

Manual de procesos internos Backend

Receiving.....	5
Heartbeat	5
Inserción y registro de totes desde Cam01.....	5
Proceso de desvío de totes	7
Proceso de confirmación de desvío de totes.....	8
Tote no desviado en todo el trayecto de Inbound Spine.....	10
Reinserción del tote al flujo inicial desde Cam02.....	11
Trayecto de totes de segundo nivel.....	13
Registro de tote	13
Desvío.....	13
Confirmación de desvío.....	14
Tote desviado a Jackpot.....	14
Tablas utilizadas para las APIs de Receiving.....	15
Diagrama Entidad-Relación cliente-db.....	16
URLS y JSON PLC de las APIs.....	16
Picking.....	18
URLS de las Apis y JSON PLC	18
Registro de un tote Picking	18
Actualización de datos para una vuelta	21
Asignación a Picking JackpotLane	21
Actualización de datos para Totes Singles	22
Picking Multi Totes	22
Creación de ola	22
Actualización del status y release del primer tote de la ola	23
Registro de escaneo de inicio de la ola (scanlogs)	23
Corrección de ola.....	23
Comprobación de finalización de la ola.....	24
Incremento del contador de la ola.....	24
Línea de desvío llena - Wave tote single	24
Disponibilidad del Commissioner.....	26
Límites para las máquinas cuando el commissioner está presente.....	27
Transferencia de totes de picking a Transfer 03 y 04.....	27
Caso para Multi Totes	29
Commissioner Presente.....	29
Commissioner Ausente.....	29
Caso de singles.....	30
Liberación del Looping.....	31
Commissioner Presente	32
Commissioner Ausente	32
Single Tote	33
SingleTote.....	34
Desvío para totes singles - Cam13.....	34
Confirmación de desvío para totes singles	36
Multi Totes	37

Destino Cam12.....	37
Desvío	37
Commissioner Presente.....	38
Commissioner Ausente.....	39
Confirmación de desvío	40
Commissioner Presente.....	41
Commissioner Ausente.....	42
Multi Totes - Destino Cam14	42
Desvío	42
Commissioner Presente.....	43
Commissioner Ausente.....	44
Confirmación de desvío	44
Multi Totes - Destino Cam15	46
Desvío	46
Confirmación de desvío	48
Commissioner Presente.....	48
Commissioner Ausente.....	49
Multi Totes - Destino Cam16	49
Desvío	49
Confirmación de desvío	51
Commissioner Presente.....	52
Commissioner Ausente.....	52
Multi Totes - Destino Cam17	53
Desvío	53
Confirmación de desvío	54
Commissioner Presente.....	55
Commissioner Ausente.....	55
PROCEDURE [UpdateToteCounts]	56
Tote desviado a picking Jackpot	58
Desvío hacia Picking Jackpot.....	58
Confirmación para Picking Jackpot.....	59
Tablas utilizadas para las APIS de Picking:.....	60
Diagrama Entidad-Relación cliente-db	62
URLS de las API's.....	62
Sorter	67
Cajas Single.....	67
Proceso realizado para cajas ingresadas por primera vez.....	67
Registro de cajas.....	67
Desvío de caja.....	68
Confirmación de cajas a línea de destino	69
Recirculación de caja.....	71
Cambio de estado de las líneas.....	71
Encendido de línea.....	71
Impresión automática de etiquetas	72

Apagado de línea	73
Cambio de estado Full	73
Caja multibox.....	77
Proceso realizado para cajas ingresadas por primera vez.....	77
Registro de cajas.....	78
Desvío de caja.....	80
Confirmación de cajas multis a línea de destino	82
Recirculación de caja.....	83
Cambio de estado de las líneas.....	83
Encendido de línea.....	84
Impresión automática de etiquetas	84
Apagado de línea	85
Cambio de estado Full.....	86
Impresión de etiquetas automáticas para cierre de líneas inesperadas	86
Tablas utilizadas para las APIs de Sorter	91
Diagrama Entidad-Relación cliente-db	93
URLS y JSON de las API's.....	94

Receiving

Heartbeat

URL: <http://10.107.92.43:8088/api/HeartBeat>

Método: GET

El método "Heartbeat" es utilizado por el cliente de manera continua para confirmar una comunicación efectiva con el sistema. Cuando el cliente hace una petición a este método cada segundo, regresará como respuesta el valor de 1, indicando que existe comunicación entre las dos partes. Si el cliente no recibe respuesta en un lapso de cinco segundos, se registra la falta de comunicación con el sistema, así como con todas sus APIs.

Respuesta:

- **1** : indica que existe disponibilidad y conexión con las APIS.
- **Sin respuesta:** Indica que no hay comunicación con las APIS, posiblemente debido a caída del sistema o se haya apagado manualmente.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
GET	http://10.107.92.43:8088/api/HeartBeat	Heartbeat	N/A	1	N/A	N/A

Inserción y registro de totes desde Cam01

Ubicación: Cam01

URL: <http://10.107.92.43:8088/api/ToteInformation/receive-flow/new-tote>

Método: POST

La entrada de los totes de Receiving se realiza por Cam01, donde son ingresados los totes cuyo LPN contienen la inicial K, H, o N. La cámara tiene asignada la API de NewTote, encargada de registrar los totes que son escaneados y la información obtenida es enviada por medio de un JSON con la siguiente estructura:

```
{  
  "camId": "Cam01", //Identificador de la cámara situada en el scanner  
  "trackingId": 10, //Identificador para el seguimiento del tote  
  "toteLpn": "HC59826023" //Código de barras perteneciente a un tote  
}
```

**TrackingId es un identificador para dar seguimiento a los totes , sirve de apoyo para el equipo de el clientePLC .*

Usando el valor del campo “toteLpn” dado por el JSON de petición, se verifica que no sea perteneciente a totes de picking, es decir, que la inicial del LPN no sea “T”.

La información de los totes que sean de Receiving debe ser obtenida a través de la API de SAP de método POST: <http://10.107.92.35:211/api/sapinfo/send/{id}>. Los datos obtenidos de la consulta a SAP se reciben a través del JSON:

```
{
  "TOTE_ID": "string",
  "V_TOTE_ID": "string",
  "ZWZONE": "string",
  "ZTIMESTAMP": "string",
  "ERROR": "string",
  "Request_ts": "string",
  "Response_ts": "string"
}
```

Cuando los datos obtenidos de la consulta a la API de SAP están completos, se realiza el registro de la información del tote en la tabla ToteInformations de la base de datos cliente-db.

Terminando la petición y como respuesta a la consulta de la API de NewTote, se devuelve el siguiente JSON:

```
{
  "tote_LPN": "HC59826023",
  "virtual_Tote": "65204687601628140522254307599307031",
  "zone_Id": 99,
  "resp_Timestamp": "16316498760012345"
}
```

Si la consulta a la API que realiza el registro de los totes “NewTote” presenta algún problema en su ejecución, como por ejemplo: el JSON de petición contenga valores no válidos para su procesamiento, fallos de conexión con la API de SAP al realizar la consulta de información, o fallos de conexión a la base de datos al realizar la inserción del tote en los registros de la tabla ToteInformations

Se devolverá la siguiente respuesta:

Status 500: Indicará el mensaje de error

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
--------	-----	------	--------------	---------------	---------------	--------------------

POST	http://10.107.92.43:8088/api/ToteInformation/register-flow/new-tote	Register	{ "camId": "string", "trackingId": 0, "toteLpn": "string" }	{ "tote_LPN": "string", "virtual_Tote": "string", "zone_Id": 0, "resp_Timestamp": "string", }	BadRequest Status 500	Camera 01
------	---	----------	--	--	-----------------------	-----------

Proceso de desvío de totes

Ubicación: Cam04 a Cam09

URL: <http://10.107.92.43:8088/divert-flow/divert>

Método: POST

En Inbound Spine, sobre la cámara 04, cámara 05, cámara 06, cámara 08 y cámara 09, se encuentra la API de desvío. Para poder realizar el desvío, se debe verificar que el JSON de petición esté completo, si contiene los campos ToteLPN, camId y TrackingId llenos, se podrá continuar con el proceso de desvío.

Debe verificar que el identificador de la cámara "CamId", parámetro que contiene el JSON de petición, coincida con el ID de la cámara asignada a la línea de desvío que activó la API para confirmar que sea la correcta.

Confirmando que la línea de desvío sea la correcta para desviar, se busca que el tote esté registrado en la tabla ToteInformation y el campo DivertStatus sea null, si encuentra el registro de dicho tote, entonces modifica los campos DivTimestamp, TrackingId y LastCam con nueva información para estos tres parámetros: el tiempo en que se ejecutó la consulta (campo DivTimestamp), actualización del TrackingId que proporciona el JSON de la petición y la cámara que escanea en ese momento. Con este procedimiento hecho de manera correcta, se puede realizar el desvío para el tote a la línea que realizó la petición de la API.

Utilizando el método POST, recibe el siguiente JSON:

```
{
  "camId": "string",
  "trackingId": 0,
  "toteLpn": "string",
  "scannerNLaneWStatus": 0,
  "scannerNLaneWFull": 0
}
```

Regresa el siguiente JSON cuando sea posible desviar:

```
{
  "divert_code": 2,
  "tracking_id": 0
}
```

El tracking devuelto será el mismo que el recibido de la petición debido a que es necesario para PLC. Y en caso de que no pueda ser desviado nos regresará el JSON:

```
{
  "divert_code": 99,
  "tracking_id": 0
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert	Divert	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNL aneWStatu s": 0, "scannerNL aneWFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadReques t { message:"s tring" } }	Camera 3 to camera 9

Proceso de confirmación de desvío de totes

Ubicación: Cam04 a Cam09

URL: <http://10.107.92.43:8088/divert-flow/confirmation>

Método: POST

Esta API se encuentra en la cámara 04, cámara 05, cámara 06, cámara 08 y cámara 09 sobre Inbound Spine. Al pasar por el sensor de confirmación, la API de confirmación envía un JSON para realizar el proceso de confirmación.

Verifica que el JSON tenga un formato válido y que el campo divertcode tenga valor de 100 (el valor de "divertcode" indica si la instrucción de confirmación debe realizarse

o no, un valor diferente a 100 indica que no será posible realizar el proceso y si es 100 se puede proceder con la confirmación del tote).

Se continúa con el proceso de confirmación, primero buscando el registro del tote en la tabla ToteInformations por medio de la fecha y hora de la petición de la API en un rango de 10 segundos y el TrackingId que se obtiene del JSON de entrada, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api. Al encontrarlo, se le actualizan los campos DivertStatus a "IN" y TrackingId se reinicia a 0.

En el mismo momento en que se realizan los procesos anteriores, se hace una inserción en la tabla externa Wcs_ZoneRouting de la base de datos E1_Receiving_WMS con los siguientes valores del tote a confirmar: Tote_ID, Virtual_Tote_ID, Zone, Insert_ts, Status.

La URL de la API es <http://10.107.92.43:8088/divert-flow/confirmation>, su método es POST y recibe el siguiente JSON:

```
{
  "camId": "string",
  "trackingId": 0,
  "divertCode": 100
}
```

Donde CamId, es el valor de la cámara que está realizando la confirmación de desvío, trackingId es el seguimiento del tote, y DivertCode es el código de desvío.

Respuestas:

Si la confirmación de desvío se realizó correctamente, devolverá un código de status indicando el fin de la operación:

- **status Ok 200:** La confirmación se realizó correctamente.

Si la confirmación de desvío no se realizó correctamente, devolverá un código de status indicando el fin de la operación:

- **status 500 Bad Request:** Regresa el siguiente json, con un mensaje indicando que no confirmó el desvío.

BadRequest:

```
{
  "message": "string"
}
```

METHOD	URL	NAME	JSON	JSON	ERROR	CAMERAS
--------	-----	------	------	------	-------	---------

			REQUEST	RESPONSE	REQUEST	ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/confirmation	Confirmation	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 3 to camera 9, camera 12 and 13

Tote no desviado en todo el trayecto de Inbound Spine

Ubicación: Cam10

URL: <http://10.107.92.43:8088/api/ToteInformation/receive-flow/borderPicking>

Método: POST

Se utiliza para desviar totes que pertenecen a receiving, los totes empiezan con la letra "H", "K" o "N". En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", //(Cámara que escanea en ese momento)
  "toteLPN": "string", //(Código LPN)
  "trackingId": 0, //(Tracking de seguimiento del tote)
  "scanner_N_lane_w_status": 0, // (Indica si la línea está activa)
  "scanner_N_lane_w_full": 0 // (Indica si la línea está llena)
}
```

En este caso, al ser totes de receiving, no podrán pasar de ninguna manera para transfer 03 y 04.

Consulta en la base de datos interna:

- En la tabla Cameras: Buscamos el registro de la camara 10(Cam10)
- En la tabla de ToteInformation: Buscamos el registro del ToteLPN y que su DivertStatus sea null.

Evaluamos el resultado de la búsqueda:

- El resultado de la búsqueda de la cámara debe ser True.
- Se valida que el ToteLPN, empiece con la letra "H", "K" o "N".

- Se valida que el resultado de la búsqueda en TotelInformations regrese un resultado con datos, si esto se cumple, actualizamos el campo TrackingId por el valor que PLC manda, agregamos un nuevo registro de scanlog a la tabla de ScanlogsReceivingPickings con el resultado encontrado, la cámara que evalúa el proceso, y el mensaje para el desvío, al final retornamos un 1.

Si alguna de estas validaciones no se cumple, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con el dato encontrado de la tabla TotelInformations, al final retornamos un valor 1.

Del valor que retornamos, el valor 1 indica que debe de desviarse y dar la vuelta hacia cam 02, con esto enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 1, //(1 indica que debe mantenerse en loop en Inbound Spine)
  "tracking_id": 0 //(TrackingId seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/api/TotInformation/receive-flow/borderPicking	Border picking(border transfer03 and 04)	{ "camId": "string", "toteLPN": "string", "trackingId": { 0, "scanner_N_lane_w_status": 0, "scanner_N_lane_w_full": 0 } }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" } }	Camera 10

Reinserción del tote al flujo inicial desde Cam02

Ubicación: Cam02

URL: <http://10.107.92.43:8088/transfer-inbound/check-tote>

Método: POST

En Cam02 se encarga de incrementar el contador de vueltas para el tote que se escanea en ese momento. Además, tiene la función de registrar el tote en caso de

que no se encuentre el registro en la tabla TotelInformations. Donde se le asignará una línea de desvío en caso de que cumpla las condiciones de mapeo configuradas en la IU y lo que se obtenga de información obtenida de SAP para dicho tote.

La URL de la API es <http://10.107.92.43:8088/transfer-inboud/check-tote>, haciendo uso del metodo POST, recibe el siguiente JSON:

```
{
  "camId": "string",
  "trackingId": 0,
  "toteLpn": "string",
  "scannerNLaneWStatus": 0,
  "scannerNLaneWFull": 0
}
```

