

Funcionamiento de Machine: Online vs Offline

□ Modos de Conectividad

La aplicación **Machine** puede operar en dos modos principales de conectividad: **Online** y **Offline**. Esta flexibilidad permite el funcionamiento continuo incluso en entornos con conectividad limitada.

□ Modo Online

Características Principales

Cuando Machine tiene acceso a internet, opera en **modo online** con las siguientes capacidades:

□ Conexión Backend

- **Sincronización automática** con servidor central
- **Actualización de datos** en tiempo real (beneficiarios, rutas, entregas)
- **Envío inmediato** de resultados de entregas
- **Descarga de actualizaciones** de modelos ML y configuraciones

□ Funcionalidades Avanzadas

- **Reconocimiento facial** con modelos actualizados
- **Verificación de peso** con calibración remota
- **Reportes en tiempo real** al backend
- **Notificaciones push** desde el servidor
- **Sincronización bidireccional** con RutaPAE via P2P

□ Beneficios

- **Datos actualizados** garantizados
- **Procesamiento remoto** para casos complejos
- **Backup automático** en la nube
- **Monitoreo centralizado** del estado de dispositivos

Flujo de Operación Online

```
graph TD
  A[Machine inicia] --> B{¿Conectividad?}
```

```
B -->|Sí| C[Modo Online]
C --> D[Conectar backend]
D --> E[Sincronizar datos]
E --> F[Procesar entregas]
F --> G[Enviar resultados]
G --> H[Continuar monitoreo]
```

☐ Modo Offline

Características Principales

Cuando no hay conectividad a internet, Machine automáticamente cambia a **modo offline**:

☐ Funcionamiento Local

- **Base de datos local** con datos sincronizados previamente
- **Procesamiento offline** de reconocimiento facial
- **Almacenamiento temporal** de resultados de entregas
- **Sincronización P2P** con dispositivos RutaPAE cercanos

☐ Capacidades Limitadas

- **Datos cacheados**: Beneficiarios y rutas disponibles localmente
- **Modelos ML locales**: Reconocimiento facial sin actualización
- **Almacenamiento limitado**: Resultados pendientes de sincronización
- **Funcionalidad básica**: Captura, verificación y almacenamiento

☐ Sincronización P2P

- **Wi-Fi Direct** para conexión directa con RutaPAE
- **Transferencia de datos** sin requerir internet
- **Sincronización automática** al detectar dispositivos compatibles
- **Protocolo DirectLink** para transferencia eficiente

Flujo de Operación Offline

```
graph TD
  A[Machine inicia] --> B{¿Conectividad?}
  B -->|No| C[Modo Offline]
  C --> D[Cargar datos locales]
  D --> E[Procesar entregas]
  E --> F[Almacenar localmente]
  F --> G{¿RutaPAE cercano?}
```

```
G -->|Sí| H[Sincronizar P2P]
G -->|No| I[Esperar conectividad]
I --> J[Reintentar conexión]
```

□ Transición entre Modos

Detección Automática

Machine monitorea continuamente la conectividad:

```
// En ConnectivityManager
fun checkConnectivity(): ConnectivityMode {
    return when {
        isInternetAvailable() -> ConnectivityMode.ONLINE
        isWifiDirectAvailable() -> ConnectivityMode.P2P_READY
        else -> ConnectivityMode.OFFLINE
    }
}
```

Estrategias de Transición

Online → Offline

1. **Detección de pérdida** de conectividad
2. **Cambio automático** a modo offline
3. **Notificación al usuario** del cambio de estado
4. **Continuación del flujo** con datos locales

Offline → Online

1. **Recuperación de conectividad** detectada
2. **Sincronización automática** de datos pendientes
3. **Actualización de modelos** y configuraciones
4. **Envío de resultados** acumulados

P2P Sincronización

1. **Detección de dispositivos** RutaPAE cercanos
2. **Establecimiento de conexión** Wi-Fi Direct
3. **Transferencia de datos** usando DirectLink
4. **Confirmación de sincronización** exitosa

☐ Gestión de Datos

En Modo Online

```
// Sincronización automática
suspend fun syncOnline() {
    val beneficiaries = api.getBeneficiaries()
    localDb.saveBeneficiaries(beneficiaries)

    val deliveries = localDb.getPendingDeliveries()
    api.sendDeliveries(deliveries)
    localDb.markAsSynced(deliveries)
}
```

En Modo Offline

```
// Almacenamiento local con marca de pendiente
suspend fun processOffline(delivery: Delivery) {
    val result = processDeliveryLocally(delivery)
    localDb.saveDelivery(result.copy(synced = false))

    // Intentar P2P si disponible
    if (p2pManager.hasNearbyDevices()) {
        p2pManager.syncWithRutaPAE(result)
    }
}
```

