



19 DE ABRIL DE 2024

2DO AVANCE DEL PROYECTO

ARQUITECTURA DE SERVICIOS 2024



1. Diseño del modelo de datos:

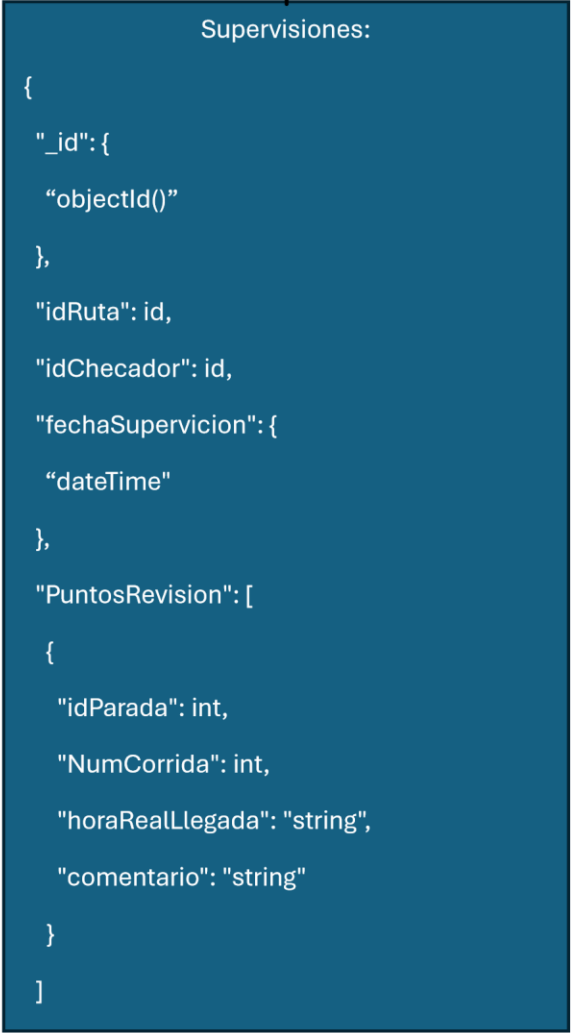
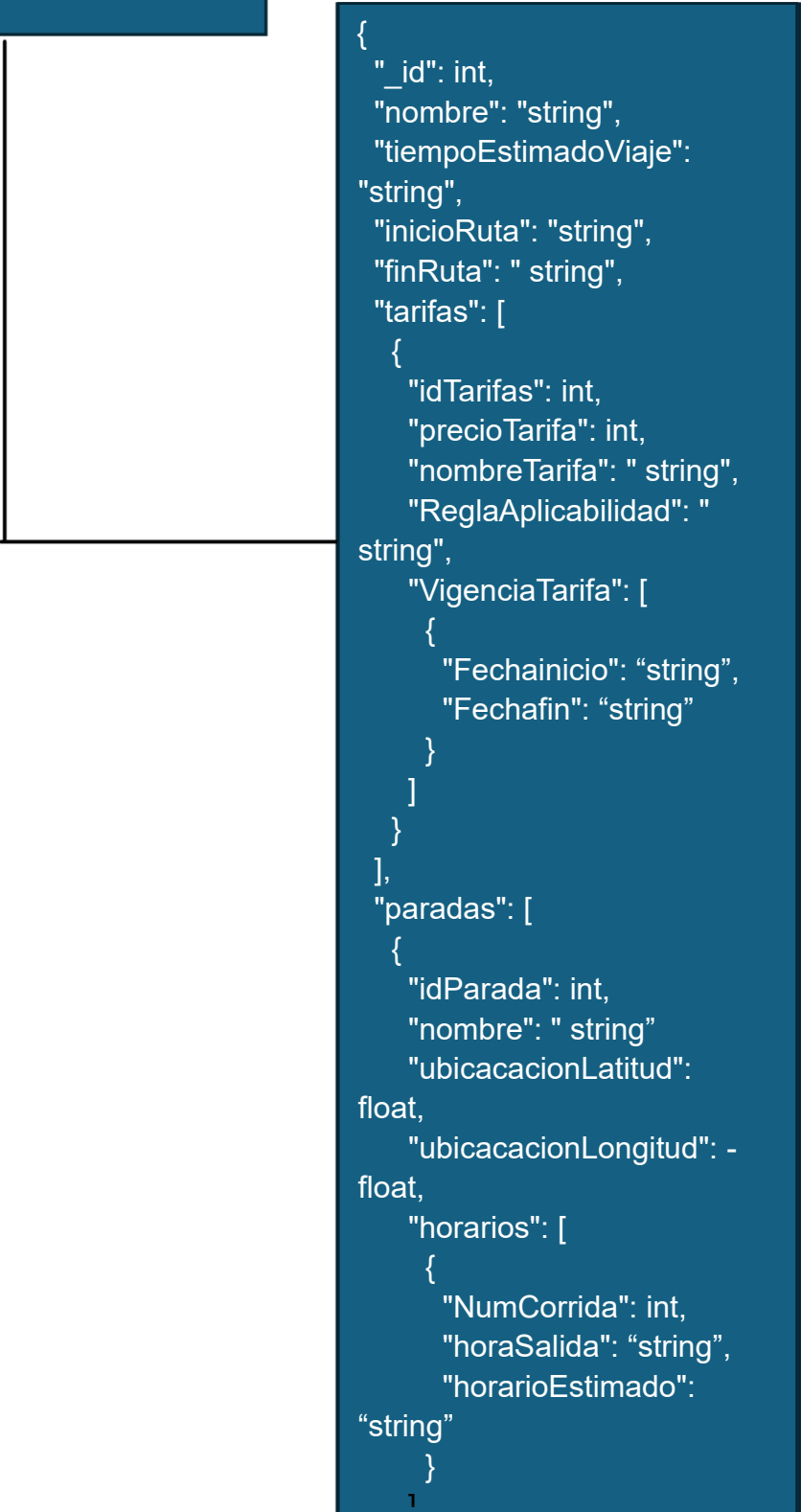
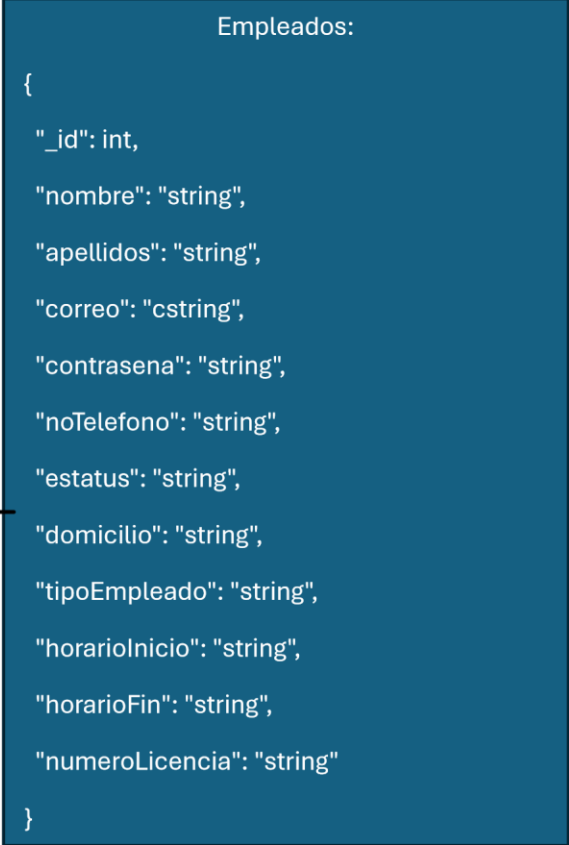
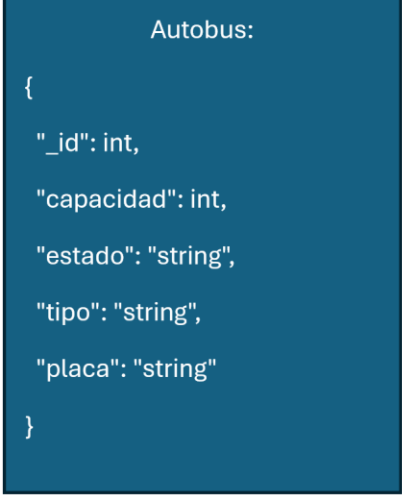
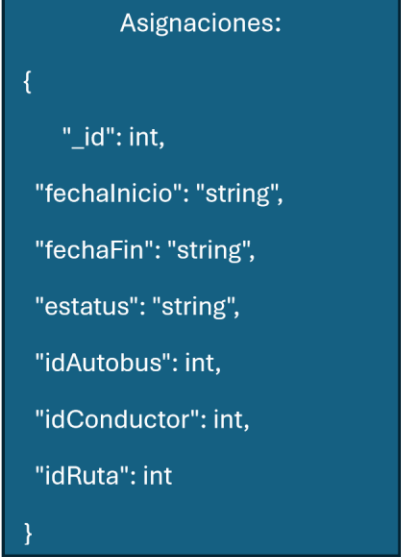
El proyecto busca resolver la problemática de la administración eficiente de una ruta de autobuses. Actualmente, se enfrentan dificultades para rastrear y gestionar la información sobre las rutas de los autobuses, los horarios de llegada, el registro de las unidades de autobuses y los choferes activos.

