

Logica Sincro Zucchetti <-> Soluziona

- [Sincronizzazione Liste di Carico Zucchetti ↔ Soluziona](#)
- [Sincronizzazione Liste di Carico Soluziona → Zucchetti](#)

Sincronizzazione Liste di Carico Zucchetti ? Soluziona

Obiettivo

La gestione delle liste di carico nasce nel gestionale **Zucchetti** e prosegue operativamente in **Soluziona**, tramite applicazione tablet.

Il flusso previsto è:

```
graph TD;
    A[Zucchetti] --> B[SincronizzaListeDaZucchettiASoluziona];
    B --> C["Soluziona - Documento TipoDocumento = 29"];
    C --> D[Compilazione tramite Tablet];
    D --> E[SincronizzaListeDaSoluzionaAZucchetti];
    E --> F[Zucchetti];
```

La sincronizzazione deve consentire a Zucchetti di continuare a modificare le liste fino alla loro conclusione, senza però perdere o sovrascrivere le informazioni già raccolte dal tablet.

Tabelle coinvolte

Zucchetti / SoluzionaComside

Testata lista

```
TestaListaDiCaricoTb
```

Chiave:

S2serial

Campi principali:

S2serial
StatoLettura
Note
Eliminato

Righe lista

RigaListaDiCaricoTb

Chiave:

S2serial + S2rownum

Campi principali:

S2serial
S2rownum
s2codcli
IdArticolo
DesArticolo
UmArticolo
Quantita
QuantitaLetta
Note
StatoLettura

Tabelle Soluziona

Le liste di carico vengono trasformate in documenti Soluziona con:

TipoDocumento = 29

Le tabelle coinvolte sono:

DocumentoTb
DettDocumentoTb
DecodificaGestionaleEsternoTb

In `DettDocumentoTb`:

QntDaLeggere

rappresenta la quantità prevista proveniente da Zucchetti.

Qnt

rappresenta la quantità effettivamente letta tramite tablet.

Regola cliente

In Zucchetti il cliente è presente sulla singola riga della lista.

In Soluziona, invece, il cliente appartiene alla testata del documento.

La regola applicativa è:

1 Documento Soluziona = 1 Cliente

Di conseguenza, se una lista Zucchetti contiene righe appartenenti a clienti diversi, vengono creati più documenti Soluziona.

Esempio:

Lista Zucchetti L00001

Riga 1 → Cliente A

Riga 2 → Cliente A

Riga 3 → Cliente B

Riga 4 → Cliente A

diventa:

Documento Soluziona A

Cliente A

└ Riga 1

└ Riga 2

└ Riga 4

Documento Soluziona B

Cliente B

└ Riga 3

Entrambi i documenti continuano comunque a derivare dalla stessa lista Zucchetti:

S2serial = L00001

Stati della lista

Gli stati utilizzati sono:

```
<table id="bkmrk-stato-significato-0-"><thead><tr><th align="right">Stato</th><th>Significato</th></tr></thead><tbody><tr><td align="right">0</td><td>ND</td></tr><tr><td align="right">1</td><td>Attivo</td></tr><tr><td align="right">2</td><td>Da convalidare</td></tr><tr><td align="right">3</td><td>Convalidato</td></tr><tr><td align="right">4</td><td>Autorizzato</td></tr></tbody></table>
```

Gli stati hanno significati differenti durante il passaggio tra Zucchetti e Soluziona.

Creazione

La lista viene inizialmente creata in Zucchetti con:

StatoLettura = 0

La sincronizzazione crea i relativi documenti Soluziona.

Finché la lista non viene iniziata dal tablet, Zucchetti rimane completamente autorevole sui dati.

Inizio lavorazione dal tablet

Alla prima lettura effettuata tramite tablet, Soluziona porta la lista a:

Stato = 2

La sincronizzazione:

SincronizzaListeDaSoluzionaAZucchetti

deve quindi portare la lista Zucchetti a:

StatoLettura = 1

Da questo momento la lista è considerata operativamente iniziata.