Devuelve el siguiente JSON como respuesta:

```
{
  "divert_code": 2
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERA S ASSIGNMENT
POST	http://10.107.92.43:8088/transfer-inboud/check-tote	TransferInboud	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLaneWStatus": 0, "scannerNLaneWFull": 0 }	{ "divert_code": 2 }	BadRequest { message:"string" } }	Camera 2

Trayecto de totes de segundo nivel

Registro de tote

Ubicación: Cam01

URL: <http://10.107.92.43:8088/api/ToteInformation/receive-flow/new-tote>

Método: POST

Los totes de segundo nivel presentan el mismo flujo de trabajo que los totes normales de receiving, son registrados en Cam01 haciendo uso del método POST de la API New Tote:

<http://10.107.92.43:8088/api/ToteInformation/receive-flow/new-tote>, son identificados por tener la inicial “N” en el campo LPN.

Los registros de los totes de segundo nivel no tienen diferencias con los totes normales en su proceso de registro, pero desde el método POST de la API de SAP <http://10.107.92.35:211/api/sapinfo/send/{id}> este tipo de totes ya contienen un valor distintivo en el campo VirtualZone con valor de 22. Este campo se refiere a la zona virtual asignada para los totes que son proporcionados por SAP.

Desvío

Ubicación: Asignación de cámara se realiza desde la UI

URL: <http://10.107.92.43:8088/divert-flow/divert>

Método: POST

El valor 22 del campo VirtualZone debe ser mapeado en la UI a una línea de desvío, en este caso la línea debe ser la que está asignada a la cámara de segundo nivel. La cámara para segundo nivel debe ser configurada desde la UI.

Estos totes siguen el flujo normal de un tote, por lo que los desvíos a las líneas se hacen usando el metodo POST perteneciente a la API: <http://10.107.92.43:8088/divert-flow/divert>, se realizan las consultas a la base de datos, comenzando por el registro del ToteLPN en la tabla ToteInformations, la consulta de la cámara, la línea a la que está asignada la cámara en la tabla CameraAssignments, y el mapeo de la línea con el VirtualZone de la tabla MapPhysicalVirtualSap. Después de hacer la búsqueda, se evalúan los resultados, validando que el Status de la línea esté activa y la línea sea igual al valor que está mapeado con el ZoneDivert 22. Si esto se cumple, se puede hacer el desvío correspondiente, después agregando un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con la información del ToteLPN de segundo nivel, y se retorna un valor 2. Si en dado caso, la línea de desvío no corresponde a la de segundo nivel, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con la información del ToteLPN, y al final, retorna un valor 99.

Confirmación de desvío

Ubicación: Se realiza una validación por parte del sistema usando la configuración realizada desde la UI para la cámara asignada a segundo nivel.

URL: <http://10.107.92.43:8088/divert-flow/confirmation>

Método: POST

Usando la API de confirmación <http://10.107.92.43:8088/divert-flow/confirmation> , se verifica que el JSON tenga un formato válido y que el campo divertcode tenga valor de 100 indicando que la confirmación puede realizarse o no.

Se continúa buscando el registro del tote en la tabla TotelInformations por medio de la fecha y hora de la petición de la API en un rango de 10 segundos y por el TrackingId que se obtiene del JSON de entrada, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api. Al encontrarlo, se le actualizan los campos DivertStatus a "IN" y TrackingId y se reinicia a 0.

Al mismo tiempo en que se realizan los procesos anteriores, se hace una inserción en la tabla externa Wcs_ZoneRouting de la base de datos E1_Receiving_WMS con los siguientes valores del tote a confirmar: Tote_ID, Virtual_Tote_ID, Zone, Insert_ts, Status.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/confirmation	Confirmation	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 3 to camera 9, camera 12 and 13

Tote desviado a Jackpot

Existen dos casos en las que un tote puede desviarse a la línea de desvío configurada como Jackpot:

- Los totes que no cuentan con la información que se obtiene en la API de SAP (campos ToteLPN, VirtualTote o VirtualZone vacíos) o en los que no se logre hacer una correcta lectura del código de barras del tote, serán asignados a la zona virtual 999, perteneciente a la línea configurada en la UI como Jackpot.
- Si no se llega a tener lectura del código de barras de un tote al momento de la petición de la API de Desvío, es decir el JSON contiene el valor de "?", en el campo toteLPN, y se comprueba que el tote está registrado correctamente en la tabla TotelInformations con una línea de desvío, se cumplen las condiciones para desviar el tote a Jackpot Line. Este proceso sucede cuando la línea que realiza la petición a la API es la que se ha configurado como Jackpot desde el UI.

Tablas utilizadas para las APIs de Receiving

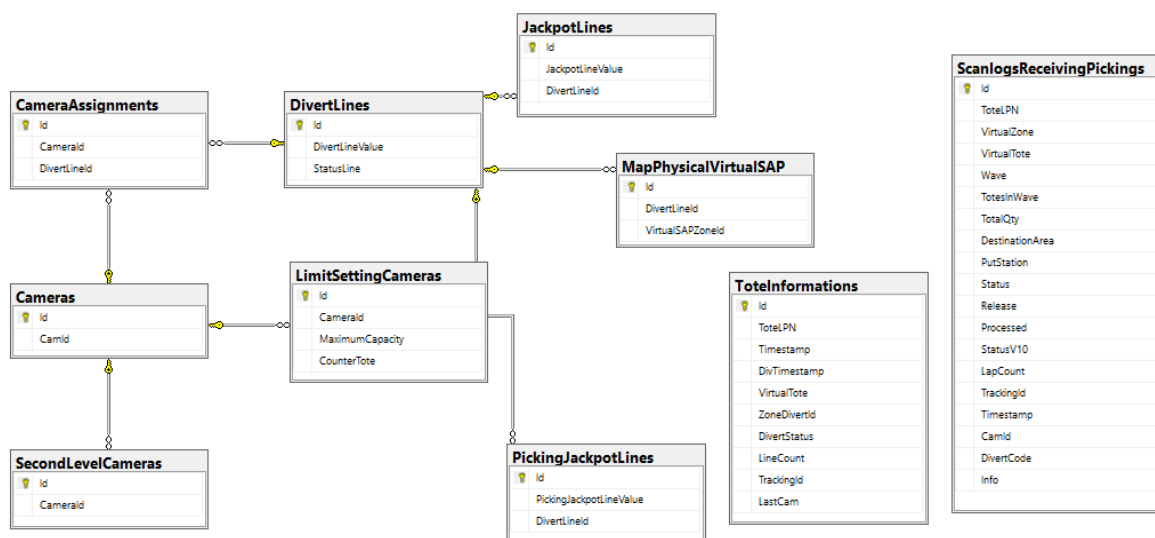
cliente-db		
CameraAssignments	Cameras	DivertLines
<ul style="list-style-type: none"> Id (PK, int, not null) CameraId (FK, int, not null) DivertLineId (FK, int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) CamId (nvarchar(max), not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) DivertLineValue (int, not null) StatusLine (bit, not null)
JackpotLines	MapPhysicalVirtualSAP	ScanlogsReceivingPickings
<ul style="list-style-type: none"> Id (PK, int, not null) JackpotLineValue (bit, not null) DivertLineId (FK, int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) DivertLineId (FK, int, not null) VirtualSAPZoneId (int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) ToteLPN (nvarchar(10), null) VirtualZone (int, null) VirtualTote (nvarchar(35), null) Wave (nvarchar(35), null) TotesInWave (int, null) TotalQty (int, null) DestinationArea (nvarchar(4), null) PutStation (int, null) Status (int, null) Release (int, null) Processed (bit, null) StatusV10 (nvarchar(2), null) LapCount (int, null) TrackingId (int, null) Timestamp (nvarchar(17), null) CamId (nvarchar(10), null) DivertCode (int, null) Info (nvarchar(100), null)
ToteInformations		
<ul style="list-style-type: none"> Id (PK, int, not null) ToteLPN (nvarchar(10), not null) Timestamp (nvarchar(17), not null) DivTimestamp (nvarchar(17), null) VirtualTote (nvarchar(35), not null) ZoneDivertId (int, not null) DivertStatus (nvarchar(2), null) LineCount (int, not null) TrackingId (int, not null) LastCam (nvarchar(6), null) 		

E1_Receiving_WMS

Wcs_ZoneRouting

Id (PK, int, not null)
BoxId (char(20), null)
Box Type (char(18), null)
CarrierCode (char(10), null)
LogisticAgent (char(4), null)
ConfirmationNumber (char(20), null)
ContainerId (char(20), not null)
ContainerType (char(1), not null)
Qty (numeric(6,0), null)
DivertLane (numeric(4,0), not null)
CurrentTs (nchar(20), not null)
Status (char(2), not null)
SAPSystem (char(4), not null)

Diagrama Entidad-Relación cliente-db



URLS y JSON PLC de las APIs

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/api/TotelInformation/register-flow/new-tote	Register	{ "camId": "string", "trackingId": 0, "toteLpn": "string" }	{ "tote_LPN": "string", "virtual_Tote": "string", "zone_Id": 0, "resp_Timestamp": "string", }	BadRequest Status 500	Camera 01

POST	http://10.107.92.43:8088/transfer-inboud/check-tote	TransferInboud	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLaneWStatus": 0, "scannerNLaneWFull": 0 }	{ "divert_code": 2 }	BadRequest { message:"string" }	Camera 2
POST	http://10.107.92.43:8088/divert-flow/divert	Divert	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLaneWStatus": 0, "scannerNLaneWFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 3 to camera 9
POST	http://10.107.92.43:8088/divert-flow/confirmation	Confirmation	{ "camId": "string", "trakingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 3 to camera 9, camera 12 and 13
POST	http://10.107.92.43:8088/api/ToteInformation/register-flow/borderPicking	Border picking(border transfer03 and 04)	{ "camId": "string", "toteLPN": "string", "trackingId":	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 10

			0, "scanner_N _lane_w_st atus": 0, "scanner_N _lane_w_full": 0 }			
--	--	--	--	--	--	--

Picking

URLS de las Apis y JSON PLC

<https://docs.google.com/spreadsheets/d/1LZuApHCdVkdTOJfWCmVKGivqXVzDnDFXSMsUaCK5aE0/edit?usp=sharing>

Registro de un tote Picking

Ubicación: Cam02

URL: <http://10.107.92.43:8088/transfer-inboud/check-tote>

Método: POST

Los totes con campo LPN iniciado con "T" pertenecen a picking, entran por las líneas de Inbound hasta llegar a Cam02 donde se tiene la API de CheckTote. En caso de que el tote ya esté procesado solo se mandara a picking jackpot si no solo seguirá el proceso normal.

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación (Cam02), el toteLPN, el tracking_ID asociado, el status de la línea y si esta full o no.

```
{
  "camId": "Cam02",
  "trackingId": 2,
  "toteLpn": "TLPN0012",
  "scannerNLaneWStatus": 1,
  "scannerNLaneWFull": 0
}
```

Parámetros

- **Tote_LPN:** Código de barras perteneciente a un tote, su valor es de tipo string
- **CamId:** Identificador de la cámara situada en el scanner, su valor es de tipo string.
- **TrackingId:** Se obtiene del PLC y sirve como identificador para cada petición, este dato puede repetirse y va del 1 al 999, se conforma por 4 dígitos, el 1er dígito se refiere a la línea de desvío.
- **ScannerNLaneWStatus:** Estado de apagado/encendido de la línea, su valor es de tipo entero y puede traer 1 o 0.
- **ScannerNLaneWFull:** Estado de llenado de la línea, su valor es de tipo entero y puede traer 1 o 0.

Si el tote se encuentra registrado en ToteInformations, sólo se actualizan los campos DivTimestamp con la fecha y hora de la petición de la API, el TrackingId proporcionado por el escáner y la cámara que escanea en ese momento. Pero si el tote no se encuentra, en el mismo proceso se hace un llamado a la API de New Tote <http://10.107.92.43:8088/api/ToteInformation/receive-flow/new-tote> para realizar el proceso de registro del tote.

Cuando un tote pertenece a picking, no se hace la búsqueda de este tote por medio de la API de SAP, sino que se busca directamente el tote en la tabla externa Tote_Hdr de la base de datos E1_EXT. La búsqueda consiste en obtener el registro más antiguo que coincida con el mismo ToteLPN y no se encuentre procesado. Agregando al registro del tote en la tabla ToteInformations, el campo ZoneDivertId será de valor 888, dando un valor distintivo a los totes pertenecientes a picking.

Posteriormente, el sistema revisa si existe el tote en la tabla Tote_Hdr de la base de datos externa. En caso de que exista se comprueba si el status está en null, en caso de que no esté en null se crea el registro, aumenta el número de vueltas en el campo LineCount dentro de la tabla ToteInformations.

Terminando con el registro del tote usando la API de New Tote, se regresa al proceso de Check Tote para terminar.

Al resultar exitosa la petición, retornamos una respuesta:

```
{
  "divert_code": 2
}
```

Y en caso de error:

```
BadRequest
{
  message:"string"
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/transfer-inboud/check-tote	Check Tote	{ "camId": "Cam02", "trackingId": 2, "toteLpn": "T43564", "scannerNLaneWStatus": 1, "scannerNLaneWFull": 0 }	{ "divert_code": 2 }	{ "message": "string" }	Cam02
POST	http://10.107.92.43:8088/api/ToteInformation/receive-flow/new-tote	New Tote	{ "camId": "Cam01", "trackingId": 2, "toteLpn": "T45345" }	{ "Tote_LPN": "T45343", "Virtual_Tote": "4653457676", "Zone_Id": 888, "Resp_Time stamp": "20240314123456789" }	{ "message": "string" }	Cam01

Actualización de datos para una vuelta

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación (Cam02), el toteLpn, el trackingId asociado, el status de la línea y si está llena o no.

```
{  
  "camId": "Cam02",  
  "trackingId": 2,  
  "toteLpn": "TLPN0012",  
  "scannerNLaneWStatus": 1,  
  "scannerNLaneWFull": 0  
}
```

Si se encuentra el registro del tote en la tabla ToteInformations, se aumenta +1 el valor del campo LineCount que se encarga de registrar las vueltas realizadas en Inbound Spine, se actualiza el campo TrackingId con el nuevo obtenido de la consulta a la API y la marca de tiempo de desvío (DivTimestamp) se sustituye con el nuevo generado al realizar la consulta a la API.

Si el tipo de registro es "T", indica que se trata de tipo de recirculación, se realizan varias verificaciones adicionales:

Se busca información en la base de datos del toteLpn en la base de datos externa (E1_EXT) (tabla Tote_Hdr).

Asignación a Picking JackpotLane

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación (Cam02), el toteLPN, el trackingId asociado, el status de la línea y si está llena o no.

```
{  
  "camId": "Cam02",  
  "trackingId": 2,  
  "toteLpn": "TLPN0012",  
  "scannerNLaneWStatus": 1,  
  "scannerNLaneWFull": 0  
}
```

Si no se encuentra la información al realizar la búsqueda del tote en la tabla Tote_Hdr de la base de datos E1_EXT, se asignará el tote a la línea de picking jackpot registrando en el campo ZoneDivertId el valor de 889 en la tabla de ToteInformations de la base de datos cliente-db indicando que es para picking jackpot.

Se registra un scanlog indicando que se ha asignado a la línea para picking jackpot y se devuelve un código de éxito (2).

Si se encuentra información al realizar la búsqueda del tote en la tabla Tote_Hdr de la base de datos E1_EXT, se realizan las siguientes verificaciones:

Si del registro encontrado el campo Wave_Nr está vacío, el tote se registrará para picking jackpot. Una vez asignado, se registra un scanlog y se devolverá el código de éxito (divert_code) con valor 2.

Actualización de datos para Totes Singles

En el cuerpo de la solicitud que realiza la API de Check Tote se debe incluir el id de la cámara que realiza la comprobación (para esta API sería "Cam02"), el toteLPN, el trackingId asociado, el status de la línea y si está llena o no.

```
{  
  "camId": "Cam02",  
  "trackingId": 2,  
  "toteLpn": "TLPN0012",  
  "scannerNLaneWStatus": 1,  
  "scannerNLaneWFull": 0  
}
```

Si el área de destino de WCS (WCS_Destination_Area) es "SPAC" o "SMC3" y tanto el campo Status como Release del registro, almacenado en la tabla Tote_Hdr de la base de datos E1_EXT, son nulos, se actualizan a valor 1 estos campos. Se registra un scanlog del proceso realizado y se devuelve un código de éxito (2) al completar la consulta a la API.

Picking Multi Totes

Creación de ola

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en la cámara en la que va en ese momento, el toteLPN, el tracking_ID asociado, el status de la línea y si está lleno o no.

```
{  
  "camId": "Cam02",  
  "trackingId": 2,  
  "toteLpn": "TLPN0012",  
  "scannerNLaneWStatus": 1,  
  "scannerNLaneWFull": 0  
}
```

Se busca el registro del toteLpn en la tabla Tote_Hdr de la base de datos externa E1_EXT, que cumpla con la condición de que los campos Status, Release y Processed sean nulos. También se ordena la búsqueda del tote por timestamp, tomando como resultado el más antiguo. Una vez encontrado el toteLpn, se buscará su waveNr en la tabla CountWaves. Si no existe un registro de conteo para la ola actual, se crea un nuevo registro en la tabla CountWaves con la información de la ola, incluyendo el número total de unidades en la ola y se establece el contador en 1. En caso de que el número total de totes para esa ola sea solo 1 pasará directamente a actualizar el status de todos los totes asociados a la ola en la base de datos externa, estableciendo su estado en 2 y registrar un scanlog indicando que la ola está completa sin hacer ningún registro en CountWaves.

Actualización del status y release del primer tote de la ola

Se ejecuta una consulta SQL a la tabla Tote_Hdr de la base de datos externa E1_EXT, para actualizar el status y el release del primer tote de la ola. Esto se realiza seleccionando al primer tote que cumple con ciertas condiciones (sin status ni release y sin processed) y se establece su status en 1 y su release en 1.

Registro de escaneo de inicio de la ola (scanlogs)

Se agrega un registro de scanlog indicando que la ola está en proceso, con información relevante como el LPN, la virtual zone, Nr Of Totes In Wave, etc.

Corrección de ola

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en la cámara en la que va en ese momento, el toteLPN, el tracking_ID asociado, el status de la línea y si está llena o no.

```
{  
  "camId": "Cam02",  
  "trackingId": 2,  
  "toteLpn": "TLPN0012",  
  "scannerNLaneWStatus": 1,  
  "scannerNLaneWFull": 0  
}
```

Se obtiene el campo waveNr del resultado de la búsqueda en Tote_Hdr y se busca en la base de datos interna si ya existe un registro de conteo de esa ola en la tabla CountWaves.

Si existe un registro de la ola en la tabla CountWaves, se realiza una corrección en el campo Count de dicho registro, asegurándose de que coincida con la cantidad de totes para ola. Se hace una consulta a la tabla Tote_Hdr para obtener la cantidad total de totes asignados a la ola, y si el contador actual del registro en CountWaves no

coincide con esta cantidad, se actualiza el contador de la ola (campo Count del registro). Si el contador coincide con el número total de totes en la ola, se toma como ola completa y actualiza el campo "Status" de los totes pertenecientes a la ola registrados en la tabla "Tote_Hdr" de la base de datos E1_EXT. Se elimina el registro de la ola en la tabla CountWaves en la base de datos cliente-db y se registra un scanlog indicando que la ola está completa.

Comprobación de finalización de la ola

Se comprueba si el contador de la ola coincide con el número total de totes en la ola. Si coincide, significa que todos los totes pertenecientes a esa ola han sido procesados, por lo que se realiza lo siguiente:

- Se actualiza el status de todos los totes asociados a la ola en la base de datos externa, estableciendo su estado en 2.
- Se elimina el registro de conteo de la ola de la tabla CountWaves.
- Se agrega un registro de scanlog indicando que la ola está completa.

Incremento del contador de la ola

Al momento de registrarse un tote en la tabla ToteInformations, se comprueba que el campo WaveNr exista en la tabla CountWaves, es decir, si la WaveNr ya está registrada en la tabla CountWaves, en ese caso solo va a aumentar en + 1 el conteo del tote para esa ola, con esto indica que se sigue construyendo la ola.

Línea de desvío llena - Wave tote single

Ubicación: Cam10

URL: <http://10.107.92.43:8088/api/ToteInformation/receive-flow/borderPicking>

Método: POST

Se usa para controlar los totes singles que pueden pasar para el área de transfer 03 y 04, si el límite de totes singles ha sido superado, no podrá dejar pasar más totes hacia Transfer 03 y Transfer 04 y se mantendrá en el Inbound Spine, hasta que haya espacio, dejará pasar más totes.

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{  
  "camId": "string", //(Cámara que escanea en ese momento)  
  "toteLPN": "string", //(Código LPN)  
  "trackingId": 0, //(Tracking de seguimiento del tote)  
  "scanner_N_lane_w_status": 0, //(Indica si la línea está activa)
```



```

"scanner_N_lane_w_full": 0 //(Indica si la línea está llena)
}

```

Al llegar un Tote de picking en cam10, consulta en tabla ToteInformations y válida que exista el registro del toteLPN. Posteriormente, se hace una consulta a la tabla Tote_Hdr y verifica que el dato que regresa de la consulta, el código de WCS_DestinationArea sea SPAC o SMC3, determinando que estos códigos son para totes Singles. También se consulta la información sobre la capacidad máxima de totes singles que puede haber en el área de transfer 03 y 04. Comparamos el contador con el límite; si este contador llega a ser mayor que el límite, no dejará pasar más totes Singles, por lo tanto, se mandará a recircular, retornando un valor 2. Hasta que el contador baje, entonces podrá dejar pasar más totes Singles y retorna un valor 99.

De los valores que retornamos, evaluamos si es un 2 o 99, donde 2 se reemplaza por el número 1 para indicar que debe de desviar y dar la vuelta hacia cam 02, y 99 indica que puede continuar derecho hacia transfer 03 y 04, con esto enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```

{
  "divert_code": 99, //(99 indica que puede pasar a Transfer 03 y 04. 1 indica que debe
mantenerse en loop en el Inbound Spine)
  "tracking_id": 0 : //(TrackingId seguimiento del tote)
}

```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/api/ToteInformation/receive-flow/borderPicking	Border picking(border transfer03 and 04)	{ "camId": "string", "toteLPN": "string", "trackingId": 0, "scanner_N_lane_w_status": 0, "scanner_N_lane_w_full": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 10

Disponibilidad del Commissioner

URL: <http://10.107.92.43:8088/api/Commissioner/GetCommissionerStatus>

Método: GET

El Commissioner cuenta con 2 estados, encendido para indicar que el commissioner está presente y apagado para indicar que el commissioner está ausente. Por medio de la api Commissioner donde se obtiene el estado del mismo. Para ello se realiza una consulta a la tabla Commissioners devolviendo una lista, una vez hecha la consulta, se verifica que el resultado devuelto contenga algún valor, si es así, se toma en cuenta el primer registro que trae el resultado de la consulta, al final retorna el valor booleano del resultado de Status del commissioner.

