

## Contratos

ID:	Nombre: Crear ruta	
Prioridad: Alta	Historia de usuario Es necesario crear las rutas para la entrega de los productos vendidos	
Estimado: 3 puntos		
Módulo: Rutas		
Criterios de Aceptación		
El microservicio retorna un código 200 con un mensaje satisfactorio al asignarle las zonas a dicha ruta		
El microservicio debe tener la siguiente configuración:		
Ruta: /rutas/crearRuta		
Método: POST		
Formato de entrada: form-data		
Payload:		
Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: - 1 - 2 Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.
Header:		
Atributo	Tipo	Descripción

Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción
nombre	cadena	identificación de la ruta <b>Campo obligatorio</b>
zonas	cadena	Arreglo de zonas asignadas <b>Campo obligatorio</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del registro del usuario administrativo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
403	Usuario sin permisos
500	Error interno del servidor
502	Error en los parámetros de solicitud

**Body de Salida:**

Atributo	Tipo	Descripción
----------	------	-------------

estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

nombre	2994902460101
zonas	[23,24,22,20,19]

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Se creó la ruta de manera exitosa"
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 403
  "error": "Usuario sin permisos",
  "mensaje": "El usuario no tiene permisos para crear rutas"
}
```

<b>ID:</b>	<b>Nombre: Obtener productos de la ruta</b>
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Es necesario que conocer los productos que tiene que irá a entregar una ruta
<b>Estimado:</b> 3 puntos	
<b>Módulo:</b> Rutas	
<b>Criterios de Aceptación</b>  El microservicio retorna un código 200 con una lista de productos que se hayan vendido con una zona de entrega igual a las zonas asignadas por la ruta.  El microservicio debe tener la siguiente configuración: <b>Ruta:</b> /rutas/obtenerProductosRuta <b>Método:</b> POST	

**Formato de entrada:** form-data

**Payload:**

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: - 1 - 2 Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción
nombre	cadena	identificación de la ruta <b>Campo obligatorio</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación

mensaje	cadena	muestra información del registro del usuario administrativo
---------	--------	---

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
403	Usuario sin permisos
500	Error interno del servidor
502	Error en los parámetros de solicitud

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

nombre	ruta 1
--------	--------

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "data": [ {"sku": 3 , "cantidad":3}, {"sku": 1000 , "cantidad":50} ]
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 403  
  "error": "Usuario sin permisos",  
  "mensaje": "El usuario no tiene permisos para obtener productos de las rutas"  
}
```

ID:	Nombre: Entregar ruta	
Prioridad: Alta	<b>Historia de usuario</b> Es necesario entregar los productos que estén relacionados con la ruta.	
Estimado: 3 puntos		
Módulo: Rutas		
<b>Criterios de Aceptación</b>  El microservicio retorna un código 200 con un mensaje satisfactorio cuando se vaya a entregar todos los productos a los compradores y se actualice el stock del inventario  El microservicio debe tener la siguiente configuración: <b>Ruta:</b> /rutas/entregarRuta <b>Método:</b> POST <b>Formato de entrada:</b> form-data  <b>Payload:</b>		
Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: - 1 - 2 Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción
nombre	cadena	identificación de la ruta <b>Campo obligatorio</b>

**Formato de salida:** JSON**Código de respuesta exitosa:** HTTP 200**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del registro del usuario administrativo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
403	Usuario sin permisos
500	Error interno del servidor
502	Error en los parámetros de solicitud

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

nombre	ruta 1
--------	--------

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "Se entregó los productos de la ruta de manera exitosa"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 403  
  "error": "Usuario sin permisos",  
  "mensaje": "El usuario no tiene permisos para entregar rutas"  
}
```

<b>ID:</b> 09	<b>Nombre:</b> Agregar usuario
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Es necesario que los usuarios administrativos sean creados a través del módulo de recursos humanos, para registrar los nuevos ingresos
<b>Estimado:</b> 3 puntos	
<b>Módulo:</b> Recursos humanos	
<b>Criterios de Aceptación</b>	



El microservicio retorna un código 200 con un mensaje satisfactorio al agregar de manera exitosa un usuario administrativos, para los siguientes casos se retornará un código de error con información del mismo:

- No se llenen los campos obligatorios
- El usuario ya existe en el sistema
- La confirmación del password no coincide.

El microservicio debe tener la siguiente configuración:

**Ruta:** /rrhh/agregarUsuario

**Método:** POST

**Formato de entrada:** form-data

**Payload:**

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: <ul style="list-style-type: none"><li>- 1</li><li>- 2</li></ul> Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción
cui	entero	identificación del usuario

		<b>Campo obligatorio</b>
correo	cadena	correo electrónico del usuario <b>Campo obligatorio</b>
nombres	cadena	nombres del usuario <b>Campo obligatorio</b>
apellidos	cadena	apellidos del usuario <b>Campo obligatorio</b>
password	cadena	contraseña del usuario <b>Campo obligatorio</b>
fechaNacimiento	cadena	fecha de nacimiento del usuario <b>Campo obligatorio</b>
confirmacionPassword	cadena	confirmación de la contraseña del usuario <b>Campo obligatorio</b>
areaAsignada	cadena	área que se le asignará al usuario administrativo <b>Campo obligatorio</b>
sueldo	decimal	sueldo del usuario administrativo <b>Campo obligatorio</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del registro del usuario administrativo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado

403	Usuario sin permisos
500	Error interno del servidor

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

cui	2994902460101
correo	m@gmail.com
nombres	Mynor Rene
apellidos	Ruiz Guerra
password	P@\$\$w0rD
confirmacionPassword	P@\$\$w0rD
areaAsignada	contabilidad
fechaNacimiento	18/08/1999
suelo	5000.00

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Se creo el usuario administrativo de manera exitosa"
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 400
  "error": "Falló la confirmación del password",
  "mensaje": "Las contraseñas deben de coincidir"
}
```

```

}

{
  "estatus": 403
  "error": "Usuario sin permisos",
  "mensaje": "El usuario no tiene permisos para crear usuarios administrativos,
solamente el personal de rrhh tiene permisos asignados"
}

```

<b>ID:</b> 10	<b>Nombre:</b> Dar de baja usuarios administrativos	
<b>Prioridad:</b> Media	<b>Historia de usuario</b> Es necesario que el personal de recursos humanos pueda dar de baja a los usuarios administrativos necesarios.	
<b>Estimado:</b> 2 puntos		
<b>Módulo:</b> Recursos humanos		
<b>Criterios de Aceptación</b>  El microservicio retorna un código 200 con un mensaje satisfactorio al dar de baja a un usuario administrativo de manera exitosa, para los siguientes casos se retornará un código de error con información del mismo: <ul style="list-style-type: none"><li>- No exista el usuario administrativo</li><li>- El usuario administrativo esté en un estado inactivo</li></ul> El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /rrhh/eliminarUsuarioAdmin <b>Método:</b> Put <b>Formato de entrada:</b> form-data   <b>Payload:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: <ul style="list-style-type: none"><li>- 1</li><li>- 2</li></ul>

		Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

#### Header:

Atributo	Tipo	Descripción
Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

#### Body:

Atributo	Tipo	Descripción
cui	entero	identificación del usuario <b>Campo obligatorio</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

#### Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información de la eliminación del usuario administrativo

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
403	Usuario sin permisos

500

Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

cui	2994902460101
-----	---------------

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "Se dio de baja a el usuario administrativo de manera exitosa"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 400  
  "error": "Falló de cui",  
  "mensaje": "El CUI no es válido, debe ser un número de 13 dígitos"  
}  
  
{  
  "estatus": 403  
  "error": "Usuario sin permisos",  
  "mensaje": "El usuario no tiene permisos para dar de baja a los usuarios  
administrativos, solamente el personal de rrhh tiene permisos asignados"  
}
```

<b>ID:</b> 11	<b>Nombre:</b> Agendar entrevistas
<b>Prioridad:</b> Media	<b>Historia de usuario</b> Es necesario que el personal de recursos humanos pueda agendar las citas de las entrevistas con los candidatos para los puestos laborales.
<b>Estimado:</b> 3 puntos	
<b>Módulo:</b> Recursos humanos	

### Criterios de Aceptación

El microservicio retorna un código 200 con un mensaje satisfactorio al agendar la entrevista y enviar un correo electrónico de manera exitosa al candidato con la información de la entrevista, para los siguientes casos se retornará un código de error con información del mismo:

- No exista el correo electrónico
- Se ingrese una fecha no válida para la cita

El microservicio debe tener la siguiente configuración:

**Ruta:** /rrhh/agendarEntrevista

**Método:** POST

**Formato de entrada:** form-data

### Payload:

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario administrativo.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: <ul style="list-style-type: none"><li>- 1</li><li>- 2</li></ul> Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

### Header:

Atributo	Tipo	Descripción
Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

### Body:

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del entrevistado <b>Campo obligatorio.</b>

nombres	cadena	nombres del candidato a entrevistar <b>Campo obligatorio</b>
apellidos	cadena	apellidos del candidato a entrevistar <b>Campo obligatorio</b>
correo	cadena	correo del candidato a entrevistar <b>Campo obligatorio</b>
fecha	cadena	fecha de la entrevista <b>Campo obligatorio</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del registro de la entrevista

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
403	Usuario sin permisos

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error



mensaje	cadena	muestra más información del error
---------	--------	-----------------------------------

**Ejemplos de parámetros de entrada:**

cui	2994902460101
nombres	Mynor Rene
apellidos	Ruiz Guerra
correo	m@gmail.com
fecha	18/04/2023

**Ejemplos de parámetros de salida exitosa:**

