

DOCUMENTO

Especificaciones, Requerimientos y Criterios de Aceptación de Software

Mi Tienda – Comisiones Backoffice

Tabla de versiones

Versión	Nombre	Fecha	Descripción
0.1	Brandon Hernández	13/06/2022	Versión inicial
0.2	Brandon Hernández	14/06/2022	Ajustes en cronograma de actividades

Tabla de contenidos

Tabla de versiones	2
Tabla de contenidos	2
Mi Tienda - Comisiones Backoffice.....	4
Accesos a GDP y API's.....	4
CU01 – Acceso Principal Backoffice.....	4
API GDP.....	5
CU02 – Registro Dependientes.....	8
Flujo de transacción.....	8
API rewardsManagement y endpoints por utilizar	9
CU03 – Administración Dependiente y Dueño.....	12
Flujo de transacción de Administración de dueño y dependientes para un PDV	12
Flujo de transacción para modificación de datos del dependiente	13
API rewardsManagement y endpoints por utilizar	14
Rutas y Look and Feel	16
Lineamientos visuales.....	16
Pantalla posterior al login.....	16
Pantalla de búsqueda	17
Registro de Dependiente.....	18
Administración de Dependiente.....	19
Documentación Técnica	20
Estructura del proyecto	20

Estructura de los errores	21
Lenguaje de programación y tecnologías implementadas.....	21
Cronograma de actividades.....	22

Mi Tienda - Comisiones Backoffice

En este documento se van a detallar cada uno de los requerimientos solicitados para dicho sistema.

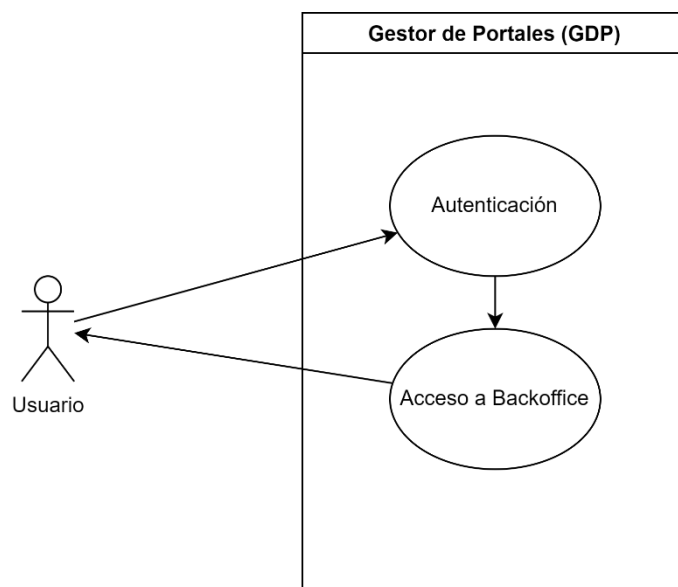
Accesos a GDP y API's

Solicito usuarios con rol de administrador y los distintos roles a manejar:

Nombre	Password	Rol
backofficettp_api	Tigo.2022	

CU01 – Acceso Principal Backoffice

El sistema debe realizar una petición y conectarse al Gestor de Portales (GDP) y autenticar que tenga los permisos necesarios para acceder. En el siguiente caso de uso se detalla el flujo del proceso.



Descripción:	El usuario realizará una petición hacia el Gestor de Portales (GDP), se autenticará y de ser correcto el ingreso se le dará el acceso a Backoffice.
Actores:	Usuario, GDP
Frecuencia de uso:	Cada vez que el usuario realice la consulta para iniciar sesión
Disparador:	Por medio de una petición que es enviada al Gestor de Portales (GDP) para validar la data y autenticar.
Pre-Condiciones:	<ul style="list-style-type: none"> - El aplicativo backoffice comisiones existe parametrizado dentro de GDP. - El usuario está creado dentro de GDP y tiene acceso al aplicativo backoffice Comisiones. - El módulo BackOffice deberá manejar roles y perfiles para los usuarios- los cuales son gestionados a nivel de GDP. - El usuario podrá gestionar dentro de backoffice comisiones a los PDV

	creados y dados de alta en DMS con perfil de comisiones.
--	--

API GDP

La aplicación Comisiones Backoffice estará integrada al aplicativo GDP manejando las siguientes funcionalidades:

1. Configuración de aplicativo Comisiones Backoffice sobre GDP.
2. Configuración de menú de aplicativo Comisiones Backoffice sobre GDP.
3. Gestión de usuarios y roles Comisiones Backoffice para autenticación sobre GDP.

Las integraciones para la autenticación de usuarios, validación de roles y carga del menú de la aplicación Comisiones Backoffice es realizada mediante las APIs que expone GDP las cuales se detallan a continuación:

El aplicativo GDP maneja 2 usuarios para una sesión de autenticación:

1. El usuario de sistema, con acceso a consumir las APIs expuestas, este debe estar configurado en el aplicativo Comisiones Backoffice para que al momento de consumir las APIs de GDP se puedan utilizar.
2. El usuario de aplicación, quien inicia sesión para poder administrar los datos del front end.