Una vez que se termina de realizar el proceso, se retorna un JSON de respuesta, donde "true" es para indicar que está activo y "false" para indicar que está apagado:

```
{  
  "status": true  
}
```

Si en dado caso no exista algún registro del commissioner, se retorna una StatusCode 404 con el mensaje de la excepción:

Not found:

```
{  
  message:"string"  
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
GET	http://10.107.92.43:8088/api/Commissioner/GetCommissionerStatus	Commissioner	N/A	<pre>{ "status": false }</pre>	<i>Not found:</i> <pre>{ message:" string" }</pre>	N/A

Límites para las máquinas cuando el commissioner está presente

URL: <http://10.107.92.43:8088/api/CommissionerPackingLimits>

Método: GET

Cuando el operador activa la máquina commissioner físicamente, se cambian los límites por los que se tienen en la tabla commissioner packing limits, cada que se registre un tote dentro de las máquinas LPUT, SPTA, estos límites se consultan en este método GET, devolviendo el siguiente JSON con los cuatro límites al usuario:

```
{
  "id": 1, // Id del registro del límite del commissioner presente
  "putStationNr": 62, // Número de estación de la máquina
  "limit": 342 // Límite actual de la máquina
}
```

El límite de totes que puede recibir cada máquina se consulta en la tabla externa *E1_EXT* de la tabla *Commissioner_Packing_Limits*. Estos límites determinan la capacidad de totes que puede manejar cada máquina cuando el commissioner está presente.

El código putStationNr 1 corresponde a Putwall1, el putStationNr 2 corresponde a Putwall 2, el putStationNr 61 corresponde a Suresort 2, y el putStationNr 62 corresponde a Suresort 1.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
GET	http://10.107.92.43:8088/api/CommissionerPackingLimits	Commissioner picking limits	N/A	[{ "id": 0, "putStationNr": 0, "limit": 0 },]	Internal Server Error: { "message": " string" }	N/A

Transferencia de totes de picking a Transfer 03 y 04

Ubicación: Cam10

URL: <http://10.107.92.43:8088/api/ToteInformation/receive-flow/borderPicking>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", // (Cámara que escanea en ese momento)
  "toteLPN": "string", // (Codigo LPN)
  "trackingId": 0, // (Seguimiento del tote)
  "scanner_N_lane_w_status": 0, // (Indica si la línea está activa)
  "scanner_N_lane_w_full": 0 // (Indica si la línea está llena)
}
```

}

Se utiliza para dejar pasar totes que pertenecen a picking, tanto multis como singles, los totes de picking empiezan con la letra "T", y tienen los códigos de área: SPTA , LPUT, SPAC y SMC3 en la columna WCS_DestinationArea en la tabla Tote_Hdr y deben de tener el campo Processed en null.

Recibimos los datos que se envían a través de la solicitud para posteriormente realizar las consultas necesarias a las bases de datos interna y externa, para determinar si es posible dejar pasar los totes hacia transfer 03 y 04 o no.

Consultas a la base de datos interna cliente-db:

- En la tabla ToteInformations: Buscamos el registro del ToteLPN y que su DivertStatus sea null.
- En la tabla Commissioner: Buscamos el registro para obtener el status del commissioner.
- En la tabla HighwayPikingLanes: Buscamos el registro para obtener el contador de totes multis.
- En la tabla LimitSettingCameras: Buscamos el registro para obtener el contador de totes singles.

Consultas a la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Buscamos el registro del Tote_LPN, Processed null y al final, ordenamos el resultado por timestamp, tomando como resultado el más antiguo. También se retorna un listado de totes que pertenecen a la misma ola.
- En la tabla SharedTable: Buscamos el registro de la ola al que pertenece el tote_LPN

Una vez obtenido las consultas procedemos a evaluar los resultados, tomando en cuenta dos casos: Caso de Multis con dos escenarios, Commissioner Presente y Commissioner Ausente y caso de Singles donde:

Se valida primero que el resultado de la consulta a la tabla Tote_Hdr, regrese datos con respecto al toteLPN que se está consultando para posteriormente poder realizar las validaciones correspondientes y se actualiza en la tabla ToteInformations la columna LastCam con la cámara que escanea en ese momento, si en dado caso de que la consulta no regrese información con respecto al toteLPN buscado, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 2.

Caso para Multi Totes

Commissioner Presente

- Verificamos que el status del commissioner sea True.

- Validamos si es un tote Reasignado, es decir Release mayor o igual a 2, si es así agregamos un nuevo registro en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 99.
- Verificamos que el código de WCS_DestinationArea sea SPTA o LPUT y tengan Status mayor o igual a 1.
- Válida que el Tote empiece con la letra "T".
- Verifica que el contador de Totes no supere el límite
- Verifica que exista el registro de la ola del Tote en la tabla SharedTable. Si esto se cumple, actualizamos el campo Release = 2 del registro del Tote en la tabla Tote_Hdr, sumamos + 1 el contador de totes para multis, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final, retornamos un valor 99.

Commissioner Ausente

- Verificamos que el status del commissioner sea False.
- Validamos si es un tote Reasignado, es decir Release mayor o igual a 2, si es así, para el caso de los multis, validamos que el WCS_DestinationArea sea SPTA o LPUT, Status sea mayor o igual a 3.
Para los singles, validamos que el WCS_DestinationArea sea SPAC o SMC3. Si estas validaciones se cumplen, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 99.
- Validamos que el Status de la ola sea 3 o 4 y la ola este completa, es decir, el número de totes del campo Nr_Of_Totes_In_Wave, debe ser igual a la lista de la ola completa, donde todos los totes que pertenecen a esa ola deben tener Status mayor o igual a 3.
- Validamos que el tote empiece con la letra "T".
- Verificamos que el contador de totes no supere el límite de totes que puede dejar pasar, si se supera el contador se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 2.
- Validamos que el código de WCS_DestinationArea sea igual a SPTA o LPUT, si esto se cumple, actualizamos a Release = 2 el registro del Tote_LPN en la Tabla Tote_Hdr, aumentamos + 1 el contador de totes para multis, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 99.

Caso de singles

- Validamos que el código de WCS_DestinationArea sea SPAC o SMC3
- Verificamos que el Tote_LPN empiece con la letra "T".

- Verificamos que el contador de totes para single, el límite no sea superado, si es así agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 2.
- Si el contador está por debajo del límite de singles, verifica que el status sea 1, si es así, actualizamos Release = 2 del registro del Tote_LPN de la tabla Tote_Hdr, aumentamos el contador de singles + 1, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 99.

Si alguna de estas validaciones no se cumple, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 2.

De los valores que retornamos, evaluamos si es un 2 o 99, donde 2 se reemplaza por el valor 1 para indicar que debe de desviar y dar la vuelta hacia cam02, y 99 indica que puede continuar derecho hacia transfer 03 y 04, con esto enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 99, // (99 indica que puede pasar a Transfer 03 y 04. 1 indica que
    debe mantenerse en loop en el Inbound Spine)
  "tracking_id": 0 // (TrackingId seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/api/ToteInformation/receive-flow/borderPicking	Border picking(border transfer03 and 04)	{ "camId": "string", "toteLPN": "string", "trackingId": 0, "scanner_N_lane_w_status": 0, "scanner_N_lane_w_full": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 10

Liberación del Looping

Ubicación: Cam11

URL: <http://10.107.92.43:8088/divert-flow/divert/TotesCam11>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", // (Cámara que escanea en ese momento)
  "trackingId": 0, // (Tracking de seguimiento del tote)
  "toteLpn": "string", // (Codigo LPN)
  "scannerNLaneWStatus": 0, // (Indica si la línea está activa)
  "scannerNLaneWFull": 0 // (Indica si la línea está llena)
}
```

Se utiliza para mantener en loop los totes pertenecientes a picking, es decir multis y singles. Si la capacidad de totes aún no ha sido superada, se mantendrán recirculando hasta que puedan desviar a la cam12 si son multis, y cam13 si son singles. Para ello, recibimos los datos que nos envían a través de la solicitud, para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determinar si es posible hacer el desvío o no.

Consultas a la base de datos interna cliente-db:

- En la tabla ToteInformations: Se busca el registro del ToteLPN y que su DivertStatus sea null.
- En la tabla Commissioner: Se busca el registro para obtener el status del commissioner.
- En la tabla HighwayPikingLanes: Se busca el registro para obtener el contador de totes multis.
- En la tabla LimitSettingCameras: Se busca el registro para obtener el contador de totes singles.

Consultas a la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Buscamos el ToteLPN, el Status diferente de 6, Release mayor o igual a 2, Processed sea null y al final ordenamos el resultado por timestamp, tomando como resultado el más antiguo.
- En la misma Tabla Tote_Hdr: Buscamos del resultado de Tote_LP, el valor de número de totes que tiene la ola, los totes que pertenecen a la misma ola y que tengan Status mayor o igual a 3, regresamos una lista con la cantidad de resultados encontrados.

Una vez obtenido las consultas procedemos a evaluar los resultados, tomando en cuenta tres escenarios: Commissioner Presente, Commissioner Ausente, Single Tote, donde:

Commissioner Presente

- Se verifica que el resultado de la búsqueda del ToteLPN en ToteInformations, regrese datos, si es así se actualiza los campos DivTimestamp con la fecha de la petición, TrackingId que manda PLC y la cámara que escanea en ese momento.
- Se verifica que el resultado de la búsqueda del Tote_LPN en Tote_Hdr, regrese datos.
- Se verifica que el Status del commissioner sea True y el código de WCS_DestiantionArea sea SPTA o LPUT
- Se verifica que el contador no supere el límite de totes que está definido, si esto se cumple actualizamos el resultado del ToteLPN de ToteInformations el campo LineCount sumandole + 1, luego agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 2.

Commissioner Ausente

- Se verifica que el resultado de la búsqueda del ToteLPN en ToteInformations, regrese datos, si es así se actualiza los campos DivTimestamp con la fecha de la petición, TrackingId que manda PLC y la cámara que escanea en ese momento.
- Se verifica que el resultado de la búsqueda del Tote_LPN en Tote_Hdr, regrese datos.
- Se verifica que el Status del commissioner sea False y el código de WCS_DestiantionArea sea SPTA o LPUT
- Se verifica que el contador no supere el límite de totes que está puesto.
- Se verifica que el status del Tote_LPN sea mayor o igual a 3 y que el número de totes de la ola, sea igual a la lista de números de olas para saber que la ola está completa, si esto se cumple actualizamos el resultado del ToteLPN de ToteInformations el campo LineCount sumandole + 1, luego agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 2.

Single Tote

- Se verifica que el resultado de la búsqueda del ToteLPN en ToteInformations, regrese datos, si es así se actualiza los campos DivTimestamp con la fecha de la petición, TrackingId que manda PLC y la cámara que escanea en ese momento.

- Se verifica que el código de WCS_DestinationArea sea SPAC o SMC3 y Status sea igual a 1.
- Se verifica que el contador no supere el límite de totes que puede haber para los singles, si esto se cumple, actualizamos el resultado del ToteLPN de TotelInformations el campo LineCount sumandole + 1, luego agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 2.

Si las validaciones que están dentro de las condiciones para Commissioner Presente, apagado o para los casos de singles, en alguno momento no se cumple, restamos el contador correspondiente - 1, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos de los resultados de la búsqueda de TotelInformations y Tote_Hdr, al final retornamos un valor 99.

Si en dado caso no entra en la validación de resultado de la búsqueda de TotelInformations o Tote_Hdr, solamente agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos de los resultados de la búsqueda de TotelInformations y Tote_Hdr, al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 2, //(2 indica que puede desviar y mantenerse en loop entre transfer
03 y 04.
// 99 indica que debe regresar al Inbound Spine)
  "tracking_id": 0 //(Tracking de seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert/TotesCam11	Tote Loop Diverts in Transfer 03 & 04	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNL aneWStatus": 0, "scannerNL	{ "divert_code": 0, "tracking_id": 0 }	BadReque st { message:" string" }	Camera 11

			<code>aneWFull":</code> <code>0</code> <code>}</code>			
--	--	--	---	--	--	--

SingleTote

Desvío para totes singles - Cam13

Ubicación: Cam13

URL : <http://10.107.92.43:8088/divert-flow/divert/singleTote>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", //(Cámara que escanea en ese momento)
  "trackingId": 0, //(Tracking de seguimiento del tote)
  "toteLpn": "string", //(Codigo LPN)
  "scannerNLaneWStatus": 0, // (Indica si la línea está activa)
  "scannerNLaneWFull": 0 // (Indica si la línea está llena)
}
```

Se utiliza para desviar totes que son singles, se identifican por el código de WCS_DestinationArea, puede ser SPAC o SMC3, para ello recibimos los datos que nos envían a través de la solicitud para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determina si es posible hacer el desvío o no.

Se consulta en la base de datos interna, cliente-db,;

- En la tabla ToteInformations: Buscamos el registro del ToteLPN y que su DivertStatus sea null.

Se consulta en la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Buscamos el registro del Tote_LPn y processed sea null, ordenamos el resultado por fecha, tomando el más antiguo.

Una vez que se concluyen las búsquedas, se evalúan los resultados;

- El resultado de la búsqueda en la tabla de Tote_Hdr, regrese información con respecto al toteLPN, si en dado caso de que la consulta no regrese información con respecto al toteLPN buscado, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 99.
- Se valida que el tamaño de la ola del tote sea de 1.
- El resultado de la búsqueda en Totelformation con divertStatus null, debe de regresar un resultado con datos, si regresa null, automáticamente se interrumpe el proceso de evaluación y se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr y retorna un valor 99.
- El código de WCS_DestinationArea sea igual a SPAC o SMC3, si esto se cumple, actualizamos los datos del registro encontrado del ToteLPN de la tabla TotelInformations, los campos: DivTimestamp con el tiempo actual de sistema al momento de hacer el desvío, TrackingId se actualiza con el nuevo valor que PLC manda y la cámara que escanea en ese momento.
- Validar que el código ToteLPN comience con "T", si esto se cumple, procede a hacer un registro nuevo de scanlog en la tabla ScanlogsReceivingPickings, para guardar la información del resultado de la búsqueda encontrada en la tabla Tote_Hdr y TotelInfomation con el valor del ToteLPN y al final se retorna un valor 2.

Si no se cumplen estas validaciones, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings donde se captura la información correspondiente de las tablas TotelInformations y Tote_Hdr, con el valor correspondiente del ToteLPN y al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 2, //(2 indica que puede pasar. 99 indica que no puede pasar y debe
dar la vuelta hacia cam11)
  "tracking_id": 0 //(Tracking de seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert/si	Divert tote single	{ "camId": "string", "trakingId": 0, "toteLpn": "string",	{ "divert_code": 0, "tracking_id"	BadReques t { message:"st	Camera 13

	ngleTote		"scannerNLaneW : 0 Status": 0, "scannerNLaneW Full": 0 }	}	ring"	
--	--------------------------	--	--	---	-------	--

Confirmación de desvío para totes singles

Ubicación: Cam13

URL: <http://10.107.92.43:8088/divert-flow/confirmation>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en la cámara en la que va en ese momento, el timestamp del toteLPN y el tracking_ID asociado.

```
{
  "camId": "Cam13", //(Cámara que escanea en ese momento)
  "trackingId": 13, //(Tracking de seguimiento del tote)
  "divertCode": 100 //(Código de desvío)
}
```

Con esa información primero se calcula un rango de tiempo utilizando la variable div_timestampPLC y se busca un registro en la base de datos ToteInformations que coincida con este rango de tiempo, el ID de seguimiento (trackingId) y que no tenga un estado de desvío (DiversStatus) asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api .

Luego, se realiza una consulta para encontrar el tote correspondiente en la tabla Tote_Hdr en la base de datos externa E1_EXT basándose en el LPN del tote encontrado anteriormente.

Se verifica si la cámara que está realizando la confirmación es la "Cam13" y si el tote tiene como destino "SPAC" o "SMC3".

Para comprobar el límite de totes Singles, se debe consultar a la tabla LimitSettingCameras que cuenta con el registro configurado por el usuario desde el UI. Como el contador está configurado en "Cam10", se debe usar su identificador (Id) para obtener el valor del campo "MaximumCapacity" del registro de la tabla LimitSettingCameras. Si el límite está por encima de cero, se decrementa el contador en -1 y se actualiza el estado del tote en la base de datos.

Depende si el destino del tote es "SPAC" o "SMC3", se actualiza el estado del tote en la tabla Tote_Hdr en la base de datos externa E1_EXT. Si es "SPAC", se establece

Status = 6, Release = 3, Processed = 1", de lo contrario, solo Status = 6, Release = 3".

Después se registra un Scanlog con información detallada sobre el tote, como ToteLPN, VirtualZone, VirtualTote, Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed, Statuscliente, LapCount, TrackingId, Timestamp, CamId, DivertCode, Info.

Se retorna un OK200 si la confirmación se realiza con éxito y el siguiente JSON si ocurre algún problema durante el proceso:

BadRequest

```
{  
  message:"string"  
}
```

Method	URL	Name	JSON Request	JSON Response	Error Request
POST	http://10.107.92.43:8088/api/divert-flow/confirmation	confirmation	<pre>{ "camId": "string", "trackingId": 0, "divertCode": 0 }</pre>	OK200	<pre>BadRequest { message:"string" }</pre>

Multi Totes

Destino Cam12

Desvío

Ubicación: Cam12

URL: <http://10.107.92.43:8088/divert-flow/divert/multiTote>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{  
  "camId": "string", //(Cámara que escanea en ese momento)  
  "trackingId": 0, //(Tracking de seguimiento del tote)
```

```

"toteLpn": "string", //(Codigo LPN)
"scannerNLaneWStatus": 0, // (Indica si la línea está activa)
"scannerNLaneWFull": 0 // (Indica si la línea está llena)
}

```

Se utiliza para desviar totes multis que se identifican con los códigos SPTA o LPUT en la columna WCS_DestinationArea, para ello recibimos los datos que nos envían a través de la solicitud de desvío, para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determina si es posible hacer el desvío o no.

Consultas a la base de datos interna cliente-db:

- En la tabla ToteInformations: Se busca el registro del ToteLPN y que su DivertStatus sea null.
- En la tabla Commissioner: Se busca el registro para obtener el status del commissioner
- En la tabla DivertOutboundLines: Se busca el primer y segundo registro para obtener el contador de totes, donde el primer registro es para Commissioner Ausente y el segundo para Commissioner Presente.

Consultas en la base de datos externa E1_EXT:

- En la tabla Tote_Hdr en la base de datos externa E1_EXT: Se busca el registro del Tote_LP_N, Processed null y al final ordenamos el resultado por timestamp, tomando como resultado el más antiguo.
- En la tabla SharedTable: Se busca el registro de la ola al que pertenece el tote_LP_N

Una vez hechas las consultas, procedemos a evaluar los resultados, tomando en cuenta dos escenarios: Commissioner Presente y Commissioner Ausente, donde:

Se valida primero que el resultado de la consulta a la tabla Tote_Hdr, regrese datos con respecto al toteLPN que se está consultando para posteriormente poder realizar las validaciones correspondientes, si en dado caso de que la consulta no regrese información con respecto al toteLPN buscado, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 99.

Commissioner Presente

- Se verifica que el resultado de la búsqueda de la tabla ToteInformations, regrese un resultado con datos.
- Se valida que el Status del commissioner sea True.
- Se valida que el código de WCS_DestinationArea sea SPTA o LPUT, si esto se cumple, actualizamos el resultado encontrado de ToteInformations, los campos DivTimestamp con la fecha del sistema, TrackingId con el valor que PLC nos manda y la cámara que escanea en ese momento.

- Se verifica si el tote es reasignado (es cuando vuelve a su última posición del trayecto después de ser sacado y colocado en otra sección anterior), es decir si Release es mayor o igual a 3, si cumple esta condición, se envía un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations en la base de datos interna cliente_bd y Tote_Hdr en la base de datos externa E1_EXT, al final retornamos un valor 2.
- Se valida que el Tote_LPN empiece con la letra "T".
- Se valida que el contador aún no supere el límite máximo puesto para totes multis, para ello se toma en cuenta el segundo registro de la tabla DivertOutboundLines.
- Se valida que exista el registro de la ola en la tabla SharedTable. Si esto se cumple, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations en la base de datos interna cliente_bd y Tote_Hdr en la base de datos externa E1_EXT, al final retornamos un valor 2.

Commissioner Ausente

- Se valida que el código de WCS_DestinationArea sea SPTA o LPUT, si esto se cumple, actualizamos el resultado encontrado de TotelInformations, los campos DivTimestamp con la fecha del sistema, TrackingId con el valor que PLC nos manda y la cámara que escanea en ese momento.
- Se verifica si el tote es reasignado (es cuando vuelve a su última posición del trayecto después de ser sacado y colocado en otra sección anterior), es decir Release es mayor o igual a 3 y Status sea 4 o 5, si cumple esta condición, enviamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations en la base de datos interna _bd y Tote_Hdr en la base de datos externa E1_EXT, al final retornamos un valor 2.
- Se valida que el resultado de la búsqueda del ToteLPN en totelInformations, regrese datos.
- Se valida que el Tote_LPN empiece con la letra "T".
- Se verifica que el status del tote sea 4 o 5, y el contador de totes no supere el límite máximo para multis. Para ello se toma en cuenta el primer registro de la tabla DivertOutboundLines. Si esto se cumple, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations en la base de datos interna cliente_bd y Tote_Hdr en la base de datos externa E1_EXT, al final retornamos un valor 2.

Si dentro de las validaciones de los escenarios del commissioner ausente y presente, no se cumplen las condiciones, agregamos un nuevo registro en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations en la base de datos interna cliente_bd y Tote_Hdr en la base de datos externa E1_EXT, al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 2, //(2 indica que puede desviar. 99 indica que no puede pasar y debe
continuar derecho)
  "tracking_id": 0 //(Tracking de seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert/multiTote	Divert tote multi	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLaneWStatus": 0, "scannerNLaneWFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" } }	Camera 12

Confirmación de desvío

Ubicación: Cam12

URL: <http://10.107.92.43:8088/divert-flow/confirmation>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en ese momento, la cámara en la que va, el timestamp del toteLPN y el tracking_ID asociado.