```
{
  "estatus": 200,
  "mensaje": "Se agendó la entrevista con el candidato de manera exitosa"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400
  "error": "Correo electrónico invalido",
  "mensaje": "El correo electrónico no se pudo enviar"
}

{
  "estatus": 403
  "error": "Usuario sin permisos",
  "mensaje": "El usuario no tiene permisos para agendar entrevistas,, solamente el personal de rrhh tiene permisos asignados"
}
```

<b>ID:</b> 13	<b>Nombre:</b> Agregar producto al carrito
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> El sistema requiere que los usuarios puedan agregar los productos de su interés a un carrito de compras
<b>Estimado:</b> 3 puntos	
<b>Módulo:</b> Compras	

### Criterios de Aceptación

Este microservicio le permitirá al cliente agregar un producto al carrito de compras, por defecto la cantidad de este producto será de 1.

El microservicio debe tener la siguiente configuración:

**Ruta:** /buy/addShoppingCart

**Método:** POST

**Formato de entrada:** form-data

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	multipart/form-data
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción
id_producto	entero	Número que identifica al producto.
cantidad	entero	Número que indica la exactitud del pedido

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo

**Código de respuesta fallida:**

Código	Descripción
400	Fallo en el proceso al intentar enviar el correo
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra más información del error
mensaje	cadena	muestra el nombre del error

**Ejemplos de parámetros de entrada:**

```
{  
  "id_producto": 1,  
  "cantidad": 3  
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "Producto agregado correctamente"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 400,  
  "mensaje": "Error con los campos."  
}
```

```
{  
  "estatus": 500,  
  "error": "Error del servidor"  
  "mensaje": "No se pudo conectar a la base de datos"  
}
```

<b>ID:</b> 14	<b>Nombre:</b> Ver Carrito de Compras	
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> El sistema debe permitirle al cliente visualizar los productos de su carrito de compras.	
<b>Estimado:</b> 3 puntos		
<b>Módulo:</b> Compras		
<b>Criterios de Aceptación</b>  Este microservicio se debe conectar con el microservicio de consultar producto específico por cada uno de los productos agregados al carrito, para mostrar la información de este.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /buy/getShoppingCart <b>Método:</b> GET <b>Header:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Authorization	header	Bearer <token>
<b>Formato de salida:</b> JSON <b>Código de respuesta exitosa:</b> HTTP 200 <b>Salida:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
estatus	entero	código devuelto al obtener una respuesta satisfactoria
mensaje	cadena	mensaje descriptivo

data	arreglo	arreglo de objetos con la información de los productos solicitados
------	---------	--

#### Código de respuesta fallida:

Código	Descripción
404	Recurso no encontrado
401	Token invalido

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra más información del error
mensaje	cadena	muestra el nombre del error

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Productos consultados exitosamente",
  "data": [ { <producto1> }, { <producto2> } ]
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 404,
  "error": "Producto inexistente"
  "mensaje": "El producto solicitado no existe"
}

{
  "estatus": 500,
  "error": "Error del servidor"
  "mensaje": "No se pudo conectar a la base de datos"
}
```

ID: 15	Nombre: Eliminar producto del carrito										
Prioridad: Media	Historia de usuario Es importante permitir que se elimine un producto del carrito de compras, ya que un cliente puede cambiar de opinion.										
Estimado: 2 puntos											
Módulo: Compras											
Criterios de Aceptación											
Este microservicio le permitirá al cliente eliminar un producto de su carrito de compras para ya no visualizarlo y evitar un cobro adicional.											
El microservicio debe tener la siguiente configuración:											
Ruta: /buy/deleteProductCart											
Método: DELETE											
Formato de entrada: form-data											
Header:											
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Content-Type</td><td>header</td><td>multipart/form-data</td></tr></table>			Atributo	Tipo	Descripción	Content-Type	header	multipart/form-data			
Atributo	Tipo	Descripción									
Content-Type	header	multipart/form-data									
Body:											
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>idProducto</td><td>entero</td><td>Número que identifica al producto.</td></tr></table>			Atributo	Tipo	Descripción	idProducto	entero	Número que identifica al producto.			
Atributo	Tipo	Descripción									
idProducto	entero	Número que identifica al producto.									
Formato de salida: JSON											
Código de respuesta exitosa: HTTP 200											
Salida:											
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>estatus</td><td>entero</td><td>Código devuelto al recibir una respuesta satisfactoria</td></tr><tr><td>mensaje</td><td>cadena</td><td>mensaje descriptivo</td></tr></table>			Atributo	Tipo	Descripción	estatus	entero	Código devuelto al recibir una respuesta satisfactoria	mensaje	cadena	mensaje descriptivo
Atributo	Tipo	Descripción									
estatus	entero	Código devuelto al recibir una respuesta satisfactoria									
mensaje	cadena	mensaje descriptivo									

**Código de respuesta fallida:**

Código	Descripción
400	Fallo en el proceso al intentar enviar el correo
500	Error interno del servidor
401	Usuario no autenticado
403	Usuario sin permisos
502	Error en parametros de entrada

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra más información del error
mensaje	cadena	muestra el nombre del error

**Ejemplos de parámetros de entrada:**

```
{  
  "idProducto": 1,  
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "Producto eliminado del carrito correctamente"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 400,  
  "error": "Error con los campos",  
  "mensaje": "Campos requeridos"  
}
```

```
{  
  "estatus": 500,  
  "error": "Error del servidor"
```

```
"mensaje": "No se pudo conectar a la base de datos"
}
```

ID: 16	Nombre: Cancelar compra										
Prioridad: Media	Historia de usuario Es requerido que el cliente pueda vaciar por completo su carrito (cancelar compra).										
Estimado: 3 puntos											
Módulo: Compras											
<b>Criterios de Aceptación</b>  Este microservicio le permitirá al cliente eliminar todos los productos de su carrito de compras para ya no visualizarlos.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /buy/emptyShoppingCart <b>Método:</b> DELETE  <b>Header:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Authorization</td><td>header</td><td>Bearer &lt;token&gt;</td></tr></table>			Atributo	Tipo	Descripción	Authorization	header	Bearer <token>			
Atributo	Tipo	Descripción									
Authorization	header	Bearer <token>									
<b>Formato de salida:</b> JSON <b>Código de respuesta exitosa:</b> HTTP 200 <b>Salida:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>estatus</td><td>entero</td><td>Código devuelto al obtener una respuesta satisfactoria</td></tr><tr><td>mensaje</td><td>cadena</td><td>mensaje descriptivo</td></tr></table>			Atributo	Tipo	Descripción	estatus	entero	Código devuelto al obtener una respuesta satisfactoria	mensaje	cadena	mensaje descriptivo
Atributo	Tipo	Descripción									
estatus	entero	Código devuelto al obtener una respuesta satisfactoria									
mensaje	cadena	mensaje descriptivo									
<b>Código de respuesta fallida:</b>											



Código	Descripción
500	Error interno del servidor
502	Error del requerimiento
401	Usuario no autenticado

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra más información del error
mensaje	cadena	muestra el nombre del error

**Ejemplos de parámetros de salida exitosa:**

```
{
  "estatus": 200,
  "mensaje": "Compra cancelada exitosamente"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400,
  "error": "Error con los campos.",
  "mensaje": "No vienen campos requeridos"
}

{
  "estatus": 500,
  "error": "Error del servidor"
  "mensaje": "No se pudo conectar a la base de datos"
}
```

## Usuarios

<b>ID:</b> 01	<b>Nombre:</b> Registrar usuario
<b>Prioridad:</b> Alta	<b>Historia de usuario</b>



		Atributo requerido
numero_grupo	entero	Es el número de grupo que representa al cliente empresarial.  Atributo requerido si el tipo = "E"
seccion_grupo	cadena	Es la sección del grupo que representa al cliente empresarial  Atributo requerido si el tipo = "E"

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del registro del usuario

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
404	Recurso no encontrado
500	Error interno del servidor
501	Si es usuario empresarial debe de ingresar numero de grupo o seccion

**Body de Salida:**

Atributo	Tipo	Descripción
----------	------	-------------

estatus	entero	indica el código de respuesta fallida
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### **Ejemplos de parámetros de entrada:**

```
{
  "cui": 2168095170407,
  "nombres": "Rafael Angel",
  "apellidos": "Chocoj Xinico",
  "contrasena": "123",
  "correo": "rafaelcliveguitar@gmail.com",
  "fecha_nacimiento": "2020-11-25",
  "tipo": "E",
  "numero_grupo": 914,
  "seccion_grupo": "N"
}
```