Para abordar estos desafíos, se propone una Arquitectura de Microservicios. Esta arquitectura permite tener varios microservicios, cada uno encargado de una tarea específica, en lugar de un sistema monolítico que maneje todas estas funciones. Esto facilita el desarrollo, despliegue y escalado independiente de cada microservicio, haciendo que el sistema sea más fácil de mantener y más resistente a los fallos.

Dada la naturaleza del problema y la solución propuesta, se ha elegido MongoDB, una base de datos NoSQL, para el proyecto. MongoDB es ideal para este tipo de aplicaciones ya que ofrece flexibilidad para cambiar el esquema y es capaz de manejar grandes volúmenes de datos. Además, al ser orientada a documentos, se adapta bien a la estructura de microservicios, permitiendo que cada servicio maneje su propia colección de documentos.

2. Scripts de la BD:

La base de datos fue generada en una orientada a documentos más específico en mongoDB y al final nos quedo de la siguiente manera:



Servicio REST Paradas

Descripción:

El servicio tendrá como propósito realizar las operaciones de: insertar, modificar y eliminar datos o información almacenada en la base de datos Zambus.

Tipo de servicio:

Este servicio se puede catalogar como un servicio de Entidad y Tarea al proporcionar el soporte a operaciones de tipo CRUD de los documentos de la entidad.

Operaciones:

Esta entidad va a contar con las siguientes operaciones:

1. Agregar Parada.
2. Quitar Parada.
3. Modificar Parada.
4. Consultar Paradas
5. Consultar Parada.
6. Agregar Horario.
7. Quitar Horario.
8. Consultar Horario por parada

Operación: Agregar Parada

Elementos	Valor
método de acceso	Put
Ruta	/rutas/Agregar/Parada
Protocolo	http
Actor(es)	Administrador
Entrada	<pre>{ "paradas": [{ "nombre": "string", "ubicacionLatitud": float, "ubicacionLongitud": float, }] }</pre>
Proceso	Acción: Registrar un parada en la BD, para realizar el registro de forma completa, se tiene que generar los siguientes datos de manera automática: <ul style="list-style-type: none">• El id de la parada.

	<ul style="list-style-type: none"> • Agregar el nombre de la parada. • Agregar los datos de ubicación, los que son Latitud y Longitud. <p>Validaciones:</p> <ol style="list-style-type: none"> 1. Los id deben de ser diferentes a los ya existentes. 2. No se debe de repetir el nombre de la parada
Salida	<pre>{ "estatus":int, "mensaje":String }</pre>

Operación: Quitar Parada

Elementos	Valor
método de acceso	Put
Ruta	/rutas/Eliminar/Parada/{idRuta:int}/{idParada:int}
Protocolo	http
Actor(es)	Administrador
Entrada	
Proceso	<p>Acción:</p> <ul style="list-style-type: none"> • Se eliminará el documento seleccionado de la BD <p>Validaciones:</p> <ol style="list-style-type: none"> 1. Se evaluará que la parada que quiera ser eliminado exista en la BD
Salida	<pre>{ "estatus": int, "mensaje": String }</pre>

Operación: Modificar Parada

Elementos	Valor
método de acceso	PUT
Ruta	/rutas/Modifica/Parada/{idRuta:int}/{idParada:int}
Protocolo	http
Actor(es)	Administrador
Entrada	<pre>{ "paradas": [{ "nombre": "string",</pre>

Proceso

```
"ubicacionLatitud": float,  
"ubicacionLongitud": float,  
}]  
}
```

Acción:

Se modificará los valores de la ubicación de la parada en caso de ser necesario.

Validaciones:

1. Evaluar que los campos ingresados sean campos validos de latitud y longitud.
2. Evaluar que el documento a modificar sea existente.

Salida

Operación: Consultar Paradas Por Rutas

Elementos	Valor
método de acceso	GET
Ruta	/rutas/Paradas/{idRuta:int}
Protocolo	http
Actor(es)	Administrador
Entrada	
Proceso	Acción: Se mostrará la parada especificada. Validaciones: <ol style="list-style-type: none">1. Evaluar que el id seleccionado sea válido o existente.
Salida	<pre>{ "paradas": [{ "idParada": int, "nombre": "string", "ubicacionLatitud": float, "ubicacionLongitud": float, }] }</pre>

Operación: Consultar una Parada por Ruta

Elementos	Valor
método de acceso	GET
Ruta	/rutas/Paradas/{idRuta:int}/{idParada:int}
Protocolo	http

Actor(es)	Administrador
Entrada	N/A
Proceso	Acción: Se mostrará todas las paradas existentes en la BD
Salida	{ "paradas": [{ "idParada": int, "nombre": "string", "ubicacionLatitud": float, "ubicacionLongitud": float, }] }

Operación: Agregar horario

Elementos	Valor
método de acceso	Put
Ruta	/rutas/Paradas/AgregarHorario/{idRuta:int}/{idParada:int}
Protocolo	http
Actor(es)	Administrador
Entrada	"horarios": [{ "horaSalida": "string", "horarioEstimado": "string" }] }
Proceso	Acción: Registrar un nuevo horario en la base de datos en el documento de parada seleccionado, para eso se tendrá que llenar los campos para poder obtener los siguientes datos <ul style="list-style-type: none"> Se tendrá que llenar el numero de corrida que sera nuestro identificador. Se nos proporcionara la hora de salida y la hora estimada de llegada. Validaciones: <ol style="list-style-type: none"> Los campos de hora tendrán un formato de reloj de 24 horas que será registrado como string, en otras palabras los campos serán validos si son ingresados de la siguiente manera: "07:30, 12:45, 16:00, 24:00"
Salida	{ "estatus":int,

```

“mensaje”:String
}