Chiusura della lista

Alla chiusura della lista tramite tablet, Soluziona imposta:

Stato = 3

La sincronizzazione inversa deve quindi portare Zucchetti a:

StatoLettura = 4

Quando Zucchetti raggiunge:

StatoLettura = 4

il processo è considerato concluso.

La sincronizzazione:

SincronizzaListeDaZucchettiASoluziona

non deve più modificare i documenti Soluziona relativi alla lista.

Sincronizzazione Zucchetti ? Soluziona

La sincronizzazione cambia comportamento in base allo stato della lista.

Lista non ancora iniziata

Condizione:

Zucchetti StatoLetture = 0

e corrispondente documento Soluziona ancora non operativo:

Stato = 0 oppure 1

In questa fase Zucchetti è completamente autorevole.

Possono quindi essere aggiornati direttamente:

Cliente
Articolo
Descrizione articolo
Unità di misura
Quantità prevista
Note
Numero riga
altri dati provenienti da Zucchetti

La quantità prevista viene sincronizzata in:

DettdocumentoTb.QntDaLeggere

La quantità letta:

DettdocumentoTb.Qnt

non deve essere valorizzata da Zucchetti.

Alla creazione della riga:

```
QntDaLeggere = Quantita Zucchetti  
Qnt = 0
```

Eliminazione di una riga prima dell'inizio

Se una riga precedentemente sincronizzata non è più presente in Zucchetti e la lista non è ancora stata iniziata, la riga Soluziona viene eliminata logicamente:

```
DettDocumentoTb.IsEliminato = 1
```

Il relativo mapping viene disattivato:

```
DecodificaGestionaleEsternoTb.IsAttivo = 0
```

Non viene effettuato un `DELETE` fisico.

Cambio articolo prima dell'inizio

Se la lista è ancora in stato iniziale, Zucchetti può modificare direttamente l'articolo della riga.

La sincronizzazione può quindi aggiornare la stessa `DettDocumentoTb`:

```
IdArticolo  
Descrizione  
IdUM  
QntDaLeggere  
Note
```

Non è necessario creare una nuova riga Soluziona perché nessuna lavorazione tablet è ancora iniziata.

Cambio cliente prima dell'inizio

Anche il cliente può essere modificato liberamente finché la lista non è stata iniziata.

Poiché in Soluziona il cliente appartiene alla testata del documento, la sincronizzazione deve assicurarsi che la riga appartenga al documento corretto.

In questa fase la riga può essere spostata sul documento del nuovo cliente.

Se il documento per:

S2serial + nuovo cliente

non esiste, viene creato.

Lista già iniziata

Quando il tablet ha iniziato la lavorazione:

Soluziona Stato = 2

e successivamente:

Zucchetti StatoLettura = 1

la sincronizzazione continua a recepire le modifiche provenienti da Zucchetti, ma deve preservare i dati operativi raccolti dal tablet.

I dati protetti sono:

Qnt
Note

Questi campi non devono più essere sovrascritti da:

SincronizzaListeDaZucchettiASoluziona

Quantità prevista modificata dopo l'inizio

`QntDaLeggere` continua invece a essere controllata da Zucchetti e può cambiare anche dopo l'inizio della lavorazione.

Esempio iniziale:

```
Zucchetti Quantita = 10
```

```
Soluziona:
```

```
QntDaLeggere = 10
```

```
Qnt = 4
```

Zucchetti modifica la quantità:

```
Quantita = 15
```

Dopo la sincronizzazione:

```
QntDaLeggere = 15
```

```
Qnt = 4
```

La quantità letta rimane invariata.

Note dopo l'inizio

Una volta iniziata la lavorazione, le note presenti in Soluziona appartengono al processo tablet.

Di conseguenza:

```
DettDocumentoTb.Note
```

non deve più essere sovrascritta dalla sincronizzazione proveniente da Zucchetti.

Cambio articolo dopo l'inizio

Dopo l'inizio della lavorazione l'articolo non può essere sostituito direttamente sulla stessa riga Soluziona.

La riga potrebbe infatti contenere quantità già lette riferite al vecchio articolo.

