

Documentación API E2

1. POST /clientes/registro

Descripción: Registra un nuevo cliente en el sistema.

Parámetros del cuerpo (Body Parameters):

nombre: Nombre del cliente (String).

contrasena: Contraseña del cliente, debe incluir al menos un número, una letra y un carácter especial (String).

email: Email del cliente, debe ser una dirección de correo válida (String).

telefono: Número de teléfono del cliente, debe ser numérico (String).

Respuestas:

201: Cliente registrado con éxito.

400: Datos no válidos o cliente ya registrado.

2. GET /clientes/buscar-por-email

Descripción: Busca a un cliente basado en su dirección de correo electrónico.

Parámetros de consulta (Query Parameters):

email: Email del cliente que se desea buscar.

Respuestas:

200: Cliente encontrado.

404: Cliente no encontrado.

400: Solicitud incorrecta o email no válido.

3. POST /admins/registro

Descripción: Registra un nuevo administrador en el sistema.

Parámetros del cuerpo (Body Parameters):

nombre: Nombre del administrador (String).

contrasena: Contraseña del administrador, debe incluir al menos un número, una letra y un carácter especial (String). Ejemplo de caracteres especiales válidos: [@ \$! % * ? &].

email: email del administrador, debe ser una dirección de correo válida (String).

Respuestas:

201: Administrador registrado con éxito.

400: Datos no válidos o administrador ya registrado.

4. GET /admins/{adminId}

Descripción: Obtiene la información de un administrador específico.

Parámetros de ruta (Path Parameters):

adminId: ID del administrador que se quiere consultar (Integer).

Respuestas:

200: Información del administrador.

404: Administrador no encontrado.

5. PUT /admins/{adminId}

Descripción: Actualiza la información de un administrador específico.

Parámetros de ruta (Path Parameters):

adminId: ID del administrador que se quiere actualizar (Integer).

Parámetros del cuerpo (Body Parameters):

nombre: (Opcional) Nombre del administrador (String).

contrasena: (Opcional) Contraseña del administrador (String).

email: (Opcional) Email del administrador (String).

Respuestas:

200: Administrador actualizado con éxito.

400: Datos no válidos.

404: Administrador no encontrado.

6. DELETE /admins/{adminId}

Descripción: Elimina un administrador específico.

Parámetros de ruta (Path Parameters):

adminId: ID del administrador que se quiere eliminar (Integer).

Respuestas:

204: Administrador eliminado con éxito.

404: Administrador no encontrado.

7. POST /choferes/registro

Descripción: Registra un nuevo chofer en el sistema.

Parámetros del cuerpo (Body Parameters):

nombre: Nombre del chofer (String).

contrasena: Contraseña del chofer, debe incluir al menos un número, una letra y un carácter especial (String). Ejemplo de caracteres especiales válidos: [@ \$! % * ? &].

email: Email del chofer, debe ser una dirección de correo válida (String).

telefono: Teléfono del chofer, debe ser un valor numérico (String).

Respuestas:

201: Chofer registrado con éxito.

400: Datos no válidos o chofer ya registrado.

8. GET /choferes/{choferId}

Descripción: Obtiene la información de un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer que se quiere consultar (Integer).

Respuestas:

200: Información del chofer.

404: Chofer no encontrado

9. PUT /choferes/{choferId}

Descripción: Actualiza la información de un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer que se quiere actualizar (Integer).

Parámetros del cuerpo (Body Parameters):

nombre: (Opcional) Nombre del chofer (String).

contrasena: (Opcional) Contraseña del chofer (String).

email: (Opcional) Email del chofer (String).

telefono: (Opcional) Teléfono del chofer (String).

Respuestas:

200: Chofer actualizado con éxito.

400: Datos no válidos.

404: Chofer no encontrado.

10. DELETE /choferes/{choferId}

Descripción: Elimina un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer que se quiere eliminar (Integer).

Respuestas:

204: Chofer eliminado con éxito.

404: Chofer no encontrado.

11. GET /choferes/{choferId}/evaluaciones

Descripción: Obtiene las evaluaciones de un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer del cual se quieren consultar sus evaluaciones (Integer).

Respuestas:

200: Lista de evaluaciones del chofer.

404: Chofer no encontrado.

12. GET /choferes/{choferId}/servicios

Descripción: Obtiene los servicios realizados por un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer del cual se quieren consultar sus servicios (Integer).

Respuestas:

200: Lista de servicios del chofer.

404: Chofer no encontrado.

13. GET /clientes/{clienteId}

Descripción: Obtiene la información de un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente que se quiere consultar (Integer).

Respuestas:

200: Información del cliente.

404: Cliente no encontrado.

14. PUT /clientes/{clienteId}

Descripción: Actualiza la información de un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente que se quiere actualizar (Integer).

Parámetros del cuerpo (Body Parameters):

nombre: (Opcional) Nombre del cliente (String).

contrasena: (Opcional) Contraseña del cliente (String).

email: (Opcional) Email del cliente (String).

telefono: (Opcional) Teléfono del cliente (String).

Respuestas:

200: Cliente actualizado con éxito.

400: Datos no válidos.

404: Cliente no encontrado.

15. DELETE /clientes/{clienteId}

Descripción: Elimina un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente que se quiere eliminar (Integer).

Respuestas:

204: Cliente eliminado con éxito.

404: Cliente no encontrado.

16. GET /clientes/{clienteId}/evaluaciones

Descripción: Obtiene las evaluaciones realizadas por un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente del cual se quieren consultar sus evaluaciones (Integer).

