

Plan de Pruebas De Aceptación Manual

Introducción

Este documento define el plan de pruebas de aceptación del Sistema de ticketing interno, basado en los requerimientos funcionales representados mediante historias de usuario. El objetivo es validar todas las funcionalidades que el sistema requiere para satisfacer las necesidades del usuario final. La herramienta principal para validar la API será Postman; podrán considerarse otras alternativas si es necesario.

Alcance del Plan de Pruebas

Se validarán las funcionalidades, por medio de los endpoints de la API para cada uno de los roles del sistema, asegurando de esta manera un funcionamiento completo y apropiado de lo que es requerido por el cliente.

Los roles son:

- Empleado
- Agente
- Admin

Ambientes de prueba

Estas pruebas serán realizadas en un entorno principal, durante el desarrollo se podrán probar las diferentes funcionalidades y se registrarán los resultados.

Criterios de Aceptación

Los criterios de aceptación están basados en las historias de usuario, de esta forma se define lo que realmente necesita el usuario principal, según cada rol en la aplicación.

En cada criterio se definió un código de estado y una salida esperada.

Estrategia de pruebas

Se utilizarán tres estrategias de pruebas por caso de prueba con los siguientes enfoques:

1. **Pruebas de solicitud correcta:** Se espera que operaciones exitosas (como el login) retornen un código de estado HTTP que indique éxito (por ejemplo: 200 OK).

2. **Pruebas de error:** Se considerarán casos alternativos, donde se espera que la API tenga un manejo adecuado de los errores.

3. **Pruebas de autenticación:** Se pondrá a prueba todo lo relacionado con el de acceso a los recursos de la API

Casos de Prueba

- CP-01
- **Caso de Prueba:** Iniciar sesión correctamente
- **Endpoint:** localhost:8080/api/auth/login
- **Método HTTP:** POST
- **Descripción:** Autentica un usuario usando email y contraseña y retorna un token valido.
- **Cuerpo de la petición:**

```
{  
  "email": "string",  
  "contrasena": "string"  
}
```

- **Headers:**
 - **Content-Type:** application/json
- **Código de Respuesta Esperado:** 200 OK
- **Respuesta esperada:**

```
{  
  "token": "<jwt-token>"  
}
```

- **Prueba Exitosa si:** La API devuelve un código 200 OK con un token valido

-
- CP-02
 - **Caso de Prueba:** Inicio de sesión fallido con credenciales incorrectas
 - **Endpoint:** localhost:8080/api/auth/login
 - **Método HTTP:** POST
 - **Descripción:** Verifica que la API rechace el inicio de sesión cuando se ingresan credenciales incorrectas, retornando un código 401 Unauthorized con un mensaje de error adecuado.
 - **Cuerpo de la petición:**

```
{  
  "email": "string",  
  "contrasena": "string"  
}
```

- **Headers:**

- **Content-Type:** application/json
- **Código de Respuesta Esperado:** 401 Unauthorized
- **Respuesta esperada:**

```
{
  "error": "Credenciales inválidas. Verifique su email y contraseña."
}
```

- **Prueba Exitosa si:** La API devuelve un código 401 Unauthorized con un mensaje adecuado

-
- CP-03
 - **Caso de Prueba:** Crear un ticket
 - **Endpoint:** localhost:8080/api/tickets
 - **Método HTTP:** POST
 - **Descripción:** Este endpoint permite a un empleado autenticado crear un nuevo ticket en el sistema, proporcionando los datos requeridos como categoría, título, descripción y prioridad.
 - **Cuerpo de la petición:**

```
{
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Error al iniciar sesión",
  "descripcion": "No puedo ingresar al sistema con mis credenciales.",
  "prioridad": "urgente"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 201 Created
- **Respuesta esperada:**

```
{
  "id": 456,
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Error al iniciar sesión",
  "descripcion": "No puedo ingresar al sistema con mis credenciales.",
  "prioridad": "urgente",
  "estado": "pendiente",
  "fecha_creacion": "2025-09-28T10:15:00Z"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve un código de estado HTTP 201 Created.
 - La respuesta contiene un objeto JSON con los datos del ticket creado, incluyendo su ID, fecha_creacion, y un estado inicial.

-
- CP-04
 - **Caso de Prueba:** Crear un ticket con datos inválidos
 - **Endpoint:** localhost:8080/api/tickets
 - **Método HTTP:** POST
 - **Descripción:** Este test valida que el sistema rechace correctamente la creación de un ticket cuando los datos enviados no cumplen con las reglas de validación (campos faltantes, vacíos o incorrectos).
 - **Cuerpo de la petición:**

```
{
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Error al iniciar sesión",
  "descripcion": "No puedo ingresar al sistema con mis credenciales.",
  "prioridad": "urgente"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Datos inválidos en la solicitud.",
  "detalles": [
    "El campo 'titulo' no puede estar vacío",
    "El campo 'prioridad' debe ser: baja, media, alta, urgente"
  ]
}
```

- **Prueba Exitosa si:**
 - El sistema rechaza la solicitud con un código de estado HTTP 400 Bad Request.
 - La respuesta contiene un mensaje de error indicando claramente qué campos son inválidos o faltan.