Estrategia de Cache

- **Beneficiarios:** Cache permanente (actualizado en online)
- **Rutas:** Cache temporal (válido por sesión)
- **Modelos ML:** Cache con versión (actualizado en online)
- **Resultados:** Almacenamiento temporal hasta sincronización

⚠ Limitaciones del Modo Offline

Funcionalidades Restringidas

Funcionalidad	Online	Offline	Notas
Reconocimiento facial	☐ Completo	☐ Completo	Modelos locales
Verificación de peso	☐ Calibrado	☐ Calibrado	Sin recalibración remota
Sincronización	☐ Inmediata ☐ P2P	☐ P2P	Conexión con dispositivo cercano independiente
Reportes	☐ Tiempo real	☐ Pendiente	Hasta próxima conexión

Funcionalidad	Online	Offline	Notas
Actualizaciones	<input type="checkbox"/> Automáticas	<input type="checkbox"/> Manuales	Requiere conexión

Gestión de Recursos

Almacenamiento

- **Límite de entregas pendientes:** Máximo 1000 registros
- **Rotación automática:** Eliminación de registros antiguos (>30 días)
- **Compresión:** Datos comprimidos para optimizar espacio

Memoria

- **Modelos ML optimizados:** Versión reducida para offline
- **Cache inteligente:** Liberación automática de recursos no usados
- **Procesamiento secuencial:** Evita sobrecarga de memoria

Configuración

Parámetros de Conectividad

```
// En build.gradle.kts o runtime
object ConnectivityConfig {
    const val SYNC_INTERVAL_ONLINE = 300_000L // 5 minutos
    const val SYNC_INTERVAL_P2P = 60_000L // 1 minuto
    const val MAX_OFFLINE_DELIVERIES = 1000
    const val CACHE_VALIDITY_DAYS = 30
}
```

Configuración de P2P

```
// En P2PManager
data class P2PConfig(
    val serviceName: String = "PAE-Machine",
    val port: Int = 8888,
    val timeout: Long = 30_000L,
    val maxRetries: Int = 3
)
```

Interfaz de Usuario

Indicadores de Estado

La UI muestra claramente el estado de conectividad:

- **Verde**: Modo online completo
- **Amarillo**: Modo P2P activo
- **Rojo**: Modo offline
- **Blanco**: Sin conectividad

Notificaciones

```
// Notificaciones automáticas
fun notifyConnectivityChange(mode: ConnectivityMode) {
    when (mode) {
        ONLINE -> showNotification("Conectado", "Sincronización activa")
        P2P_READY -> showNotification("P2P disponible", "Sincronizando con RutaPAE")
        OFFLINE -> showNotification("Modo offline", "Funcionalidad limitada")
    }
}
```

Monitoreo y Diagnóstico

Logs de Conectividad

```
// Logging detallado
logger.info("Connectivity changed: $previousMode -> $currentMode")
logger.debug("P2P devices found: ${devices.size}")
logger.warn("Offline mode activated - limited functionality")
```

Métricas de Rendimiento

- **Tiempo de sincronización** online vs P2P
- **Tasa de éxito** de transferencias P2P
- **Uso de almacenamiento** en modo offline
- **Latencia de procesamiento** local vs remoto

Troubleshooting

Problemas Comunes

No cambia a modo online

- Verificar conectividad de red
- Revisar configuración de proxy/firewall
- Comprobar credenciales de API

P2P no funciona

- Verificar Wi-Fi Direct habilitado
- Comprobar dispositivos RutaPAE cercanos
- Revisar permisos de ubicación

Sincronización falla

- Verificar espacio de almacenamiento
- Comprobar integridad de base de datos local
- Revisar logs para errores específicos

Comandos de Diagnóstico

```
# Ver estado de conectividad
adb shell dumpsys connectivity

# Ver logs de P2P
adb logcat | grep -i p2p

# Ver estado de sincronización
adb shell run-as co.ada.machine sqlite3
/data/data/co.ada.machine/databases/app.db "SELECT COUNT(*) FROM deliveries
WHERE synced = 0;"
```

☐ Mejores Prácticas

Para Desarrollo

1. **Probar ambos modos** en desarrollo
2. **Simular desconexión** para testing offline
3. **Implementar graceful degradation** para funcionalidades críticas
4. **Optimizar modelos ML** para funcionamiento offline

Para Producción

1. **Monitorear conectividad** continuamente
2. **Configurar alertas** para cambios de modo

3. **Planificar capacidad** de almacenamiento offline
4. **Documentar procedimientos** de recuperación

Para Usuarios

1. **Mantener dispositivos actualizados** cuando hay conectividad
2. **Verificar sincronización** antes de desconectar
3. **Reportar problemas** de conectividad inmediatamente
4. **Entender limitaciones** del modo offline

Referencias

- [Arquitectura P2P](#)
- [Flujo P2P](#)
- [Almacenamiento Local](#)
- [Gestión de Estado](#)
- [Hardware Integration](#)

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

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

Last update: **2026/04/07 19:55**