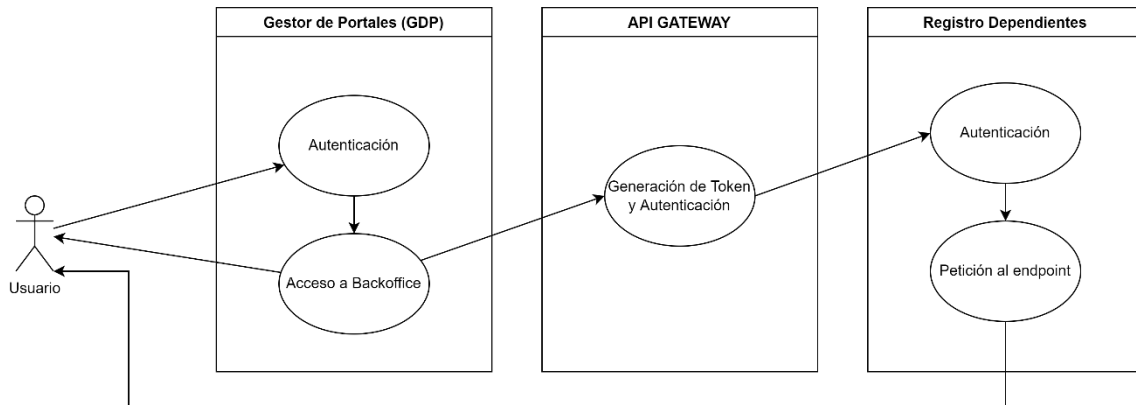
Tipo	Descripción API REST
Autenticación Usuario Sistema	<p>Esta API permite la autenticación del usuario de sistema para generación de token que permitirá llamar al API de autenticación de usuario de aplicación.</p> <p>URL: http://gdp-api.apps.csddev.comcel.com.gt/login/validate</p> <p>Petición: POST</p> <p>json body:</p> <pre>{ "usuario": "backofficettp_api", "password": "xxxxx" }</pre> <p>json response:</p> <pre>{ "respuesta": "Validate", "apps": [{ "icono": "", "nombre": "BackOffice TTP", "id": "16", "context": "url redirect aplicativo", "token": "xxxxxxxxxxxxxxxxxxxxxxx" }] }</pre>

	<pre> }], "nombreUsuario": "backofficettp_api", "idUsuario": "65", "email": "", "token": " xxxxxxxxxxxxxxxxxxxxxxxx" "mensajeAdicional": "", } </pre> <p>Nota: los datos de usuario y password deberán estar configurados en el aplicativo Backoffice para la interacción con GDP, este usuario es creado en GDP con permisos de consumo de APIs.</p>
Consulta de menú usuario aplicativo	<p>Una vez se consuma el API /login/validate, con el token obtenido se envía en el siguiente API, la cual permitirá validar los accesos y devolver el menú al cual tiene acceso el usuario de aplicativo.</p> <p>URL: http://gdp-api.apps.csddev.comcel.com.gt/apps/searchMenuUser</p> <p>Petición: POST</p> <p>Header:</p> <p>Authorization Bearer {token generado en operación /login/validate}</p> <p>json body:</p> <pre> { "appId": "16", "token": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxx" } </pre> <p>Notas: AppId, debe estar parametrizado en el Backoffice ya que es el id que da GDP al momento de configurar la aplicación como módulo.</p> <p>token: token de sesión que genera GDP al momento de hacer login el usuario de aplicación.</p> <p>El token de sesión viene en la url http://backofficettp-app.apps.csddev.comcel.com.gt/init/{hashsesionGDP} la cual es generada al momento que GDP redirecciona.</p> <p>json response:</p> <pre> { "respuesta": "{ \"user\": { \"ldap\": 1, \"name\": \"Jose Javier Vasquez\", \"aplicacion\": \"homeactivator\", \"id\": 25, \"email \": \"jjvasquez@tigo.com.gt\", \"username\": \"jjvasquez\", \"pe rfil\": \"Admin\" }, \"menu\": [{ \"nombre\": \"Solicitudes\", \"ur </pre>