#### **Ejemplos de parámetros de salida exitosa:**

```
{
  "estatus": 200,
  "mensaje": "Se ha registrado el usuario exitosamente"
}
```

#### **Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 401,
  "mensaje": "Si es usuario empresarial debe de ingresar numero de grupo o seccion"
}

{
  "estatus": 500, "error": "Error interno del servidor",
  "mensaje": "Error, ya existe usuario con el mismo cui. "
}

{
  "estatus": 400, "error": "error", "mensaje": "error al enviar correo"
}

{
  "estatus": 400, "error": "error server", "mensaje": "Error al interpretar la solicitud"
}
```

**Servicio de usuario consume envio de correo:**

/notification/sendEmail

```
{
  "para": "rafael@gmail.com",
  "asunto": "CONFIRMAR CUENTA",
  "mensaje": "bienvenido ",
  "plantilla": "confirmEmail",
  "data": {"code": "enlace"}
}
```

<b>ID:</b> 03	<b>Nombre:</b> Verificar correo													
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Es importante verificar que la dirección de correo electrónico exista y el usuario tenga acceso a ella.													
<b>Estimado:</b> 1 punto														
<b>Módulo:</b> Usuario														
<b>Criterios de Aceptación</b>  Se deberá ingresar al link que fue enviado por correo, para verificar que este exista correo.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/verifyemail <b>Método:</b> POST <b>Formato de entrada:</b> JSON  <b>Header:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Content-Type</td><td>header</td><td>application/json</td></tr></table> <b>Body:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>token</td><td>cadena</td><td>cui del usuario encriptado para validar ya activar cuenta</td></tr></table>			Atributo	Tipo	Descripción	Content-Type	header	application/json	Atributo	Tipo	Descripción	token	cadena	cui del usuario encriptado para validar ya activar cuenta
Atributo	Tipo	Descripción												
Content-Type	header	application/json												
Atributo	Tipo	Descripción												
token	cadena	cui del usuario encriptado para validar ya activar cuenta												

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información de la verificación del correo

Código	Descripción
400	Error al interpretar la solicitud
404	no existe usuario
500	JWT ya expiro

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

```
{  
  "token":  
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjdWkiOiJxNjgwOTUxNzA0MDcsImV4cCI6MTY4MjMzMzI5OX0.zzV7IShwElgG6iRe5D11dMFIMuNTONWIBBYwKNfaYeo"  
}
```

```
{
  "estatus": 200,
  "mensaje": "se activó cuenta correctamente"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 404, "error": "activacion cuenta",
  "mensaje": "no existe usuario"
}

{
  "estatus": 300,
  "error": "JWT ya expiro",
  "mensaje": "ya expiro invitacion",
}

{
  "estatus": 400,
  "error": "Error en token enviado",
  "mensaje": "Error en token invalido",
}

{
  "estatus": 400, "error": "error servidor", "mensaje": "Error al interpretar la solicitud"
}
```

<b>ID:</b> 04	<b>Nombre:</b> Iniciar sesión
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Es importante tener un medio por el cual los usuarios puedan ingresar a la plataforma de manera segura.
<b>Estimado:</b> 2 punto	
<b>Módulo:</b> Usuario	
<b>Criterios de Aceptación</b> Al validar las credenciales se deberá, mediante al estándar JWT, crear un token válido que será devuelto para que el usuario esté autenticado.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/login <b>Método:</b> POST <b>Formato de entrada:</b> JSON  <b>Payload:</b>	

Atributo	Tipo	Descripción
cui	entero	el número que identifica al usuario en el sistema.
id_grupo	entero	el grupo si el usuario es empresarial
seccion	cadena	seccion si el usuario es empresarial

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	application/json

**Body:**

Atributo	Tipo	Descripción
correo	cadena	correo electrónico del usuario
contrasena	cadena	contraseña del usuario

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del inicio de sesión
token	cadena	token válido para la autenticación

**Código de respuesta fallida:**

Código	Descripción
--------	-------------



400	Error al interpretar la solicitud
404	correo o contraseña no existen

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error
id_grupo	entero	nodel grupo si es empresarial
seccion	cadena	seccion del grupo si es usuario empresarial
data	array	datos del usuario
token	cadena	datos codificados del usuario

#### Ejemplos de parámetros de entrada:

```
{
  "contrasena": "123",
  "correo": "rafaelcliveguitar@gmail.com"
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{
  "data": {
    "apellidos": "Chocoj Xinico",
    "cui": 2168095170407,
    "estado": 1,
    "fecha_nacimiento": "Wed, 25 Nov 2020 00:00:00 GMT",
    "nombres": "Rafael Angel",
    "numero_grupo": 914,
    "seccion_grupo": "N",
    "tipo": "E"
  },
  "estatus": 200,
  "id_grupo": 914,
  "mensaje": "Se inició sesión correctamente",
  "seccion": "N",
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdWkiOiJlNjgwOTUxNzA0MDcslmlkX2
```

```
dydXBvljo5MTQslnNIY2Npb24iOiJOLIwiZXhwljoxNjgyNDY1ODI2fQ.tV4amtW0gxWH  
GBJ3RcfXtoQvedOXy18rcnOUBPlwj6k"
```

```
}
```

**Ejemplo de parámetros de salida fallida:**

```
{'estatus': 404, 'error': 'Falló la autenticación',  
'mensaje': 'Error, correo o contraseña no existen'
```

```
}
```

```
{
```

```
"estatus": 400,  
'error': 'Falló la autenticación',  
"mensaje": "el usuario no esta activo"
```

```
}
```

```
{
```

```
"estatus": 400, "error": "error servidor", "mensaje": "Error al interpretar la solicitud"
```

```
}
```

<b>ID:</b> 05	<b>Nombre:</b> Ver Perfil	
<b>Prioridad:</b> Media	<b>Historia de usuario</b> Es importante que el usuario pueda ver la información que fue ingresada para ser revisada por él.	
<b>Estimado:</b> 2 puntos		
<b>Módulo:</b> Usuario		
<b>Criterios de Aceptación</b>  Se solicitará la información del usuario logueado a la BD, y se mostrará la información que sea necesaria para el usuario.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/profile/<cui> <b>Método:</b> GET <b>Formato de entrada:</b> JSON  <b>parametro:</b>		
Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	application/json
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción

**Formato de salida:** JSON**Código de respuesta exitosa:** HTTP 200**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta
data	objeto	Objeto que contiene la información del usuario
mensaje	cadena	mensaje devuelto por el servidor

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	no existe usuario
300	Token ha expirado

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta

error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error
correcto	booleano	si ocurre error en la autorización

#### **Ejemplos de parámetros de entrada:**

/user/profile/2168095170407

#### **Ejemplos de parámetros de salida exitosa:**

```
{
  "data": {
    "apellidos": "Chocoj Xinico",
    "cui": 2168095170407,
    "estado": 1,
    "fecha_nacimiento": "Wed, 25 Nov 2020 00:00:00 GMT",
    "nombres": "Rafael Angel",
    "numero_grupo": 914,
    "seccion_grupo": "N",
    "tipo": "E"
  },
  "estatus": 200,
  "mensaje": "correcto"
}
```

#### **Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400,
  "error": "Error al hacer la petición",
  "mensaje": "Se produjo un error al realizar la petición"
}
```

```
{
  'estatus': 404, 'error': 'cuenta',
  'mensaje': 'no existe usuario'
}
```

```
{
  'correcto': False, 'estatus': 401,
  'error': 'Usuario no autenticado',
  'mensaje': 'Error, usuario no autorizado'
}
```

```
{
  'correcto': False, 'estatus': 300,
  'error': 'Token ha expirado, iniciar sesión de nuevo', 'mensaje': 'Error, con el
```

```
token'
}

{
  'correcto': False, "estatus": 400,   'error': 'Token invalidó, iniciar sesión de
nuevo',
  'mensaje': 'Error, con el token'
}
```

<b>ID:</b> 06	<b>Nombre:</b> Verificar Token										
<b>Prioridad:</b> Media	<b>Historia de usuario</b> Es importante que el usuario pueda ser autorizado para realizar acciones requeridas solo para usuarios autenticados.										
<b>Estimado:</b> 2 puntos											
<b>Módulo:</b> Usuario											
<b>Criterios de Aceptación</b>  Se solicitará la información del usuario logueado verificando el token que exista y que no esté vencido  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/verificartoken <b>Método:</b> GET <b>Formato de entrada:</b> JSON  <b>parametro:</b>											
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td> </td><td> </td><td> </td></tr></table>			Atributo	Tipo	Descripción						
Atributo	Tipo	Descripción									
<b>Header:</b>											
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Content-Type</td><td>header</td><td>application/json</td></tr><tr><td>Authorization</td><td>header</td><td>Bearer &lt;token&gt;</td></tr></table>			Atributo	Tipo	Descripción	Content-Type	header	application/json	Authorization	header	Bearer <token>
Atributo	Tipo	Descripción									
Content-Type	header	application/json									
Authorization	header	Bearer <token>									
<b>Body:</b>											