Esempio:

```
Riga Soluziona
Articolo A
QntDaLeggere = 10
Qnt = 4
```

Zucchetti modifica la stessa riga in:

```
Articolo B
Quantita = 8
```

La sincronizzazione deve:

1. mantenere la vecchia riga;
2. impostare:

```
QntDaLeggere = 0
```

sulla vecchia riga;

3. non modificare:

```
Qnt
Note
Articolo
```

della vecchia riga;

4. creare una nuova `DettdocumentoTb` per il nuovo articolo:

```
Articolo B
QntDaLeggere = 8
```

Qnt = 0

La nuova riga diventa la rappresentazione attiva della riga Zucchetti.

Cambio cliente dopo l'inizio

Il cambio cliente segue la stessa filosofia del cambio articolo.

La riga già lavorata non deve essere spostata da un documento all'altro.

Esempio iniziale:

Documento Cliente A

Riga X

Articolo A

QntDaLeggere = 10

Qnt = 4

Zucchetti cambia il cliente della riga:

Cliente B

La sincronizzazione deve lasciare la vecchia riga nel documento del Cliente A:

Documento Cliente A

Riga X

QntDaLeggere = 0

Qnt = 4

e creare una nuova riga nel documento del Cliente B:

Documento Cliente B

Nuova Riga

QntDaLeggere = 10

Qnt = 0

Se il documento del Cliente B non esiste, deve essere creato.

In questo modo non viene alterato lo storico delle letture già effettuate.

Eliminazione di una riga dopo l'inizio

Se una riga viene eliminata da Zucchetti quando la lavorazione è già iniziata, non deve essere eliminata in Soluziona.

La sincronizzazione deve invece impostare:

QntDaLeggere = 0

La riga mantiene:

Qnt
Note
Articolo
Descrizione

e gli altri dati storici necessari a rappresentare ciò che il tablet aveva già lavorato.

Il mapping della riga viene disattivato:

IsAttivo = 0

Riapertura di una lista già completata

Una lista già completata può essere riaperta da Zucchetti per aggiungere o modificare articoli, quantità, clienti o altre informazioni previste.

La situazione iniziale è:

Soluziona Stato = 3
Zucchetti StatoLettura = 4

Se l'utente Zucchetti riporta volontariamente la testata a:

Zucchetti StatoLettura = 0

la variazione viene interpretata come **riapertura della lista**.

La riapertura non deve creare una nuova lavorazione da zero e non deve perdere le letture già effettuate dal tablet.

Il documento Soluziona passa quindi:

Soluziona Stato 3 → 2

e non:

Soluziona Stato 3 → 0

In questo modo la lista torna modificabile, ma continua a essere considerata **già iniziata**.

Dati conservati alla riapertura

Le letture precedenti devono rimanere intatte:

DettDocumentoTb.Qnt

non viene azzerata.

Anche le note operative già presenti in:

DocumentoTb.NoteInterne

rimangono invariate.

Esempio:

Prima della chiusura:

Articolo A

QntDaLeggere = 10

```
Qnt = 10
```

```
Articolo B
```

```
QntDaLeggere = 5
```

```
Qnt = 5
```

```
Documento Stato = 3
```

Zucchetti riapre la lista e aggiunge un nuovo articolo:

```
Documento Stato = 2
```

```
Articolo A
```

```
QntDaLeggere = 10
```

```
Qnt = 10
```

```
Articolo B
```

```
QntDaLeggere = 5
```

```
Qnt = 5
```

```
Articolo C
```

```
QntDaLeggere = 8
```

```
Qnt = 0
```

Le letture degli articoli A e B vengono preservate. Solo la nuova riga parte con `Qnt = 0`.

Modifiche gestite dopo la riapertura

La riapertura non è limitata alla sola aggiunta di nuovi articoli.

Poiché Soluziona torna a stato `2`, la lista rientra automaticamente nelle regole della **lista già iniziata**.