	<pre> l\":"#\","editar\":false,\"active\":true,\"opciones\":[],\ \"icono\":"\", \"hijos\":[{\"nombre\":"Solicitudes recibidas\", \"url\":"request\", \"editar\":false,\"active\ :true,\"opciones\":[],\"visibilidad\":true,\"selected\":fals e,\"id\":"bxmpw5wqg\"},{\"nombre\":"Solicitudes pendientes\", \"url\":"request/pendientes\", \"editar\":fals e,\"active\":true,\"opciones\":[],\"visibilidad\":true,\"sel ected\":false,\"id\":"tfcl6xsi\"},{\"nombre\":"Solicitude s finalizadas\", \"url\":"request/finalizadas\", \"editar\":fa lse,\"active\":true,\"opciones\":[],\"visibilidad\":true,\"s elected\":false,\"id\":"s5hpx4r\"},{\"nombre\":"Solicitu des rechazadas\", \"url\":"request/rechazadas\", \"editar\":fals e,\"active\":true,\"opciones\":[],\"visibilidad\":true,\"sel ected\":false,\"id\":"0qndkqp6e\"}],\"visibilidad\":true,\" selected\":false,\"id\":"4cd52bvds\"},{\"nombre\":"Vendedo res y territorios\", \"url\":"\", \"editar\":false,\"active\":true ,\"opciones\":[],\"icono\":"fa-address- book\", \"hijos\":[{\"nombre\":"Vendedores\", \"url\":"sale sperson\", \"editar\":false,\"active\":true,\"opciones\":[],\ \"visibilidad\":true,\"selected\":false,\"id\":"tvcvti374\"} ,{\"nombre\":"Territorios\", \"url\":"territories\", \"edit ar\":false,\"active\":true,\"opciones\":[],\"visibilidad\":t rue,\"selected\":false,\"id\":"5c67m8rtj\"},{\"nombre\":"V endedores y territorios\", \"url\":"salespersonprofile\", \"editar\":fal se,\"active\":true,\"opciones\":[],\"visibilidad\":true,\"se lected\":false,\"id\":"mr5t1y3je\"}],\"visibilidad\":true,\ \"selected\":false,\"id\":"dizjkttaf\"},{\"nombre\":"Config uración\", \"url\":"\", \"editar\":false,\"active\":true,\"o pciones\":[],\"icono\":"fa-cogs \", \"hijos\":[{\"nombre\":"Celdas\", \"url\":"cells\", \"ed itar\":false,\"active\":true,\"opciones\":[],\"visibilidad\ \":true,\"selected\":false,\"id\":"rwrngxsg4w\"},{\"nombre\":" Tecnologías\", \"url\":"technologies\", \"editar\":false,\" active\":true,\"opciones\":[],\"visibilidad\":true,\"selecte d\":false,\"id\":"2abdu0wek\"},{\"nombre\":"Configuración\ \", \"url\":"configuration\", \"editar\":false,\"active\":tru e,\"opciones\":[],\"visibilidad\":true,\"selected\":false,\" id\":"hcagofse6\"}],\"visibilidad\":true,\"selected\":false ,\"id\":"cs5x0c7x9\"},{\"nombre\":"Reportería\", \"url\":" \", \"editar\":false,\"active\":true,\"opciones\":[],\"icono \":"fa-bar-chart\", \"hijos\":[{\"nombre\":"Reporte de vendedores\", \"url\":"reports\", \"editar\":false,\"active\ \":true,\"opciones\":[],\"visibilidad\":true,\"selected\":fal se,\"id\":"g7cov6vqx\"},{\"nombre\":"Reporte de solicitudes\", \"url\":"requestreports\", \"editar\":false,\ \"active\":true,\"opciones\":[],\"visibilidad\":true,\"select ed\":false,\"id\":"lrm09jj8f\"}],\"visibilidad\":true,\"sel ected\":false,\"id\":"ujfelr4n2\"}}\", \"codigo\": \"100\" </pre>
	}

CU02 – Registro Dependientes

En el siguiente caso de uso se detalla el flujo del proceso.



Descripción:	Se realizará una petición para registrar dependientes, teniendo en cuenta que se debe estar autenticado con anterioridad dentro del sistema (CU01) .
Actores:	Usuario, API Gateway, API REST
Frecuencia de uso:	Cada vez que el usuario realice la consulta para determinado endpoint
Disparador:	Petición POST que se realiza al endpoint /pointOfSale/{dmsId}/sellers
Pre-Condiciones:	<ul style="list-style-type: none"> - El usuario cuenta con acceso y permisos al módulo de Backoffice. - El PDV existe dentro del módulo comisiones.

Flujo de transacción

No.	Descripción
1	Se realizará el proceso de autenticación del CU01 .
2	El usuario ingresa a la opción del menú “Registro dependientes”
3	El sistema presenta pantalla de búsqueda para poder ubicar al PDV sobre el que se quiere registrar al dependiente nuevo.
4	El usuario ingresa datos de búsqueda idDMS y presiona “Buscar”.
5	El sistema presenta el resultado de búsqueda.
6	El usuario selecciona el epin sobre el cual desea realizar la operación de registro de dependiente, presiona “Registrar Dependiente”.
7	El sistema muestra pantalla para ingresar datos de registro.
8	El usuario ingresa datos y selecciona “Guardar”.
9	El sistema valida que sea un dependiente nuevo, de serlo realiza el registro hacia el epin seleccionado de forma exitosa y presenta un mensaje de resultado exitoso.

10	Si el sistema indica que el dependiente que se trata de registrar ya existe, presenta un mensaje de error.
----	--

API rewardsManagement y endpoints por utilizar

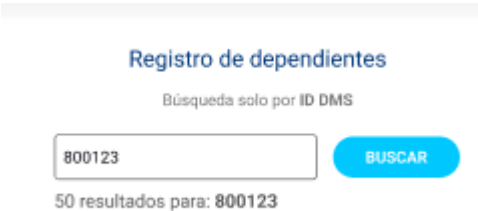
Petición http	Endpoint	Descripción
GET	/pointOfSale/{dmsId}/sellers?epin={val}	Consulta de lista de dependientes activos para puntos de venta.
POST	/pointOfSale/{dmsId}/sellers	Registra a dependiente dentro de un punto de venta.
GET	/pointOfSale/{dmsId}/stores?user={val}&epin={val}	Consulta de puntos de venta activos.

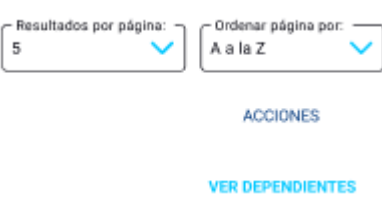
Para las operaciones respectivas tomar en cuenta lo siguiente:

- Toda interacción pasa por el API Gateway.
- El **dmsId** es obligatorio en todas las operaciones.
- El **epin** es obligatorio en la operación GET **sellers** y es opcional en GET **stores**.
- El **dmsId** y **epin** son de tipo integer y **user** es de tipo string.
- El formato de **epin** es 502XXXXXXX, se debe agregar el 502 al consumir la operación.
- El parámetro **user** es el usuario autenticado en la aplicación.

