

Caso de Uso Automatización de Procesos Cafetito

Comunicación Web Service

Versión 1.2

**Elaborado por Melannie Paniagua y
Anderson Suruy**

Fecha Elaboración 15/03/2023

Historial Revisiones

[illegible]

1. Introducción.

1.1. Objetivo.

El presente CU tiene como objetivo describir la comunicación que habrá entre los diferentes actores y los servicios disponibles del web service y los flujos que pueden realizar los actores.

2. Definición Caso de Uso

2.1 Actores.

2.1.1. Web Service de la empresa Cafetito.

2.1.2. Usuario Cafetito **RN01**

2.1.3. Usuario Agricultor **RN01**

2.1.4. Usuario Peso Cabal **RN01**

2.2 Precondiciones.

2.2.1. Que el Web Service se encuentre activo.

2.2.2. Que la persona cuente con un usuario activo dentro de la aplicación.

2.2.3. Los usuarios de "Cafetito" deben contar con el rol "ROLE_BENEFICIO", los usuarios de "Agricultor" deben contar con el rol "ROLE_AGRICULTOR" y los usuarios de "Peso Cabal" deben contar con el rol "ROLE_PESOCABAL".

2.3 Flujo Normal Básico

1. El sistema se conecta al Web Service.
2. El sistema realiza el consumo de un servicio del WS. FA01, FA02, FA03, FA04, FA05, FA06, FA07, FA08, FA09, FA10, FA11, FA12, FA13, FA14, FA15, FA16, FA17, FA18, FA19, FA20, FA21, FA22, FA23, FA24, FA25, FA26, FA27, FA28, FA29, FA30, FA31, FA32, FA33, FA34, FA35.
3. Fin del caso de uso.

2.4 Flujos Alternos.

FA0: Consultar estado del WS

GET /beneficio/status



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO".
RN16.
2. El beneficio realiza la consulta del estado del web service.
3. El servicio devuelve el estado "Activo" o "Inactivo".
4. Fin del flujo alterno.

FA01: Activar o Inactivar un transporte

PUT /beneficio/updateTransport/{placa} Activar o Inactivar un transporte



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO".
RN16.
2. Se envían los parámetros con la información para actualización del registro.
RN06 No. 2
3. Al realizar la actualización. **RN07**
4. Fin del flujo alterno.

FA02: Activar o Inactivar un transportista

PUT /beneficio/updateCarrier/{DPI} Activar o Inactivar un transportista



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO".
RN16.
2. Se envían los parámetros con la información para actualización del registro.
RN06 No.3
3. Al realizar la actualización. **RN08**
4. Fin del flujo alterno.



FA03: Cambiar el estado de una cuenta a Cuenta Confirmada

PUT /beneficio/state/confirmed/account/{idCuenta} Actualizar el estado de una cuenta a 'Cuenta Confirmada'





1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO".
RN16.
2. El beneficio realiza el cambio de estado a Cuenta Confirmada. **RN02 y**
RN06 No.4
3. Al realizar la actualización. **RN09**
4. Fin del flujo alterno.

FA04: Cambiar el estado de una cuenta a Cuenta Cerrada

PUT /beneficio/state/close/account/{idCuenta} Actualizar el estado de una cuenta a 'Cuenta Cerrada'  



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio realiza el cambio de estado a Cuenta Cerrada. **RN02 y RN06 No.5**
3. Al realizar la actualización. **RN10**
4. Fin del flujo alterno.

FA05: Aceptar el ingreso de una parcialidad

PUT /beneficio/income/parts Aceptar el ingreso de una parcialidad  



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio acepta el ingreso de una parcialidad. **RN06 No.6**
3. Al realizar la actualización. **RN11**
4. Fin del flujo alterno.

FA06: Rechazar una parcialidad

PUT /beneficio/decline/part Rechazar una parcialidad  


1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio rechaza el ingreso de una parcialidad. **RN06 No.7**
3. Al realizar el rechazo. **RN12**
4. Fin del flujo alterno.

FA07: Rechazar una cuenta

PUT /beneficio/decline/account Rechazar una cuenta  

1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio rechaza una cuenta. **RN06 No.7**
3. Al realizar el rechazo. **RN13**
4. Fin del flujo alterno.

FA08: Crear un agricultor

POST /beneficio/create/farmer Crear un agricultor  

1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El agricultor crea un registro para agricultor con la información necesaria **RN06 No.8**
3. Se crea el registro en las bases de datos del beneficio y guarda la información en bitácora.
4. El servicio devuelve el código RN03 201 y muestra el MSG20.
5. Fin del flujo alterno.