Sono quindi gestiti anche:

```
modifica quantità prevista
```

```
modifica descrizione
```

```
modifica unità di misura
```

```
aggiunta di nuove righe
```

```
cambio articolo
```

cambio cliente
eliminazione di righe

Le regole restano quelle già definite per una lista iniziata:

Quantità modificata

- aggiorno QntDaLeggere
- mantengo Qnt

Cambio articolo

- vecchia riga QntDaLeggere = 0
- mapping vecchio IsAttivo = 0
- nuova DettDocumentoTb con Qnt = 0

Cambio cliente

- vecchia riga resta nel vecchio documento
- QntDaLeggere = 0
- nuova riga nel documento del nuovo cliente

Riga eliminata

- QntDaLeggere = 0
- Qnt preservato
- mapping IsAttivo = 0

Nuova riga

- nuova DettDocumentoTb
- QntDaLeggere = Quantita Zucchetti
- Qnt = 0

Ordine della sincronizzazione in caso di riapertura

La riapertura deve essere riconosciuta **prima** della sincronizzazione inversa.

Se l'inversa venisse eseguita subito mentre Soluziona è ancora a stato **3**, riporterebbe nuovamente Zucchetti a stato **4**, annullando la riapertura.

Il flusso deve quindi essere:

1. Rilevo eventuali riaperture Zucchetti

Zucchetti 0 + Soluzione 3

→ Soluzione 2

2. Importo eventuali nuove anagrafiche

Clienti

Articoli

3. Sincronizzo Zucchetti → Soluzione

recepisco aggiunte e modifiche

preservando le letture precedenti

4. Sincronizzo Soluzione → Zucchetti

Soluzione 2

→ Zucchetti 1

Dopo una nuova chiusura dal tablet:

Soluzione 3

↓

Zucchetti 4

e la lavorazione torna nuovamente conclusa.

Lista multicliente e riapertura

Una `S2serial` può corrispondere a più documenti Soluzione.

Poiché lo stato di riapertura viene espresso sulla testata unica Zucchetti, la riapertura della lista riguarda la `S2serial` nel suo complesso.

I documenti Soluzione collegati e già conclusi vengono quindi riportati da:

Stato 3 → Stato 2

mantenendo comunque invariati `Qnt` e `NoteInterne`.

Tabella di mapping

La relazione tra i record Zucchetti e quelli Soluziona viene gestita tramite:

```
DecodificaGestionaleEsternoTb
```

La tabella deve permettere di risalire sia ai documenti sia alle singole righe.

Per le righe il riferimento esterno è:

```
S2serial + S2rownum
```

La tabella contiene almeno:

```
IdSoluziona  
IdGestionaleEsterno  
IdRigaGestionaleEsterno  
TipoDocumento  
TipoEntita  
GestionaleEsterno  
IdAzienda  
IsAttivo
```

dove:

```
IdGestionaleEsterno = S2serial  
IdRigaGestionaleEsterno = S2rownum  
TipoDocumento = 29
```

Mapping storico delle righe

Dopo l'inizio della lavorazione, una stessa riga Zucchetti può generare più righe Soluziona.

Esempio:

```
S2serial = L00001  
S2rownum = 10
```

può avere:

```
DettDocumento A → Articolo A → IsAttivo = 0  
DettDocumento B → Articolo B → IsAttivo = 0  
DettDocumento C → Articolo C → IsAttivo = 1
```

Questo accade, ad esempio, quando Zucchetti modifica più volte l'articolo o il cliente dopo l'inizio della lavorazione.

Per questo motivo:

```
S2serial + S2rownum
```

non deve essere vincolato a una sola `DettDocumentoTb`.

Deve invece esistere una sola associazione corrente:

```
IsAttivo = 1
```

mentre le precedenti rimangono nello storico con:

```
IsAttivo = 0
```

Disattivazione mapping

È stata scelta la seguente regola:

“ Quando una riga non esiste più in Zucchetti, il relativo mapping corrente viene disattivato.

Quindi:

```
IsAttivo = 0
```

La scelta vale sia quando la riga viene eliminata logicamente da Soluziona prima dell'inizio, sia quando viene mantenuta storicamente con `QntDaLeggere = 0` dopo l'inizio.