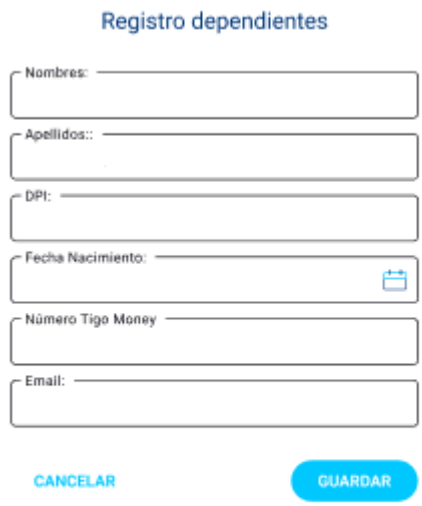
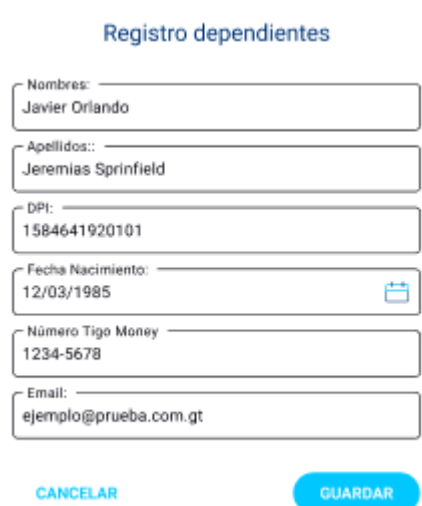
Descripción de las peticiones a ejecutar y sus respectivos endpoints:

Búsqueda:

	<ul style="list-style-type: none"> • Al presionar el botón “BUSCAR” se debe consumir la operación GET stores. • Si en el response de la operación GET stores el campo storeOwner = false significa que no existe un dueño en el punto de venta, se deberá mostrar un pop up con el mensaje “El punto de venta no tiene un dueño asignado”. • Los datos de respuesta de la operación GET stores se deben de guardar en sesión ya que se utilizarán en la página stores-management para listar los iddms y epin.
---	--

	<ul style="list-style-type: none"> Al presionar el botón “VER DEPENDIENTES” se debe consumir la operación GET sellers. Los datos de respuesta de la operación GET sellers se deben de guardar en sesión ya que se utilizarán en la página sellers-management para listar los # Tigo Money.
---	---

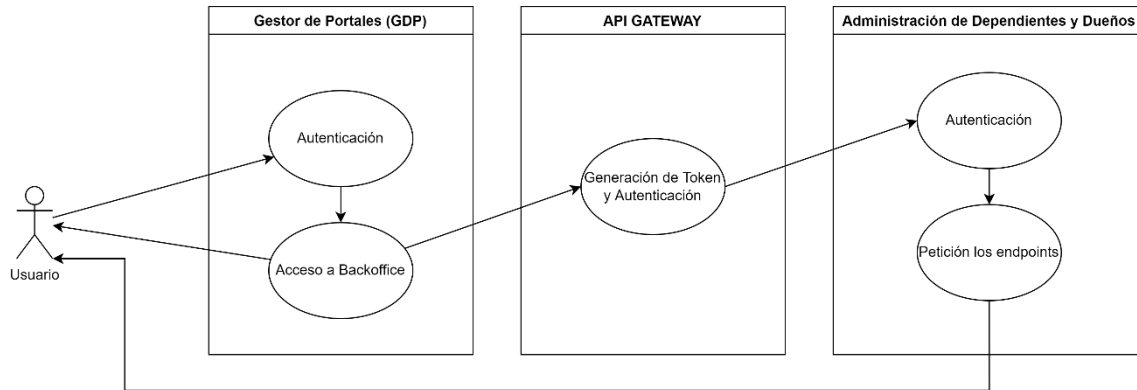
Registro de Dependientes

	<ul style="list-style-type: none"> Para la opción “Registro dependientes” del menú, al completar los datos y presionar el botón “GUARDAR” se debe consumir la operación POST sellers. <p>El campo # Tigo Money es de tipo integer y de formato XXXXXXXX.</p> <p>El campo Fecha Nacimiento es de tipo date y de formato dd/mm/yyyy.</p> <p>El campo Nombre, Apellido, email y DPI son de tipo string.</p> <p>Todos los campos son obligatorios.</p> <p>Se debe enviar el dmsId y epin.</p>
	<ul style="list-style-type: none"> Para la opción “Administración dependientes” una vez se seleccione un dependiente y se ingrese a modificar, al presionar el botón “GUARDAR” se debe consumir la operación PATCH wallets, si no se cambia el valor del campo # Tigo Money no se deberá de consumir la operación. <p>El campo # Tigo Money es de tipo integer y de formato XXXXXXXX.</p>

Petición HTTP - Endpoint	JSON Request o Response
GET – sellers	response: <pre> { "sellerList": { "name": "nombre", "lastName": "apellido", "personalId": "1122334450101", "birthDate": "28/12/1997", "msisdnTigoMoney": 50252015426, "email": "desbloqueo2tigo@gmail.com", "isOwner": "true", "storeName": "nombre punto de venta", "address": "direccion punto de venta", "latitude": 13.9226, "longitude": -90.819091 } } </pre>
POST – sellers	request: <pre> { "epin": 50246089513, "name": "nombre", "lastName": "apellido", "personalId": "1122334460101", "birthDate": "28/12/1998", "msisdnTigoMoney": 50248214774, "email": "desbloqueo2tigo@gmail.com" } </pre>
GET - stores	response: <pre> { "storeOwner": "true", "storeList": { "dmsId": 787960, "epin": 50246089513 } } </pre>

CU03 – Administración Dependiente y Dueño

En el siguiente caso de uso se detalla el flujo del proceso.



