# Definición de Endpoint

## Caso de uso: Ver Agenda.

Endpoint: /vet/:id/agenda

Verbo: GET

Parámetros:

* id: ID del veterinario de quien se quiere obtener su agenda.

Resultados:

* status 200: Lista de Citas que componen la agenda del veterinario, puede ser una lista vacía.
* status 404: Error el id del veterinario no se encuentra.

## Caso de uso: Ver Cita.

Endpoint: /cita/:id

Verbo: GET

Parámetros:

* id: ID de la cita a obtener.

Resultados:

* status 200: Cita requerida.
* status 404: Error el id de la cita no se encuentra.

## Caso de uso: Ver datos Mascota.

Endpoint: /mascota/:id

Verbo: GET

Parámetros:

* id: ID de la mascota de quien se requieren los datos.

Resultados:

* status 200: Mascota requerida.
* status 404: Error el id de la mascota no se encuentra.

## Caso de uso: Añadir diagnóstico.

Endpoint: /cita/:id

Verbo: PATCH

Parámetros:

* id: ID de la cita a la que añadir el diagnostico

Body:

* diagnóstico: String que contiene el diagnóstico de la cita.

Resultados:

* 200: OK.
* 404: Error el id de la cita no se encuentra.

## Caso de uso: Ver productos tienda.

Endpoint: /productos

Verbo: GET

Resultados:

* 200: Lista de Productos habilitados en el sistema.

## Caso de uso: Crear Producto.

Endpoint: /producto

Verbo: POST

Body:

* identificador: String con el identificador que se muestra al usuario.
* precio: double con el precio de venta del producto.
* inventario: entero con la cantidad de producto disponible.
* descripcion: String descripcion del producto.
* marca: String con marca del producto.

Resultados:

* 200: OK.

## Caso de uso: Obtener Producto.

Endpoint: /producto/:id

Verbo: GET

Parámetros:

* id: ID del producto que se requiere.

Resultados:

* 200: OK.

## Caso de uso: Actualizar Producto.

Endpoint: /producto/:id

Verbo: PATCH

Parámetros:

* id: ID del producto a actualizar.

Body:

* identificador
* precio
* inventario
* descripcion
* marca

Resultados:

* 200: OK.
* 404: Error el id del producto no existe.

## Caso de uso: Eliminar Producto.

Endpoint: /producto/:id

Verbo: DELETE

Parámetros:

* id: ID del producto a eliminar.

Resultados:

* 200: OK.

## Caso de uso: Ver pedidos realizados por usuarios.

Endpoint: /pedidos

Verbo: GET

Resultados:

* 200: Lista de pedidos vigentes.

## Caso de uso: Enviar Pedido.

Endpoint: /pedido/:id

Verbo: PATCH

Parámetros:

* id: ID del pedido a enviar.

Resultados:

* 200: OK.

## Caso de uso: Autenticarse.

Endpoint: /login

Verbo: POST

Body:

* email: String correo con el que se quiere identificar el usuario.
* Pass: String contraseña del usuario encriptada.

Resultados:

* 200: Token JWT para autenticar las siguientes peticiones.

## Caso de uso: Hacer Registro.

Endpoint: /registroUsuario

Verbo: POST

Body:

* nombre: String con el nombre completo del usuario.
* fechaNacimiento: Date milisegundos equivalentes a la fecha de nacimiento del usuario.
* cedula: String con la cedula del usuario.
* email: String correo electrónico del usuario.
* pass: String contraseña del usuario encriptada.
* telefono: String numero de contacto del usuario.
* direccion: String dirección de entrega de los pedidos del usuario.

Resultados:

* 200: OK.

## Caso de uso: Ver empleados/veterinarios.

Endpoint: /empleados

Verbo: GET

Resultados:

* 200: Lista de Empleados/Veterinarios en el sistema.

## Caso de uso: Ver Empleado/Veterinario.

Endpoint: /empleado/:id

Verbo: GET

Parámetros:

* id: ID empleado/veterinario requerido.

Resultados:

* 200: empleado/veterinario requerido.

## Caso de uso: Crear Empleado/Veterinario.

Endpoint: /empleado

Verbo: POST

Body:

* nombre: String con el nombre completo del usuario.
* fechaNacimiento: Date milisegundos equivalentes a la fecha de nacimiento del empleado.
* cedula: String con la cedula del empleado.
* email: String correo electrónico del empleado.
* pass: String contraseña del empleado encriptada.
* telefono: String numero de contacto del empleado.
* direccion: String dirección de entrega de los pedidos del empleado.
* tarjetaProfesional? (opcional): String tarjeta profesional del veterinario.