La stessa regola viene applicata quando un cambio articolo o cliente produce una nuova versione della riga.

Il vecchio mapping viene disattivato e ne viene creato uno nuovo per la nuova `DettDocumentoTb`.

Regole riassuntive

Lista non iniziata

Zucchetti Stato = 0

È possibile:

modificare cliente
modificare articolo
modificare descrizione
modificare UM
modificare QntDaLeggere
modificare note
eliminare logicamente righe
spostare righe tra documenti

`Qnt` non viene mai alimentata dalla sincronizzazione Zucchetti → Soluziona.

Lista iniziata

Soluziona Stato = 2

Zucchetti Stato = 1

È possibile aggiornare:

QntDaLeggere
Descrizione
UM
altri dati provenienti da Zucchetti

Non possono essere sovrascritti:

Qnt

Note

Cambio articolo:

vecchia riga → QntDaLeggere = 0

nuova riga → nuovo articolo

Cambio cliente:

vecchia riga → QntDaLeggere = 0

nuova riga → documento del nuovo cliente

Eliminazione riga:

QntDaLeggere = 0

IsAttivo mapping = 0

Lista completata

Soluziona Stato = 3

La sincronizzazione inversa porta:

Zucchetti Stato = 4

Da quel momento:

SincronizzaListeDaZucchettiASoluziona

non deve più effettuare modifiche sulla lista.

Il processo è considerato concluso.

Principio generale

La sincronizzazione segue questo principio:

“ Prima dell'inizio della lavorazione, Zucchetti è completamente autorevole sulla struttura della lista. Dopo l'inizio, Zucchetti continua a essere autorevole sui dati previsti, mentre Soluziona diventa autorevole sui dati operativi raccolti dal tablet.

In particolare:

```
Zucchetti
  ↓
QntDaLeggere
Articolo
Cliente
UM
Descrizione
struttura prevista della lista
```

mentre:

```
Tablet / Soluziona
  ↓
Qnt
Note
storico della lavorazione effettuata
```

Le modifiche strutturali effettuate da Zucchetti dopo l'inizio non alterano retroattivamente le righe già lavorate: le vecchie righe vengono mantenute come storico e neutralizzate tramite `QntDaLeggere = 0`, mentre vengono create nuove righe per rappresentare la nuova situazione.

Regola finale sulla riapertura

“ La chiusura a stato `4` blocca la sincronizzazione Zucchetti → Soluziona finché la lista rimane conclusa. Se Zucchetti riporta esplicitamente la lista a stato `0`, la lista viene riaperta in Soluziona a stato `2`, senza azzerare le letture precedenti. Da quel momento tornano applicabili tutte le regole previste per una lista già

iniziata.

Sincronizzazione Liste di Carico Soluziona ? Zucchetti

Obiettivo

La sincronizzazione inversa ha lo scopo di riportare su Zucchetti i dati operativi prodotti dal tablet in Soluziona.

Il flusso è:

```
graph TD; Z1[Zucchetti] --> S1[SincronizzaListeDaZucchettiASoluziona]; S1 --> D[DocumentoTb / DettaglioDocumentoTb]; D --> C[Compilazione tramite Tablet]; C --> S2[SincronizzaListeDaSoluzionaAZucchetti]; S2 --> Z2[Zucchetti];
```

La sincronizzazione inversa deve aggiornare:

```
QuantitaLetta
Note
StatoLettura delle righe
StatoLettura della testa
```

La logica deve tenere conto del fatto che una singola lista Zucchetti può essere divisa in più documenti Soluziona quando contiene righe appartenenti a clienti differenti.

Tabelle coinvolte

Soluziona

DocumentoTb

DettDocumentoTb

DecodificaGestionaleEsternoTb

I documenti interessati hanno:

TipoDocumento = 29

Quantità

In

DettDocumentoTb

:

QntDaLeggere

rappresenta la quantità prevista proveniente da Zucchetti.

Qnt

rappresenta la quantità effettivamente letta tramite tablet.

