

Estados de flujo de entrega en MachineDomain

Visión general

En MachineDomain, el dispositivo máquina ejecuta una máquina de estados (StateManager) que controla el flujo completo de cada entrega. Los estados son secuenciales y cada uno es responsable de un aspecto diferente del procesamiento de entregas.

Antes de que esa máquina de estados arranque, la app ahora resuelve una **jornada operativa visible** a partir de MinutaMenuDue + MachineEnrollmentShift. Esto deja la experiencia alineada con la Máquina #1: el operador elige una jornada del día y no una asignación técnica cruda.

Archivo principal:

- MachineDomain/src/main/java/co/ada/domain/state/StateManager.kt

Interfaz "State"

Cada estado implementa la interfaz State:

```
interface State {
    val name: String
    suspend fun run(
        next: () -> Unit,
        retry: (String) -> Unit
    ): Unit
}
```

- name: identificador del estado.
- run(): ejecuta la lógica del estado.
 - next(): avanza al siguiente estado.
 - retry(): retrocede y emite un mensaje de error.

Estados de la máquina de entregas

Los estados se ejecutan en este orden:

1. "WaitingForWeight"

- Espera a que se coloque un peso en la balanza.
- Verifica que el flujo anterior esté completado.

2. "CaptureImages"

- Captura imágenes del producto/entrega desde la cámara.
- Almacena las fotos localmente en rutas del dispositivo.

3. "CaptureFace"

- Captura la cara del beneficiario mediante la cámara frontal.
- Genera datos biométricos para verificación.

4. "ComparingWeights"

- Compara el peso capturado con el peso esperado.
- Valida que el peso esté dentro de rangos aceptables.
- Si hay desviación, puede requerir recaptura.

5. "GenerateEmbedding"

- Genera un embedding vectorial de 512 dimensiones de la cara del beneficiario.
- Usa modelo de IA local para la extracción de características.
- Almacena el embedding en la base de datos local.

6. "VerifyInDatabase"

- Consulta la base de datos local para verificar que el beneficiario existe.
- Si hay múltiples coincidencias de embeddings similares, puede requerir confirmación.
- Calcula similitud entre el embedding capturado y los registrados.

7. "SaveDelivery"

- Guarda la entrega en la base de datos local.
- Registra timestamp, fotos, peso, embedding, beneficiario.
- Marca la entrega como lista para sincronizar.

8. "WaitForWeightRemoved"

- Espera a que se retire el peso de la balanza.
- Valida que el ciclo esté completo antes de reiniciar.

Flujo de control del "StateManager"

Inicio

```
↓
[Selección de jornada operativa]
↓
[Estado 1] → [Estado 2] → ... → [Estado 8]
  ↓ (next)                ↓ (next)
Avanza                    Reinicia desde Estado 1
  ↓ (retry)
Retrocede a Estado 1 + emite error
```

Comportamiento

- Cuando un estado llama `next()`, el `StateManager` avanza al siguiente.
- Cuando llama `retry(reason)`, retrocede a Estado 1 y emite mensaje de error.
- Si hay excepción no controlada, se registra y se retrocede.
- Los estados son síncronos pero se ejecutan en contexto de corrutina.

Emisores asociados

- `StateNameEmitter`: emite el nombre del estado actual para la UI.
- `StateMessageEmitter`: emite mensajes de error o información.
- `StateContextEmitter`: almacena contexto compartido entre estados.

Reintento y recuperación

Si en algún punto hay un error:

1. Se invoca `retry(reason)`.
2. Se resetea `currentIndex = 0`.
3. Se retira al usuario a `WaitingForWeight`.
4. Se muestra el mensaje de error.
5. El usuario puede reintentar.

Integración con base de datos

- `MachineData/database.db` almacena entregas, beneficiarios, asignaciones técnicas (`MachineEnrollmentShift`) y jornadas operativas del día (`MinutaMenuDue`).
- `VectorDB` (ORM local) indexa embeddings para búsquedas rápidas.

Nota sobre jornadas

- `MachineEnrollmentShift` sigue siendo la referencia interna para guardar entregas, beneficiarios y actividad.
- `OperationalShift` es una proyección en memoria usada por la UI.
- Si varias asignaciones corresponden a la misma jornada del día, la UI las agrupa y activa un

representante para mantener intacto el flujo interno de la Maquina #2.

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

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

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