Resultados:

* 200: OK.

## Caso de uso: Actualizar Empleado/Veterinario.

Endpoint: /empleado/:id

Verbo: PATCH

Parámetros:

* id: ID del empleado/veterinario a actualizar.

Body:

* nombre: String con el nombre completo del usuario.
* fechaNacimiento: Date milisegundos equivalentes a la fecha de nacimiento del empleado.
* cedula: String con la cedula del empleado.
* email: String correo electrónico del empleado.
* pass: String contraseña del empleado encriptada.
* telefono: String numero de contacto del empleado.
* direccion: String dirección de entrega de los pedidos del empleado.
* tarjetaProfesional? (opcional): String tarjeta profesional del veterinario.

Resultados:

* 200: OK.
* 404: Error el id del empleado/veterinario no existe.

## Caso de uso: Ver administradores.

Endpoint: /admin

Verbo: GET

Resultados:

* 200: Lista de administradores.

## Caso de uso: Crear Administrador.

Endpoint: /admin

Verbo: POST

Body:

* nombre: String con el nombre completo del admin.
* fechaNacimiento: Date milisegundos equivalentes a la fecha de nacimiento del admin.
* cedula: String con la cedula del admin.
* email: String correo electrónico del admin.
* pass: String contraseña del admin encriptada.
* telefono: String numero de contacto del admin.
* direccion: String dirección del admin.

Resultados:

* 200: OK.

## Caso de uso: Actualizar Administrador.

Endpoint: /admin/:id

Verbo: PATCH

Parámetros:

* id: ID del admin a actualizar.

Body:

* nombre.
* fechaNacimiento.
* cedula.
* email.
* pass.
* telefono.
* direccion.

Resultados:

* 200: OK.
* 404: Error el id del empleado/veterinario no existe.

## Caso de uso: Notificar asignación de cita.

Endpoint: /mail/cita/:id

Verbo: POST

Parámetros:

* id: ID de la cita que se quiere notificar.

Resultados:

* 200: OK.
* 404: Error el id del empleado/veterinario no existe.

## Caso de uso: Ver citas agendadas por el usuario.

Endpoint: /citas/usuario/:id

Verbo: GET

Parámetros:

* id: ID del usuario que desea ver sus citas agendadas.

Resultados:

* 200: Lista de citas agendadas.
* 404: Error el id del usuario no existe.

## Caso de uso: Agendar cita.

Endpoint: /cita

Verbo: POST

Body:

* idMascota: String id de la mascota que recibirá la cita.
* fecha: Date milisegundos equivalentes a la fecha de la cita.
* idVeterinario: String id del veterinario que atenderá la cita.

Resultados:

* 200: OK.
* 404: Error el id de la mascota/veterinario no existe.

## Caso de uso: Eliminar Cita.

Endpoint: /cita/:id

Verbo: DELETE

Parámetros:

* id: ID de la cita a cancelar.

Resultados:

* 200: OK.
* 404: Error el id de la cita no existe.

## Caso de uso: Ver tienda.

Endpoint: /productos

Verbo: GET

Resultados:

* 200: Lista de productos que componen la tienda.

## Caso de uso: Buscar producto.

Endpoint: /producto/?busqueda

Verbo: GET

Parámetros:

* busqueda: String que contiene la búsqueda del usuario.

Resultados:

* 200: Lista de productos relevantes a la busqueda.

## Caso de uso: Ver producto.

Endpoint: /producto/:id

Verbo: GET

Parámetros:

* id: ID del producto requerido.

Resultados:

* 200: Producto requerido.
* 404: Error el id del producto no existe.

## Caso de uso: Ver mascotas.

Endpoint: /mascota/usuario/:id

Verbo: GET

Parámetros:

* id: ID del usuario quiere ver a sus mascotas.

Resultados:

* 200: Lista de las mascotas del usuario.
* 404: Error el id del usuario no existe.

## Caso de uso: Agregar al carrito.

Endpoint: /carrito/usuario/:id

Verbo: POST

Parámetros:

* id: ID del usuario a actualizar.

Body:

* idProducto: String id del producto que el usuario quiere agregar al carrito.
* cantidad: entero cantidad de producto a agregar al carrito.

Resultados:

* 200: OK.
* 404: Error el id del empleado/veterinario no existe.