```
{
  "camId": "Cam12", //(Cámara que escanea en ese momento)
  "trackingId": 12, //(Tracking de seguimiento del tote)
  "divertCode": 100 //(Código de desvío)
}
```


Con esa información se utiliza el timestamp del toteLPN para calcular un rango de tiempo dentro del cual se espera que el tote haya sido registrado en la base de datos.

Se verifica si el tote es reasignado (es cuando vuelve a su última posición del trayecto después de ser sacado y colocado en otra sección anterior), en dado caso de que sea, lo dejará pasar sin afectar los contadores de DivertOutboundLine y highwayPikingLane a su vez agregando un scanlog de tote reasignado. En dado caso de que no sea así, se continuará el proceso normal.

Se realiza una consulta a la tabla ToteInformations para encontrar un registro que coincida con: este rango de tiempo, el tracking de seguimiento del tote (trackingId), y que no tenga un estado de desvío (DivertStatus) asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api.

Si coinciden las condiciones de búsqueda con un registro, se realiza una consulta a la tabla Tote_Hdr en la base de datos externa E1_Ext para encontrar el tote correspondiente utilizando el LPN del tote obtenido anteriormente.

Se verifica que el status del tote no sea 6 (indicando que ya ha sido confirmado) y que el proceso no haya sido completado.

Se verifican varias condiciones para determinar si el desvío puede ser confirmado:

- Se verifica si la cámara que está realizando la confirmación es la "Cam12".
- Se verifica si existe el commissioner presente o ausente en el sistema.

Commissioner Presente

Se verifica si existe el límite para totes para DivertOutboundLine para commissioner presente (Segundo registro de la tabla "DivertOutboundLine"), en ese caso de que exista se incrementa el contador en +1 de totes.

Se decrementa el contador de totes en el carril de salida del desvío (highwayPikingLane) para dejar espacio por si hay totes disponibles queriendo entrar en transfer 03 y 04.

La consulta SQL para actualizar el estado del tote en la tabla Tote_Hdr en la base de datos externa E1_Ext se ejecuta bajo las siguientes condiciones:

Se actualiza el release a 3 del tote solo si el status actual es menor que 6 (indicando que no se ha completado el proceso) y no ha sido procesado aún.

Commissioner Ausente

- Se verifica si existe el límite para totes para DivertOutboundLine para commissioner ausente (Primer registro de la tabla "DivertOutboundLine"), en ese caso de que exista se incrementa el contador en +1 de totes.

- Se decrementa el contador de totes en el carril de salida del desvío (highwayPikingLane) para dejar espacio por si hay totes disponibles queriendo entrar en transfer 03 y 04.
- La consulta SQL para actualizar el estado del tote en la tabla Tote_Hdr en la base de datos externa E1_Ext cuando el commissioner está ausente se ejecuta bajo las siguientes condiciones adicionales:
- Se actualiza el release a 3 del tote solo si el status actual es 4 o 5 (indicando que el proceso de lanzamiento está en espera) y no ha sido procesado aún.

Se registra el evento de confirmación en el registro de escaneo (Scanlog) con información detallada sobre el tote, como ToteLPN, VirtualZone, VirtualTote, Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed, Statuscliente, LapCount, TrackingId, Timestamp, CamId, DivertCode, Info.

Se retorna un OK200 si la confirmación se realiza con éxito y el siguiente JSON si ocurre algún problema durante el proceso:

BadRequest

```
{
  message:"string"
}
```

Method	URL		JSON Request	JSON Response	Error Request
POST	http://10.107.92.43:8088/api/divert-flow/confirmation	confirmation/	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }

Multi Totes - Destino Cam14

Desvío

Ubicación: Cam14

URL: <http://10.107.92.43:8088/divert-flow/divert/multiToteCam14>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", //(Cámara que escanea en ese momento)
  "trackingId": 0, //(Tracking de seguimiento del tote)
  "toteLpn": "string", //(Codigo LPN)
  "scannerNLaneWStatus": 0, // (Indica si la línea está activa)
  "scannerNLaneWFull": 0 // (Indica si la línea está llena)
}
```

Se utiliza para desviar los totes multis que tiene Código SPTA y LPUT en la columna WCS_DestinationArea en la tabla Tote_Hdr en la base de datos externa E1_Ext, para ello recibimos los datos que nos envían a través de la solicitud para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determina si es posible hacer el desvío o no.

Consultas en la tabla interna cliente-db:

- En la tabla TotelInformations: Buscamos el registro del ToteLPN y que su DivertSatus sea null.
- En la tabla Commissioners: Buscamos el registro del commissioner.

Consulta a la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Buscamos el registro del toteLPN, que su campo Release sea menor o igual a 4, Processed sea null y ordenamos el registro por fecha, tomando como resultado el más antiguo.

Una vez hechas las consultas, procedemos a evaluar los resultados, tomando en cuenta dos escenarios: Commissioner Presente y Commissioner Ausente, donde:

Commissioner Presente

- Se verifica que el resultado de la consulta del comissioner tenga status en True para que puedan desviar los totes que lleguen a pasar.
- El resultado de la búsqueda de la tabla TotelInformations debe de regresar un resultado con datos, si esto se cumple, actualizamos el resultado encontrado de TotelInformations, los campos DivTimestamp con la fecha del sistema, TrackingId con el valor que PLC nos manda y la cámara que escanea en ese momento.
- El resultado de la búsqueda en Tote_Hdr debe regresar datos, si se cumple agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 2.

Commissioner Ausente

- En ningún momento entrará en alguna de las validaciones para commissioner True y enviará un nuevo registro de scanlog en la tabla

ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 99.

Si no cumple con ninguna de las condiciones, se agrega un nuevo de scanlog registro en la tabla ScanlogsReceivingPickings con los datos de los resultados de la búsqueda de ToteInformations y Tote_Hdr, al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 2, //(2 indica que puede pasar. 99 indica que no puede pasar y debe
continuar derecho)
  "tracking_id": 0 //(Tracking de seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert/multiToteCam14	Diversión Comisionado estado Verdadero	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNewStatus": 0, "scannerNewFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 14

Confirmación de desvío

Ubicación: Cam14

URL: <http://10.107.92.43:8088/divert-flow/confirmation/Cam14>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en ese momento, la cámara en la que va, el timestamp del toteLPN y el tracking_ID asociado.

```
{
  "camId": "Cam14", //(Cámara que escanea en ese momento)
```

```



```

Con esa información se utiliza el timestamp del toteLPN para calcular un rango de tiempo dentro del cual se espera que el tote haya sido registrado en la base de datos. Se verifica si el tote es reasignado (es cuando vuelve a su última posición del trayecto después de ser sacado y colocado en otra sección anterior) en dado caso de que sea este lo dejara pasar sin afectar los contadores (DivertOutboundLine) dependiendo el estado del commissioner en dado caso de que no sea se continuará el proceso normal.

Se realiza una consulta a la tabla ToteInformations para encontrar un registro que coincida con: este rango de tiempo, el tracking de seguimiento del tote (trackingId), y que no tenga un estado de desvío (DivertStatus) asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api.

Antes de confirmar el desvío del tote, se verifican varias condiciones:

- Se busca el tote en la tabla Tote_Hdr en la base de datos externa E1_EXT basado en el LPN. Se asegura de que el status del tote no sea 6 (completado) y que el proceso no haya sido realizado.
- Se obtienen los límites de totes presentes en las DivertOutboundLines.
- Se comprueba si la cámara que está realizando la confirmación es la "Cam14".

Si todas las condiciones anteriores se cumplen, se procede a confirmar el desvío del tote. Esto implica actualizar el estado del tote en la tabla Tote_Hdr mediante una consulta SQL a Tote_Hdr en la base de datos externa E1_Ext, estableciendo el release en 4. También se actualiza el conteo de totes en la tabla DivertOutboundLines, restando - 1 al contador del segundo registro que hay en la tabla.

Luego se registra el evento de confirmación en el registro de escaneo (Scanlog) con información detallada sobre el tote, ToteLPN, VirtualZone, VirtualTote, Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed, Statuscliente, LapCount, TrackingId, Timestamp, CamId, DivertCode, Info.

Se retorna un OK200 si la confirmación se realiza con éxito y el siguiente JSON si ocurre algún problema durante el proceso:

```

BadRequest
{
  message:"string"
}

```

Method	URL		JSON Request	JSON Response	Error Request
POST	http://10.107.92.43:8088/api/confirmation/Cam14	Cam14	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	<i>BadRequest</i> { message:"string" }

Multi Totes - Destino Cam15

Desvío

Ubicación: Cam15

URL: <http://10.107.92.43:8088/divert-flow/divert/multiToteCam15>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", //(Cámara que escanea en ese momento)
  "trackingId": 0, //(Tracking de seguimiento del tote)
  "toteLpn": "string", //(Codigo LPN)
  "scannerNLaneWStatus": 0, // (Indica si la línea está activa)
  "scannerNLaneWFull": 0 // (Indica si la línea está llena)
}
```

Se utiliza para desviar Totes Multis que tienen código LPUT en la columna WCS_DestinationArea y número de estación 1 o 2 en la columna PutStation, para ello recibimos los datos que nos envían a través de la solicitud para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determina si es posible hacer el desvío o no.

Consultas a la base de datos interna cliente-db:

- En la tabla ToteInformations: Se busca el registro del ToteLPN y que su DivertStatus sea null.

Consultas a la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Se busca el ToteLPN, Processed sea null y al final ordenamos el resultado por timestamp, tomando como resultado el más antiguo.

Una vez obtenido las consultas procedemos a evaluar los resultados:

- Se verifica que el resultado de la búsqueda del ToteLPN en totelInformations regrese información relacionada con el ToteLPN que se está buscando.
- El código de WCS_DestinationArea sea LPUT en el registro del Tote_LPN en la tabla Tote_Hdr, si esto se cumple, actualizamos el resultado de TotelInformations, los campos DivTimestamp por la fecha del sistema, TrackingId que manda PLC y la cámara que escanea en ese momento.
- Se valida que el resultado de la búsqueda del Tote_LPN en ToteHdr tenga datos, Put_Station 1 o 2, Status 4 o 5, si esto se cumple, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 2.
- Si el valor de Put_Station es nulo, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con el mensaje indicando que no se puede desviar debido a que no tiene número de estación, y los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 99.

Si no cumple con ninguna de las condiciones, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos de los resultados de la búsqueda de TotelInformations y Tote_Hdr, al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 2, //(2 indica que puede pasar. 99 indica que no puede pasar y debe
continuar derecho)
  "tracking_id": 0 //(Tracking de seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert-multiToteCam15	Divert Put to Wall	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLaneWStatus": 0, "scannerNLaneWFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" } }	Camera 15

			}			
--	--	--	---	--	--	--

Confirmación de desvío

Ubicación: Cam15

URL: <http://10.107.92.43:8088/divert-flow/confirmation/Cam15>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en ese momento, la cámara en la que va, el timestamp del toteLPN y el tracking_ID asociado.

```
{
  "camId": "Cam15", //(Cámara que escanea en ese momento)
  "trackingId": 15, //(Tracking de seguimiento del tote)
  "divertCode": 100 //(Código de desvío)
}
```

Con esa información se utiliza el timestamp del toteLPN para calcular un rango de tiempo dentro del cual se espera que el tote haya sido registrado en la base de datos.

Se realiza una consulta a la tabla ToteInformations para encontrar un registro que coincida con este rango de tiempo, el trackingId, y que no tenga DivertStatus asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api.

Se busca en la tabla Tote_Hdr que el LPN contenga las condiciones, el status del tote sea 4 o 5 y que el processed esté en null.

Commissioner Presente

- Si se encuentra un tote en la tabla Tote_Hdr, se procede a verificar si hay límites de totes presentes en DivertOutboundLines y también si el commissioner está presente. Si todas estas condiciones se cumplen, se decrementa en -1 el límite de totes presentes en el segundo registro almacenado en la tabla DivertOutboundLines.

Commissioner Ausente

- Si se encuentra un tote en la tabla Tote_Hdr, se verifica el límite de DivertOutboundLines y también si el commissioner está ausente. Si todas estas condiciones se cumplen, se decrementa en -1 el límite de totes presentes en el primer registro almacenado en la tabla DivertOutboundLines .

También se actualiza el release del tote a 15 en la tabla Tote_Hdr así como también se actualiza como desviado el tote en la tabla interna Toteinformations en el campus status y se le coloca como desviado ("IN")

Después, se registra un Scanlog con información detallada sobre el tote, que incluye los siguientes campos: ToteLPN, VirtualZone, VirtualTote, Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed, Status, LapCount, TrackingId, Timestamp, CamId, DivertCode, Info.

Se retorna un OK200 si la confirmación se realiza con éxito y el siguiente JSON si ocurre algún problema durante el proceso:

BadRequest

```
{
  message:"string"
}
```

Method	URL	NAME	JSON Request	JSON Response	Error Request
POST	http://10.107.92.43:8088/api/confirmation/Cam15	Cam15	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }

Multi Totes - Destino Cam16

Desvío

Ubicación: Cam16

URL: <http://10.107.92.43:8088/divert-flow/divert/multiToteCam16>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", //(Cámara que escanea en ese momento)
  "trackingId": 0, //(Tracking de seguimiento del tote)
  "toteLpn": "string", //(Codigo LPN)
```

```

"scannerNLaneWStatus": 0, // (Indica si la línea está activa)
"scannerNLaneWFull": 0 // (Indica si la línea está llena)
}

```

Se utiliza para desviar Totes Multis que tienen código SPTA en la columna WCS_DestinationArea y número de estación 62 en la columna PutStation, para ello recibimos los datos que nos envían a través de la solicitud para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determina si es posible hacer el desvío o no.

Consultas a la base de datos interna cliente-db:

- En la tabla ToteInformations: Buscamos el registro del ToteLPN y que su DivertStatus sea null.

Consultas a la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Buscamos el ToteLPN, Processed sea null y al final ordenamos el resultado por timestamp, tomando como resultado el más antiguo.

Una vez obtenido las consultas procedemos a evaluar los resultados:

- Verificamos que el resultado de la búsqueda del ToteLPN en toteInformations, regrese datos
- El código de WCS_DestinationArea sea SPTA en el registro del Tote_LPN en la tabla Tote_Hdr, si esto se cumple, actualizamos el resultado de ToteInformations los campos DivTimestamp por la fecha del sistema, TrackingId que manda PLC y la cámara que escanea en ese momento.
- Validamos que el resultado de la búsqueda del ToteLPN en ToteHdr tenga datos, Put_Station 62, Status 4 o 5, si esto se cumple, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 2.
- Si el valor de Put_Station es nulo, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con el mensaje indicando que no se puede desviar debido a que no tiene número de estación, y los datos encontrados de la tabla ToteInformations y Tote_Hdr, al final retornamos un valor 99.

Si no cumple con ninguna de las condiciones, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos de los resultados de la búsqueda de ToteInformations y Tote_Hdr, al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```

{

```

```

    "divert_code": 2, //(2 indica que puede pasar. 99 indica que no puede pasar y debe
continuar derecho)
    "tracking_id": 0 //(Tracking de seguimiento del tote)
}

```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert-multiToteConfirmation16	Divert SureSort1	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLaneWidth": 0, "scannerNLaneWidthFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 16

Confirmación de desvío

Ubicación: Cam16

URL:<http://10.107.92.43:8088/divert-flow/confirmation/Cam16>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en ese momento, la cámara en la que va, el divertcode del toteLPN y el tracking_ID asociado.