Descripción:	Administración de dueños y dependientes para un punto de venta específico realizando peticiones a los endpoints para eliminar, modificar o asignar un dueño.
Actores:	Usuario, API Gateway, API REST
Frecuencia de uso:	Cada vez que se realice una operación de modificación de dependiente o asignación de dueño.
Disparador:	Serie de peticiones que se realizan a los endpoints respectivos.
Pre-Condiciones:	<ul style="list-style-type: none"> - El usuario cuenta con acceso y permisos al módulo de Backoffice. - El PDV existe dentro del módulo comisiones y tiene registrados dependientes de acuerdo con el CU02.

Flujo de transacción de Administración de dueño y dependientes para un PDV

No.	Descripción
1	El usuario realiza el proceso de autenticación del CU01
2	El usuario ingresa a la opción del menú “Administración dependientes”
3	El sistema presenta pantalla de búsqueda para poder ubicar al PDV sobre el que se quiere realizar la acción de administración
4	El usuario ingresa datos de búsqueda idDMS y presiona “Buscar”
5	El sistema valida si para ese PDV no existe un dueño asignado, muestra un mensaje de notificación
6	El usuario se informa mediante el mensaje y procede a cerrarlo
7	El sistema presenta el resultado de búsqueda
8	El usuario selecciona el epin sobre el cual desea realizar la operación administración dueño y dependientes, presiona “Ver Dependientes”

9	El sistema muestra la lista de dependientes para ese epin seleccionado. Si el epin seleccionado ya cuenta con un dueño asignado, se presenta bloqueado el registro del dueño
10	El usuario realiza la acción de cambio de dueño en la columna “Dueño” o bien realiza la baja de algún dependiente presionando el botón con icono de “X”, presiona “Guardar”
11	El sistema muestra un mensaje de cambio exitoso. Si la modificación enviada no es exitosa, el sistema muestra mensaje de error.

Flujo de transacción para modificación de datos del dependiente

No.	Descripción
1	El usuario realiza el proceso de autenticación del CU01
2	El usuario ingresa a la opción del menú “Administración dependientes”
3	El sistema presenta pantalla de búsqueda para poder ubicar al PDV sobre el que se quiere realizar la acción de administración
4	El usuario ingresa datos de búsqueda idDMS y presiona “Buscar”
5	El sistema valida si para ese PDV no existe un dueño asignado, muestra un mensaje de notificación
6	El usuario se informa mediante el mensaje y procede a cerrarlo
7	El sistema presenta el resultado de búsqueda
8	El usuario selecciona el epin sobre el cual desea realizar la operación administración dueño y dependientes, presiona “Ver Dependientes”
9	El sistema muestra la lista de dependientes para ese epin seleccionado. Si el epin seleccionado ya cuenta con un dueño asignado, se presenta bloqueado el registro del dueño
10	El usuario selecciona un dependiente de la lista y presiona el botón con icono de “Lápiz”
11	El sistema muestra la pantalla de actualización de datos. Para este escenario solo se puede actualizar el número Tigo Money los otros datos mostrados se presentan bloqueados
12	El usuario realizar el cambio y presiona “Guardar”
13	El sistema valida que el número Tigo Money que se manda no exista ya asignado a otro epin, de no existir, el sistema guarda el cambio y presenta un mensaje de resultado exitoso
14	Si el numero Tigo Money ya existe, el sistema presenta un mensaje de error

Petición http	Endpoint	Descripción
GET	/pointOfSale/{dmsId}/sellers?epin={val}	Consulta de lista de dependientes activos para puntos de venta.
PATCH	/pointOfSale/{dmsId}/sellers	Baja a dependiente en punto de venta actualizando su estado.
PATCH	/pointOfSale/{dmsId}/wallets	Actualiza dato del número TMY para el pago de comisiones a dependiente dentro de un punto de venta.
POST	/pointOfSale/{dmsId}/owners	Asigna rol de dueño a un vendedor dentro de un punto de venta.
GET	/pointOfSale/{dmsId}/stores?user={val}&epin={val}	Consulta de puntos de venta activos.

Para las operaciones respectivas tomar en cuenta lo siguiente:

- Toda interacción pasa por el API Gateway.
- El **dmsId** es obligatorio en todas las operaciones.
- El **epin** es obligatorio en la operación GET **sellers** y es opcional en GET **stores**.
- El **dmsId** y **epin** son de tipo integer y **user** es de tipo string.
- El formato de **epin** es 502XXXXXXX, se debe agregar el 502 al consumir la operación.
- El parámetro **user** es el usuario autenticado en la aplicación.