## Caso de uso: Crear mascota.

Endpoint: /mascota

Verbo: POST

Body:

* nombre: String con el nombre de la mascota.
* edad: entero con el número de años de vida del animal.
* tipo: String tipo de mascota.
* raza: String raza del animal.
* peso: entero con el peso del animal.
* dueño: String id del usuario dueño del animal.

Resultados:

* 200: OK.

## Caso de uso: Actualizar mascota.

Endpoint: /mascota/:id

Verbo: PATCH

Parámetros:

* id: ID de la mascota a actualizar.

Body:

* nombre: String con el nombre de la mascota.
* edad: entero con el número de años de vida del animal.
* tipo: String tipo de mascota.
* raza: String raza del animal.
* peso: entero con el peso del animal.
* dueño: String id del usuario dueño del animal.

Resultados:

* 200: OK.
* 404: Error el id del empleado/veterinario no existe.

## Caso de uso: Ver carrito.

Endpoint: /carrito/usuario/:id

Verbo: GET

Parámetros:

* id: ID del usuario que quiere ver el carrito.

Resultados:

* 200: Contenido del carrito.
* 404: Error el id del empleado/veterinario no existe.

## Caso de uso: Actualizar elemento del carrito.

Endpoint: /carrito/usuario/:id/idProducto

Verbo: PATCH

Parámetros:

* id: ID del usuario que quiere actualizar del carrito.
* idProducto: String con el id del producto a actualizar.

Body:

* cantidad: entero cantidad con la que se quiere actualizar el producto.

Resultados:

* 200: OK.
* 404: Error el id del usuario/producto no existe.

## Caso de uso: Eliminar elemento del carrito.

Endpoint: /carrito/usuario/:id/idProducto

Verbo: DELETE

Parámetros:

* id: ID del usuario que quiere eliminar del carrito.
* idProducto: String con el id del producto a eliminar.

Body:

* cantidad: entero cantidad con la que se quiere actualizar el producto.

Resultados:

* 200: OK.
* 404: Error el id del usuario/producto no existe.

## Caso de uso: Ver información usuario.

Endpoint: /usuario/:id

Verbo: GET

Parámetros:

* id: ID del usuario requerido.

Resultados:

* 200: Usuario requerido.
* 404: Error el id del usuario no existe.

## Caso de uso: Actualizar información del usuario.

Endpoint: /usuario/:id

Verbo: PATCH

Parámetros:

* id: ID del usuario que quiere actualizar del carrito.

Body:

* nombre: String con el nombre completo del usuario.
* fechaNacimiento: Date milisegundos equivalentes a la fecha de nacimiento del usuario.
* cedula: String con la cedula del usuario.
* email: String correo electrónico del usuario.
* pass: String contraseña del usuario encriptada.
* telefono: String numero de contacto del usuario.
* direccion: String dirección de entrega de los pedidos del usuario.

Resultados:

* 200: OK.
* 404: Error el id del usuario/producto no existe.

## Caso de uso: Ver pedidos del usuario.

Endpoint: /pedidos/:id

Verbo: GET

Parámetros:

* id: ID del usuario que quiere ver sus pedidos.

Resultados:

* 200: Lista de pedidos.
* 404: Error el id del usuario/producto no existe.

## Caso de uso: Finalizar pedido.

Endpoint: /pedido/:id

Verbo: Delete

Parámetros:

* id: ID del pedido a finalizar.

Resultados:

* 200: OK.
* 404: Error el id del pedido no existe.

## Caso de uso: Actualizar elemento del carrito.

Endpoint: /carrito/usuario/:id/idProducto

Verbo: PATCH

Parámetros:

* id: ID del usuario que quiere actualizar del carrito.
* idProducto: String con el id del producto a actualizar.

Body:

* cantidad: entero cantidad con la que se quiere actualizar el producto.

Resultados:

* 200: OK.
* 404: Error el id del usuario/producto no existe.

## Caso de uso: Realizar pedido.

Endpoint: /pedido/?usuario

Verbo: POST

Parámetros:

* usuario: ID del usuario que quiere realizar el pedido.

Resultados:

* 200: OK.
* 404: Error el id del usuario/producto no existe.

## Caso de uso: Notificar pedido.

Endpoint: /mail/pedido/:id

Verbo: POST

Parámetros:

* id: ID del pedido que quiere notificarse.

Resultados:

* 200: OK.
* 404: Error el id del usuario/producto no existe.