```

{
  "camId": "Cam16", //(Cámara que escanea en ese momento)
  "trackingId": 16, //(Tracking de seguimiento del tote)
  "divertCode": 100 //(Código de desvío)
}

```

Con esa información se utiliza el timestamp del toteLPN para calcular un rango de tiempo dentro del cual se espera que el tote haya sido registrado en la base de datos.

Se realiza una consulta a la tabla ToteInformations para encontrar un registro que coincida con este rango de tiempo, el trackingId, y que no tenga DivertStatus asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api.

Se busca en la tabla Tote_Hdr que el LPN contenga las condiciones de que el status del tote sea 4 o 5 y que el processed esté en null.

Commissioner Presente

- Si se encuentra un tote en la tabla Tote_Hdr, se procede a verificar si hay límites de totes presentes en DivertOutboundLines y también si el commissioner está presente. Si todas estas condiciones se cumplen, se decrementa en -1 el límite de totes presentes en el segundo registro almacenado en la tabla DivertOutboundLines.

Commissioner Ausente

- Si se encuentra un tote en la tabla Tote_Hdr, se verifica el límite de DivertOutboundLines y también si el commissioner está ausente. Si todas estas condiciones se cumplen, se decrementa en -1 el límite de totes presentes en el primer registro almacenado en la tabla DivertOutboundLines .

También se actualiza el release del tote a 16 en la tabla Tote_Hdr así como también se actualiza como desviado el tote en la tabla interna Toteinformations en el campus status y se le coloca como desviado ("IN")

Después, se registra un Scanlog con información detallada sobre el tote, que incluye los siguientes campos: ToteLPN, VirtualZone, VirtualTote, Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed, Status, LapCount, TrackingId, Timestamp, CamId, DivertCode, Info.

Se retorna un OK200 si la confirmación se realiza con éxito y el siguiente JSON si ocurre algún problema durante el proceso:

```
BadRequest
{
  message:"string"
}
```

Method	URL	NAME	JSON Request	JSON Response	Error Request
POST	http://10.107.92.43:8088/api/confirmation/Cam16	Cam16	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	<i>BadRequest</i> { message:"string" }

Multi Totes - Destino Cam17

Desvío

Ubicación: Cam17

URL: <http://10.107.92.43:8088/divert-flow/divert/multiToteCam17>

Método: POST

En el cuerpo de la solicitud se debe de enviar un JSON donde PLC llena los datos que se piden, como es el número de cámara, el código LPN que llega a escanear la cámara, el trackingId para el seguimiento del tote, el status de la línea si está activa y el status de la línea si está llena:

```
{
  "camId": "string", //(Cámara que escanea en ese momento)
  "trackingId": 0, //(Tracking de seguimiento del tote)
  "toteLpn": "string", //(Codigo LPN)
  "scannerNLaneWStatus": 0, // (Indica si la línea está activa)
  "scannerNLaneWFull": 0 // (Indica si la línea está llena)
}
```

Se utiliza para desviar Totes Multis que tienen código SPTA en la columna WCS_DestinationArea y número de estación 62 en la columna PutStation, para ello recibimos los datos que nos envían a través de la solicitud para posteriormente realizar las consultas necesarias a las bases de datos correspondientes y determina si es posible hacer el desvío o no.

Consultas a la base de datos interna cliente-db:

- En la tabla ToteInformations: Buscamos el registro del ToteLPN y que su DivertStatus sea null.

Consultas a la base de datos externa E1_EXT:

- En la tabla Tote_Hdr: Buscamos el ToteLPN, Processed sea null y al final ordenamos el resultado por timestamp, tomando como resultado el más antiguo.

Una vez obtenido las consultas procedemos a evaluar los resultados:

- Verificamos que el resultado de la búsqueda del ToteLPN en toteInformations, regrese datos.
- El código de WCS_DestinationArea sea SPTA en el registro del Tote_LPn en la tabla Tote_Hdr, si esto se cumple, actualizamos el resultado de ToteInformations los campos DivTimestamp por la fecha del sistema y TrackingId que manda PLC y la cámara que escanea en ese momento.
- Validamos que el resultado de la búsqueda del ToteLPN en ToteHdr tenga datos, Put_Station 61, Status 4 o 5, si esto se cumple, agregamos un nuevo

registro de scanlog en la tabla ScanlogsReceivingPickings con los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 2.

- Si el valor de Put_Station es nulo, agregamos un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con el mensaje indicando que no se puede desviar debido a que no tiene número de estación, y los datos encontrados de la tabla TotelInformations y Tote_Hdr, al final retornamos un valor 99.

Si no cumple con ninguna de las condiciones, se agrega un nuevo registro de scanlog en la tabla ScanlogsReceivingPickings con los datos de los resultados de la búsqueda de TotelInformations y Tote_Hdr, al final retornamos un valor 99.

De los valores que retornamos, enviamos un JSON de respuesta con los siguientes datos que PLC lee y toma las decisiones correspondientes:

```
{
  "divert_code": 2, //(2 indica que puede pasar. 99 indica que no puede pasar y debe
continuar derecho)
  "tracking_id": 0 : //(Tracking de seguimiento del tote)
}
```

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/divert-flow/divert-multiToteCam17	Divert SureSort2	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLane WStatus": 0, "scannerNLane WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" } }	Camera 17

Confirmación de desvío

Ubicación: Cam17

URL:<http://10.107.92.43:8088/divert-flow/confirmation/Cam17>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en ese momento, la cámara en la que va, el timestamp del toteLPN y el tracking_ID asociado.

```
{
  "camId": "Cam17", //(Cámara que escanea en ese momento)
  "trackingId": 17, //(Tracking de seguimiento del tote)
  "divertCode": 100 //(Código de desvío)
}
```

Con esa información se utiliza el timestamp del toteLPN para calcular un rango de tiempo dentro del cual se espera que el tote haya sido registrado en la base de datos. Se realiza una consulta a la tabla ToteInformations para encontrar un registro que coincida con este rango de tiempo, el trackingId, y que no tenga DivertStatus asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api.

Se busca en la tabla Tote_Hdr que el LPN contenga las condiciones de que el status del tote sea 4 o 5 y que el processed este en null.

Commissioner Presente

- Si se encuentra un tote en la tabla Tote_Hdr, se procede a verificar si hay límites de totes presentes en DivertOutboundLines y también si el commissioner está presente. Si todas estas condiciones se cumplen, se decrementa en -1 el límite de totes presentes en el segundo registro almacenado en la tabla DivertOutboundLines.

Commissioner Ausente

- Si se encuentra un tote en la tabla Tote_Hdr, se verifica el límite de DivertOutboundLines y también si el commissioner está ausente. Si todas estas condiciones se cumplen, se decrementa en -1 el límite de totes presentes en el primer registro almacenado en la tabla DivertOutboundLines .

También se actualiza el release del tote a 17 en la tabla Tote_Hdr así como también se actualiza como desviado el tote en la tabla interna Toteinformations en el campus status y se le coloca como desviado ("IN")

Después, se registra un Scanlog con información detallada sobre el tote, que incluye los siguientes campos: ToteLPN, VirtualZone, VirtualTote, Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed, Status, LapCount, TrackingId, Timestamp, CamId, DivertCode, Info.

Se retorna un OK200 si la confirmación se realiza con éxito y el siguiente JSON si ocurre algún problema durante el proceso:

BadRequest

```
{
  message:"string"
}
```

METHOD	URL	NAME	JSON Request	JSON Response	Error Request
POST	http://10.10.7.92.43:8088/api/confirmation/Cam17	Cam17	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }

PROCEDURE [UpdateToteCounts]

El procedure UpdateToteCounts tiene la funcionalidad de corregir los contadores de HighwayPikingLanes, LimitSettingCameras y DivertOutboundLines cada 3 minutos a través de una tarea programada en MS SQL Server:


```

CREATE OR ALTER PROCEDURE [dbo].[UpdateToteCounts]
AS
BEGIN
    BEGIN TRY
        -- Iniciar la transacción
        BEGIN TRANSACTION ;

        DECLARE @ContadorTote INT;
        DECLARE @ContadorMultiTotesHighway INT;
        DECLARE @ContadorMultiTotesDivert INT;
        DECLARE @StatusCommissioner BIT;

        -- Contador para la condición en Singles
        SELECT @ContadorTote = COUNT(*)
        FROM E1_EXT.dbo.Tote_Hdr WITH(NOLOCK)
        WHERE (WCS_Destination_Area = 'SPAC' OR WCS_Destination_Area = 'SMC3')
            AND Release = 2
            AND Processed IS NULL;

        -- Contador para la nueva condición en HighwayPikingLane
        SELECT @ContadorMultiTotesHighway = COUNT(*)
        FROM E1_EXT.dbo.Tote_Hdr WITH(NOLOCK)
        WHERE (WCS_Destination_Area = 'LPUT' OR WCS_Destination_Area = 'SPTA')
            AND Release = 2
            AND Processed IS NULL;

        -- Contador para la nueva condición en DivertOutboundLine
        SELECT @ContadorMultiTotesDivert = COUNT(*)
        FROM E1_EXT.dbo.Tote_Hdr WITH(NOLOCK)
        WHERE (WCS_Destination_Area = 'LPUT' OR WCS_Destination_Area = 'SPTA')
            AND (Release = 3 OR Release = 14)
            AND Processed IS NULL;

        -- Obtener el status de Commissioner
        SELECT @StatusCommissioner = Status
        FROM [v10-db].dbo.Commissioners
        WHERE Id = (SELECT TOP 1 Id FROM [v10-db].dbo.Commissioners ORDER BY Id);

        -- Actualizar la tabla LimitSetting en CounterTote
        UPDATE [v10-db].dbo.LimitSettingCameras
        SET CounterTote = @ContadorTote
        WHERE Id = (SELECT TOP 1 Id FROM [v10-db].dbo.LimitSettingCameras ORDER BY Id);

        -- Actualizar la tabla HighwayPikingLane en CountMultiTotes
        UPDATE [v10-db].dbo.HighwayPikingLanes
        SET CountMultiTotes = @ContadorMultiTotesHighway
        WHERE Id = (SELECT TOP 1 Id FROM [v10-db].dbo.HighwayPikingLanes ORDER BY Id);

        -- Verificar y actualizar la tabla DivertOutboundLine
        IF @StatusCommissioner = 1
        BEGIN
            -- Actualizar el primer registro si el status es true
            UPDATE [v10-db].dbo.DivertOutboundLines
            SET CountMultiTotes = @ContadorMultiTotesDivert
            WHERE Id = (SELECT TOP 1 Id FROM [v10-db].dbo.DivertOutboundLines ORDER BY Id desc);
        END
        ELSE
        BEGIN
            -- Actualizar el segundo registro si el status es false
            UPDATE [v10-db].dbo.DivertOutboundLines
            SET CountMultiTotes = @ContadorMultiTotesDivert
            WHERE Id = (SELECT TOP 1 Id FROM [v10-db].dbo.DivertOutboundLines ORDER BY Id);
        END;

        -- Confirmar la transacción
        COMMIT;
    END TRY
    BEGIN CATCH
        -- En caso de error, revertir la transacción
        IF @@TRANCOUNT > 0
            ROLLBACK ;
    END CATCH;
END;

```

- **Inicio del procedimiento almacenado y transacción:** El procedimiento comienza estableciendo una transacción para garantizar la consistencia de los datos durante las operaciones de actualización.
- **Recuperación de contadores:** Se ejecutan tres consultas para contar registros en la tabla Tote_Hdr basándose en diferentes condiciones relacionadas con el área de destino y el estado de procesamiento. Estos contadores se utilizan para actualizar diferentes tablas más adelante.
- **Obtención del estado del Commissioner:** Se recupera el estado del "Commissioner" de la base de datos.
- **Actualización de tablas:** Dependiendo del estado obtenido del Commissioner, se actualizan dos tablas diferentes, LimitSettingCameras y HighwayPikingLanes, con los contadores obtenidos anteriormente. Además, se actualiza la tabla DivertOutboundLines con uno de los contadores dependiendo del estado del Commissioner.
- **Confirmación de la transacción:** Si todas las operaciones se han ejecutado sin errores, se confirma la transacción, lo que implica que todos los cambios se hacen permanentes.
- **Manejo de errores:** Se incluye un bloque CATCH para manejar cualquier error que pueda ocurrir durante la ejecución del procedimiento. Si se produce un error, se revierte la transacción para asegurar que los datos no queden en un estado inconsistente.

Tote desviado a picking Jackpot

Desvío hacia Picking Jackpot

URL: <http://10.107.92.43:8088/divert-flow/divert>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en la cámara en la que va en ese momento, el toteLPN, el tracking_ID asociado, el status de la línea y si está llena o no.

```
{
  "camId": "Cam06",
  "trakingId": 6,
  "toteLpn": "TLPN0012",
  "scannerNLaneWStatus": 1,
  "scannerNLaneWFull": 0
}
```

Con estos datos, se consulta a la tabla ToteInformations en búsqueda del registro ToteLPN. Del resultado de la búsqueda, se valida que el ToteLPN empiece con la letra "T", el campo ZoneDivert tenga el valor 889, a su vez, válida que la cámara sea a la que corresponde a la línea asignada para Picking Jackpot. Si esto se cumple,

entonces se puede hacer el desvío correspondiente, agregando un registro de scanlog.

Este registro incluye información relevante como ToteLPN, VirtualZone, detalles específicos del tote (Wave, TotesInWave, TotalQty, DestinationArea, PutStation, Status, Release, Processed), el estado del desvío (Status), el recuento de vueltas (LapCount), el identificador de seguimiento (TrackingId), la marca de tiempo (Timestamp), el identificador de la cámara (CamId), el código de desvío (DivertCode), y un mensaje informativo indicando que el tote ha sido desviado a la línea de selección de jackpot.

En este caso, se devuelve el valor 2, lo que indica el éxito de la operación de desvío a la línea de selección de jackpot. Si en este caso, si la cámara que evalúa en ese momento, no es la que está asignada a picking jackpot, se devuelve el valor 99, lo que indica que no se puede realizar el desvío correspondiente.

Confirmación para Picking Jackpot

URL: <http://10.107.92.43:8088/divert-flow/confirmation>

Método: POST

En el cuerpo de la solicitud, se debe incluir el número de la cámara que realiza la comprobación en la cámara en la que va en ese momento, el timestamp del toteLPN y el tracking_ID asociado.

```
{  
  "camId": "Cam06", //(Cámara que escanea en ese momento)  
  "trackingId": 06, //(Tracking de seguimiento del tote)  
  "divertCode": 100 //(Código de desvío)  
}
```

Con esa información primero se calcula un rango de tiempo utilizando la variable div_timestampPLC y se busca un registro en la base de datos ToteInformations que coincida con este rango de tiempo, el ID de seguimiento (trackingId) y que no tenga un estado de desvío (DivertStatus) asignado, además que coincida el último registro de lastCam con el parámetro camId que nos llega en la api.

Comprueba si el ToteLPN comienza con la letra "T". Si es así, verifica si la propiedad ZoneDivertId sea igual a 889. Si esta condición también se cumple, procede a determinar si es una línea de picking jackpot.

Si todas las condiciones anteriores son verdaderas, se ejecutan las siguientes acciones:

- Se establece el estado del desvío (DivertStatus) como "IN".
- Se actualiza el TrackingId del objeto result a 0.

- Se actualiza el objeto result en el contexto de la base de datos.
- Se guardan los cambios realizados en la base de datos.
- Se agrega un nuevo registro de scanlogs. Este registro incluye información relevante como el ToteLPN, la zona virtual (VirtualZone), el recuento de vueltas (LapCount), TrackingId, la marca de tiempo (Timestamp), el identificador de la cámara (CamId), el código de desvío (DivertCode), y detalles adicionales (Info) que indican que se ha confirmado que el tote está siendo seleccionado para un picking jackpot.

Se retorna un OK 200 si la confirmación se realiza con éxito en caso de que sea picking jackpot y el siguiente JSON si ocurre algún problema durante el proceso:

BadRequest