-
- CP-05
 - **Caso de Prueba:** Crear un ticket con recurso no encontrado
 - **Endpoint:** localhost:8080/api/tickets
 - **Método HTTP:** POST
 - **Descripción:** Este caso de prueba verifica que el sistema retorne un error 404 Not Found cuando se intenta crear un ticket con referencias a recursos inexistentes, como un usuario_id, categoria_id o prioridad que no existen en la base de datos.
 - **Cuerpo de la petición:**

```
{
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Error al iniciar sesión",
  "descripcion": "No puedo ingresar al sistema con mis credenciales.",
  "prioridad": "urgente"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{
  "error": "Recurso no encontrado.",
  "detalles": "El usuario_id 123 no existe en la base de datos."
}
```

- **Prueba Exitosa si:**
 - El sistema retorna el código de estado HTTP 404 Not Found.
 - El cuerpo de la respuesta indica claramente cuál recurso no fue encontrado.

-
- CP-06
 - **Caso de Prueba:** Error del servidor al crear un ticket
 - **Endpoint:** localhost:8080/api/tickets
 - **Método HTTP:** POST
 - **Descripción:** Verificar que, ante un fallo interno al crear un ticket, el sistema devuelva 500 Internal Server Error con un mensaje genérico sin exponer detalles técnicos.
 - **Cuerpo de la petición:**

```
{
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Error al iniciar sesión",
  "descripcion": "No puedo ingresar al sistema con mis credenciales.",
  "prioridad": "urgente"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.
-

- CP-07
- **Caso de Prueba:** Obtener tickets creados por un empleado
- **Endpoint:** localhost:8080/api/tickets?estado=enProgreso&prioridad=urgente&usuario_id=123
- **Método HTTP:** GET
- **Descripción:** Este caso de prueba valida que el sistema devuelva correctamente la lista de tickets creados por los empleados
- **Cuerpo de la petición:** No aplica.
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 200 OK
- **Respuesta esperada:**

```
[
  {
    "id": 101,
    "usuario_id": 123,
    "categoria_id": 4,
    "titulo": "Falla en sistema de correo",
    "descripcion": "No puedo enviar correos desde ayer.",
    "prioridad": "importante",
    "estado": "enProgreso",
    "fecha_creacion": "2025-09-25T09:00:00Z",
    "fecha_cierre": null
  },
  {
    "id": 102,
    "usuario_id": 123,
    "categoria_id": 1,
    "titulo": "Solicitud de acceso VPN",
    "descripcion": "Necesito acceso remoto para trabajar desde casa.",
    "prioridad": "urgente",
    "estado": "cerrado",
    "fecha_creacion": "2025-09-20T14:30:00Z",
    "fecha_cierre": "2025-09-21T10:15:00Z"
  }
]
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta contiene un arreglo JSON con todos los tickets disponibles para el empleado autenticado
 - Si no hay ningún ticket entonces devuelve arreglo vacío
-

- CP-08
- **Caso de Prueba:** Consultar tickets con Cuerpo de la petición inválidos
- **Endpoint:** localhost:8080/api/tickets?estado=enProgreso&prioridad=urgente&usuario_id=123
- **Método HTTP:** GET
- **Descripción:** Este caso de prueba valida que el sistema maneje correctamente los errores cuando se envían parámetros de consulta con valores no permitidos o en formato incorrecto, como una prioridad inexistente o un estado no válido.
- **Cuerpo de la petición:** No aplica.
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Parámetros de consulta inválidos.",
  "detalles": [
    "El valor 'urgenteee' no es una prioridad válida",
    "El estado 'enProgreso' no es reconocido"
  ]
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - Se rechaza la solicitud sin ejecutar la consulta.
 - La respuesta contiene un mensaje de error que describe los parámetros inválidos.
 - No se devuelve ninguna lista de tickets.

-
- CP-09
 - **Caso de Prueba:** Error del servidor al consultar tickets
 - **Endpoint:** localhost:8080/api/tickets?estado=enProgreso&prioridad=urgente&usuario_id=123
 - **Método HTTP:** GET
 - **Descripción:** Este caso de prueba valida que el sistema maneje correctamente los errores internos inesperados.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.
-

- CP-10
- **Caso de Prueba:** Obtener información completa de un ticket
- **Endpoint:** localhost:8080/api/tickets/{idTicket}
- **Método HTTP:** GET
- **Descripción:** Este caso de prueba valida que el sistema devuelva correctamente la información completa de un ticket específico al consultarlo por su ID, siempre que el ticket exista y el usuario tenga permisos para verlo.
- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 200 OK
- **Respuesta esperada:**

```
{
  "id": 456,
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Falla en impresora",
  "descripcion": "La impresora no responde a los comandos de impresión.",
  "prioridad": "urgente",
  "estado": "enProgreso",
  "fecha_creacion": "2025-09-25T09:00:00Z",
  "fecha_cierre": null
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta contiene un objeto JSON con todos los detalles del ticket