Descripción de las operaciones a ejecutar y sus peticiones:

Petición HTTP - Endpoint	JSON Request o Response
GET – sellers	response: <pre>{ "sellerList": { "name": "nombre", "lastName": "apellido", "personalId": "1122334450101", "birthDate": "28/12/1997", "msisdTigoMoney": 50252015426, "email":</pre>

	<pre>"desbloqueotigo@gmail.com", "isOwner": "true", "storeName": "nombre punto de venta", "address": "direccion punto de venta", "latitude": 13.9226, "longitude": -90.819091 } }</pre>
GET - stores	response: <pre>{ "storeOwner": "true", "storeList": { "dmsId": 787960, "epin": 50246089513 } }</pre>
PATCH – sellers	request: <pre>{ "sellerList": [{ "msisdnTigoMoney": 50248214772 }, { "msisdnTigoMoney": 50249308484 }] }</pre>
POST - owners	request: <pre>{ "msisdnTigoMoney": 50248214772, "name": "nombre", "storeName": "nombre punto de venta", "address": "direccion punto de venta", "latitude": 13.9226, "longitude": -90.819091 }</pre>
PATCH - wallets	request: <pre>{ "oldMsisdnTigoMoney": 50248214772, "newMsisdnTigoMoney": 50248214773, "epin": 50246089513, "name": "nombre", "lastName": "apellido", "personalId": "1122334450101", }</pre>

	<pre>"birthDate": "28/12/1997", "email": "desbloqueoTigo@gmail.com" }</pre>
--	---

Rutas y Look and Feel

Lineamientos visuales

La aplicación debe mantener el diseño y Brand de la marca TIGO utilizando atomic design. El diseño se encuentra en: <https://brand.tigo.com/d/Y1SQ9kbC0nxf/atomic-design#/>

Pantalla posterior al login

- Ruta: <http://backofficettp-app.apps.csddev.comcel.com.gt/init/>



Pantalla de búsqueda

Registro de Dependientes

- Ruta: <http://backofficettp-app.apps.csddev.comcel.com.gt/stores-search>

Registro de dependientes

Búsqueda solo por ID DMS

BIUSCAR

50 resultados para: 800123

REGISTRAR DEPENDIENTE

Resultados por página: 5

Ordenar página por: A a la Z

	EPIN	ID DMS
<input type="checkbox"/>	50212345678	800123
<input type="checkbox"/>	50212345678	800123
<input type="checkbox"/>	50212345678	800123
<input type="checkbox"/>	50212345678	800123
<input type="checkbox"/>	50212345678	800123

<

1

2

3

4

>

Administración de Dependientes

- Ruta: <http://backofficettp-app.apps.csddev.comcel.com.gt/stores-search>

Registro de dependientes

Búsqueda solo por ID DMS

BIUSCAR

50 resultados para: 800123

Resultados por página: 5

Ordenar página por: A a la Z

EPIN	ID DMS	ACCIONES
50212345678	800123	VER DEPENDIENTES
50212345678	800123	VER DEPENDIENTES
50212345678	800123	VER DEPENDIENTES
50212345678	800123	VER DEPENDIENTES
50212345678	800123	VER DEPENDIENTES

<

1

2

3

4

>

Lineamientos por seguir

- El campo ID DMS es de tipo integer.
- El campo ID DMS es obligatorio.
- Si la búsqueda da como resultado que para el PDV indicado no hay un dependiente con rol de dueño, debe mostrarse un warning comunicando ese dato.
- Si la página **stores-search** es invocada desde la opción **“Registro dependientes”** se debe mostrar el botón con el texto **“Registrar Dependiente”**.
- Si la página **stores-search** es invocada desde la opción **“Administración dependientes”** se debe mostrar el botón con el texto **“Ver Dependientes”**.
- Se podrá seleccionar únicamente un registro.
- Al presionar el botón **“Registrar Dependiente”** se debe mostrar la página **sellers-register** con el formulario en blanco, el dmsId y epin se deben guardar en sesión y enviar en formulario.
- Al presionar el botón **“Ver Dependientes”** se debe mostrar la página **sellers-management**.
- El número de epin debe formatearse en pantalla sin código de área.

Registro de Dependiente

- Ruta: <http://backofficettp-app.apps.csddev.comcel.com.gt/sellers-register>

<u>Registro Dependiente</u>	<u>Administración Dependiente</u>
<div> <div>Registro dependientes</div> <div> <div>Nombres:</div> <div></div> </div> <div> <div>Apellidos:</div> <div></div> </div> <div> <div>DPI:</div> <div></div> </div> <div> <div>Fecha Nacimiento:</div> <div></div> </div> <div> <div>Número Tigo Money</div> <div></div> </div> <div> <div>Email:</div> <div></div> </div> <div> <div>CANCELAR</div> <div>GUARDAR</div> </div> </div>	<div> <div>Registro dependientes</div> <div> <div>Nombres:</div> <div>Javier Orlando</div> </div> <div> <div>Apellidos:</div> <div>Jeremias Springfield</div> </div> <div> <div>DPI:</div> <div>1584641920101</div> </div> <div> <div>Fecha Nacimiento:</div> <div>12/03/1985</div> </div> <div> <div>Número Tigo Money</div> <div>1234-5678</div> </div> <div> <div>Email:</div> <div>ejemplo@prueba.com.gt</div> </div> <div> <div>CANCELAR</div> <div>GUARDAR</div> </div> </div>

Lineamientos por seguir:

- Si se ingresa a esta página desde la opción **“Registro dependientes”** se debe mostrar el formulario en blanco.
 - El campo Número Tigo Money es de tipo integer y de formato XXXXXXXX.
 - El campo Fecha Nacimiento es de tipo date y de formato dd/mm/yyyy.
 - El campo Nombre, Apellido, email y DPI son de tipo string.