FA09: Obtener transportes por agricultor

GET /beneficio/transport/list/{nitAgricultor} Lista los transportes de un agricultor



1. El beneficio consulta una lista de transportes por agricultor.
2. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 9**
3. El servicio devuelve una lista de transportes por agricultor. **RN06 No. 10**
4. Fin del flujo alterno.

FA10: Obtener transportistas por agricultor

GET /beneficio/carrier/list/{nitAgricultor} Lista los transportistas de un agricultor



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta una lista de transportistas por agricultor.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 11**
4. El servicio devuelve una lista de transportes por agricultor. **RN06 No. 12**
5. Fin del flujo alterno.

FA11: Obtener el detalle de una parcialidad

GET /beneficio/part/detail/{idParcialidad} Muestra el detalle de una parcialidad



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta el detalle de una parcialidad.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 13**
4. El servicio devuelve el detalle de la parcialidad. **RN06 No. 14**
5. Fin del flujo alterno.

FA12: Obtener los agricultores registrados en el beneficio

GET /beneficio/farmer/listar Lista los agricultores



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta los agricultores registrados.
3. El beneficio realiza la consulta a la base de datos.
4. El servicio devuelve una lista de agricultores. **RN06 No. 15**
5. Fin del flujo alterno.

FA13: Obtener cuentas por agricultor y estado

GET /beneficio/counts/agricultor/{nitAgricultor}/{estado} Lista las cuenta de un agricultor por NIT y estado

1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta una lista de cuentas por agricultor y estado.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 16**
4. El servicio devuelve una lista de cuentas. **RN06 No. 17**
5. Fin del flujo alterno.

FA14: Obtener parcialidades de una cuenta

GET /beneficio/count/parts/{idCuenta} Lista las parcialidades de una cuenta

1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta una lista parcialidad por cuenta.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 18**
4. El servicio devuelve una lista de parcialidades. **RN06 No. 19**
5. Fin del flujo alterno.

FA15: Obtener el detalle de una cuenta

GET /beneficio/count/detail/{idCuenta} Muestra la información de una cuenta

1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta el detalle de una cuenta.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 20**
4. El servicio devuelve el detalle de la cuenta. **RN06 No. 21**
5. Fin del flujo alterno.

FA16: Obtener cuentas por agricultor

GET /beneficio/count/agricultor/{nitAgricultor} Lista las cuentas de un agricultor por NIT

1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta una lista de cuentas por agricultor y estado.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 22**
4. El servicio devuelve una lista de cuentas. **RN06 No. 23**
5. Fin del flujo alterno.

FA17: Obtener catalogo por código

GET /beneficio/catalogue/{catalogo} Lista los catalogos por su codigo_catalogo



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio consulta una lista de catálogos por código.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 24**
4. El servicio devuelve una lista de catálogos. **RN06 No. 25**
5. Fin del flujo alterno.

FA18: Eliminar lógicamente un transporte

DELETE /beneficio/deleteTransport/{placa} Borrar lógicamente un transporte



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio realiza la eliminación de un transporte enviando los parámetros necesarios **RN06 No. 26**
3. Al realizar la eliminación. **RN14**
4. Fin del flujo alterno.

FA19: Eliminar lógicamente un transportista

DELETE /beneficio/deleteCarrier/{DPI} Borrar lógicamente un transportista



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_BENEFICIO". **RN16.**
2. El beneficio realiza la eliminación de un transportista enviando los parámetros necesarios **RN06 No. 27**
3. Al realizar la eliminación. **RN15**
4. Fin del flujo alterno.
- 5.

FA20: Crear un nuevo pesaje

POST /Agricultor/create/weighing Crear un nuevo pesaje



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El agricultor crea un registro para pesaje con la información necesaria **RN06 No.28**
3. crea el nuevo registro. **RN17**
4. Fin del flujo alterno.

FA21: Crear un nuevo transporte

POST /Agricultor/create/transport Crear un nuevo transporte



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El agricultor crea un registro para transporte con la información necesaria **RN06 No.29**
3. Se crea el nuevo registro. **RN18**
4. Fin del flujo alterno.

FA22: Crear un nuevo transportista

POST /Agricultor/create/carrier Crear un transportista



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El agricultor crea un registro para transportista con la información necesaria **RN06 No.30**
3. Se crea el nuevo registro. **RN19**
4. Fin del flujo alterno.

FA23: Crear una nueva parcialidad

POST /Agricultor/count/parts/create Crear una nueva parcialidad



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El agricultor crea un registro para transportista con la información necesaria **RN06 No.31**
3. Se crea el nuevo registro. **RN20**
4. Fin del flujo alterno.

FA24: Obtener los datos del agricultor

GET /Agricultor/show/farmer Muestra los datos del agricultor



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El Agricultor consulta su información.
3. El beneficio realiza la consulta a la base de datos.
4. El servicio devuelve la información. **RN06 No. 32**
5. Fin del flujo alterno.