Atributo	Tipo	Descripción

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta
data	objeto	Objeto que contiene la información del usuario decodificada
mensaje	cadena	mensaje devuelto por el servidor
correcto	booleano	si el token es correcto devuelve True, de lo contrario devuelve False

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	no existe usuario
300	Token ha expirado

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

correcto	booleano	si ocurre error en la autorización
----------	----------	------------------------------------

**Ejemplos de parámetros de entrada:**

/user/profile/2168095170407

**Ejemplos de parámetros de salida exitosa:**

```
{
  "data": {
    "apellidos": "Chocoj Xinico",
    "cui": 2168095170407,
    "estado": 1,
    "fecha_nacimiento": "Wed, 25 Nov 2020 00:00:00 GMT",
    "nombres": "Rafael Angel",
    "numero_grupo": 914,
    "seccion_grupo": "N",
    "tipo": "E"
  },
  "estatus": 200,
  "mensaje": "correcto"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400,
  "error": "Error al hacer la petición",
  "mensaje": "Se produjo un error al realizar la petición"
}

{
  'estatus': 404, 'error': 'cuenta',
  'mensaje': 'no existe usuario'
}

{
  'correcto': False, 'estatus': 401,
  'error': 'Usuario no autenticado',
  'mensaje': 'Error, usuario no autorizado'
}

{
  'correcto': False, 'estatus': 300,
  'error': 'Token ha expirado, iniciar sesión de nuevo', 'mensaje': 'Error, con el token'
}

{
  'correcto': False, 'estatus': 400, 'error': 'Token invalidó, iniciar sesión de
```

```
nuevo',  
    'mensaje': 'Error, con el token'  
}
```

ID: 01	Nombre: Registrar usuario																						
Prioridad: Alta	Historia de usuario Se quiere poder almacenar la información del usuario para la creación de su perfil.																						
Estimado: 8 puntos																							
Módulo: Usuario																							
Criterios de Aceptación																							
El usuario ingresará la información requerida para crear su perfil a través de un formulario.																							
El microservicio debe tener la siguiente configuración:																							
Ruta: /user/signin																							
Método: POST																							
Formato de entrada: application/json																							
Header:																							
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Content-Type</td><td>header</td><td>application/json</td></tr></table>			Atributo	Tipo	Descripción	Content-Type	header	application/json															
Atributo	Tipo	Descripción																					
Content-Type	header	application/json																					
Body:																							
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>cui</td><td>BigInt</td><td>cui del usuario</td></tr><tr><td>nombres</td><td>cadena</td><td>nombre del usuario</td></tr><tr><td>apellidos</td><td>cadena</td><td>apellido del usuario</td></tr><tr><td>correo</td><td>cadena</td><td>correo electrónico del usuario</td></tr><tr><td>contrasena</td><td>cadena</td><td>contraseña para el usuario Atributo requerido</td></tr><tr><td>fecha_nacimiento</td><td>cadena</td><td>fecha de nacimiento del usuario,</td></tr></table>			Atributo	Tipo	Descripción	cui	BigInt	cui del usuario	nombres	cadena	nombre del usuario	apellidos	cadena	apellido del usuario	correo	cadena	correo electrónico del usuario	contrasena	cadena	contraseña para el usuario Atributo requerido	fecha_nacimiento	cadena	fecha de nacimiento del usuario,
Atributo	Tipo	Descripción																					
cui	BigInt	cui del usuario																					
nombres	cadena	nombre del usuario																					
apellidos	cadena	apellido del usuario																					
correo	cadena	correo electrónico del usuario																					
contrasena	cadena	contraseña para el usuario Atributo requerido																					
fecha_nacimiento	cadena	fecha de nacimiento del usuario,																					



		el formato a utilizar será el siguiente: "YYYY-mm-dd"
tipo	cadena	confirmación de la contraseña <ul style="list-style-type: none"> <li>- "C" = Usuario cliente</li> <li>- "E" = Usuario empresarial</li> <li>- "A" = Usuario administrativo</li> </ul> <b>Atributo requerido</b>
numero_grupo	entero	Es el número de grupo que representa al cliente empresarial.  Atributo requerido si el tipo = "E"
seccion_grupo	cadena	Es la sección del grupo que representa al cliente empresarial  Atributo requerido si el tipo = "E"

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información del registro del usuario

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
404	Recurso no encontrado

500	Error interno del servidor
501	Si es usuario empresarial debe de ingresar numero de grupo o seccion

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallida
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

```
{
  "cui": 2168095170407,
  "nombres": "Rafael Angel",
  "apellidos": "Chocoj Xinico",
  "contrasena": "123",
  "correo": "rafaelcliveguitar@gmail.com",
  "fecha_nacimiento": "2020-11-25",
  "tipo": "E",
  "numero_grupo": 914,
  "seccion_grupo": "N"
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Se ha registrado el usuario exitosamente"
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 401,
  "mensaje": "Si es usuario empresarial debe de ingresar numero de grupo o seccion"
}
{
  "estatus": 500, "error": "Error interno del servidor",
```

<pre> "mensaje": "Error, ya existe usuario con el mismo cui. " }  {   "estatus": 400, "error": "error", "mensaje": "error al enviar correo" }  {   "estatus": 400, "error": "error server", "mensaje": "Error al interpretar la solicitud" }  <b>Servicio de usuario consume envio de correo:</b> /notification/sendEmail {   "para": "rafael@gmail.com",   "asunto": "CONFIRMAR CUENTA",   "mensaje": "bienvenido ",   "plantilla": "confirmEmail",   "data": {"code": "enlace"} } </pre>
--

<b>ID:</b> 03	<b>Nombre:</b> Verificar correo				
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Es importante verificar que la dirección de correo electrónico exista y el usuario tenga acceso a ella.				
<b>Estimado:</b> 1 punto					
<b>Módulo:</b> Usuario					
<b>Criterios de Aceptación</b>  Se deberá ingresar al link que fue enviado por correo, para verificar que este exista correo.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/verifyemail <b>Método:</b> POST <b>Formato de entrada:</b> JSON  <b>Header:</b>  <table><tr><td><b>Atributo</b></td><td><b>Tipo</b></td><td><b>Descripción</b></td></tr></table>			<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>			

Content-Type	header	application/json
--------------	--------	------------------

**Body:**

Atributo	Tipo	Descripción
token	cadena	cui del usuario encriptado para validar ya activar cuenta

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información de la verificación del correo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
404	no existe usuario
500	JWT ya expiro

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

### Ejemplos de parámetros de salida exitosa:

### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 400, "error": "error servidor", "mensaje": "Error al interpretar la solicitud"
}
```

<b>ID:</b> 04	<b>Nombre:</b> Iniciar sesión
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Es importante tener un medio por el cual los usuarios puedan ingresar a la plataforma de manera segura.
<b>Estimado:</b> 2 punto	
<b>Módulo:</b> Usuario	
<b>Criterios de Aceptación</b> Al validar las credenciales se deberá, mediante al estándar JWT, crear un token	

válido que será devuelto para que el usuario esté autenticado.

El microservicio debe tener la siguiente configuración:

**Ruta:** /user/login

**Método:** POST

**Formato de entrada:** JSON

**Payload:**

Atributo	Tipo	Descripción
cui	entero	el número que identifica al usuario en el sistema.
id_grupo	entero	el grupo si el usuario es empresarial
seccion	cadena	seccion si el usuario es empresarial

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	application/json

**Body:**

Atributo	Tipo	Descripción
correo	cadena	correo electrónico del usuario
contrasena	cadena	contraseña del usuario

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación

mensaje	cadena	muestra información del inicio de sesión
token	cadena	token válido para la autenticación

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
404	correo o contraseña no existen

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error
id_grupo	entero	no del grupo si es empresarial
seccion	cadena	seccion del grupo si es usuario empresarial
data	array	datos del usuario
token	cadena	datos codificados del usuario

#### Ejemplos de parámetros de entrada:

```
{
  "contrasena": "123",
  "correo": "rafaelcliveguitar@gmail.com"
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{
  "data": {
    "apellidos": "Chocoj Xinico",
    "cui": 2168095170407,
    "estado": 1,
    "fecha_nacimiento": "Wed, 25 Nov 2020 00:00:00 GMT",
  }
}
```

**Ejemplo de parámetros de salida fallida:**

```
{'estatus': 404, 'error': 'Falló la autenticación',  
'mensaje': 'Error, correo o contraseña no existen'  
}  
  
{  
  "estatus": 400,  
  'error': 'Falló la autenticación',  
  "mensaje": "el usuario no esta activo"  
}  
  
