

Arquitectura P2P

Contrato P2P

El contrato P2P define nombres, topics y rutas compartidas entre máquina y ruta.

Archivo principal:

- `Contract/src/main/java/co/ada/contract/P2PContract.kt`

Nombres de peer

- `P2PContract.Names.MACHINE_NAME = "MACHINE-"`
- `P2PContract.Names.OPERATOR_NAME = "OPERATOR-"`

Topic principal

- `P2PContract.Topics.MACHINE_STATE = "p2p.machine.state"`

Paths P2P

- `MACHINE_WEIGHT = "/p2p/machine/weight"`
- `MACHINE_CONFIGURATION = "/p2p/machine/configuration"`
- `DELIVERIES_PAGE(page, size) = "/p2p/machine/deliveries/$page/$size"`
- `DELIVERY(id) = "/p2p/machine/delivery/$id"`
- `DELIVERY_SYNCHRONIZED(id) = "/p2p/machine/delivery/synchronized/$id"`

Modelos compartidos

Archivo principal:

- `Contract/src/main/java/co/ada/contract/P2PModels.kt`

Modelos clave

- `P2PMachineState`
 - `id`, `name`, `unsyncedDeliveries`, `machineStatus`, `stateName`, `shiftName`, etc.
- `P2PWeight`
 - `weight` para actualización de peso.
- `P2PDeliveriesPageResponse`
 - lista de `deliveryIds` y metadatos de paginación.
- `P2PDeliveryData`
 - detalle completo de una entrega.

- P2PDeliveryDeletionResponse
 - respuesta de confirmación de eliminación.
- P2PMachineConfiguration
 - parámetros de la máquina: umbrales, debug, tiempos.
- P2PMachineConfigurationUpdateRequest / Response
 - payload para la actualización de configuración.

Gestión de conexión

Archivo principal:

- Contract/src/main/java/co/ada/contract/P2PGestor.kt

"P2PGestor"

Responsabilidades:

- inicializar `DirectConnection`.
- establecer el nombre local de peer.
- habilitar auto-conexión.
- exponer peers, machines y operators.
- recoger eventos de descubrimiento.

"machines" y "operators"

- machines: peers que comienzan con MACHINE-.
- operators: peers que comienzan con OPERATOR-.

Implementación del gestor de RutaPAE

Archivo principal:

- RutaPAEDomain/src/main/java/co/ada/rutapaedomain/p2p/P2PManager.kt

Funciones principales

- `start()`: arranca el P2P, consolida alias de máquinas y suscribe a eventos.
- `close()`: detiene el gestor y cancela jobs.
- `discoverableMachineIds()`: IDs locales de máquinas detectables.
- `discoveredMachineCandidates()`: máquinas detectadas sin registro local.
- `discoveredMachineHotspots()`: hotspots de máquina disponibles.
- `connectMachine(machineDatabaseId)`: conecta a una máquina conocida.
- `connectDiscoveredMachine(address)`: conecta a un peer descubierto.
- `connectMachineHotspot(ssid)`: une y resuelve un hotspot de máquina.

- `getMachineConfiguration(machineDatabaseId)`: obtiene configuración desde la máquina.
- `updateMachineConfiguration(...)`: actualiza parámetros de configuración.
- `notifyDeliverySynchronized(...)`: marca una entrega remota como sincronizada.
- `syncDeliveriesFromMachine(machineDatabaseId)`: descarga entregas remotas desde la máquina.

Reconocimiento de identidad

- `P2PManager` extrae IDs de peer basándose en prefijos `MACHINE-`.
- Aplica normalización de IDs temporales (`tmp-...`) y IDs oficiales.
- Cuando el dispositivo reporta un `P2PMachineState.id` oficial, reconcilia alias temporales.

Conexiones P2P admitidas

- `DirectConnection.connect(peer)`: conecta con un peer ya descubierto.
- `DirectConnection.connectHotspot(hotspot)`: une a un hotspot reportado por la máquina.

Fallback hotspot

Si el hotspot no puede unirse automáticamente, la app busca un peer descubierto que represente la misma máquina y lo usa como respaldo.

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

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

Last update: 2026/04/07 19:56