Stato

Il tablet gestisce lo stato esclusivamente sulla testa del documento:

DocumentoTb.StatusDocumento

Non viene gestito uno stato operativo sulla singola

DettDocumentoTb

.

Note

Anche le note vengono gestite dal tablet sulla testa del documento Soluziona.

Zucchetti

Testata

TestaListaDiCaricoTb

Campi interessati:

S2serial
Note
StatoLettura
DataModifica
UtenteModifica

Righe

RigaListaDiCaricoTb

Campi interessati:

S2serial
S2rownum
QuantitaLetta
Note
StatoLettura
DataModifica
UtenteModifica

Mapping

La relazione tra le righe Soluziona e le righe Zucchetti viene gestita tramite:

DecodificaGestionaleEsternoTb

Per le righe:

IdSoluziona	=	DettDocumentoTb.Id
IdGestionaleEsterno	=	S2serial

IdRigaGestionaleEsterno	= S2rownum
TipoEntita	= 2
TipoDocumento	= 29
IsAttivo	= 1

La sincronizzazione inversa deve utilizzare esclusivamente mapping attivi:

IsAttivo = 1

I mapping storici con:

IsAttivo = 0

non devono partecipare alla sincronizzazione verso Zucchetti.

Aggiornamento quantità lette

La quantità effettivamente letta dal tablet è memorizzata in:

DettDocumentoTb.Qnt

e deve essere riportata in:

RigaListaDiCaricoTb.QuantitaLetta

Il mapping è quindi:

DettDocumentoTb.Qnt
↓
RigaListaDiCaricoTb.QuantitaLetta

L'aggiornamento deve essere eseguito esclusivamente sulla riga Zucchetti corrispondente al mapping attivo:

S2serial + S2rownum

Esempio:

Soluziona

Qnt = 25

↓

Zucchetti

QuantitaLetta = 25

Aggiornamento stati

Gli stati definiti sono:

Stato		Significato
0	ND	
1	Attivo	
2	Da convalidare	
3	Convalidato	
4	Autorizzato	

Il tablet modifica solamente:

DocumentoTb.StatusDocumento

La sincronizzazione inversa deve propagare lo stato della testa Soluziona alle righe Zucchetti appartenenti a quel documento.

Stati Soluziona 0 e 1

Se:

DocumentoTb.StatusDocumento = 0

oppure:

DocumentoTb.StatusDocumento = 1

la sincronizzazione inversa non deve modificare lo stato Zucchetti.

Quindi:

Soluziona 0 / 1

↓

nessun aggiornamento StatoLettura

Non devono essere effettuati reset dello stato Zucchetti.

Passaggio a Da_Convalidare

Quando il tablet inizia la lavorazione, Soluziona porta il documento a:

StatusDocumento = 2

La sincronizzazione inversa deve impostare:

StatoLettura = 1

su tutte le righe Zucchetti collegate a quel documento.

Quindi:

Soluziona StatusDocumento = 2

↓

Righe Zucchetti StatoLettura = 1

La testa Zucchetti viene successivamente ricalcolata sulla base dello stato complessivo delle sue righe.

Passaggio a Convalidato

Quando il tablet chiude la lavorazione, Soluziona porta il documento a:

StatusDocumento = 3

La sincronizzazione inversa deve impostare:

StatoLettura = 4

su tutte le righe Zucchetti appartenenti a quel documento.

Quindi:

Soluziona StatusDocumento = 3

↓

Righe Zucchetti StatoLettura = 4

Liste Zucchetti monocliente

Se tutte le righe di una lista Zucchetti appartengono allo stesso cliente, viene creato un solo documento Soluziona.

Esempio:

Lista Zucchetti L00001

Cliente 0001246

↓

Documento Soluziona

Cliente 0001246

In questo caso lo stato del documento Soluziona può essere propagato a tutte le righe Zucchetti della lista.

Liste Zucchetti multicliente

Se una lista Zucchetti contiene clienti differenti, Soluziona crea un documento separato per ogni cliente.