```

Operación: Quitar horario

Elementos	Valor
método de acceso	Put
Ruta	/rutas/Paradas/QuitarHorario/{idRuta:int}/{idParada:int}/{numCorrida:int}
Protocolo	http
Actor(es)	Administrador
Entrada	
Proceso	<p>Acción: Registrar un nuevo horario en la base de datos en el documento de parada seleccionado, para eso se tendrá que llenar los campos para poder obtener los siguientes datos</p> <ul style="list-style-type: none"> Se tendrá que llenar el numero de corrida que sera nuestro identificador. Se nos proporcionara la hora de salida y la hora estimada de llegada. <p>Validaciones:</p> <ol style="list-style-type: none"> Los campos de hora tendrán un formato de reloj de 24 horas que será registrado como string, en otras palabras los campos serán validos si son ingresados de la siguiente manera: “07:30, 12:45, 16:00, 24:00”
Salida	<pre> { “estatus”:int, “mensaje”:String } </pre>

Operación: Consultar Horario por parada

Elementos	Valor
método de acceso	Get
Ruta	/rutas/Paradas/Horario/{idRuta:int}/{idParada:int}"
Protocolo	http
Actor(es)	Administrador
Entrada	
Proceso	<p>Acción: Registrar un nuevo horario en la base de datos en el documento de parada seleccionado, para eso se</p>

Salida

tendrá que llenar los campos para poder obtener los siguientes datos

- Se tendrá que llenar el numero de corrida que sera nuestro identificador.
- Se nos proporcionara la hora de salida y la hora estimada de llegada.

Validaciones:

3. Los campos de hora tendrán un formato de reloj de 24 horas que será registrado como string, en otras palabras los campos serán validos si son ingresados de la siguiente manera: “07:30, 12:45, 16:00, 24:00”

```
{
  "paradas": [
    {
      "nombre": "string",
      "ubicacionLatitud": float,
      "ubicacionLongitud": float,
    },
    "horarios": [
      {
        "horaSalida": "string",
        "horarioEstimado": "string"
      }
    ]
  ]
}
```

Operación: Consultar Horario específico

Elementos	Valor
método de acceso	Get
Ruta	/rutas/Paradas/Horario/{idRuta:int}/{idParada:int}/{numCorrida:int}
Protocolo	http
Actor(es)	Administrador
Entrada	
Proceso	Acción: Registrar un nuevo horario en la base de datos en el documento de parada seleccionado, para eso se tendrá que llenar los campos para poder obtener los siguientes datos <ul style="list-style-type: none">• Se tendrá que llenar el numero de corrida que sera nuestro identificador.• Se nos proporcionara la hora de salida y la hora estimada de llegada. Validaciones:

Salida

4. Los campos de hora tendrán un formato de reloj de 24 horas que será registrado como string, en otras palabras los campos serán validos si son ingresados de la siguiente manera: “07:30, 12:45, 16:00, 24:00”

```
{
  "paradas": [
    {
      "nombre": "string",
      "ubicacionLatitud": float,
      "ubicacionLongitud": float,
    },
    "horarios": [
      {
        "horaSalida": "string",
        "horarioEstimado": "string"
      }
    ]
  }
}
```

Servicio REST Tarifa

Descripción:

El servicio tendrá como propósito realizar las operaciones de: insertar, modificar y eliminar datos o información almacena en la base de datos Zambus.

Tipo de servicio:

Este servicio se puede catalogar como un servicio de Entidad y Tarea al proporcionar el soporte a operaciones de tipo CRUD de los documentos de la entidad.

Operaciones:

Esta entidad va a contar con las siguientes operaciones:

1. Crear Tarifa.
2. Modificar Tarifa.
3. Eliminar Tarifa.
4. Consultar tarifa.
5. Consultar tarifas.
6. Agregar Vigencia.
7. Quitar Vigencia.
8. Consultar Vigencia de Tarifa.

Operación: Agregar tarifa

Elementos	Valor
método de acceso	PUT
Ruta	/rutas/Agregar/Tarifa
Protocolo	http
Actor(es)	Administrador
Entrada	<pre>{ "tarifas": [{ "precioTarifa": int, "nombreTarifa": "string", "ReglaAplicabilidad": "string" }] }</pre>
Proceso	<p>Acción: Registrar una nueva tarifa en la base de datos en la parada, para eso se tendrá que llenar los campos para poder obtener los siguientes datos</p> <ul style="list-style-type: none">• El id de la tarifa.• Agregar el precio de la tarifa.• Agregar el nombre de la tarifa.• Agregar la regla de aplicabilidad que tendrá la tarifa. <p>Validaciones:</p> <ol style="list-style-type: none">1. Los id deben de ser diferentes a los ya existentes.
Salida	<pre>{ "estatus":int, "mensaje":String }</pre>

Operación: Modificar tarifa

Elementos	Valor
método de acceso	PUT
Ruta	/rutas/Modificar/Tarifas/{idRuta:int}/{idTarifas:int}
Protocolo	http
Actor(es)	Administrador
Entrada	<pre>{ "tarifas": [{ "precioTarifa": int, "nombreTarifa": "string", "ReglaAplicabilidad": "string" }] }</pre>

Proceso

```
}  
}
```

Acción:

Se modificará los valores de la tarifa en caso de necesitar hacer cambios en una ya existente.

