /p2p/machine/delivery/<id>]
    I --> L[GET /p2p/machine/delivery/synchronized/<id>]
    L --> M[Eliminar entrega local sincronizada]
```

Ejemplo 1: conectar una máquina conocida

1. El usuario elige una máquina de la lista.
2. RutaPAE llama `P2PManager.connectMachine(machineDatabaseId)`.
3. `P2PManager` intenta resolver la dirección del peer conectado.
4. Si no hay peer conectado, busca en `discoveredPeers`.
5. Si hay hotspot, intenta `connectHotspot`.
6. Una vez conectado, `syncDeliveriesFromMachine()` puede iniciar.

Ejemplo 2: conectar una máquina descubierta

1. Aparece un peer desconocido en `discoveredPeers`.
2. Se muestra como candidato en la UI.

3. El usuario solicita conexión.
4. Se invoca `P2PManager.connectDiscoveredMachine(address)`.
5. El gestor intenta conectar y espera que el peer pase a `machines`.

Ejemplo 3: unir a un hotspot de máquina

1. El sistema detecta `hotspotCandidates`.
2. Si el hotspot es abierto, o tiene `passphrase`, se intenta conectar automáticamente.
3. Si no puede auto-unir, busca un peer descubierto como fallback.
4. Tras unirse, espera que aparezca el peer en `machines`.

Tipos de descubrimiento

- `discoveredPeers`: peers P2P que han sido detectados pero no necesariamente conectados.
- `machines`: peers ya reconocidos como máquinas (`MACHINE-...`).
- `operators`: peers reconocidos como operadores (`OPERATOR-...`).
- `hotspotCandidates`: redes Wi-Fi creadas por una máquina.

Sincronización de entregas

Iteración de páginas

- `P2PManager` solicita `DELIVERIES_PAGE(page, size)`.
- En el proyecto actual, `pageSize` se fija en 1.
- Obtiene `deliveryIds` y procesa cada `DELIVERY(id)`.

Creación local

Para cada entrega remota:

- Si no existe localmente, se descarga `P2PDeliveryData`.
- Se crea un `Delivery` local con `idLocalDeliveryT = remoteDeliveryId`.
- Si ya existe, se marca como `alreadyPresent`.

Confirmación de entrega sincronizada

Una vez que `RutaPAE` confirma una entrega remota sincronizada:

- Llama `notifyDeliverySynchronized(machineDatabaseId, remoteDeliveryId)`.
- Esto invoca `DELIVERY_SYNCHRONIZED(id)`.
- Si la máquina responde `deleted: true`, la entidad local puede borrarse.

Resiliencia de flujo

- El proceso monitorea la conexión con `monitorMachineSyncConnection()`.
- Si se pierde la conexión, se pausa la sincronización y mantiene el estado de UI.
- Los mensajes de estado se muestran en la UI usando `DeliverySyncUiEmitter`.

Ejemplo de actualización de configuración

1. El operador solicita la configuración de la máquina.
2. `getMachineConfiguration(machineDatabaseId)` llama a GET `/p2p/machine/configuration`.
3. El operador modifica valores locales.
4. `updateMachineConfiguration(machineDatabaseId, configuration)` hace PUT `/p2p/machine/configuration`.
5. La máquina responde con `P2PMachineConfigurationUpdateResponse`.

From:
<http://wiki.adacsc.co/> - Wiki

Permanent link:
<http://wiki.adacsc.co/doku.php?id=ada:howto:sicoferp:factory:new-migracion-sicoferp:p2p-flow>

Last update: **2026/04/07 20:01**