{  
  "estatus": 400, "error": "error servidor", "mensaje": "Error al interpretar la solicitud"  
}
```

<b>ID:</b> 05	<b>Nombre:</b> Ver Perfil
<b>Prioridad:</b> Media	<b>Historia de usuario</b> Es importante que el usuario pueda ver la información que fue ingresada para ser revisada por él.
<b>Estimado:</b> 2 puntos	
<b>Módulo:</b> Usuario	
<b>Criterios de Aceptación</b>  Se solicitará la información del usuario logueado a la BD, y se mostrará la información que sea necesaria para el usuario.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/profile/<cui>	



**Método:** GET

**Formato de entrada:** JSON

**parametro:**

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	application/json
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta
data	objeto	Objeto que contiene la información del usuario
mensaje	cadena	mensaje devuelto por el servidor

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud

401	Usuario no autenticado
404	no existe usuario
300	Token ha expirado

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error
correcto	booleano	si ocurre error en la autorización

#### Ejemplos de parámetros de entrada:

/user/profile/2168095170407

#### Ejemplos de parámetros de salida exitosa:

```
{
  "data": {
    "apellidos": "Chocoj Xinico",
    "cui": 2168095170407,
    "estado": 1,
    "fecha_nacimiento": "Wed, 25 Nov 2020 00:00:00 GMT",
    "nombres": "Rafael Angel",
    "numero_grupo": 914,
    "seccion_grupo": "N",
    "tipo": "E"
  },
  "estatus": 200,
  "mensaje": "correcto"
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 400,
  "error": "Error al hacer la petición",
  "mensaje": "Se produjo un error al realizar la petición"
}
{
  'estatus': 404, 'error': 'cuenta',
```

```

'mensaje': 'no existe usuario'
}

{
  'correcto': False,'estatus': 401,
  'error': 'Usuario no autenticado',
  'mensaje': 'Error, usuario no autorizado'
}

{
  'correcto': False, "estatus": 300,
  "error": 'Token ha expirado, iniciar sesión de nuevo', 'mensaje': 'Error, con el
token'
}

{
  'correcto': False, "estatus": 400,   'error': 'Token invalidó, iniciar sesión de
nuevo',          'mensaje': 'Error, con el token'
}

```

<b>ID:</b> 06	<b>Nombre:</b> Verificar Token	
<b>Prioridad:</b> Media	<b>Historia de usuario</b> Es importante que el usuario pueda ser autorizado para realizar acciones requeridas solo para usuarios autenticados.	
<b>Estimado:</b> 2 puntos		
<b>Módulo:</b> Usuario		
<b>Criterios de Aceptación</b>  Se solicitará la información del usuario logueado verificando el token que exista y que no esté vencido  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /user/verificartoken <b>Método:</b> GET <b>Formato de entrada:</b> JSON  <b>parametro:</b>		
Atributo	Tipo	Descripción

**Header:**

Atributo	Tipo	Descripción
Content-Type	header	application/json
Authorization	header	Bearer <token>

**Body:**

Atributo	Tipo	Descripción

**Formato de salida:** JSON**Código de respuesta exitosa:** HTTP 200**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta
data	objeto	Objeto que contiene la información del usuario decodificada
mensaje	cadena	mensaje devuelto por el servidor
correcto	booleano	si el token es correcto devuelve True, de lo contrario devuelve False

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	no existe usuario
300	Token ha expirado

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error
correcto	booleano	si ocurre error en la autorización

**Ejemplos de parámetros de entrada:**

/user/profile/2168095170407

**Ejemplos de parámetros de salida exitosa:**

```
{
  "data": {
    "apellidos": "Chocoj Xinico",
    "cui": 2168095170407,
    "estado": 1,
    "fecha_nacimiento": "Wed, 25 Nov 2020 00:00:00 GMT",
    "nombres": "Rafael Angel",
    "numero_grupo": 914,
    "seccion_grupo": "N",
    "tipo": "E"
  },
  "estatus": 200,
  "mensaje": "correcto"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400,
  "error": "Error al hacer la petición",
  "mensaje": "Se produjo un error al realizar la petición"
}
{
  'estatus': 404, 'error': 'cuenta',
  'mensaje': 'no existe usuario'
}
{
  'correcto': False, 'estatus': 401,
  'error': 'Usuario no autenticado',
  'mensaje': 'Error, usuario no autorizado'
}
```

```

}

{
  'correcto': False, "estatus": 300,
  "error": 'Token ha expirado, iniciar sesión de nuevo', 'mensaje': 'Error, con el token'
}

{
  'correcto': False, "estatus": 400,   'error': 'Token invalidó, iniciar sesión de nuevo',
                                     'mensaje': 'Error, con el token'
}

```

## Inventario

ID: 19	Nombre: Agregar Producto													
Prioridad: Baja	Historia de usuario Como usuario quiero poder consultar y visualizar todos los productos que se tiene en inventario.													
Estimado: 3 puntos														
Módulo: Inventario														
<b>Criterios de Aceptación</b> El microservicio para poder agregar nuevo producto en el inventario  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /inventario/addproducto <b>Método:</b> POST <b>Formato de entrada:</b>  <b>Header:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Content-Type</td><td>header</td><td>multipart/form-data</td></tr></table> <b>Body:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>nombre</td><td>cadena</td><td>nombre del producto</td></tr></table>			Atributo	Tipo	Descripción	Content-Type	header	multipart/form-data	Atributo	Tipo	Descripción	nombre	cadena	nombre del producto
Atributo	Tipo	Descripción												
Content-Type	header	multipart/form-data												
Atributo	Tipo	Descripción												
nombre	cadena	nombre del producto												

descripcion	cadena	descripcion del producto
precio_costo	decimal	costo del producto
archivo	file	archivo de la foto
unidades_disponibles	entero	stock del producto
precio_unitario	decimal	precio del producto
id_categoria	entero	categoría del producto
sku	entero	código del producto a agregar

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo
data	arreglo	listado de productos

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
mensaje	cadena	muestra el nombre del error
error	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

```
nombre = StringField('nombres', [DataRequired('Este campo es requerido.')])
descripcion = StringField('descripcion', [DataRequired('Este campo es
requerido.')])
precio_costo = DecimalField('precio_costo',
validators=[validators.DataRequired("Este campo es requerido.")])
archivo = MultipleFileField('archivo')
unidades_disponibles =
IntegerField('sueldo',validators=[validators.DataRequired("Este campo es
requerido.")])
precio_unitario = DecimalField('precio_unitario',
validators=[validators.DataRequired("Este campo es requerido.")])
id_categoria =
IntegerField('id_categoria',validators=[validators.DataRequired("Este campo es
requerido.")])
sku = IntegerField('sku',validators=[validators.DataRequired("Este campo es
requerido.")])
```

**Ejemplos de parámetros de salida exitosa:**

```
{
"estatus": 401,
"mensaje": "Si es usuario empresarial debe de ingresar numero de grupo o seccion"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{'estatus': 400, 'error':'error', 'mensaje': 'error al enviar correo'}, 400
{
"estatus": 200,
"mensaje": "Se ha registrado el usuario exitosamente"
}
```

<b>ID:</b> 19	<b>Nombre:</b> Consultar productos
<b>Prioridad:</b> Baja	<b>Historia de usuario</b> Como usuario quiero poder consultar y visualizar todos los productos que se tiene en inventario.
<b>Estimado:</b> 3 puntos	
<b>Módulo:</b> Inventario	
<b>Criterios de Aceptación</b> El microservicio por medio de una petición de tipo get, solicita la información de todos los productos en inventario con código 200, en caso de algún error retornará un código de error y mas información del mismo.  El microservicio debe tener la siguiente configuración:	



**Ruta:** /inventario/getproductos

**Método:** GET

**Formato de entrada:**

**Header:**

Atributo	Tipo	Descripción

**Body:**

Atributo	Tipo	Descripción

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo
data	arreglo	listado de productos

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta

mensaje	cadena	muestra el nombre del error
error	cadena	muestra más información del error

### Ejemplos de parámetros de entrada:

### Ejemplos de parámetros de salida exitosa:

```
{
  "data": [
    {
      "descripcion": "Donas con azucar glass",
      "id_categoria": 1,
      "imagen": null,
      "nombre": "Donas bimbo",
      "precio_costo": 5.85,
      "precio_unitario": 10.0,
      "sku": 1,
      "unidades_disponibles": 40
    },
    {
      "descripcion": "Pan sabor vainilla relleno de chocolate y cubierto de chocolate",
      "id_categoria": 1,
      "imagen": null,
      "nombre": "Negrito",
      "precio_costo": 7.85,
      "precio_unitario": 15.0,
      "sku": 2,
      "unidades_disponibles": 50
    },
    {
      "descripcion": "Nachos Diana",
      "id_categoria": 1,
      "imagen": "https://www.guatemala.com/fotos/2021/08/Nachos-Diana.jpg",
      "nombre": "Nachos Diana",
      "precio_costo": 3.0,
      "precio_unitario": 8.0,
      "sku": 5000,
      "unidades_disponibles": 80
    },
    {
      "descripcion": "Elotitos Diana",
      "id_categoria": 1,
      "imagen": "https://diana.sv/wp-content/uploads/2020/07/Elotitos-Barbacoa.png",

```