Validaciones:

1. Evaluar que los campos ingresados sean campos válidos.
2. Evaluar que el documento a modificar sea existente.

Salida

```
{  
  "estatus":int,  
  "mensaje":String  
}
```

Operación: Quitar tarifa

Elementos	Valor
método de acceso	Put
Ruta	/rutas/Eliminar/Tarifas/{idRuta:int}/{idTarifas:int}
Protocolo	http
Actor(es)	Administrador
Entrada	
Proceso	Acción: <ul style="list-style-type: none">• Se eliminará el documento seleccionado de la BD Validaciones: <ol style="list-style-type: none">1. Se evaluará que la parada que quiera ser eliminado exista en la BD
Salida	<pre>{ "estatus": int, "mensaje": String }</pre>

Operación: Consultar Tarifa Por Rutas

Elementos	Valor
método de acceso	GET
Ruta	/rutas/Tarifas/{idRuta:int}
Protocolo	http
Actor(es)	Administrador, Conductor, Checador

Entrada	
Proceso	Acción: Se mostrará la tarifa especificada. Validaciones: <ol style="list-style-type: none"> 1. Evaluar que el id seleccionado sea válido o existente.
Salida	<pre>{ "tarifas": [{ "idTarifas": int, "precioTarifa": int, "nombreTarifa": "string", "ReglaAplicabilidad": "string" }] }</pre>

Operación: Agregar nueva vigencia

Elementos	Valor
método de acceso	PUT
Ruta	/rutas/Agregar/VigenciaTarifa/{idRuta:int}/{idTarifas:int}
Protocolo	http
Actor(es)	Administrador
Entrada	<pre>{ "VigenciaTarifa": [{ "Fechainicio": "string", "Fechafin": "string" }] }</pre>
Proceso	Acción: Registrar una nueva vigencia en la base de datos en la tarifa/vigenciaTrifa, para eso se tendrá que llenar los campos para poder obtener los siguientes datos <ul style="list-style-type: none"> • Se tendrá que llenar los campos de fecha Inicio y fecha Fin. Validaciones: <ol style="list-style-type: none"> 1. Las fechas contarán con formato año-mes-día
Salida	<pre>{ "estatus":int, "mensaje":String }</pre>

Operación: Quitar vigencia

<i>Elementos</i>	<i>Valor</i>
<i>método de acceso</i>	PUT
<i>Ruta</i>	/rutas/Eliminar/VigenciaTarifa/{idRuta:int}/{idTarifas:int}/{idVigencia:int}
<i>Protocolo</i>	http
<i>Actor(es)</i>	Administrador, conductor y checador
<i>Entrada</i>	
<i>Proceso</i>	Acción: Consultara todas las vigencias del id de la tarifa seleccionada <ul style="list-style-type: none">• Se que proporcionar el id de la tarifa Validaciones: <ol style="list-style-type: none">1. El id debe de ser valido y existente
<i>Salida</i>	{ “estatus”:int, “mensaje”:String }

Servicio REST Autobús

Descripción:

Este servicio permite realizar operaciones relacionadas con la gestión de autobuses, rutas y asignaciones.

Tipo de servicio:

Entidad y Tarea, ya que soporta operaciones específicas para las entidades Autobús.

Operaciones:

1. Agregar
2. Modificar
3. Consultar Unidad
4. Consultar Unidades
5. Eliminar

Operación: Agregar Autobús

Elementos	Valor
Método de acceso	Post
Ruta	/autobuses/Agregar
Protocolo	HTTP
Actor	Administrador
Entrada	{ "capacidad": int, "estado": "string", "tipo": "string", "placa": "string" }
Proceso	*Se valida que la capacidad del autobús sea un valor positivo. *Se valida que el estado del autobús sea uno de los valores predefinidos. *Se valida que el tipo de autobús sea uno de los valores predefinidos.