```
{
  message:"string"
}
```

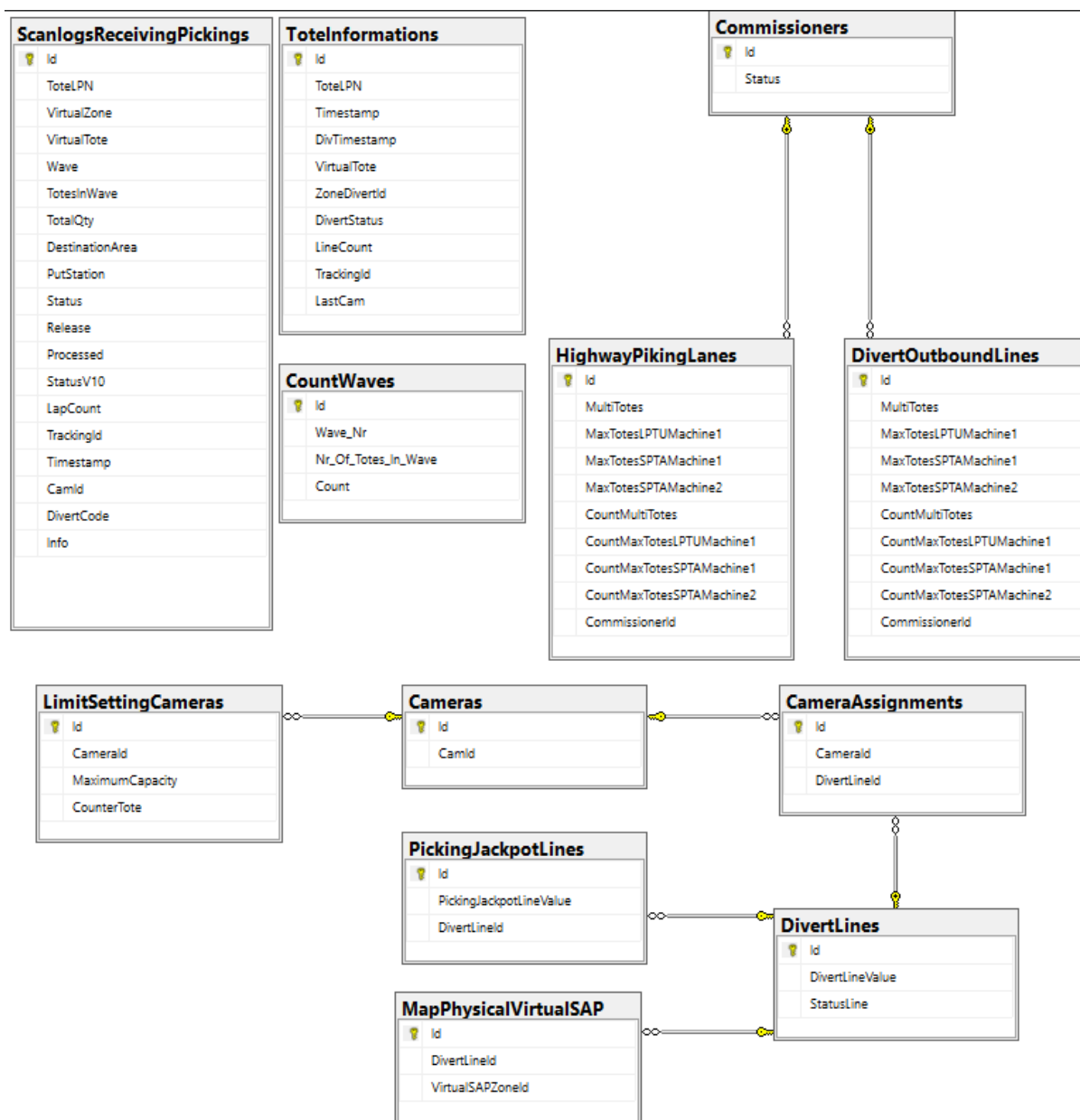
Tablas utilizadas para las APIS de Picking:

cliente-db		
ToteInformations	HighwayPikingLanes	DivertOutboundLines
<ul style="list-style-type: none"> Id (PK, int, not null) ToteLPN (nvarchar(10), not null) Timestamp (nvarchar(17), not null) DivTimestamp (nvarchar(17), null) VirtualTote (nvarchar(35), not null) ZoneDivertId (int, not null) DivertStatus (nvarchar(2), null) LineCount (int, not null) TrackingId (int, not null) LastCam (nvarchar(6), null) 	<ul style="list-style-type: none"> Id (PK, int, not null) MultiTotes (int, not null) MaxTotesLPTUMachine1 (int, not null) MaxTotesSPTAMachine1 (int, not null) MaxTotesSPTAMachine2 (int, not null) CountMultiTotes (int, not null) CountMaxTotesLPTUMachine1 (int, not null) CountMaxTotesSPTAMachine1 (int, not null) CountMaxTotesSPTAMachine2 (int, not null) CommissionerId (FK, int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) MultiTotes (int, not null) MaxTotesLPTUMachine1 (int, not null) MaxTotesSPTAMachine1 (int, not null) MaxTotesSPTAMachine2 (int, not null) CountMultiTotes (int, not null) CountMaxTotesLPTUMachine1 (int, not null) CountMaxTotesSPTAMachine1 (int, not null) CountMaxTotesSPTAMachine2 (int, not null) CommissionerId (FK, int, not null)
LimitSettingCameras	Commissioners	CountWaves
<ul style="list-style-type: none"> Id (PK, int, not null) CameralId (FK, int, not null) MaximumCapacity (int, not null) CounterTote (int, null) 	<ul style="list-style-type: none"> Id (PK, int, not null) Status (bit, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) Wave_Nr (nvarchar(max), not null) Nr_Of_Totes_In_Wave (int, not null) Count (int, not null)
ScanlogsReceivingPickings	PickingJackpotLines	MapPhysicalVirtualSAP

<ul style="list-style-type: none"> Id (PK, int, not null) ToteLPN (nvarchar(10), null) VirtualZone (int, null) VirtualTote (nvarchar(35), null) Wave (nvarchar(35), null) TotesInWave (int, null) TotalQty (int, null) DestinationArea (nvarchar(4), null) PutStation (int, null) Status (int, null) Release (int, null) Processed (bit, null) StatusV10 (nvarchar(2), null) LapCount (int, null) TrackingId (int, null) Timestamp (nvarchar(17), null) CamId (nvarchar(10), null) DivertCode (int, null) Info (nvarchar(100), null) 	<ul style="list-style-type: none"> Id (PK, int, not null) PickingJackpotLineValue (bit, not null) DivertLineId (FK, int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) DivertLineId (FK, int, not null) VirtualSAPZoneId (int, not null)
Cameras	CameraAssignments	DivertLines
<ul style="list-style-type: none"> Id (PK, int, not null) CamId (nvarchar(max), not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) CameraId (FK, int, not null) DivertLineId (FK, int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) DivertLineValue (int, not null) StatusLine (bit, not null)

E1_EXT	
Tote_Hdr	SharedTable
<ul style="list-style-type: none"> Id (int, not null) Tote_LPN (PK, char(35), not null) Wave_Nr (PK, char(35), not null) Wave_Priority (int, not null) Nr_Of_Totes_In_Wave (int, not null) Tote_Total_Qty (int, not null) WCS_Destination_Area (char(4), null) RFID_Relevance (nchar(1), null) Timestamp (char(50), not null) Put_Station_Nr (int, null) Status (int, null) Release (int, null) Processed (bit, null) 	<ul style="list-style-type: none"> Id (int, not null) Wave_Nr (PK, varchar(35), not null) ReleasedAt (datetime, null)

Diagrama Entidad-Relación cliente-db



URLS de las API's

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	http://10.107.92.43:8088/transfer-inboud/check-tote	TransferInbound	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLaneWStatus": 0, }	{ "divert_code": 2 }	BadRequest { message:"string" }	Camera 2

			"scannerNLANe WFull": 0 }			
POST	http://10.107.92.43:8088/divert-flow/divert	Divert	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLANe WStatus": 0, "scannerNLANe WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string "	Assignmenfrom UI
POST	http://10.107.92.43:8088/api/ToteInformation/receive-flow/borderPicking	Border picking(border transfer03 and 04)	{ "camId": "string", "toteLPN": "string", "trackingId": 0, "scanner_N_lan e_w_status": 0, "scanner_N_lan e_w_full": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string "	Camera 10
POST	http://10.107.92.43:8088/divert-flow/divert/TotesCam11	Tote Loop Diverts in Transfer03 & 04	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLANe WStatus": 0, "scannerNLANe WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string "	Camera 11
POST	http://10.107.92.43:8088/divert-flow/confirmation	Confirmatio n	{ "camId": "string", "trakingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string "	Camera 4 to camera 9,camera 12 and 13
POST	http://10.107.92.43:8088/divert-flow/divert/singleTote	Divert tote single	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLANe WStatus": 0, }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string "	Camera 13

			"scannerNLane WFull": 0 }			
POST	http://10.107.92.43:8088/divert-flow/divert/multiTote	Divert tote multi	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLane WStatus": 0, "scannerNLane WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 12
POST	http://10.107.92.43:8088/divert-flow/divert/multiToteCam14	Divert Commissioner status TRUE	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLane WStatus": 0, "scannerNLane WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 14
POST	http://10.107.92.43:8088/divert-flow/confirmation/Cam14	Confirmation	{ "camId": "string", "trakingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 14
POST	http://10.107.92.43:8088/divert-flow/divert/multiToteCam15	Divert Put to Wall	{ "camId": "string", "trakingId": 0, "toteLpn": "string", "scannerNLane WStatus": 0, "scannerNLane WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 15

POST	http://10.107.92.43:8088/divert-flow/confirmation/Cam15	Confirmation	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 15
POST	http://10.107.92.43:8088/divert-flow/divert/multiToteCam16	Divert SureSort1	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLane WStatus": 0, "scannerNLane WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 16
POST	http://10.107.92.43:8088/divert-flow/confirmation/Cam16	Confirmation	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 16
POST	http://10.107.92.43:8088/divert-flow/divert/multiToteCam17	Divert SureSort2	{ "camId": "string", "trackingId": 0, "toteLpn": "string", "scannerNLane WStatus": 0, "scannerNLane WFull": 0 }	{ "divert_code": 0, "tracking_id": 0 }	BadRequest { message:"string" }	Camera 17
POST	http://10.107.92.43:8088/divert-flow/confirmation/Cam17	Confirmation	{ "camId": "string", "trackingId": 0, "divertCode": 0 }	OK200	BadRequest { message:"string" }	Camera 17
GET	http://10.107.92.43:8088/api/Commissioner/GetCommissionerStatus	Commissioner	N/A	{ "status": false }	Not found: { message:"string" }	N/A
GET	http://10.107.92.43:8088/api/CommissionerPackingLimits	Commissioner picking limits	N/A	[{ "id": 0, "putStationNr":	Internal Server Error: { message:"string"	N/A

				<i>0,</i> <i>"limit": 0</i> <i>},</i> <i>J</i>	<i>"</i> <i>}</i>	
--	--	--	--	---	----------------------	--

Sorter

Cajas Single

Proceso realizado para cajas ingresadas por primera vez

Ubicación: Cam25

URL: <http://hostname:8880/api/DivertBox/Destination>

Método: POST

En el Scanner 25 se tiene la API “**Destination**”, que se encarga de registrar en sistema cliente y asignar una línea de desvío a las cajas de acuerdo a la información obtenida por SAP y por la configuración creada en los mapeos del UI. De la misma manera, se encarga de realizar los desvíos una vez registrada la caja en el sistema.

Registro de cajas

Se verifica que lo escaneado por la cámara no sea “?”, o “111111” (errores de escaneo, “?”, para no lecturas y “111111” para apilado de 2 o más cajas en el escaneo por parte de PLC). En estos casos, la caja se enviará al hospital o si no se encuentra disponible la línea (status OFF o Full) se enviará a recircular.

- Si la caja es escaneada correctamente, se realiza el proceso de registro de caja. Para iniciar, si la caja ya existe en la tabla WCSRouting entonces solo se actualizan los campos Count (aumento del contador de vueltas) y TrackingId del registro. Sin embargo, si la caja no se encuentra registrada, entonces se realiza la búsqueda por BoxId en la base de datos de SAP para encontrar información de la caja.
- Si la información obtenida de SAP se encuentra incompleta, la caja será desviada y registrada para ser enviada a Pallet. Independientemente de si es multi (multicaja es una modalidad donde un pedido está compuesto por más de una caja) o no. Pero si la información es correctamente proporcionada por SAP, se realiza la búsqueda de una línea de desvío disponible mapeada por el usuario desde el UI.
- Si la búsqueda por algún mapeo que coincida con la información obtenida por SAP no es encontrada (que alguna de las características de la caja no está mapeada en ninguna línea, como puede ser el boxType o carrierCode), la caja será desviada y registrada para ser enviada a Pallet.
- Si no se encuentran líneas que estén configuradas para las características de la caja, será enviada a recircular para esperar que alguna línea coincida con su configuración y en WCSRouting será registrada como una caja recirculada.
- En caso de que la caja escaneada no cuente con un ConfirmationNumber entonces se procede a realizar el “RoundRobin”, un sistema para distribuir las

cajas 1:1 entre las distintas líneas de desvío mapeadas con la misma configuración de la caja escaneada.

Teniendo la caja seleccionada después del 1:1, se realizan validaciones para comprobar que sea la línea de desvío correcta y no haya sufrido cambios en el mapeo de la misma. Si no se cuenta con un ContainerId y la línea obtenida de la selección fue alguna de las líneas de Truck entonces se va a registrar la caja como recirculación. De no ser así, entonces la caja se registra normalmente como caja con destino.

Desvío de caja

Después de pasar por el proceso de registro, se llega al de desvío de caja. En esa sección lo primero que se verifica es si se pudo escanear la caja correctamente o se obtuvieron valores del escáner como pueden ser "?", o "111111". De ser el caso, estas cajas son enviadas a la línea de "Hospital".

- Si el escáner obtiene lecturas de la caja, se busca su registro en la tabla WCSRouting a través de su BoxId y el Status "IN" para saber que se encuentra dentro del sistema. Si no existe el registro, se realizará el desvío a la línea de Pallet. Pero si existe, se realizan una serie de comprobaciones antes de obtener la línea a la cual desviarse.

Se actualizan los campos TrackingId y DivertTs (DivertTs se refiere a la marca de tiempo hecha en cada intento de desvío) del registro de la caja, almacenado en la tabla WCSRouting, con el TrackingId obtenido de la consulta a la API Destination y se generará una nueva marca de tiempo para DivertTs.

Para obtener el límite de vueltas que puede dar una caja en Sorter, se consulta de la tabla RecirculationLimits la información del campo CountLimit. Este valor es configurado por el usuario desde el UI.

Comprobando la existencia del registro de la caja en el sistema, se confirma si fue registrada para recircular a través del campo DivertLane, debe tener de valor 999. En la misma validación se verifica si el contador de vueltas de la caja ha excedido el configurado, usando la información obtenida de la tabla RecirculationLimits.

- Si la caja está registrada para recircular y ha superado el límite de vueltas, entonces se realizarán modificaciones en su registro de la tabla WCSRouting para asignarle Pallet como línea de desvío. Pero si no ha superado el límite de vueltas y pertenece a los registros de cajas por recircular, entonces solo se mandará a recircular enviando código de desvío 99.

Antes de que continúe el proceso de desvío de cajas a sus destinos, se valida que la línea a la que está destinada la caja se encuentre aún activa y funcional.

En caso de que no se encuentre un ContainerId (Identificador de los paquetes contenedores situados al final de las líneas de desvío) registrado para la línea de desvío que se asignó a la caja, se eliminará de los registros de la tabla WCSRouting y se enviará a recircular para volver a registrarla e intentar darle otra línea de desvío. Esto se hace debido a que puede que la línea haya sido modificada en el transcurso de registrarla hasta desplazarla a una línea de desvío.

Se tiene una validación para verificar la condición en la línea de desvío. Si la línea se encuentra apagada o llena, se realiza otro proceso para las cajas. Consiste en primero verificar si el valor de recirculación que contiene el registro de la caja se ha superado, y desviar la caja a Pallet.

En caso de que no se encuentre un ContainerId registrado para la línea de desvío que se escogió para la caja, se eliminará de los registros de la tabla WCSRouting y se enviará a recircular para volver a registrarla e intentar darle otra línea de desvío. Esto se hace debido a que puede que la línea haya sido modificada en el transcurso de registrarla hasta desplazarla a una línea de desvío.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	<i>http://35.192.47.202:8880/api/DivertBox/Destination</i>	<i>Destination</i>	<pre>{ "cam_Id": "Cam25", "boxId": "C1006881659", "trackingId": 100 }</pre>	<pre>{ "trackingId": 0, "divertCode" : 0, "boxId": "C1006881659" }</pre>	<pre>{ message:"s tring", "trackingId": 0, "divertCode ": 99 }</pre>	<i>Cam25</i>

Confirmación de cajas a línea de destino

Ubicación: Controlado por clientePLC

URL: <http://localhost:8880/api/DivertBox/Confirmation>

Método: POST

La API "**Confirmation**" se encarga de confirmar las cajas enviadas a desviarse a una línea de destino.

Se verifica el DivertCode que se recibe del JSON y que el modelo JSON se encuentre correctamente creado. Si estas condiciones no llegan a cumplirse, se envía un mensaje de error y se devuelve el valor del DivertCode.

De manera interna, al momento de mandar a llamar la API, se genera un Timestamp que es usado para encontrar el registro de la caja en la tabla WCSRouting en un rango de tiempo de 38 segundos antes de que se manda a llamar la API hasta el momento exacto. Esto se debe a que no se nos envía el valor del BoxId de la caja a confirmar. Y en adición al Timestamp, se usa el valor del TrackingId que se nos proporciona en el JSON. Con estos dos valores es posible encontrar un registro que coincida.

Si no se encuentra el registro en el rango de tiempo, entonces no se realiza ningún proceso.

Pero en caso de encontrarlo se realizan las siguientes comprobaciones:

- Si el resultado tiene como destino la línea de Pallet o línea normal de desvío (Truck o Gaylord), entonces verifica que la línea siga activa y no se haya llenado. Si no se cumplen las condiciones, entonces se cancela el proceso de confirmación y no realiza ninguna acción.
- Al tener activa y sin llenar la línea de desvío, el valor del TrackingId del registro de la caja será cambiado a 0, así como el valor de Status cambiará a "NA" para determinar que la caja se ha confirmado.
- Antes de concluir con la confirmación de la caja, se tiene que identificar su línea, de qué tipo es. Esto con el fin de registrar la caja en la tabla WCS_Routing en la base externa E1_Shipping_WMS, por lo que los tipos de línea deben ser Truck o Gaylord. El registro se realiza a través de querys donde se insertan los valores del registro creado en la tabla WCSRouting.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST
POST	<i>http://35.192.47.202:8880/api/DivertBox/Confirmation</i>	<i>Confirmation</i>	<pre>{ "trackingId": 0, "divertCode": 0 }</pre>	<i>OK200</i>	<pre>{ divertCode:0 }</pre>

Recirculación de caja

Una vez escaneada la caja y obteniendo su registro de la tabla WCSRouting, se comprueba el campo DivertLane para saber si la caja está configurada para recirculación, esto comprobando que el valor del campo DivertLane sea 999. En caso de ser así, entra en una sección de procesos prácticamente iguales del proceso de registro de cajas. La diferencia de esta sección es al momento de la modificación del registro cuando se designa a recirculación. Pues cuando se cumplen las condiciones para establecer el registro de la caja como recirculada, al registro no se le aplica ningún cambio.

Para esta sección, se aplica la misma lógica descrita para registro de cajas por primera vez.

Cambio de estado de las líneas

Ubicación: Controlado por clientePLC

URL: <http://35.192.47.202:8880/api/DivertBox/Destination>

Método: POST

La API “LaneStatus”, se encarga principalmente de actualizar los estatus de las líneas de desvío. De esta acción se desencadenan otras dependiendo si la línea a actualizar se va a apagar o encender.

Lo primero que se encarga de procesar esta API es que el JSON de petición se encuentre bien construido o en su defecto, que la petición contenga información. De no ser así, se envía un mensaje de error BadRequest.

Este proceso obtiene el JSON de petición y lo transforma en datos legibles para el programa, pues como tal no se puede distinguir por el propio JSON todas las líneas de destino y sus atributos. Entonces, para hacer sencillo los pasos, se realiza Regex al JSON para identificar la línea a actualizar.

Una vez identificada la línea, se tiene que identificar qué propiedad se va a actualizar, Status o Full. Para actualizar la propiedad Status, existen dos secciones que tener en cuenta, al encender y al apagar la línea.

Encendido de línea

Para saber cuándo entra al proceso de encender la línea, se tiene que obtener el Status actual de la línea de desvío y compararlo con el valor obtenido del JSON para esa línea. Si coincide que el Status de la línea es falso (la línea está apagada) y el valor del JSON contiene el valor de 1 (la línea debe encender) entonces se debe realizar el proceso de encendido de línea.

El proceso de encendido de línea inicia con la búsqueda del ContainerId de la línea a encender. Si existe, se realiza la impresión de un ticket que contiene la información de la línea (su mapeo y ContainerId). Si no existe, no realiza ningún proceso.

Impresión automática de etiquetas

Al momento de encender una línea que ha sido apagada, el sistema, además de modificar el status de la DivertLane, hace llamado al método para impresión de etiquetas. Donde, por parámetro, se envía la siguiente entidad con los siguientes atributos: el valor del Id del ContainerId nuevo para la línea que se acaba de encender, el valor del Id de la DivertLaneId que se acaba de encender.

Entidad que se envía:

```
{  
  "containerId": 10, //INT (Id del container que está en la DivertLane)  
  "divertLaneId": 23 // INT (Id de la DivertLane que se ha cambiado de status)  
}
```

Con estos datos, se puede realizar la búsqueda correspondiente por medio de los Id's que se recibe del parámetro del método para impresión de la etiqueta, obteniendo los datos de las tablas "Containers", "MappingSorters", "CarrierCodes", "LogisticAgents" y "DivertLane". El Id que se recibe, se tiene que buscar el dato en la tabla correspondiente y devolver la información correcta del Id que se está buscando

Consultas en la base interna:

- En la tabla Containers se busca por medio del Id el container del cual se imprimirán los datos.
- En la tabla MappingSorters busca la información por medio del DivertLaneId, verificando que exista en la tabla.
- Después en la tabla de CarrierCodes y LogisticAgents busca los datos necesarios para imprimir el CarrierCodes y el LogisticAgent que la DivertLane tenga asignados en la tabla MappingSorters.
- Al hacer todas las validaciones, busca la configuración de la impresora, la cual la DivertLane tenga mapeada.

Evaluación de datos para impresión:

- Al obtener los datos necesarios para la impresión, y también a qué impresora está configurada para realizarlo, el método mandará a imprimir una etiqueta con los siguientes datos:
 1. ContainerId
 2. Fecha y hora de impresión
 3. CarrierCode
 4. Logistic Agent
 5. DivertLane.

Apagado de línea

Para saber cuándo entra al proceso de apagado de línea, se tiene que obtener el Status actual de la línea de desvío y compararlo con el valor obtenido del JSON para esa línea. Si coincide que el Status de la línea es verdadero (la línea está encendida) y el valor del JSON contiene el valor de 0 (la línea debe apagarse) entonces se debe realizar el proceso de apagado de línea.

El proceso de apagado de línea inicia con la búsqueda del ContainerId de la línea a apagar. Si existe, se realiza el recuento de cajas que se desviaron correctamente a la línea a cerrar, tanto en la tabla externa WCS_Routing como en la interna WCSRouting con algunas diferencias.

Para el recuento que se registra en la tabla externa se tienen que buscar todas las cajas pertenecientes a esa línea de desvío a través del ContainerId en la tabla WCSRouting, tanto las cajas registradas a Pallet como las cajas con destino que no alcanzaron a desviarse. Las cajas escaneadas deben ser registradas en la tabla WCS_Routing aun si no se desviaron correctamente a la línea. Por lo que, si alguna de las cajas registradas en WCSRouting no se encuentra en la tabla externa WCS_Routing, se agrega al momento de realizar el recuento de cajas.

El recuento de cajas es un registro ingresado en las tablas que tiene como función informar sobre cuantas cajas fueron desviadas correctamente a la línea que está por cerrar. Este registro cuenta con el campo Qty que será llenado por la cantidad de cajas, el campo ContainerId y DivertLane para identificar la línea de desvío que fue cerrada.

Por último, para las líneas de desvío pertenecientes a Truck se les elimina el ContainerId al cierre, pero para las líneas pertenecientes a Gaylord se realiza el proceso de eliminar el ContainerId actual y generar un nuevo ContainerId.

En caso de que no exista un ContainerId registrado para la línea de desvío que está por cerrarse y esa línea sea una perteneciente a Gaylord, se le genera y asigna un nuevo ContainerId. Y si la línea pertenece a Truck entonces solo se le elimina el ContainerId si es que al momento de cierre tiene asignado algún ContainerId.

Cambio de estado Full

Para saber cuándo una línea se llena, se tiene que consultar la propiedad Full del JSON de entrada que pertenezca a la línea de desvío de ese momento. Este proceso es sencillo, pues solo actualiza el valor actual del registro por el que obtenga del JSON.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST
--------	-----	------	--------------	---------------	---------------

POST	<i>http://35.192.47.202:8880/api/LaneStatus</i>	<i>LaneStatus</i>	<pre> { "lane_1_status": 1, "lane_1_full": 0, "lane_2_status": 1, "lane_2_full": 0, "lane_3_status": 1, "lane_3_full": 0, "lane_4_status": 1, "lane_4_full": 0, "lane_5_status": 1, "lane_5_full": 0, "lane_6_status": 1, "lane_6_full": 0, "lane_7_status": 1, "lane_7_full": </pre>	OK200	<pre> { message:"string" } </pre>
------	---	-------------------	---	-------	-------------------------------------

			: 0, "lane_8_stat us": 1, "lane_8_full" : 0, "lane_9_stat us": 1, "lane_9_full" : 0, "lane_10_st atus": 1, "lane_10_ful l": 0, "lane_11_st atus": 1, "lane_11_ful l": 0, "lane_12_st atus": 1, "lane_12_ful l": 0, "lane_13_st atus": 1, "lane_13_ful l": 0, "lane_14_st atus": 1, "lane_14_ful l": 0,		
--	--	--	--	--	--

			<div>"lane_15_st atus": 1, "lane_15_ful l": 0, "lane_16_st atus": 1, "lane_16_ful l": 0, "lane_17_st atus": 1, "lane_17_ful l": 0, "lane_18_st atus": 1, "lane_18_ful l": 0, "lane_19_st atus": 1, "lane_19_ful l": 0, "lane_20_st atus": 1, "lane_20_ful l": 0, "lane_21_st atus": 1, "lane_21_ful l": 0,</div>		
--	--	--	---	--	--

			<div>"lane_22_st atus": 1, "lane_22_ful l": 0, "lane_23_st atus": 1, "lane_23_ful l": 0, "lane_24_st atus": 1, "lane_24_ful l": 0, "lane_25_st atus": 1, "lane_25_ful l": 0, "lane_26_st atus": 1, "lane_26_ful l": 0, "lane_27_st atus": 1, "lane_27_ful l": 0, "lane_28_st atus": 1, "lane_28_ful l": 0, "lane_29_st</div>	
--	--	--	--	--

			<pre> atus": 1, "lane_29_ful l": 0, "lane_30_st atus": 1, "lane_30_ful l": 0, "lane_31_st atus": 1, "lane_31_ful l": 0, "lane_32_st atus": 1, "lane_32_ful l": 0, } </pre>		
--	--	--	---	--	--

Caja multibox

Proceso realizado para cajas ingresadas por primera vez

En el Scanner 25 se tiene la API "**Destination**", que se encarga de registrar en nuestro sistema y asignar una línea de desvío a las cajas de acuerdo a la información obtenida por SAP y por la configuración creada en los mapeos del UI. De la misma manera, se encarga de realizar los desvíos una vez registrada la caja en el sistema.

Ubicación: Cam25

URL: <http://35.192.47.202:8880/api/DivertBox/Destination>

Método:POST

Registro de cajas

Se verifica que lo escaneado por la cámara no sea "?" o "111111" (errores de escaneo, "?", para no lecturas y "111111" cuando se escanean juntas dos o más cajas en el

escáner por parte de PLC). En estos casos, la caja se enviará al hospital o si no se encuentra disponible la línea (status OFF o Full) se enviará a recircular.

Si la caja es escaneada correctamente, se realiza el proceso de registro de caja. Para iniciar, si la caja ya existe en la tabla WCSRouting entonces solo se actualizan los campos Count (aumento del contador de vueltas) y TrackingId del registro. Sin embargo, si la caja no se encuentra registrada, entonces se realiza la búsqueda por BoxId en la base de datos de SAP para encontrar información de la caja.

Si la información obtenida de SAP se encuentra incompleta, la caja será desviada y registrada para ser enviada a Pallet. Independientemente de si es multi o no. Pero si la información es correctamente proporcionada por SAP, se realiza la búsqueda de una línea de desvío disponible mapeada por el usuario desde el UI.

Si la búsqueda por algún mapeo que coincida con la información obtenida por SAP no es encontrada (que alguna de las características de la caja no está mapeada en ninguna línea, como puede ser el boxType o carrierCode), la caja será desviada y registrada para ser enviada a Pallet.

Si no se encuentran líneas que estén configuradas para las características de la caja, será enviada a recircular para esperar que alguna línea coincida con su configuración y en WCSRouting será registrada como una caja recirculada.

En caso de que la caja contenga ConfirmationNumber y no se encuentren líneas configuradas para las características de la caja, se realiza la búsqueda del registro de la ola en la tabla multiBox_Wave a través del campo ConfirmationNumber. Si la ola se encuentra registrada y el Status de la misma es "IN", la caja será desviada y registrada para ser enviada a Pallet. Si la ola se encuentra registrada, pero el Status es nulo, se verificará que la primera línea que se encontró anteriormente mapeada para la configuración de la caja sea perteneciente a las líneas de Truck. Si no lo es, la caja será desviada y registrada para ser enviada a Pallet. Pero si lo es, entonces el contador de la ola será aumentado (esto será así hasta que la ola se complete).

Si la caja de la que se obtuvo información de SAP tiene ConfirmationNumber, se verifica la configuración de límite de cantidad de cajas de una ola (campo MaxBoxCount de tabla ConfigurationsSetting en base de datos E1_Shipping_WMS). Si la cantidad de cajas de una ola supera la cantidad configurada, esto lo verificamos comparando el campo Qty del registro de la caja en la tabla WCSRouting con el campo MaxBoxCount de la tabla ConfigurationsSettings, entonces la caja será desviada y registrada para ser enviada a Pallet. En caso de no superarlo, se obtiene la primera línea de desvío de la lista anteriormente mencionada para coincidir con la configuración de la caja. Si esta línea obtenida no es perteneciente a las líneas de Truck (2 o 4) entonces se buscará el registro de la ola en la tabla multiBox_Wave para saber si se encuentra registrada. En caso de estarlo y a su vez tener Status en "IN" o

en nulo, se enviará la caja a pallet. Si no se cumple el criterio anterior, entonces la caja será registrada para recircular.

Después de pasar estos filtros, si la caja cuenta con línea a la cual desviarse y esta se encuentra mapeada correctamente, entonces se procederá a registrar la caja en el contador de olas de la tabla multiBox_Wave. Si la ola no se encuentra registrada, se registra y a su vez se posiciona el contador en 1 para señalar que se ha registrado una caja en el sistema. Pero en caso de que la ola ya se encuentre registrada y el Status esté en nulo, entonces se realiza la comprobación para confirmar si la línea de desvío asignada a la ola no haya sido apagada y encendida con un nuevo ContainerId configurado, o si la cantidad de cajas (campo QtyCount de la tabla multiBox_Wave) se ha sobrepasado del configurado para la ola (campo Qty de la tabla multiBox_Wave). En caso de que se cumplan estas condiciones, se registrará la caja en el contador de olas y registrada para desviar en Pallet para su posterior confirmación manual.

En caso de que se cuente con la línea de desvíos sin cambios desde el inicio del registro de la ola, entonces se procederá a registrar la caja en el contador de olas de la tabla multiBox_Wave en el campo QtyCount.

Si en lo anterior, se llega a completar la ola y cambiar el Status a “IN” entonces se obtienen todas las cajas pertenecientes que se encuentren recirculando para cambiar su registro de la tabla WCSRouting de recirculadas a cajas con destino. En caso de aún no haber completado la ola, se registrará la caja para recircular.

En caso de que la caja escaneada no cuente con un ConfirmationNumber entonces se procede a realizar el “RoundRobin”, un sistema para distribuir las cajas 1:1 entre las distintas líneas de desvío mapeadas con la misma configuración de la caja escaneada.

Teniendo la caja seleccionada después del 1:1, se realizan validaciones para comprobar que sea la línea de desvío correcta y no haya sufrido cambios en el mapeo de la misma. Si no se cuenta con un ContainerId y la línea obtenida de la selección fue alguna de las líneas de Truck entonces se va a registrar la caja como recirculación. De no ser así, entonces la caja se registra normalmente como caja con destino.

Desvío de caja

Después de pasar por el proceso de registro, se llega al de desvío de caja. En esa sección lo primero que se verifica es si se pudo escanear la caja correctamente o se obtuvieron valores del escáner como pueden ser “?” o “111111”. De ser el caso, estas cajas son enviadas a la línea de Hospital.

Si el escáner obtiene lecturas de la caja, se busca su registro en la tabla WCSRouting a través de su BoxId y el Status “IN” para saber que se encuentra dentro del sistema.

Si no existe el registro, se realizará el desvío a la línea de Pallet. Pero si existe, se realizan las siguientes comprobaciones antes de obtener la línea a la cual desviarse:

Se actualizan los campos TrackingId y DivertTs (DivertTs se refiere a la marca de tiempo hecha en cada intento de desvío) del registro de la caja, almacenado en la tabla WCSRouting, con el TrackingId obtenido de la consulta a la API Destination y se generará una nueva marca de tiempo para DivertTs.

Para obtener el límite de vueltas que puede dar una caja en Sorter, se consulta de la tabla RecirculationLimits la información del campo CountLimit. Este valor es configurado por el usuario desde el UI.

Si existe un registro de la caja en el sistema (registro de caja en tabla WCSRouting), se confirma si fue registrada para recircular a través del campo DivertLane, debe tener de valor 999. En la misma validación se verifica si el contador de vueltas de la caja ha excedido el configurado, usando la información obtenida de la tabla RecirculationLimits.

Si la caja está registrada para recircular y ha superado el límite de vueltas, entonces se realizarán modificaciones en su registro de la tabla WCSRouting para asignarle Pallet como línea de desvío. Pero si no ha superado el límite de vueltas y pertenece a los registros de cajas por recircular, entonces solo se mandará a recircular enviando código de desvío 99.

Para continuar con el flujo de desvío de cajas a sus destinos, se valida que la línea a la que está destinada la caja se encuentre aún activa y funcional. Si esta línea de desvío está en condiciones de aceptar cajas (se encuentra encendida y no llena), entonces se chequea si la línea pertenece a Truck y la caja contiene ConfirmationNumber en su registro de la tabla WCSRouting. Si se cumple, se valida si se encuentra registrada la ola en la tabla multiBox_Wave, la ola cuente con Status "IN" y la línea de desvío configurada para la ola no sea la misma que la línea a la cual se va a desviar la caja, para enviarla entonces a Pallet cambiando sus valores registrados en WCSRoutings.

En caso de que no se encuentre un ContainerId registrado para la línea de desvío que se escogió para la caja, se eliminará de los registros de la tabla WCSRoutings y se enviará a recircular para volver a registrar y asignar a la caja una nueva línea de desvío. Esto se hace debido a que puede que la línea haya sido modificada durante el transcurso de registrarla hasta desplazarla a una línea de desvío, es decir, que la línea pueda sufrir cambios como asignación de un nuevo ContainerId, apagado repentino de la línea o estar llena; situaciones que pueden suceder desde el momento en que se registra la caja en el sistema hasta.

Como se mencionó antes, hay una validación para verificar la condición de la línea de desvío. Si la línea se encuentra apagada o llena, se realiza otro proceso para las

cajas. Consiste en primero verificar si el valor de recirculación que contiene el registro de la caja se ha superado, y desviar la caja a Pallet.

Si lo anterior no se cumple, se realiza la misma verificación para saber si la línea de desvío pertenece a Truck y si la caja contiene ConfirmationNumber en su registro de la tabla WCSRouting. Si se cumple, se valida si se encuentra registrada la ola en la tabla multiBox_Wave, la ola cuente con Status "IN" y la línea de desvío configurada para la ola no sea la misma que la línea a la cual se va a desviar la caja, para enviarla entonces a Pallet cambiando sus valores registrados en WCSRouting.

En caso de que no se encuentre un ContainerId registrado para la línea de desvío que se escogió para la caja, se eliminará de los registros de la tabla WCSRouting y se enviará a recircular para volver a registrarla e intentar darle otra línea de desvío. Esto se hace debido a que puede que la línea haya sido modificada en el transcurso de registrarla hasta desplazarla a una línea de desvío.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST	CAMERAS ASSIGNMENT
POST	<i>http://35.192.47.202:8880/api/DivertBox/Destination</i>	<i>Destination</i>	<i>{ "cam_Id": "Cam25", "boxId": "C1006881659", "trackingId": 100 }</i>	<i>{ "trackingId": 0, "divertCode": 0, "boxId": "C1006881659" }</i>	<i>{ message:" string", "trackingId": 0, "divertCode": 99 }</i>	<i>Cam25</i>

Confirmación de cajas multis a línea de destino

La API "**Confirmation**" se encarga de confirmar las cajas enviadas a desviarse a una línea de destino.

Ubicación: Controlado por PLC

URL: <http://35.192.47.202:8880/api/DivertBox/Confirmation>

Método: POST

Se verifica que el campo DivertCode que se recibe del JSON y que el modelo en general se encuentre correctamente creado. Si estas condiciones no llegan a cumplirse, se envía un mensaje de error y se devuelve el valor del DivertCode.

De manera interna, al momento de mandar a llamar la API, se genera un Timestamp que es usado para encontrar el registro de la caja en la tabla WCSRouting en un rango de tiempo de 38 segundos antes de que se manda a llamar la API hasta el momento exacto. Esto se debe a que no se nos envía el valor del BoxId de la caja a confirmar. Además del valor del campo Timestamp, se usa el valor del campo TrackingId que se proporciona en el JSON. Con estos dos valores es posible encontrar un registro que coincida.

Si no se encuentra el registro en el rango de tiempo, entonces no se realiza ningún proceso. Pero en caso de encontrarlo se realizan las siguientes comprobaciones:

Si el resultado tiene como destino una línea de Truck, se tiene que verificar que el registro cuente con ConfirmationNumber para poder hacer la búsqueda de la ola en la tabla multiBox_Wave. Si el registro de la ola existe en dicha tabla y el campo Status de la ola es nulo o con valor "NA", entonces termina el proceso y no realiza ninguna acción. Pero si el Status se encuentra en "IN", entonces se aumenta el contador de cajas del registro de la ola. En caso de que la ola alcance a completarse, el Status será cambiado a "NA" en el registro de la ola.

Si el resultado tiene como destino la línea de Pallet o línea normal de desvío (Truck o Gaylord), entonces verifica que la línea siga activa y no se haya llenado. Si no se cumplen las condiciones, entonces se cancela el proceso de confirmación y no realiza ninguna acción.

Al tener activa y sin llenar la línea de desvío, el valor del TrackingId del registro de la caja será cambiado a 0, así como el valor de Status cambiará a "NA" para determinar que la caja se ha confirmado.

Antes de concluir con la confirmación de la caja, se tiene que identificar su línea de qué tipo es. Esto con el fin de registrar la caja en la tabla WCS_Routing en la base externa E1_Shipping_WMS, por lo que los tipos de línea deben ser Truck o Gaylord. El registro se realiza a través de queries donde se insertan los valores del registro creado en la tabla WCSRouting.

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST
--------	-----	------	--------------	---------------	---------------

POST	http://35.192.47.202:8880/api/Diver tBox/Confirmation	Confirmation	{ "trackingId" : 0, "divertCod e": 0 }	OK200	{ divertCode:0 }
------	--	--------------	---	-------	------------------------

Recirculación de caja

Una vez escaneada la caja y obteniendo su registro de la tabla WCSRouting, se comprueba el campo DivertLane para saber si la caja está configurada para recirculación. En caso de ser así, entra en una sección de procesos prácticamente copias del registro de cajas. La diferencia de esta sección es al momento de la modificación del registro cuando se designa a recirculación. Pues cuando se cumplen las condiciones para establecer el registro de la caja como recirculada, al registro no se le aplica ningún cambio.

Para este proceso se aplica la misma lógica descrita para registro de cajas por primera vez.

Cambio de estado de las líneas

La API "**LaneStatus**" se encarga principalmente de actualizar los estados de las líneas de desvío. De esta acción se desencadenan otras dependiendo si la línea a actualizar se va a apagar o encender.

Ubicación: Controlado por PLC

URL: <http://35.192.47.202:8880/api/DiverLanes/LaneStatus>

Método: POST

Lo primero que se encarga de procesar esta API es que el JSON de petición se encuentre bien construido o en su defecto, que la petición contenga información. De no ser así, se envía un mensaje de error BadRequest.

Este proceso obtiene el JSON de petición y lo transforma en datos legibles para el programa, pues como tal no se puede distinguir por el propio JSON todas las líneas de destino y sus atributos. Entonces, para hacer sencillo los pasos se realiza Regex al JSON para identificar la línea a actualizar.

Después de identificar la línea, se tiene que identificar qué propiedad se va a actualizar, Status o Full. Para actualizar la propiedad Status, existen dos secciones que tener en cuenta, al encender y al apagar la línea.

Encendido de línea

Para saber cuándo entra al proceso de encender línea se tiene que obtener el Status actual de la línea de desvío y compararlo con el valor obtenido del JSON para esa línea. Si coincide que el Status de la línea es falso (la línea está apagada) y el valor del JSON contiene el valor de 1 (la línea debe encender) entonces se debe realizar el proceso de encendido de línea.

El proceso de encendido de línea inicia con la búsqueda del ContainerId de la línea a encender. Si existe, se realiza la impresión de un ticket que contiene la información de la línea (su mapeo y ContainerId). Si no existe, no realiza ningún proceso.

Impresión automática de etiquetas

Al momento de encender una línea que ha sido apagada, el sistema, además de modificar el status de la DivertLane, hace llamado al método para impresión de etiquetas. Donde, por parámetro, se envía la siguiente entidad con los siguientes atributos: el valor del Id del ContainerId nuevo para la línea que se acaba de encender, el valor del Id de la DivertLaneId que se acaba de encender.

Entidad que se envía:

```
{
  "containerId": 10, //INT (Id del container que está en la DivertLane)
  "divertLaneId": 23 // INT (Id de la DivertLane que se ha cambiado de status)
}
```

Con estos datos, se puede realizar la búsqueda correspondiente por medio de los Id's que se recibe del parámetro del método para impresión de la etiqueta, obteniendo los datos de las tablas "Containers", "MappingSorters", "CarrierCodes", "LogisticAgents" y "DivertLane". El Id que se recibe, se tiene que buscar el dato en la tabla correspondiente y devolver la información correcta del Id que se está buscando

Consultas en la base interna:

- En la tabla Containers se busca por medio del Id el container del cual se imprimirán los datos.
- En la tabla MappingSorters busca la información por medio del DivertLaneId, verificando que exista en la tabla.
- Después en la tabla de CarrierCodes y LogisticAgents busca los datos necesarios para imprimir el CarrierCodes y el LogisticAgent que la DivertLane tenga asignados en la tabla MappingSorters.

- Al hacer todas las validaciones, busca la configuración de la impresora, la cual la DivertLane tenga mapeada.
-

Evaluación de datos para impresión:

- Al obtener los datos necesarios para la impresión, y también a que impresora está configurada para realizarlo, el método mandará a imprimir una etiqueta con los siguientes datos:
 6. ContainerId
 7. Fecha y hora de impresión
 8. CarrierCode
 9. Logistic Agent
 10. DivertLane.

Apagado de línea

Para saber cuándo entra al proceso de apagado de línea, se tiene que obtener el Status actual de la línea de desvío y compararlo con el valor obtenido del JSON para esa línea. Si coincide que el Status de la línea es verdadero (la línea está encendida) y el valor del JSON contiene el valor de 0 (la línea debe apagarse) entonces se debe realizar el proceso de apagado de línea.

El proceso de apagado de línea, comienza con la búsqueda del ContainerId de la línea a apagar. Si existe se eliminan las olas incompletas de la tabla multiBox_Waves, es decir, las olas que no alcanzaron a crearse y tenían como destino la línea a apagar. Al eliminar estas olas, también se tiene que eliminar los registros de las cajas que pertenecen a estas olas. El criterio para eliminarlas, es que estas cajas registradas en la tabla WCSRouting sean recirculadas o con destino (Status "IN", ContainerType "S" o "T" y DivertLane con valor 999 o con el valor de la línea que está por cerrarse). Estas eliminaciones solo están estrictamente limitadas a las líneas de desvíos 2 o 4.

Se realiza el recuento de cajas que se desviaron correctamente a la línea a cerrar, tanto en la tabla externa WCS_Routing como en la interna WCSRouting con algunas diferencias. Para el recuento que se registra en la tabla externa se tienen que buscar todas las cajas pertenecientes a esa línea de desvío a través del ContainerId en la tabla WCSRouting, tanto las cajas registradas a Pallet como las cajas con destino que no alcanzaron a desviarse. Todas las cajas deben de registrarse en la tabla WCS_Routing aun si no se desviaron correctamente a la línea. Por lo que, si alguna de las cajas registradas en WCSRouting no se encuentra en la tabla externa WCS_Routing, se agrega al momento de realizar el recuento de cajas.

El recuento de cajas es un registro ingresado en las tablas que tiene como función informar sobre cuantas cajas fueron desviadas correctamente a la línea que está por cerrar. Este registro cuenta con el campo Qty que será llenado por la cantidad de cajas, el campo ContainerId y DivertLane para identificar la línea de desvío que fue cerrada.

Se realiza también la impresión de las cajas pertenecientes a olas que no se desviaron completamente en la línea que se va a cerrar. Esto para informar a través de los BoxId qué cajas se desviaron correctamente y que cajas faltaron por desviar a su destino.

Por último, para las líneas de desvío pertenecientes a Truck se les elimina el ContainerId al cierre, pero para las líneas pertenecientes a Gaylord se realiza el proceso de eliminar el ContainerId actual y generar un nuevo ContainerId.

En caso de que no exista un ContainerId registrado para la línea de desvío que está por cerrarse y esa línea sea una perteneciente a Gaylord, se le genera y asigna un nuevo ContainerId. Y si la línea pertenece a Truck entonces solo se le elimina el ContainerId si es que al momento de cierre tiene asignado algún ContainerId.

Cambio de estado Full

Para saber cuándo una línea se llena, se tiene que consultar la propiedad Full del JSON de entrada que pertenezca a la línea de desvío de ese momento. Este proceso es sencillo, pues solo actualiza el valor actual del registro por el que obtenga del JSON.

Impresión de etiquetas automáticas para cierre de líneas inesperadas

A través de la API “**LaneStatus**” se obtendrán los estatus de las líneas de desvío. De esta acción se desencadenan otras dependiendo si la línea a actualizar se va a apagar o encender.

Ubicación: Controlado por PLC

URL: <http://35.192.47.202:8880/api/DivertLanes/LaneStatus>

Método: POST

Al momento de un cierre inesperado de una línea para truck, el sistema, además de modificar el status de la DivertLane, hace llamado al método para impresión de etiquetas para multiboxes. Donde, por parámetro, se envía la siguiente entidad con los siguientes atributos: el valor del Id del ContainerId de la línea que se acaba de apagar, el valor del Id de la DivertLaneId que se acaba de apagar.

```
{  
  "containerId": 10, //INT (Id del container que está en la DivertLane)  
  "divertLaneId": 23 // INT (Id de la DivertLane que se ha cambiado de status)  
}
```

Con estos datos, se puede realizar la búsqueda correspondiente por medio de los Id's que se recibe del parámetro del método para impresión de la etiqueta para multiboxes,

obteniendo los datos (ConfirmationNumber, BoxesNotDeliveredDueToClosure, BoxesProperlyDelivered).

Consultas en la base interna:

- En la tabla Containers se busca por medio del Id el container del cual se imprimirán los datos.
- En la tabla DivertLanes se busca la información por medio del DivertLaneId, verificando que exista en la tabla.
- En la tabla de multiBox_Wave, se busca la información de la divertlane, containerId, ConfirmationNumber de la ola, status diferente de null y QtyCount mayor o igual a 1
- Al hacer todas las validaciones, busca la configuración de la impresora, la cual la DivertLane tenga mapeada.
- En la tabla wCSRouting, se selecciona todas las cajas que estén desviadas y pertenecen a la ola que se está buscando, es decir ConfirmationNumber, Divertlane y ContainerId correspondiente a TruckLane

Consultas en la base de datos externa SAP-db:

- En la tabla SAP_Orders, se busca la información de ConfirmationNumber al que pertenece la ola.

Evaluación de datos para impresión:

- Se obtiene la lista de cajas que se han desviado a la línea de TruckLane y se compara con todas las cajas que pertenecen a la misma ola, separado las cajas que sí han desviado a TruckLane y las que aún faltan por desviar después del cierre de línea inesperadamente.
- Una vez separadas las listas, se agregan los datos de las cajas(BoxId) a las secciones correspondientes, es decir; BoxesNotDeliveredDueToClosure: Tendrá la lista de las cajas que no se ha desviado a la línea de TruckLane después del cierre inesperado. BoxesProperlyDelivered: Tendrá la lista de las cajas que se desviaron a la línea de TruckLane que corresponde antes del cierre inesperado.
- Al obtener los datos necesarios para la impresión, y también a qué impresora está configurada para realizarlo, el método mandará a imprimir una etiqueta con los siguientes datos:
 - 11. ConfirmationNumber
 - 12. BoxesNotDeliveredDueToClosure
 - 13. BoxesProperlyDelivered

METHOD	URL	NAME	JSON REQUEST	JSON	ERROR
--------	-----	------	--------------	------	-------

				RESPONSE	REQUEST
POST	http://35.192.47.202:8880/api/DivertLanes/LaneStatus	LaneStatus	<pre>{ "lane_1_status": 1, "lane_1_full": 0, "lane_2_status": 1, "lane_2_full": 0, "lane_3_status": 1, "lane_3_full": 0, "lane_4_status": 1, "lane_4_full": 0, "lane_5_status": 1, "lane_5_full": 0, "lane_6_status": 1, "lane_6_full": 0, "lane_7_status": 1, "lane_7_full": 0, "lane_8_status": 1, "lane_8_full": 0 }</pre>	OK200	<pre>{ message:"string" }</pre>

			0, "lane_9_stat us": 1, "lane_9_full": 0, "lane_10_sta tus": 1, "lane_10_full ": 0, "lane_11_sta tus": 1, "lane_11_full ": 0, "lane_12_sta tus": 1, "lane_12_full ": 0, "lane_13_sta tus": 1, "lane_13_full ": 0, "lane_14_sta tus": 1, "lane_14_full ": 0, "lane_15_sta tus": 1, "lane_15_full ": 0,		
--	--	--	---	--	--

			<div>"lane_16_sta tus": 1,</div> <div>"lane_16_full ": 0,</div> <div>"lane_17_sta tus": 1,</div> <div>"lane_17_full ": 0,</div> <div>"lane_18_sta tus": 1,</div> <div>"lane_18_full ": 0,</div> <div>"lane_19_sta tus": 1,</div> <div>"lane_19_full ": 0,</div> <div>"lane_20_sta tus": 1,</div> <div>"lane_20_full ": 0,</div> <div>"lane_21_sta tus": 1,</div> <div>"lane_21_full ": 0,</div> <div>"lane_22_sta tus": 1,</div> <div>"lane_22_full ": 0,</div> <div>"lane_23_sta</div>		
--	--	--	---	--	--

			tus": 1, "lane_23_full": 0, "lane_24_status": 1, "lane_24_full": 0, "lane_25_status": 1, "lane_25_full": 0, "lane_26_status": 1, "lane_26_full": 0, "lane_27_status": 1, "lane_27_full": 0, "lane_28_status": 1, "lane_28_full": 0, "lane_29_status": 1, "lane_29_full": 0, "lane_30_status": 1,		
--	--	--	--	--	--

			"lane_30_full ": 0, "lane_31_sta tus": 1, "lane_31_full ": 0, "lane_32_sta tus": 1, "lane_32_full ": 0, }		
--	--	--	---	--	--

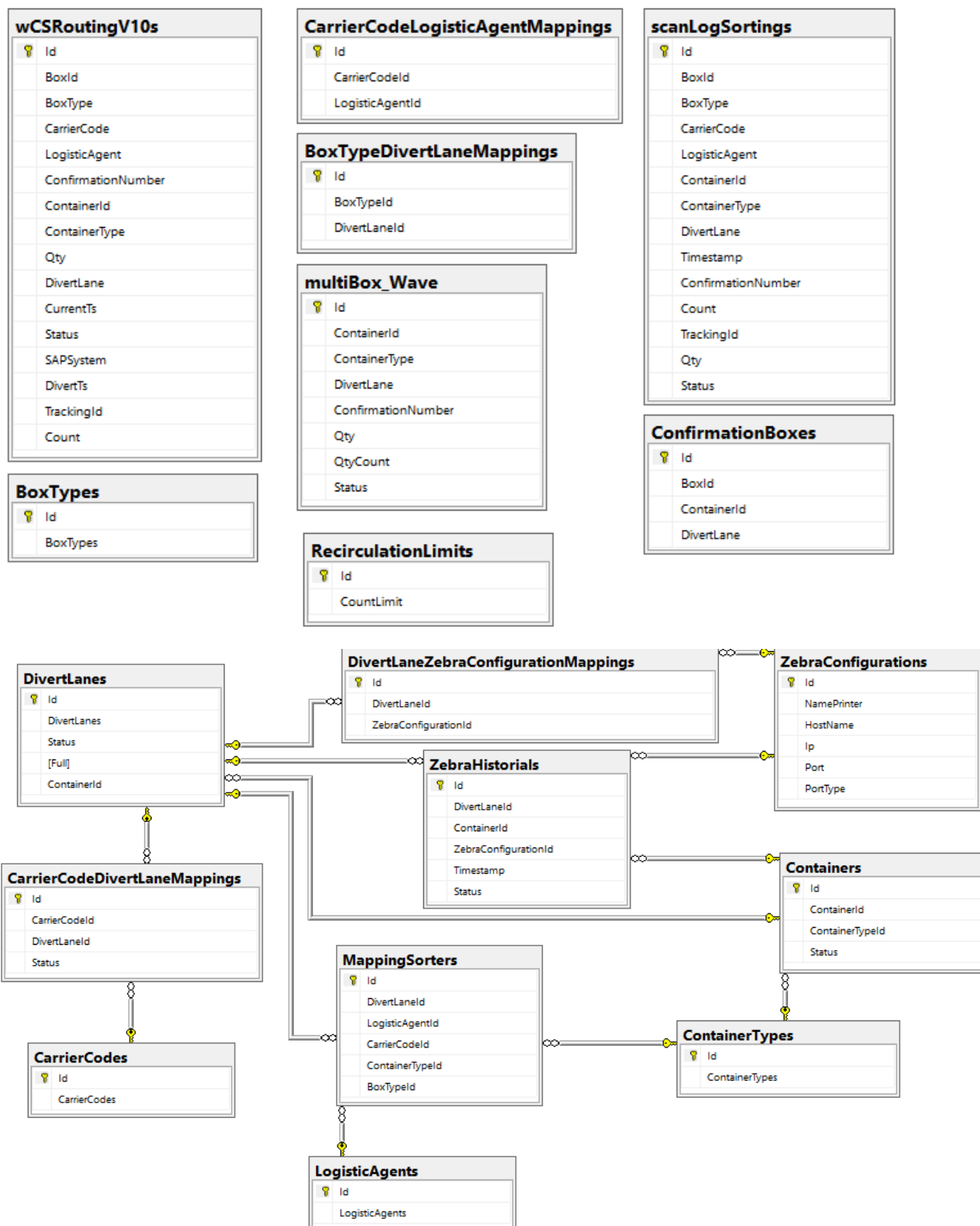
Tablas utilizadas para las APIs de Sorter

cliente-db		
CarrierCodes	BoxTypes	ContainerTypes
<ul style="list-style-type: none"> Id (PK, int, not null) CarrierCodes (nvarchar(max), not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) BoxTypes (nvarchar(max), not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) ContainerTypes (nvarchar(max), not null)
DivertLanes	LogisticAgents	CarrierCodeDivertLaneMappings
<ul style="list-style-type: none"> Id (PK, int, not null) DivertLanes (int, not null) Status (bit, not null) Full (bit, not null) ContainerId (FK, int, null) 	<ul style="list-style-type: none"> Id (PK, int, not null) LogisticAgents (nvarchar(max), not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) CarrierCodeId (FK, int, not null) DivertLaneId (FK, int, not null) Status (bit, not null)
BoxTypeDivertLaneMappings	CarrierCodeLogisticAgentMappings	Containers
<ul style="list-style-type: none"> Id (PK, int, not null) BoxTypeId (int, not null) DivertLaneId (int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) CarrierCodeId (int, not null) LogisticAgentId (int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) ContainerId (nvarchar(max), not null) ContainerTypeId (FK, int, not null) Status (bit, null)
MappingSorters	ZebraHistorials	ZebraConfigurations
<ul style="list-style-type: none"> Id (PK, int, not null) DivertLaneId (FK, int, not null) LogisticAgentId (FK, int, null) CarrierCodeId (nvarchar(max), null) ContainerTypeId (FK, int, null) BoxTypeId (nvarchar(max), null) 	<ul style="list-style-type: none"> Id (PK, int, not null) DivertLaneId (FK, int, not null) ContainerId (FK, int, not null) ZebraConfigurationId (FK, int, not null) Timestamp (nvarchar(20), not null) Status (bit, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) NamePrinter (nvarchar(35), not null) HostName (nvarchar(75), null) Ip (nvarchar(15), null) Port (int, not null) PortType (nvarchar(3), not null)

RecirculationLimits	DivertLaneZebraConfigurationMappings	ConfirmationBoxes
<ul style="list-style-type: none"> Id (PK, int, not null) CountLimit (int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) DivertLaneId (FK, int, not null) ZebraConfigurationId (FK, int, not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) BoxId (int, not null) ContainerId (nvarchar(max), not null) DivertLane (int, not null)
scanLogSortings	multiBox_Wave	wCSRoutingclientes
<ul style="list-style-type: none"> Id (PK, int, not null) BoxId (nvarchar(max), null) BoxType (nvarchar(max), null) CarrierCode (nvarchar(max), null) LogisticAgent (nvarchar(max), null) ContainerId (nvarchar(max), null) ContainerType (nvarchar(max), null) DivertLane (int, null) Timestamp (nvarchar(max), null) ConfirmationNumber (nvarchar(max), null) Count (int, null) TrackingId (int, null) Qty (int, null) Status (nvarchar(max), null) 	<ul style="list-style-type: none"> Id (PK, int, not null) ContainerId (nvarchar(20), not null) ContainerType (nvarchar(1), not null) DivertLane (int, not null) ConfirmationNumber (nvarchar(20), not null) Qty (int, not null) QtyCount (int, null) Status (nvarchar(2), null) 	<ul style="list-style-type: none"> Id (PK, int, not null) BoxId (nvarchar(20), null) BoxType (nvarchar(18), null) CarrierCode (nvarchar(10), null) LogisticAgent (nvarchar(4), null) ConfirmationNumber (nvarchar(20), null) ContainerId (nvarchar(20), not null) ContainerType (nvarchar(1), not null) Qty (int, not null) DivertLane (int, not null) CurrentTs (nvarchar(20), not null) Status (nvarchar(2), not null) SAPSystem (nvarchar(4), not null) DivertTs (nvarchar(max), null) TrackingId (int, null) Count (int, not null)

SAP-db		
WCS_Routing	SAP_Orders	ConfigurationsSetting
<ul style="list-style-type: none"> Id (PK, int, not null) BoxId (char(20), null) BoxType (char(18), null) CarrierCode (char(10), null) LogisticAgent (char(4), null) ConfirmationNumber (char(20), null) ContainerId (char(20), not null) ContainerType (char(1), not null) Qty (numeric(6,0), null) DivertLane (numeric(4,0), not null) CurrentTs (nchar(20), not null) Status (char(2), not null) SAPSystem (char(4), not null) 	<ul style="list-style-type: none"> Id (int, not null) BoxId (char(20), not null) BoxType (char(18), not null) CarrierCode (char(10), not null) LogisticAgent (char(4), null) ConfirmationNumber (char(20), null) Qty (numeric(6,0), null) CurrentTs (char(20), not null) Status (char(2), not null) SAPSystem (char(4), not null) 	<ul style="list-style-type: none"> Id (PK, int, not null) MaxBoxCount (int, null) MaxRecirculationCount (int, null) RetentionPeriod (int, null) EnableArchive (int, null) ArchiveTime (datetime, null) Archive_retention (int, null)

Diagrama Entidad-Relación cliente-db



URLS y JSON de las API's

METHOD	URL	NAME	JSON REQUEST	JSON RESPONSE	ERROR REQUEST
POST	http://35.192.47.202:8880/api/DivertBox/Destination	Destination	{ "cam_Id": "string", "boxId": "string", "trackingId": 0 }	{ "trackingId": 0, "divertCode": 0, "boxId": "string" }	BadRequest { message:"string", "trackingId": 0, "divertCode": 99 }
POST	http://35.192.47.202:8880/api/DivertBox/Confirmation	Confirmation	{ "trackingId": 0, "divertCode": 0 }	OK200	{ divertCode:0 }

POST	http://35.192.47.202:8880/api/DivertLanes/LaneStatus	Lane Status	{ "lane_1_ status": 1, "lane_1_ full": 0, "lane_2_ status": 1, "lane_2_ full": 0, "lane_3_ status": 1, "lane_3_ full": 0, "lane_4_ status": 1, "lane_4_ full": 0, "lane_5_ status": 1, "lane_5_ full": 0, "lane_6_ status": 1, "lane_6_ full": 0, "lane_7_ status": 1, "lane_7_ full": 0, "lane_8_ status": 1, "lane_8_ full": 0,	OK200	BadRequest { message:"string" }
------	--	-------------	---	-------	--