```
    "nombre": "Elotitos Diana",
    "precio_costo": 11.45,
    "precio_unitario": 17.0,
    "sku": 1000,
    "unidades_disponibles": 151
  },
  {
    "descripcion": "Tocinitos Diana",
    "id_categoria": 1,
    "imagen":
"https://diana.sv/wp-content/uploads/2020/07/Tocinitos-1-225x300.png",
    "nombre": "Tocinitos Diana",
    "precio_costo": 15.85,
    "precio_unitario": 20.0,
    "sku": 2000,
    "unidades_disponibles": 78
  },
  {
    "descripcion": "pizza de jamon",
    "id_categoria": 1,
    "imagen":
"http://res.cloudinary.com/dm4pe5xwn/image/upload/v1682559624/SA_proyecto/x
nyaqeg0rrojspe0xakn.png",
    "nombre": "pizza jamon",
    "precio_costo": 100.0,
    "precio_unitario": 125.0,
    "sku": 10,
    "unidades_disponibles": 9
  },
  {
    "descripcion": "Cereza Diana",
    "id_categoria": 1,
    "imagen": "https://diana.co.cr/wp-content/uploads/2020/07/Cereza-1.png",
    "nombre": "Cereza Diana",
    "precio_costo": 4.85,
    "precio_unitario": 10.0,
    "sku": 3000,
    "unidades_disponibles": 560
  },
  {
    "descripcion": "Gomitas Diana",
    "id_categoria": 1,
    "imagen": "https://diana.sv/wp-content/uploads/2020/07/Gomitas-1.png",
    "nombre": "Gomitas Diana",
    "precio_costo": 6.6,
    "precio_unitario": 13.5,
```

```

        "sku": 4000,
        "unidades_disponibles": 550
    }
],
"estatus": 200,
"mensaje": "datos obtenidos"
}

```

**Ejemplo de parámetros de salida fallida:**

```

{
  "estatus":400,
  "mensaje": "Error al interpretar la solicitud"
  "error": "error interno servidor"
}

```

ID: 21	Nombre: Consultar stock	
Prioridad: Baja	Historia de usuario Como usuario quiero poder consultar el stock de un producto en específico en tiempo real..	
Estimado: 3 puntos		
Módulo: Inventario		
<b>Criterios de Aceptación</b> El microservicio retorna el stock de un producto en específico del inventario y un código 200, en caso de algún error retornará un código de error y mas información del mismo.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /inventario/getstock/<sku> <b>Método:</b> GET <b>Formato de entrada:</b>  <b>Header:</b>		
Atributo	Tipo	Descripción
<b>Variables en el URL:</b>		
Atributo	Tipo	Descripción

sku	entero	Identificador del producto
-----	--------	----------------------------

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo
unidades_disponibles	numero	stock del producto

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
mensaje	cadena	muestra el nombre del error
error	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:****Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "datos obtenidos",  
  "unidades_disponibles": 9  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 400,  
  "mensaje": "Error al interpretar la solicitud"  
  "error": "error interno"  
}
```

<b>ID:</b> 191	<b>Nombre:</b> Consultar categorías	
<b>Prioridad:</b> Baja	<b>Historia de usuario</b> Como usuario quiero poder consultar y visualizar todos las categorías	
<b>Estimado:</b> 3 puntos		
<b>Módulo:</b> Inventario		
<b>Criterios de Aceptación</b> El microservicio por medio de una petición de tipo get, solicita la información de todos las categorías  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /inventario/getcategorias <b>Método:</b> GET <b>Formato de entrada:</b>  <b>Header:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Body:</b>		

Atributo	Tipo	Descripción

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo
data	arreglo	listado de productos

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
mensaje	cadena	muestra el nombre del error
error	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

**Ejemplos de parámetros de salida exitosa:**

```
{
  "data": [
    {
      "id_categoria": 1,
      "nombre": "Alimentos"
    }
  ]
}
```

```
],  
"estatus": 200,  
"mensaje": "datos obtenidos"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus":400,  
  "mensaje": "Error al interpretar la solicitud"  
  "error": "error interno servidor"  
}
```

<b>ID:</b> 21	<b>Nombre:</b> Consultar stock							
<b>Prioridad:</b> Baja	<b>Historia de usuario</b> Como usuario quiero poder consultar el stock de un producto en específico en tiempo real..							
<b>Estimado:</b> 3 puntos								
<b>Módulo:</b> Inventario								
<b>Criterios de Aceptación</b> El microservicio retorna el stock de un producto en específico del inventario y un código 200, en caso de algún error retornará un código de error y mas información del mismo.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /inventario/getstock/<sku> <b>Método:</b> GET <b>Formato de entrada:</b>  <b>Header:</b>								
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td> </td><td> </td><td> </td></tr></table>			Atributo	Tipo	Descripción			
Atributo	Tipo	Descripción						
<b>Variables en el URL:</b>								
<table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>sku</td><td>entero</td><td>Identificador</td></tr></table>			Atributo	Tipo	Descripción	sku	entero	Identificador
Atributo	Tipo	Descripción						
sku	entero	Identificador						



		del producto
--	--	--------------

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo
unidades_disponibles	numero	stock del producto

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
mensaje	cadena	muestra el nombre del error
error	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:****Ejemplos de parámetros de salida exitosa:**

```
{
  "estatus": 200,
  "mensaje": "datos obtenidos",
  "unidades_disponibles": 9
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400,
  "mensaje": "Error al interpretar la solicitud"
  "error": "error interno"
}
```

ID: 199	Nombre: Actualizar stock	
Prioridad: Baja	Historia de usuario Como usuario quiero poder actualizar stock para tener un control de inventario	
Estimado: 3 puntos		
Módulo: Inventario		
<b>Criterios de Aceptación</b> El microservicio para poder actualizar stock  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /inventario/actualizarstock <b>Método:</b> PUT <b>Formato de entrada:</b>  <b>Header:</b>		
Atributo	Tipo	Descripción
Content-Type	header	application/json
<b>Body:</b>		
Atributo	Tipo	Descripción

tipo	string	“-” para restar, “+” para sumar
unidades	entero	unidades que se va actualizar
sku	entero	codigo del producto a agregar

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
mensaje	cadena	muestra el nombre del error
error	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

```
{
  "sku": 10,
  "unidades": 2,
  "tipo": "+"
}
```

```
"sku": 10,  
"unidades": 3,  
"tipo": "-"  
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "Se ha actualizaron las unidades correctamente"  
}  
{  
  "estatus": 200,  
  "mensaje": "Se ha actualizaron las unidades correctamente"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{'estatus': 400, 'error': 'error', 'mensaje': 'error al enviar correo'}, 400
```

<b>ID:</b> 18	<b>Nombre:</b> Subir objetos al bucket	
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Como institución quiero poder almacenar objetos estáticos necesarios para el aplicativo y también para los usuarios.	
<b>Estimado:</b> 3 puntos		
<b>Módulo:</b> Almacenamiento de objetos		
<b>Criterios de Aceptación</b> El servicio debe conectarse a un servicio de almacenamientos de objetos, retornar un código 200, un mensaje satisfactorio y la ruta del objeto donde fue almacenado, caso contrario retorna un código de error y descripción del mismo.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /bucket/upload <b>Método:</b> POST <b>Formato de entrada:</b> form-data  <b>Header:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

Content-Type	header	multipart/form-data
--------------	--------	---------------------

**Body:**

Atributo	Tipo	Descripción
archivo	archivo o arreglo de archivos	Archivos a almacenar. <b>Atributo requerido.</b>
directorio	texto	Directorio donde se almacenará el o los archivos. <b>Atributo opcional.</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	estatus del envío de correo electrónico
mensaje	cadena	mensaje descriptivo
data	cadena o arreglo de cadena	Direcciones URL públicas del archivo o archivos subidos.