Validaciones	*Se valida que el conductor con el ID proporcionado exista en la base de datos. *Se valida que la placa del autobús no esté ya registrada en la base de datos.
	La dirección de correo debe ser única
	Salida { "estatus": int, "mensaje": "string" }

Operación: Modificar Autobús

Elementos	Valor
Método de acceso	Put
Ruta	/autobuses/Modificar/{idAutobus}
Protocolo	HTTP
Actor	Administrador
Entrada	{ "capacidad": int, "estado": "string", "tipo": "string", "placa": "string" }
Proceso	Modificar los datos de un autobús existente en la base de datos.
Validación	*Se valida que la capacidad del autobús sea un valor positivo. *Se valida que el estado del autobús sea uno de los valores predefinidos. *Se valida que el tipo de autobús sea uno de los valores predefinidos. *Se valida que el conductor con el ID proporcionado exista en la base de datos. *Se valida que la placa del autobús no esté ya registrada en la base de datos con un ID diferente.
Salida	{ "estatus": int, "mensaje": "string" }


```
}
```

Operación: Consultar Unidad

<i>Elementos</i>	<i>Valor</i>
<i>Métodos de acceso</i>	Get
<i>Ruta</i>	/autobuses/Consultar/{idAutobus}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Administrar
<i>Entrada</i>	
<i>Proceso</i>	Consultar los datos de un autobús específico en lavase de datos
<i>Validación</i>	El autobús con el ID especificado debe existir en la base de datos
<i>Salida</i>	{ "capacidad": int, "estado": "string", "tipo": "string", "placa": "string" }

Operación: Consultar Unidades

Elementos	Valor
Métodos de acceso	Get
Ruta	/autobuses/Consultar/{idAutobus}
Protocolo	HTTP
Actor	Administrar
Entrada	
Proceso	Consultar los datos de un autobús específico en lavase de datos
Validación	El autobús con el ID especificado debe existir en la base de datos
Salida	{ "capacidad": int, "estado": "string", "tipo": "string", "placa": "string" }

Operación: Eliminar Autobús

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Delete
<i>Ruta</i>	/autobuses/Eliminar/{idAutobus}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Administrador
<i>Entrada</i>	
<i>Proceso</i>	Eliminar un autobús de la base de datos
<i>Validación</i>	El autobús con el ID específico debe existir en la base de datos
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }

Servicio REST Ruta

Descripción:

Este servicio permite realizar operaciones relacionadas con las supervisiones de las rutas.

Tipo de servicio:

Entidad y Tarea, ya que soporta operaciones específicas para la entidad Ruta.

Operaciones:

1. Agregar Ruta
2. Modificar Ruta
3. Consultar Ruta
4. Consultar Rutas
5. Eliminar Ruta

Operación: Agregar Ruta

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post

<i>Ruta</i>	/rutas/Agregar
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	<pre>{ "nombre": "string", "tiempoEstimadoViaje": int, "inicioRuta": "string", "finRuta": "string", "idChecador": int, }</pre>
<i>Proceso</i>	Acción: Registrar una nueva ruta en la base de datos.
<i>Validaciones</i>	<p>*Se valida que el nombre de la ruta no esté ya registrado en la base de datos.</p> <p>*Se valida que el tiempo estimado de viaje sea un valor positivo.</p> <p>*Se valida que los puntos de inicio y fin de la ruta sean válidos.</p> <p>*Se valida que las tarifas sean valores positivos.</p> <p>*Se valida que el checador con el ID proporcionado exista en la base de datos.</p>
<i>Salida</i>	<pre>{ "estatus": int, "mensaje": "string" }</pre>

Operación: Modificar Ruta

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/rutas/Modificar/{idRuta}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	<pre>{ "nombre": "string", "tiempoEstimadoViaje": int, "inicioRuta": "string", "finRuta": "string", "idChecador": int }</pre>

<i>Proceso</i>	Acción: Modificar los datos de una ruta existente en la base de datos.
<i>Validaciones</i>	*Se valida que el nombre de la ruta no esté ya registrado en la base de datos. *Se valida que el tiempo estimado de viaje sea un valor positivo. *Se valida que los puntos de inicio y fin de la ruta sean válidos. *Se valida que las tarifas sean valores positivos. *Se valida que el checador con el ID proporcionado exista en la base de datos.
<i>Salida</i>	<pre>{ "estatus": int, "mensaje": "string" }</pre>

Operación: Consultar Ruta

<i>Elementos</i>	Valor
<i>Método de acceso</i>	Post
<i>Ruta</i>	/rutas/Consultar/{idRuta}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	
<i>Proceso</i>	Acción: Consulta los datos de una ruta existente en la base de datos.
<i>Validaciones</i>	*Se valida que el nombre de la ruta no esté ya registrado en la base de datos. *Se valida que el tiempo estimado de viaje sea un valor positivo. *Se valida que los puntos de inicio y fin de la ruta sean válidos.