Esempio:

Lista Zucchetti L00001

Riga 1 → Cliente A

Riga 2 → Cliente A

Riga 3 → Cliente B

Riga 4 → Cliente B

diventa:

Documento Soluziona A

Cliente A

└─ Riga 1

└─ Riga 2

Documento Soluziona B

Cliente B

└─ Riga 3

└─ Riga 4

La sincronizzazione inversa deve modificare esclusivamente le righe Zucchetti appartenenti al documento Soluziona interessato.

Stato testa Zucchetti in caso di multicliente

Lo stato della testa Zucchetti non può essere copiato direttamente dallo stato di un singolo documento Soluziona.

Deve essere calcolato analizzando tutte le righe appartenenti alla stessa:

S2serial

La regola è:

se tutte le righe sono a 4

→ testa = 4

se almeno una riga è a 1 oppure 4

e non tutte sono a 4

→ testa = 1

altrimenti

→ non modificare lo stato

Esempio stato parziale

Lista Zucchetti:

S2serial = L00001

divisa in:

Documento A

Documento B

Situazione:

Documento A = StatusDocumento 2

Documento B = StatusDocumento 0

La sincronizzazione porta le righe del Documento A a:

StatoLettura = 1

Le righe del Documento B non vengono modificate.

La testa Zucchetti diventa:

StatoLettura = 1

perché almeno una parte della lista è stata iniziata.

Esempio autorizzazione parziale

Situazione:

Documento A = 3

Documento B = 2

Le righe di A diventano:

StatoLettura = 4

Le righe di B diventano:

StatoLettura = 1

La testa Zucchetti rimane:

StatoLettura = 1

perché non tutte le righe sono ancora autorizzate.

Esempio autorizzazione completa

Situazione:

Documento A = 3

Documento B = 3

Tutte le righe Zucchetti diventano:

StatoLettura = 4

A quel punto anche la testa viene impostata a:

StatoLettura = 4

Il processo può essere considerato completato.

Gestione note

Il tablet gestisce le note esclusivamente sulla testa del documento Soluziona.

La modalità con cui vengono riportate su Zucchetti cambia tra lista monocliente e lista multICliente.

Note su lista monocliente

Quando una lista Zucchetti genera un solo documento Soluziona:

```
1 lista Zucchetti
1 cliente
1 DocumentoTb
```

la nota della testa Soluziona viene riportata direttamente sulla testa Zucchetti.

Quindi:

```
DocumentoTb.Note...
      ↓
TestaListaDiCaricoTb.Note
```

Le note delle righe Zucchetti non devono essere modificate per questo scopo.

Esempio:

```
Soluziona:

Materiale verificato, manca un collo.

      ↓

Zucchetti Testa.Note:

Materiale verificato, manca un collo.
```

Note su lista multICliente

Quando una lista Zucchetti genera più documenti Soluziona, ogni documento può avere una propria nota.

In questo caso la nota della singola testa Soluziona deve essere riportata sulle sole righe Zucchetti appartenenti a quel documento.

Esempio:

Documento Cliente 0001246

Nota:

Materiale verificato, manca un collo.

Le righe Zucchetti appartenenti al Cliente 0001246 ricevono:

Note = Materiale verificato, manca un collo.

Un secondo documento:

Documento Cliente 0000219

Nota:

Consegna parziale autorizzata.

porta la propria nota esclusivamente sulle righe Zucchetti appartenenti al Cliente 0000219.

Nota concatenata sulla testa Zucchetti

Nel caso multicliente, la testa Zucchetti deve contenere la concatenazione delle note delle singole sottoliste Soluziona.

Ogni nota deve essere preceduta dal codice cliente.

Il formato deciso è:

[Cliente 0001246] Materiale verificato, manca un collo. [Cliente 0000219] Consegna parziale autorizzata.

Quindi la struttura è:

[Cliente CODICE] NOTA

ripetuta per ogni documento Soluziona derivato dalla stessa lista Zucchetti.

Esempio completo:

[Cliente 0001246] Materiale verificato, manca un collo. [Cliente 0000219] Consegna parziale autorizzata.

La concatenazione deve contenere una sola voce per ogni documento/cliente Soluziona collegato alla lista.