**Código de respuesta fallida:**

Código	Descripción
400	Fallo en el proceso
413	Error por exceder el límite de peso de los archivos adjuntos
500	error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	Indica el código de respuesta
error	cadena	muestra más información del

		error
mensaje	cadena	muestra el nombre del error

#### Ejemplos de parámetros de entrada:

clave	valor
archivo	imagen.png

clave	valor
archivo	[imagen.png, imagen2.jpg]

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Objeto almacenado",
  "data": "https://bucket.com/archivos/imagen.png"
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "error": "No hay archivos por subir",
  "estatus": 400,
  "mensaje": "Archivo es requerido"
}
{
  "error": "413 Request Entity Too Large: The data value transmitted exceeds the capacity limit.",
  "estatus": 413,
  "mensaje": "Error al almacenar el objeto"
}
```

## BILLETERA

<b>ID:</b> 02	<b>Nombre:</b> Recargar billetera
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se desea poder recargar el monto de una billetera virtual para tener saldo y evitar usar la tarjeta de credito para realizar compras.
<b>Estimado:</b> 5 puntos	

**Módulo:** Pagos**Criterios de Aceptación**

Se generará una notificación al correo del usuario indicando el monto de recarga en la tarjeta y el mismo monto debitado de la tarjeta de crédito.

El microservicio debe tener la siguiente configuración:

**Ruta:** /payments/rechargetwallet

**Método:** POST

**Formato de entrada:** JSON

**Header:**

Atributo	Tipo	Descripción
Authorization	header	Bearer <token>
Content-Type	header	application/json

**Variables en el URL:**

Atributo	Tipo	Descripción
idBilletera	entero	el número de identificación de la billetera del usuario. <b>Atributo requerido</b>
monto	entero	La cantidad a recargar en la billetera: 1. 50 2. 100 3. 250 4. 500 <b>Atributo requerido</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de

		aceptación
mensaje	cadena	mensaje descriptivo

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

/payments/rechargeWallet

```
{
  "monto":50,
  "idBilletera":3
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Recarga de billetera exitosa. "
}
```



**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400
  "error": "Parámetro requerido",
  "mensaje": "Faltó algún parámetro"
}
```

```
{
  "estatus": 404
  "error": "Tarjeta sin fondos,
  "mensaje": "No se pudo hacer la recarga del monto de la billetera por que no hay
suficientes fondos en al tarjeta"
}
```

<b>ID:</b> 03	<b>Nombre:</b> Crear billetera										
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se desea poder recargar el monto de una billetera virtual para tener saldo y evitar usar la tarjeta de credito para realizar compras.										
<b>Estimado:</b> 5 puntos											
<b>Módulo:</b> Pagos											
<b>Criterios de Aceptación</b>  Se generará una notificación al correo del usuario indicando el monto de recarga en la tarjeta y el mismo monto debitado de la tarjeta de crédito.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> payments/createWallet <b>Método:</b> POST <b>Formato de entrada:</b> JSON  <b>Header:</b> <table><tr><th>Atributo</th><th>Tipo</th><th>Descripción</th></tr><tr><td>Authorization</td><td>header</td><td>Bearer &lt;token&gt;</td></tr><tr><td>Content-Type</td><td>header</td><td>application/json</td></tr></table>			Atributo	Tipo	Descripción	Authorization	header	Bearer <token>	Content-Type	header	application/json
Atributo	Tipo	Descripción									
Authorization	header	Bearer <token>									
Content-Type	header	application/json									
<b>Variables en el URL:</b>											

Atributo	Tipo	Descripción
cui	entero	el número de identificación del usuario. <b>Atributo requerido</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	mensaje descriptivo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

```
{  
  "cui":2994902460101  
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "data": [  
    {  
      "idBilletera": 3,  
      "idUsuario": 2994902460101,  
      "monto": 0  
    }  
  ],  
  "estatus": 200,  
  "mensaje": "Billetera creada con éxito"  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 400  
  "error": "Parámetro requerido",  
  "mensaje": "Faltó algún parámetro"  
}
```

<b>ID:</b> 03	<b>Nombre:</b> Obtener billetera
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se desea poder recargar el monto de una billetera virtual para tener saldo y evitar usar la tarjeta de credito para realizar compras.
<b>Estimado:</b> 5 puntos	
<b>Módulo:</b> Pagos	
<b>Criterios de Aceptación</b>  Se generará una notificación al correo del usuario indicando el monto de recarga en la tarjeta y el mismo monto debitado de la tarjeta de crédito.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /payments/getwallet <b>Método:</b> POST <b>Formato de entrada:</b> JSON  <b>Header:</b>	

Atributo	Tipo	Descripción
Authorization	header	Bearer <token>
Content-Type	header	application/json

#### Variables en el URL:

Atributo	Tipo	Descripción
cui	entero	el número de identificación del usuario. <b>Atributo requerido</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	mensaje descriptivo

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

#### Body de Salida:

Atributo	Tipo	Descripción
----------	------	-------------

estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

```
{
  "cui":2994902460101
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{
  "data": [
    {
      "idBilletera": 3,
      "idUsuario": 2994902460101,
      "monto": 0
    }
  ],
  "estatus": 200,
  "mensaje": "Billetera encontrada"
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400
  "error": "Parámetro requerido",
  "mensaje": "Faltó algún parámetro"
}
```

## PAGOS

<b>ID:</b> 04	<b>Nombre:</b> Registrar pago
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se desea poder hacer el pago correspondiente a la compra, poder pagar con la billetera virtual o con la tarjeta de credito.
<b>Estimado:</b> 5 puntos	
<b>Módulo:</b> Pagos	
<b>Criterios de Aceptación</b>	

El sistema debe permitir al usuario realizar el pago correspondiente a su compra de manera efectiva. El usuario debe tener la opción de pagar con su billetera virtual. El usuario debe tener la opción de pagar con su tarjeta de crédito. El proceso de pago debe ser seguro y confiable para proteger la información del usuario y su dinero.

El microservicio debe tener la siguiente configuración:

**Ruta:** /payments/recordpayment

**Método:** POST

**Formato de entrada:** JSON

**Header:**

Atributo	Tipo	Descripción
Authorization	header	Bearer <token>
Content-Type	header	application/json

**Variables en el URL:**

Atributo	Tipo	Descripción
montowallet	double	La cantidad a pagar con la billetera virtual <b>Atributo requerido</b>
id_usuario	entero	Identificación del usuario que esta realizando la compra. <b>Atributo requerido</b>
id_billetera	entero	Id de la billetera del usuario <b>Atributo requerido</b>
id_tarjeta	entero	Id de la tarjeta de credito del usuario. <b>Atributo requerido</b>
montotarjeta	double	La cantidad a pagar con la tarjeta <b>Atributo requerido</b>
tipopago	entero	tipo de pago que se realiza: 0: tarjeta, 1: wallet , 2: mixto

tipopago		<b>Atributo requerido</b>
----------	--	---------------------------

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	mensaje descriptivo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

```
{
  "montowallet":60,
  "id_usuario":2994902460109,
  "id_billetera":4,
  "id_tarjeta":3,
  "montotarjeta":20,
  "tipopago":2
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{
  "estatus": 200,
  "mensaje": "Pago registrado exitosamente. "
}
```

**Ejemplo de parámetros de salida fallida:**

```
{
  "estatus": 400
  "error": "Parámetro requerido",
  "mensaje": "Faltó algún parámetro"
}
```

<b>ID:</b> 04	<b>Nombre:</b> Aceptar pago
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se desea poder validar el pago correspondiente a la compra, poder pagar con la billetera virtual o con la tarjeta de credito.
<b>Estimado:</b> 5 puntos	
<b>Módulo:</b> Pagos	
<b>Criterios de Aceptación</b> El sistema debe permitir al usuario aceptar el pago correspondiente a su compra de manera efectiva  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /payments/recordpayment <b>Método:</b> POST <b>Formato de entrada:</b> JSON	



**Header:**

Atributo	Tipo	Descripción
Authorization	header	Bearer <token>
Content-Type	header	application/json

**Variables en el URL:**

Atributo	Tipo	Descripción
id_usuario	entero	Identificación del usuario que esta realizando la compra. <b>Atributo requerido</b>
total	float	total a cancelar <b>Atributo requerido</b>

**Formato de salida:** JSON**Código de respuesta exitosa:** HTTP 200**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	mensaje descriptivo

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

```
{
  "id_usuario":2994902460109,
  "total":40
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{
  "estatus": 200,
  "mensaje": "Pago registrado exitosamente. "
}
```

#### Ejemplo de parámetros de salida fallida:

```
{
  "estatus": 400
  "error": "Parámetro requerido",
  "mensaje": "Faltó algún parámetro"
}
```

## VENTAS

**ID:** 04

**Nombre:** Obtener ventas



Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

```
{
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{
  "data": [
    {
      "estado": "P",
      "id_usuario": 2168095170407,
      "nombre": "Rafael Angel",
      "nombre_ruta": "direccion1",
      "productos": [
        {
          "cantidad": 2,
          "id_detalle_carrito": 14,
          "id_producto": 1,
          "id_ruta": 1,

```

```
        "nombre": "Donas bimbo",
        "precio_total": 20
    },
    {
        "cantidad": 2,
        "id_detalle_carrito": 15,
        "id_producto": 1,
        "id_ruta": 1,
        "nombre": "Donas bimbo",
        "precio_total": 20
    },
    {
        "cantidad": 2,
        "id_detalle_carrito": 16,
        "id_producto": 1,
        "id_ruta": 1,
        "nombre": "Donas bimbo",
        "precio_total": 20
    }
],
"total": 60
},
{
    "estado": "P",
    "id_usuario": 2994902460101,
    "nombre": "Mynor",
    "nombre_ruta": "direccion1",
    "productos": [
        {
            "cantidad": 2,
            "id_detalle_carrito": 19,
            "id_producto": 1,
            "id_ruta": 1,
            "nombre": "Donas bimbo",
            "precio_total": 20
        },
        {
            "cantidad": 2,
            "id_detalle_carrito": 20,
            "id_producto": 1,
            "id_ruta": 1,
            "nombre": "Donas bimbo",
            "precio_total": 20
        },
        {
            "cantidad": 2,
```

```

        "id_detalle_carrito": 21,
        "id_producto": 1,
        "id_ruta": null,
        "nombre": "Donas bimbo",
        "precio_total": 20
    }
],
"total": 60
}
],
"estatus": 200,
"mensaje": "Ventas encontradas"
}