Salida

*Se valida que las tarifas sean valores positivos.
*Se valida que el checador con el ID proporcionado exista en la base de datos.

```
{  
  "id": int,  
  "nombre": "string",  
  "tiempoEstimadoViaje": int,  
  "inicioRuta": "string",  
  "finRuta": "string",  
  "idChecador": int  
}
```

Operación: Consultar Rutas

Elementos	Valor
Método de acceso	Post
Ruta	/rutas/Consultar
Protocolo	HTTP
Actor	Supervisor
Entrada	
Proceso	Acción: Consulta los datos de una ruta existente en la base de datos.
Validaciones	<p>*Se valida que el nombre de la ruta no esté ya registrado en la base de datos.</p> <p>*Se valida que el tiempo estimado de viaje sea un valor positivo.</p> <p>*Se valida que los puntos de inicio y fin de la ruta sean válidos.</p>

	<p>*Se valida que las tarifas sean valores positivos.</p> <p>*Se valida que el checador con el ID proporcionado exista en la base de datos.</p>
Salida	<pre>{ "id": int, "nombre": "string", "tiempoEstimadoViaje": int, "inicioRuta": "string", "finRuta": "string", "idChecador": int }</pre>

Operación:Eliminar Ruta

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/rutas/Eliminar/{idRuta}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	
<i>Proceso</i>	Acción: Modificar los datos de una ruta existente en la base de datos.
<i>Validaciones</i>	<p>*Se valida que el nombre de la ruta no esté ya registrado en la base de datos.</p> <p>*Se valida que el tiempo estimado de viaje sea un valor positivo.</p>

*Se valida que los puntos de inicio y fin de la ruta sean válidos.
*Se valida que las tarifas sean valores positivos.
*Se valida que el checador con el ID proporcionado exista en la base de datos.

Salida

```
{  
  "estatus": int,  
  "mensaje": "string"  
}
```

Servicio REST Asignación

Descripción:

Este servicio permite realizar operaciones relacionadas con las supervisiones de las Asignaciones .

Tipo de servicio:

Entidad y Tarea, ya que soporta operaciones específicas para la entidad Asignación.

Operaciones:

1. Asignar Ruta
2. Quitar Asignación de Ruta
3. Modificar Asignación
4. Consultar Asignaciones
5. Consultar Asignación

Operación: Asignar Ruta

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/asignaciones/AsignarRuta
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor

<i>Entrada</i>	{ "fechaInicio": "string", "fechaFin": "string", "estatus": "string", "idAutobus": int, "idConductor": int, "idRuta": int }
<i>Proceso</i>	Acción: Asignar un autobús a una ruta específica.
<i>Validaciones</i>	*Se valida que el autobús con el ID proporcionado exista en la base de datos. *Se valida que la ruta con el ID proporcionado exista en la base de datos. *Se valida que el autobús no esté ya asignado a otra ruta.
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }

Operación: Quitar Asignación de Ruta

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Delete
<i>Ruta</i>	/asignaciones/QuitarAsignacion/{_id}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	
<i>Proceso</i>	Acción: Quitar un autobús de una ruta específica.
<i>Validaciones</i>	*Se valida que el autobús con el ID proporcionado exista en la base de datos.

	*Se valida que la ruta con el ID proporcionado exista en la base de datos. *Se valida que el autobús no esté ya asignado a otra ruta.
Salida	<pre>{ "estatus": int, "mensaje": "string" }</pre>

Operación: Modificar Asignación

Elementos	Valor
Método de acceso	Put
Ruta	/asignaciones/ModificarAsignacion/{_id}
Protocolo	HTTP
Actor	Supervisor
Entrada	<pre>{ "fechaInicio": "string", "fechaFin": "string", "estatus": "string", "idAutobus": int, "idConductor": int, "idRuta": int }</pre>
Proceso	Acción: Consulta los datos de una asignación a autobuses existente en la base de datos
Validaciones	*Se valida que el autobús con el ID proporcionado exista en la base de datos. *Se valida que la ruta con el ID proporcionado exista en la base de datos. *Se valida que el autobús no esté ya asignado a otra ruta.
Salida	<pre>{ "estatus": int, "mensaje": "string" }</pre>

Operación: Consultar Asignaciones

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Get
<i>Ruta</i>	/asignaciones/Consultar
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	
<i>Proceso</i>	Acción: Consulta los datos de una asignación a una ruta existente en la base de datos
<i>Validaciones</i>	*Se valida que el autobús con el ID proporcionado exista en la base de datos. *Se valida que la ruta con el ID proporcionado exista en la base de datos. *Se valida que el autobús no esté ya asignado a otra ruta.
<i>Salida</i>	{ "fechaInicio": "string", "fechaFin": "string", "estatus": "string", "idAutobus": int, "idConductor": int, "idRuta": int }

Operación: Consultar Asignación

Elementos	Valor
Método de acceso	Get
Ruta	/asignaciones/Consultar/{_id}
Protocolo	HTTP
Actor	Supervisor

Entrada	
Proceso	Acción: Consulta los datos de una asignación a una ruta existente en la base de datos
Validaciones	<p>*Se valida que el autobús con el ID proporcionado exista en la base de datos.</p> <p>*Se valida que la ruta con el ID proporcionado exista en la base de datos.</p> <p>*Se valida que el autobús no esté ya asignado a otra ruta.</p>
Salida	<pre>{ "fechaInicio": "string", "fechaFin": "string", "estatus": "string", "idAutobus": int, "idConductor": int, "idRuta": int }</pre>

Servicio REST Empleado

Descripción:

El servicio de Empleado permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los registros de empleados en la base de datos.