- Todos los campos son obligatorios.
- b. Si se ingresa desde la opción **“Administración dependientes”** se debe mostrar los datos del dependiente seleccionado para poder modificar la información.
 - Los datos del dependiente a excepción del Número Tigo Money se muestran bloqueados.
 - El campo Número Tigo Money es de tipo integer de formato XXXXXXXX y obligatorio.
- c. Agregar un botón de Cancelar por si no se desean hacer los cambios.

Administración de Dependiente

- Ruta: <http://backofficettp-app.apps.csddev.comcel.com.gt/sellers-management>



The screenshot shows a web application titled "Registro de dependientes". It includes a search bar with the text "Puedes buscar por: ID DMS / EPIN". There are two buttons: "GUARDAR" (with a save icon) and "CANCELAR" (with a cancel icon). Below these are two dropdown menus: "Resultados por página:" set to "5" and "Ordenar página por:" set to "A a la Z". The main content is a table with the following columns: NOMBRES, APELLIDOS, DPI, FECHA NAC., # TIGO MONEY, EMAIL, DUEÑO, and ACCIONES. The table contains five rows of data. The first row has a checked checkbox in the "DUEÑO" column. The "ACCIONES" column contains icons for edit (pencil) and delete (X). At the bottom of the table, there is a pagination control showing "1" as the current page and "2", "3", "4" as other pages.

NOMBRES	APELLIDOS	DPI	FECHA NAC.	# TIGO MONEY	EMAIL	DUEÑO	ACCIONES
Javier Orlando	Jeremias Springfield	1584641920101	12/03/1985	1234-5678	ejemplo@prueba.com.gt	<input checked="" type="checkbox"/>	
Juan Javier	Mendoza Matias	1584641920101	12/03/1985	1234-5678	ejemplo@prueba.com.gt	<input type="checkbox"/>	
Xavier José	Perez Xoc	1584641920101	12/03/1985	1234-5678	ejemplo@prueba.com.gt	<input type="checkbox"/>	
Yolanda Maria	Medina Marroquin	1584641920101	12/03/1985	1234-5678	ejemplo@prueba.com.gt	<input type="checkbox"/>	
Zintya Xiomara	Muniz Beltran	1584641920101	12/03/1985	1234-5678	ejemplo@prueba.com.gt	<input type="checkbox"/>	

Lineamientos por seguir:

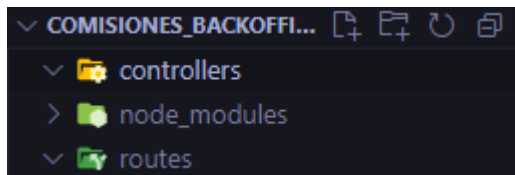
- Se podrá seleccionar únicamente un dueño.
- Al presionar el botón “X” se debe eliminar de la página el registro y mantener en memoria los registros a los que se les dio “X” para que al presionar Guardar se realice la acción sobre todos los registros.
- Si el dependiente es dueño el botón “X” debe estar deshabilitado, deberá primero cambiar de dueño para poder dar de baja a ese dependiente.
- Al presionar el botón con el icono de “Lápiz” se debe mostrar la página **sellers-register** y mostrar los datos del número Tigo Money a actualizar.
- Cambio de dueño y baja de usuarios se pueden hacer en la misma acción.
- Si se está realizando alguna acción de cambio de dueño y/o baja de dependiente, se deberá deshabilitar el botón con el icono de “Lápiz” y habilitar un botón de “Cancelar”, si no se realiza alguna de estas acciones el botón “Cancelar” deberá permanecer deshabilitado.
- El número Tigo Money debe formatearse en pantalla sin código de área.
- Agregar un botón de Cancelar por si no se desean hacer los cambios.

Documentación Técnica

Estructura del proyecto

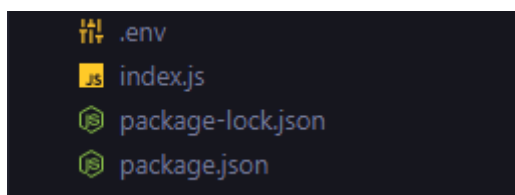
Para este aplicativo se utilizará **NodeJS y Angular**. Se procederá a crear un nuevo proyecto de **NodeJS** para consumir la información de los endpoints y crear el proyecto en **Angular** para el Frontend.

Para **NodeJS**, se utilizará la siguiente estructura:



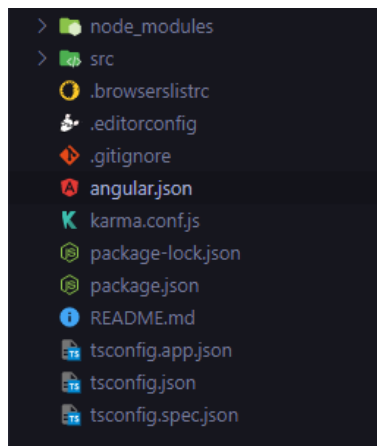
- **controllers:** controladores del servicio. Todos los archivos dentro llevaran el prefijo [nombre].controller.js
- **routes:** rutas del servicio. Todos los archivos dentro llevaran el prefijo [nombre].route.js
- **node_modules:** carpeta que se genera al correr el comando npm install en consola, este contiene las librerías y todos los módulos que utiliza NodeJS para su funcionamiento, dentro de este se encuentra Express.