Respuestas:

200: Lista de evaluaciones realizadas por el cliente.

404: Cliente no encontrado.

17. GET /clientes/{clienteId}/servicios

Descripción: Obtiene los servicios solicitados por un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente del cual se quieren consultar sus servicios (Integer).

Respuestas:

200: Lista de servicios solicitados por el cliente.

404: Cliente no encontrado.

18. GET /clientes/{clienteId}/chats

Descripción: Obtiene los chats en los que ha participado un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente del cual se quieren consultar sus chats (Integer).

Respuestas:

200: Lista de chats del cliente.

404: Cliente no encontrado.

19. POST /evaluaciones

Descripción: Registra una nueva evaluación en el sistema.

Parámetros del cuerpo (Body Parameters):

clienteID: ID del cliente que realiza la evaluación (Integer).

choferID: ID del chofer que es evaluado (Integer).

comentario: Comentario sobre el chofer, debe tener entre 10 y 500 caracteres (String).

calificacion: Calificación para el chofer, entre 1 y 10 (Integer).

Respuestas:

201: Evaluación registrada con éxito.

400: Datos no válidos.

20. GET /evaluaciones/{evaluacionId}

Descripción: Obtiene la información de una evaluación específica.

Parámetros de ruta (Path Parameters):

evaluacionId: ID de la evaluación que se quiere consultar (Integer).

Respuestas:

200: Información de la evaluación.

404: Evaluación no encontrada.

21. PUT /evaluaciones/{evaluacionId}

Descripción: Actualiza una evaluación específica.

Parámetros de ruta (Path Parameters):

evaluacionId: ID de la evaluación que se quiere actualizar (Integer).

Parámetros del cuerpo (Body Parameters):

comentario: (Opcional) Comentario sobre el chofer (String).

calificacion: (Opcional) Calificación para el chofer (Integer).

Respuestas:

200: Evaluación actualizada con éxito.

400: Datos no válidos.

404: Evaluación no encontrada.

22. DELETE /evaluaciones/{evaluacionId}

Descripción: Elimina una evaluación específica.

Parámetros de ruta (Path Parameters):

evaluacionId: ID de la evaluación que se quiere eliminar (Integer).

Respuestas:

204: Evaluación eliminada con éxito.

404: Evaluación no encontrada.

23. GET /choferes/{choferId}/evaluaciones

Descripción: Obtiene todas las evaluaciones relacionadas con un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer del cual se quieren consultar las evaluaciones (Integer).

Respuestas:

200: Lista de evaluaciones relacionadas con el chofer.

404: Chofer no encontrado.

24. GET /clientes/{clienteId}/evaluaciones

Descripción: Obtiene todas las evaluaciones realizadas por un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente del cual se quieren consultar sus evaluaciones (Integer).

Respuestas:

200: Lista de evaluaciones realizadas por el cliente.

404: Cliente no encontrado.

25. POST /servicios

Descripción: Registra un nuevo servicio en el sistema.

Parámetros del cuerpo (Body Parameters):

clienteID: ID del cliente que solicita el servicio (Integer).

choferID: ID del chofer que atenderá el servicio (Integer).

precio: Precio del servicio (Integer).

tipo: Tipo de servicio (String).

hora: Hora en que se brindará el servicio (Time).

fecha: Fecha en que se brindará el servicio (Date).

estado: Estado actual del servicio (String).

origen: Lugar de origen donde se iniciará el servicio (String).

destino: Lugar de destino a donde se llevará al cliente (String).

Respuestas:

201: Servicio registrado con éxito.

400: Datos no válidos.

26. GET /servicios/{servicioId}

Descripción: Obtiene la información de un servicio específico.

Parámetros de ruta (Path Parameters):

servicioId: ID del servicio que se quiere consultar (Integer).

Respuestas:

200: Información del servicio.

404: Servicio no encontrado.

27. PUT /servicios/{servicioId}

Descripción: Actualiza la información de un servicio específico.

Parámetros de ruta (Path Parameters):

servicioId: ID del servicio que se quiere actualizar (Integer).

Parámetros del cuerpo (Body Parameters):

precio: (Opcional) Precio del servicio (Integer).

tipo: (Opcional) Tipo de servicio (String).

hora: (Opcional) Hora del servicio (Time).

fecha: (Opcional) Fecha del servicio (Date).

estado: (Opcional) Estado del servicio (String).

origen: (Opcional) Lugar de origen del servicio (String).

destino: (Opcional) Lugar de destino del servicio (String).

Respuestas:

200: Servicio actualizado con éxito.

400: Datos no válidos.

404: Servicio no encontrado.

28. DELETE /servicios/{servicioId}

Descripción: Elimina un servicio específico.

Parámetros de ruta (Path Parameters):

servicioId: ID del servicio que se quiere eliminar (Integer).

Respuestas:

204: Servicio eliminado con éxito.

404: Servicio no encontrado.

29. GET /choferes/{choferId}/servicios

Descripción: Obtiene todos los servicios atendidos o por atender de un chofer específico.

Parámetros de ruta (Path Parameters):

choferId: ID del chofer del cual se quieren consultar sus servicios (Integer).

Respuestas:

200: Lista de servicios relacionados con el chofer.

404: Chofer no encontrado.

30. GET /clientes/{clienteId}/servicios

Descripción: Obtiene todos los servicios solicitados por un cliente específico.

Parámetros de ruta (Path Parameters):

clienteId: ID del cliente del cual se quieren consultar sus servicios (Integer).

Respuestas:

200: Lista de servicios solicitados por el cliente.

404: Cliente no encontrado.