Tipo de servicio:

El servicio para la entidad Empleado se puede clasificar como un servicio de Entidad y Tarea, ya que proporciona soporte para operaciones CRUD (Crear, Leer, Actualizar, Eliminar) relacionadas con los registros de empleado en la base de datos.

Operaciones:

1. Agregar
2. Modificar
3. Consultar
4. Eliminar

Operación: Agregar Empleado

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/Empleado/Agregar
<i>Protocolo</i>	HTTP
<i>Actor</i>	Administrador
<i>Entrada</i>	{ "nombre": "string", "apellidos": "string", "correo": "string", "contraseña": "string", "noTelefono": "string", "estatus": "string", "domicilio": "string", "tipoEmpleado": "string", "horarioInicio": "string", "horarioFin": "string", "numeroLicencia": "string" }

<i>Proceso</i>	Agrega un nuevo empleado a la base de datos con la información proporcionada
<i>Validaciones</i>	La dirección de correo debe ser única
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }

Operación: Modificar Empleado

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Put
<i>Ruta</i>	/Empleado/Modificar/{id}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Administrador
<i>Entrada</i>	{ "nombre": "string", "apellidos": "string", "correo": "string", "contraseña": "string", "noTelefono": "string", "estatus": "string", "domicilio": "string", "tipoEmpleado": "string", "horarioInicio": "string", "horarioFin": "string", "numeroLicencia": "string" }
<i>Proceso</i>	Modificar los datos de un empleado existente en la base de datos
<i>Validación</i>	El empleado con el ID especificado debe existir en la base de datos
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }

Operación: Consultar Empleado

<i>Elementos</i>	<i>Valor</i>
<i>Métodos de acceso</i>	Get

<i>Ruta</i>	/Empleado/Consultar/{id}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Administrar
<i>Entrada</i>	{ "id": int }
<i>Proceso</i>	Consultar los datos de un empleado específico en lavase de datos
<i>Validación</i>	El empleado con el ID especificado debe existir en la base de datos
<i>Salida</i>	{ "id": int, "nombre": "string", "apellidos": "string", "correo": "string", "noTelefono": "string", "estatus": "string", "domicilio": "string", "tipoEmpleado": "string", "horarioInicio": "string", "horarioFin": "string", "numeroLicencia": "string" }

Operación: Eliminar Empleado

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Delete
<i>Ruta</i>	/Empleado/Eliminar/{id}
<i>Protocolo</i>	HTTP
<i>Actor</i>	Administrador
<i>Entrada</i>	{ "id": int }
<i>Proceso</i>	Eliminar un empleado de la base de datos
<i>Validación</i>	El empleado con el ID específico debe existir en la base de datos
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }

Servicio REST Supervisiones

Descripción:

Este servicio permite realizar operaciones relacionadas con las supervisiones de empleados.

Tipo de servicio:

Entidad y Tarea, ya que soporta operaciones específicas para la entidad Supervisiones.

Operaciones:

1. Agregar Fecha de Supervisión

Operación: Agregar Fecha de Supervisión

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/Supervisiones/AgregarFechaSupervision
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	{ "idEmpleado": int, "fechaSupervision": "string (formato de fecha)" }
<i>Proceso</i>	Acción: Agregar la fecha de supervisión para un empleado. Registro de la fecha de supervisión en la base de datos.
<i>Validaciones</i>	Se valida que la fecha de supervisión sea válida Se valida que el empleado con el ID proporcionado exista en la base de datos
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }

Servicio REST puntosRevision

Descripción:

Este servicio permite realizar operaciones relacionadas con los puntos de revisión de empleados.

Tipo de servicio:

Entidad y Tarea, ya que soporta operaciones específicas para la entidad puntosRevision.

Operaciones:

1. Agregar Hora de Llegada
2. Agregar Comentario

Operación: Agregar Hora de Llegada

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/puntosRevision/AgregarHoraLlegada
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	{ "idEmpleado": int, "horaRealLlegada": "string (formato de hora)" }
<i>Proceso</i>	Acción: Registrar la hora real de llegada de un empleado en un punto de revisión Actualización del registro en la base de datos
<i>Validaciones</i>	Se valida que la hora de llegada sea válida y esté en un formato correcto Se valida que el empleado con el ID proporcionado exista en la base de datos
<i>Salida</i>	{ "estatus": int, "mensaje": "string"


```
}
```

Operación: Agregar Comentario

<i>Elementos</i>	<i>Valor</i>
<i>Método de acceso</i>	Post
<i>Ruta</i>	/puntosRevision/AgregarComentario
<i>Protocolo</i>	HTTP
<i>Actor</i>	Supervisor
<i>Entrada</i>	{ "idEmpleado": int, "comentario": "string" }
<i>Proceso</i>	Acción: Agregar un comentario sobre la revisión de un empleado en un punto específico Registro del comentario en la base de datos
<i>Validaciones</i>	Se valida que el comentario no esté vacío Se valida que el empleado con el ID proporcionado exista en la base de datos
<i>Salida</i>	{ "estatus": int, "mensaje": "string" }