```

**Ejemplo de parámetros de salida fallida:**

```

{
  "estatus": 500
  "error": "Error al acceder a la base de datos",
  "mensaje": "Registro no existente"
}

```

<b>ID:</b> 04	<b>Nombre:</b> Aprobar venta	
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se desea poder aprobar la venta, pasar del estado pendiente a un estado aprobado de todas las ventas pendientes de autorizar.	
<b>Estimado:</b> 5 puntos		
<b>Módulo:</b> Pagos		
<b>Criterios de Aceptación</b> El sistema debe permitir al usuario aceptar y aprobar las ventas que se encuentran en estado pendiente.  El microservicio debe tener la siguiente configuración:  <b>Ruta:</b> /sales/approvesale <b>Método:</b> POST <b>Formato de entrada:</b> JSON  <b>Header:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

Authorization	header	Bearer <token>
Content-Type	header	application/json

#### Variables en el URL:

Atributo	Tipo	Descripción
cui	entero	Identificación del usuario que esta realizando la compra. <b>Atributo requerido</b>
total	float	total a cancelar <b>Atributo requerido</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	mensaje descriptivo

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
401	Usuario no autenticado
404	Tarjeta sin fondos
403	Usuario sin permisos
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallido
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

**Ejemplos de parámetros de entrada:**

```
{  
  "cui":2994902460109,  
  "total":40  
}
```

**Ejemplos de parámetros de salida exitosa:**

```
{  
  "estatus": 200,  
  "mensaje": "Venta aprobada con exito."  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 400  
  "error": "Parámetro requerido",  
  "mensaje": "Faltó algún parámetro"  
}
```

## VENTAS

<b>ID:</b> 23	<b>Nombre:</b> Registro de planilla
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se quiere poder registrar una planilla para cada trabajador de la organización, y poder llevar el control
<b>Estimado:</b> 5	



<b>Módulo:</b> Contabilidad	de los mismos.
-----------------------------	----------------

### Criterios de Aceptación

Poder visualizar de manera detallada cuál es la retribución que el empleado que se registra, por las tareas que realiza dentro de la compañía,

El microservicio debe tener la siguiente configuración:

**Ruta:** /accounting/payrollrecord

**Método:** POST

**Formato de entrada:** JSON

### Payload:

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: - 1 - 2 Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

### Header:

Atributo	Tipo	Descripción
Content-Type	header	application/json
Authorization	header	Bearer <token>

### Body:

Atributo	Tipo	Descripción
cui	entero	identificación del usuario <b>Atributo requerido</b>

idEmpleo	entero	identificador del tipo de empleo de la persona <b>Atributo requerido</b>
bonificación	arreglo	Arreglo con las actividades acreedoras correspondientes a un empleado correspondientes según la ley. las actividades tendrán los siguientes atributos: <ul style="list-style-type: none"> <li>- tipo</li> <li>- monto</li> </ul> Si no hay actividades, se enviara un arreglo vacío <b>Atributo requerido</b>
extras	arreglo	arreglo con las actividades acreedoras correspondientes a un empleado las actividades tendrán los siguientes atributos: <ul style="list-style-type: none"> <li>- tipo</li> <li>- monto</li> </ul> Si no hay actividades, se enviara un arreglo vacío <b>Atributo requerido</b>
retención	arreglo	arreglo con las actividades que ameriten un descuento o retención monetaria del empleado. las actividades tendrán los siguientes atributos: <ul style="list-style-type: none"> <li>- tipo</li> <li>- monto</li> </ul> Si no hay actividades, se enviara un arreglo vacío <b>Atributo requerido</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 201

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de

		aceptación
mensaje	cadena	muestra información de registro de de planilla exitoso

#### Código de respuesta fallida:

Código	Descripción
400	Error al interpretar la solicitud
401	El usuario no está autenticado
403	Usuario sin permisos
500	Error interno del servidor

#### Body de Salida:

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallida
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

```
{
  "cui": "123456789",
  "idEmpleo": 4567,
  "bonificacion": [
    {
      "tipo": "productividad",
      "monto": 1000.0
    },
    {
      "tipo": "asistencia",
      "monto": 500.0
    }
  ],
  "extras": [
    {
```

```

    "tipo": "capacitación",
    "monto": 100
  },
  {
    "tipo": "proyecto especial",
    "monto": 500
  }
],
"retencion": [
  {
    "tipo": "impuestos",
    "monto": 800.0
  },
  {
    "tipo": "préstamos",
    "monto": 300.0
  }
]
}

```

**Ejemplos de parámetros de salida exitosa:**

```

{
  "estatus": 201,
  "mensaje": "Registro de planilla realizado con éxito"
}

```

**Ejemplo de parámetros de salida fallida:**

```

{
  "estatus": 401,
  "mensaje": "Acceso denegado",
  "error": "Credenciales inválidas"
}

```

<b>ID:</b> 25	<b>Nombre:</b> Pago de planilla
<b>Prioridad:</b> Alta	<b>Historia de usuario</b> Se requiere poder realizar el pago de planilla de cada trabajador de la empresa. y llevar un control de los pagos.
<b>Estimado:</b> 5	
<b>Módulo:</b> Contabilidad	
<b>Criterios de Aceptación</b>  Poder generar los pagos de planilla de cada trabajador y llevar el respectivo control de los mismos.	

El microservicio debe tener la siguiente configuración:

**Ruta:** /accounting/payroll

**Método:** POST

**Formato de entrada:**

**Payload:**

Atributo	Tipo	Descripción
cui	cadena	el número de identificación del usuario.
tipoUsuario	entero	El tipo de usuario dentro de la plataforma, este puede ser: - 1 - 2 Donde: 1 representa administradores 2 clientes normales
exp	entero	Este representa la fecha de expiración del token.

**Header:**

Atributo	Tipo	Descripción
Authorization	header	Bearer <token>
Content-Type	header	application/json

**Body:**

Atributo	Tipo	Descripción
cui	entero	id del empleado a pagar. <b>Atributo requerido</b>
metodo	entero	método de pago: transferencia bancaria = 1 cheque = 2 <b>Atributo requerido</b>

mes	entero	Número del mes a pagar <b>Atributo requerido</b>
anio	entero	año del pago en que se hace el pago. <b>Atributo requerido</b>
moneda	cadena	tipo de moneda para realizar el pago. <b>Atributo requerido</b>
descripcion	cadena	Descripción del pago de planilla. <b>Atributo requerido</b>

**Formato de salida:** JSON

**Código de respuesta exitosa:** HTTP 200

**Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta de aceptación
mensaje	cadena	muestra información de la planilla pagada con éxito.
data	objeto	contiene la información de toda la planilla de un empleado, el nombre, puesto, saldo líquido, y los detalles del pago.

**Código de respuesta fallida:**

Código	Descripción
400	Error al interpretar la solicitud
401	El usuario no está autenticado
403	Usuario sin permisos
500	Error interno del servidor

**Body de Salida:**

Atributo	Tipo	Descripción
estatus	entero	indica el código de respuesta fallida
error	cadena	muestra el nombre del error
mensaje	cadena	muestra más información del error

#### Ejemplos de parámetros de entrada:

```
{  
  "cui": "1234567890123",  
  "metodo": "1",  
  "mes": 12,  
  "anio": 2023,  
  "moneda": "GTQ",  
  "descripcion": "Pago de planilla del mes de diciembre 2023"  
}
```

#### Ejemplos de parámetros de salida exitosa:

```
{  
  "estatus": 201,  
  "mensaje": "Planilla pagada con éxito",  
  "data": {  
    "nombreEmpresa": "ACME Corporation",  
    "mes": "enero",  
    "año": 2023,  
    "empleado": {  
      "nombre": "Juan Pérez",  
      "cui": "1234567890123",  
      "puesto": "desarrollador frontend",  
      "tipoPago": "transferencia bancaria",  
      "sueldoBase": 5000,  
      "bonificacion": [  
        {  
          "tipo": "productividad",  
          "monto": 1000.0  
        },  
        {  
          "tipo": "asistencia",  
          "monto": 500.0  
        }  
      ],  
    },  
  },  
}
```

```
"extras": [  
  {  
    "tipo": "capacitación",  
    "descripcion": "Curso de ventas"  
  },  
  {  
    "tipo": "proyecto especial",  
    "descripcion": "Desarrollo de una nueva aplicación"  
  }  
],  
"retencion": [  
  {  
    "tipo": "impuestos",  
    "monto": 800.0  
  },  
  {  
    "tipo": "préstamos",  
    "monto": 300.0  
  }  
],  
"total": 8000  
}  
}
```

**Ejemplo de parámetros de salida fallida:**

```
{  
  "estatus": 401,  
  "error": "Usuario no autenticado",  
  "mensaje": "La sesión del usuario a expirado o no se a iniciado sesión"  
}
```