Individuazione del codice cliente

Il codice cliente Zucchetti è memorizzato in Soluziona nel campo:

ClienteTb.CodiceTessera

Per la composizione della nota della testa deve quindi essere utilizzato:

ClienteTb.CodiceTessera

e non:

ClienteTb.Codice

Righe storiche e mapping inattivi

Una stessa riga Zucchetti può aver generato più righe Soluziona nel tempo, ad esempio in caso di cambio articolo o cambio cliente dopo l'inizio della lavorazione.

Esempio:

S2serial = L00001

S2rownum = 10

Dettdocumento A → IsAttivo = 0

Dettdocumento B → IsAttivo = 0

Dettdocumento C → IsAttivo = 1

La sincronizzazione inversa deve considerare esclusivamente:

Dettdocumento C

perché rappresenta la versione attuale della riga Zucchetti.

Quindi:

```
IsAttivo = 0  
→ ignorato  
  
IsAttivo = 1  
→ sincronizzato
```

Cambio articolo o cliente

Se dopo l'inizio della lista Zucchetti cambia articolo o cliente, la sincronizzazione Zucchetti → Soluziona:

```
azzerà QntDaLeggere sulla vecchia riga  
disattiva il vecchio mapping  
crea una nuova DettDocumentoTb  
crea un nuovo mapping attivo
```

La sincronizzazione inversa deve quindi riportare verso Zucchetti solamente i dati della nuova riga attiva.

Non devono essere sommati o trasferiti automaticamente i dati delle vecchie righe storiche.

In particolare:

```
QuantitaLetta Zucchetti  
=  
Qnt della DettDocumentoTb con mapping attivo
```

e:

```
Note associate alla sottolista  
=  
note della testa del documento Soluziona corrente
```

Righe eliminate da Zucchetti

Quando una riga Zucchetti viene eliminata dopo l'inizio della lavorazione, la sincronizzazione diretta:

```
QntDaLeggere = 0  
IsAttivo mapping = 0
```

La sincronizzazione inversa non deve più tentare di aggiornare questa riga.

Ordine logico della sincronizzazione inversa

La procedura:

```
SincronizzaListeDaSoluzionaAZucchetti
```

deve eseguire logicamente questi passaggi.

1. Aggiornamento quantità

Per ogni mapping attivo:

```
DettDocumentoTb.Qnt  
↓  
RigaListaDiCaricoTb.QuantitaLetta
```

2. Aggiornamento note

Lista monocliente

Nota DocumentoTb

↓

TestaListaDiCaricoTb.Note

Lista multicliente

Nota Documento Cliente A

↓

righe Cliente A

Nota Documento Cliente B

↓

righe Cliente B

e sulla testa:

[Cliente A] Nota A [Cliente B] Nota B

3. Aggiornamento stato righe

Solo per:

StatusDocumento = 2

oppure:

StatusDocumento = 3

Mapping:

Soluziona 2 → Zucchetti 1

Soluziona 3 → Zucchetti 4

Gli stati Soluziona e non modificano Zucchetti.

4. Ricalcolo stato testa

Per ogni `S2serial` coinvolta:

```
tutte righe = 4
    → testa = 4

almeno una riga = 1 o 4
    → testa = 1

nessuna riga iniziata
    → nessuna modifica
```

Principio generale

La sincronizzazione inversa segue questo principio:

“ Soluziona è autorevole sui dati prodotti dal tablet; Zucchetti continua a rappresentare la lista originaria e deve ricevere quantità lette, note e avanzamento della lavorazione senza perdere la distinzione tra le eventuali sottoliste create per cliente.

In particolare:

```
Soluziona
    ↓
Qnt
Note testa
StatusDocumento
```

vengono trasformati in:

```
Zucchetti
    ↓
QuantitaLetta
Note testa / righe
StatoLettura righe
StatoLettura testa
```

tenendo sempre conto della relazione:

1 lista Zucchetti

↓

1 o più documenti Soluziona

e utilizzando esclusivamente i mapping correnti:

`DecodificaGestionaleEsternoTb.IsAttivo = 1`