En la raíz del proyecto se encontrará la estructura siguiente:



- **index.js:** contiene lo necesario para inicializar todo el proyecto.
- **package.json y package-lock.json:** contiene toda la metadata del proyecto, dentro de ello cada una de las dependencias a utilizar en el proyecto que serán detalladas más adelante.
- **.env:** variable de entorno donde estará ubicada toda la información sensible.

Para **AngularJS**, se utilizará la siguiente estructura:



- **node_modules:** carpeta que se genera al correr el comando npm install en consola, este contiene las librerías y todos los módulos que utiliza NodeJS para su funcionamiento, dentro de este se encuentra Express.
- **angular.json, package.json, package-lock.json, karma.conf.js, tsconfig.app.json, tsconfig.json, tsconfig.spec.json:** contiene toda la metadata y ajustes del proyecto.
- **src:** carpeta que contiene todo el maquetado del proyecto y la lógica del proyecto.

Estructura de los errores

Los errores serán mostrados por medio de una respuesta JSON de la siguiente manera:

```
{
  "name": "",
  "apiKey": "",
  "uri": "",
  "responseCode": "",
  "responseTime": "",
  "hostname": "",
  "pid": 0,
  "level": 0,
  "msg": "",
  "time": "",
  "v": 0,
  "clientIp": ""
}
```

Lenguaje de programación y tecnologías implementadas

- Lenguaje de programación
 - JavaScript
- Entorno en tiempo de ejecución multiplataforma para la capa del servidor
 - NodeJS v14.17.0
- Expuesto sobre
 - Express v4.17.1
- Puerto expuesto de NodeJS
 - 3000
- Frontend
 - AngularJS
- Puerto expuesto de Angular
 - 4200
- Librerías instaladas
 - body-parser
 - jsonwebtoken
 - dotenv
 - cors
 - express
 - nodemon
 - Morgan

- Control de versiones
 - Git

Cronograma de actividades

		Nombre	Trabajo	Duracion	Inicio	Terminado
1		Ambiente de Desarrollo	16 horas	2 days	14/06/22 08:00 AM	15/06/22 05:00 PM
2		Creación del repositorio	1 hora	0.125 days	14/06/22 08:00 AM	14/06/22 09:00 AM
3		Crear Proyecto	15 horas	1.875 days	14/06/22 09:00 AM	15/06/22 05:00 PM
4		Crear Proyecto con NodeJS y Express	7 horas	0.875 days	14/06/22 09:00 AM	14/06/22 05:00 PM
5		Crear Proyecto con Angular	8 horas	1 day	15/06/22 08:00 AM	15/06/22 05:00 PM
6		CU01 - Acceso Principal Backoffice	16 horas	2 days?	16/06/22 08:00 AM	17/06/22 05:00 PM
7		Validación e integración de sesiones	8 horas	1 day?	16/06/22 08:00 AM	16/06/22 05:00 PM
8		Obtención de accesos y pintado de menús	8 horas	1 day?	17/06/22 08:00 AM	17/06/22 05:00 PM
9		CU02 - Registro Dependientes	36 horas	4.5 days?	20/06/22 08:00 AM	24/06/22 01:00 PM
10		Búsqueda de usuarios	8 horas	1 day?	20/06/22 08:00 AM	20/06/22 05:00 PM
11		Administración de usuarios	12 horas	1.5 days?	21/06/22 08:00 AM	22/06/22 01:00 PM
12		Registro de usuarios	12 horas	1.5 days?	22/06/22 01:00 PM	23/06/22 05:00 PM
13		Manejo de respuestas y errores	4 horas	0.5 days?	24/06/22 08:00 AM	24/06/22 01:00 PM
14		CU03 - Administración Dependiente y Dueño	62 horas	7.75 days?	24/06/22 01:00 PM	7/07/22 10:00 AM
15		Definir perfil de dueño para el PDV	16 horas	2 days?	24/06/22 01:00 PM	28/06/22 01:00 PM
16		Cambiar el dueño para un PDV	12 horas	1.5 days?	28/06/22 01:00 PM	29/06/22 05:00 PM
17		Desasignar a un dependiente del PDV	14 horas	1.75 days?	30/06/22 08:00 AM	1/07/22 03:00 PM
18		Actualización de datos de dependiente	16 horas	2 days?	1/07/22 03:00 PM	6/07/22 03:00 PM
19		Manejo de respuesta y errores	4 horas	0.5 days?	6/07/22 03:00 PM	7/07/22 10:00 AM
20		No funcionales	24 horas	3 days?	6/07/22 10:00 AM	11/07/22 10:00 AM
21		Pruebas	24 horas	3 days?	6/07/22 10:00 AM	11/07/22 10:00 AM
22		Documentación	32 horas	4 days?	11/07/22 10:00 AM	15/07/22 10:00 AM
23		Documentación	32 horas	4 days?	11/07/22 10:00 AM	15/07/22 10:00 AM

Nota: La documentación se trabajará paralelamente con las pruebas UAT