FA25: Obtener los pesajes del agricultor

GET /Agricultor/list/weighing Lista los pesajes



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El Agricultor consulta la lista de pesajes.
3. El beneficio realiza la consulta a la base de datos.
4. El servicio devuelve una lista de pesajes. **RN06 No. 33**
5. Fin del flujo alterno.

FA26: Obtener los transportes del agricultor

GET /Agricultor/list/transport Lista todos los transportes



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El Agricultor consulta la lista de transportes.
3. El beneficio realiza la consulta a la base de datos.
4. El servicio devuelve una lista de transportes. **RN06 No. 34**
5. Fin del flujo alterno.

FA27: Obtener los transportistas del agricultor

GET /Agricultor/list/carrier Lista todos los transportistas



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El Agricultor consulta la lista de transportistas.
3. El beneficio realiza la consulta a la base de datos.
4. El servicio devuelve una lista de transportistas. **RN06 No. 35**
5. Fin del flujo alterno.

FA28: Obtener las parcialidades de un pesaje

GET /Agricultor/count/parts/{idPesaje} Lista las parcialidades de un pesaje



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El beneficio consulta una lista de parcialidades de un pesaje.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 36**
4. El servicio devuelve una lista de parcialidades. **RN06 No. 37**
5. Fin del flujo alterno.

FA29: Obtener transportes disponibles o vinculados a un pesaje

GET /Agricultor/assignment/transport/{idPesaje} Lista de transportes disponible o que pertenezcan a un pesaje



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El beneficio consulta una lista de transportes disponibles o vinculados a un pesaje.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 38**
4. El servicio devuelve una lista de transportes. **RN06 No. 39**
5. Fin del flujo alterno.

FA30: Obtener transportistas disponibles o vinculados a un pesaje

GET /Agricultor/assignment/carrier/{idPesaje} Lista de transportistas disponible o que pertenezcan a un pesaje



1. El sistema valida que el usuario logueado cuente con el rol "ROLE_AGRICULTOR". **RN16.**
2. El beneficio consulta una lista de transportistas disponibles o vinculados a un pesaje.
3. El beneficio realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 40**
4. El servicio devuelve una lista de transportistas. **RN06 No. 41**
5. Fin del flujo alterno.

FA31: Obtener cuentas de peso cabal

GET /peso/count Lista las cuenta de peso cabal



1. El sistema valida que el usuario logueado cuente con rol "ROLE_PRESOCABAL". **RN16.**
2. Peso cabal consulta una lista de cuentas.
3. El servicio devuelve una lista de cuentas que tengan algunos de los siguientes estados: "Cuenta Abierta", "Pesaje Iniciado" o "Pesaje Finalizado". **RN06 No. 42**
4. Fin del flujo alterno.

FA32: Obtener parcialidades de una cuenta

GET /peso/count/parts/{idCuenta} Lista las parcialidades de una cuenta



1. El sistema valida que el usuario logueado cuente con rol "ROLE_PRESOCABAL". **RN16.**
2. Peso cabal consulta una lista de parcialidades.
3. Peso cabal realiza la consulta a la base de datos enviando los parámetros necesarios. **RN06 No. 43**
4. El servicio devuelve una lista de parcialidades que se encuentren aceptadas. **RN06 No. 44**
5. Fin del flujo alterno.

FA33: Ingresar pesaje a una parcialidad

PUT /peso/update/weight Ingresar pesaje a una parcialidad



1. El sistema valida que el usuario logueado cuente con rol "ROLE_ PRESOCABAL". **RN16.**
2. Peso cabal ingresa el pesaje a una parcialidad. **RN06 No.45**
3. Al realizar el ingreso del pesaje. **RN21**
4. Fin del flujo alterno.

FA34: Crear un usuario

POST /api/auth/signup



1. El usuario crea un nuevo usuario con la información necesaria **RN06 No.46**
2. Se crea el nuevo registro. **RN22**
3. Fin del flujo alterno.

FA35: Inicio de sesión

POST /api/auth/signin



1. El usuario ingresa sus credenciales. **RN06 No.47**
2. El sistema valida las credenciales. **RN23**
3. El servicio retorna la información del usuario. **RN06 No.48**
4. Fin del flujo alterno.

2.5 Poscondiciones.

3. Requerimientos Suplementarios o no Funcionales

4. Anexos

4.1. Reglas de Negocio:

4.1.1. Ver Documento: Reglas de negocio (última versión formalizada)

5. Firmas Necesarias

<i>Persona</i>	<i>Sector</i>	<i>Firma</i>
Edy Rigoberto Ramírez	Jefe de departamento	