```
{
  "id": 456,
  "usuario_id": 123,
  "categoria_id": 2,
  "titulo": "Falla en impresora",
  "descripcion": "La impresora no responde a los comandos de impresión.",
  "prioridad": "urgente",
  "estado": "enProgreso",
  "fecha_creacion": "2025-09-25T09:00:00Z",
  "fecha_cierre": null
}
```

-
- CP-11
 - **Caso de Prueba:** Solicitud inválida al consultar detalles del ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}
 - **Método HTTP:** GET

- **Descripción:** Este caso de prueba valida que el sistema maneje correctamente los errores cuando el idTicket es inválido (por ejemplo, no es un número, es nulo o tiene un formato incorrecto).
- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "ID de ticket inválido.",
  "detalles": "El parámetro idTicket debe ser un número válido."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta contiene un mensaje de error claro indicando que el ID del ticket no es válido.

-
- CP-12
 - **Caso de Prueba:** Ticket no encontrado
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}
 - **Método HTTP:** GET
 - **Descripción:** Este caso de prueba valida que el sistema responda adecuadamente cuando se solicita el detalle de un ticket que no existe en la base de datos.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 404 Not Found
 - **Respuesta esperada:**

```
{
  "mensaje": "No se encontró el ticket solicitado."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta contiene un mensaje claro que indica que el ticket no fue encontrado.
 - No se devuelve ningún objeto de ticket.

-
- CP-13
 - **Caso de Prueba:** Error del servidor al obtener detalles del ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}
 - **Método HTTP:** GET
 - **Descripción:** Este caso de prueba valida que el sistema maneje correctamente los errores internos inesperados al intentar obtener el detalle de un ticket.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error

- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**

- El sistema responde con código HTTP 500 Internal Server Error.
- La respuesta incluye un mensaje de error genérico sin exponer detalles.

- CP-14

- **Caso de Prueba:** Ver proceso del ticket (comentarios)

- **Endpoint:** localhost:8080/api/tickets/{idTicket}/comentarios

- **Método HTTP:** GET

- **Descripción:** Este caso de prueba valida que el sistema devuelva correctamente la lista de comentarios asociados a un ticket específico, mostrando el historial del proceso desde su creación hasta la resolución.

- **Cuerpo de la petición:** No aplica

- **Headers:** Authorization: Bearer

- **Código de Respuesta Esperado:** 200 OK

- **Respuesta esperada:**

```
[
  {
    "id": 301,
    "ticket_id": 456,
    "usuario_id": 789,
    "comentario": "Estamos revisando el problema.",
    "fecha_creacion": "2025-09-25T10:30:00Z"
  },
  {
    "id": 302,
    "ticket_id": 456,
    "usuario_id": 789,
    "comentario": "Nuevo informe, se ha encontrado...",
    "fecha_creacion": "2025-09-26T08:15:00Z"
  }
]
```

- **Prueba Exitosa si:**

- El sistema responde con código HTTP 200 OK.
- La respuesta contiene un arreglo JSON con todos los comentarios registrados para el ticket.

- CP-15

- **Caso de Prueba:** Cuerpo de la petición inválidos al consultar proceso del ticket

- **Endpoint:** localhost:8080/api/tickets/{idTicket}/comentarios
- **Método HTTP:** GET
- **Descripción:** Este caso de prueba valida que el sistema maneje correctamente los errores cuando el parámetro idTicket es inválido
- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "ID de ticket inválido.",
  "detalles": "El parámetro idTicket debe ser un número válido."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta contiene un mensaje de error claro indicando que el parámetro idTicket no es válido.
 - No se devuelve ninguna lista de comentarios.

- CP-16
- **Caso de Prueba:** Proceso del ticket no encontrado
- **Endpoint:** localhost:8080/api/tickets/{idTicket}/comentarios
- **Método HTTP:** GET
- **Descripción:** Este caso de prueba valida que el sistema responda adecuadamente cuando se consultan los comentarios de un ticket que no existe en la base de datos
- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{
  "mensaje": "No se encontro un ticket con este id"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta contiene un mensaje claro que indique que no se encontró el ticket
 - No se devuelve ningún arreglo.

- CP-17
- **Caso de Prueba:** Error del servidor al consultar proceso del ticket
- **Endpoint:** localhost:8080/api/tickets/{idTicket}/comentarios
- **Método HTTP:** GET
- **Descripción:** Este caso de prueba valida que el sistema maneje correctamente errores internos inesperados al intentar obtener los comentarios asociados a un ticket, como fallas en la base de datos, excepciones no controladas o

problemas de conexión.

- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-18
 - **Caso de Prueba:** Ver notificaciones del sistema
 - **Endpoint:** localhost:8080/api/usuarios/{idUserio}/notificaciones
 - **Método HTTP:** GET
 - **Descripción:** Este caso de prueba valida que un usuario autenticado pueda consultar la lista de notificaciones generadas por el sistema, relacionadas con sus tickets, asignaciones o actualizaciones.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**

```
[
  {
    "id": 501,
    "titulo": "Ticket actualizado",
    "descripcion": "El ticket #456 ha cambiado de estado a En Progreso",
    "fecha_creacion": "2025-09-26T12:00:00Z"
  },
  {
    "id": 502,
    "titulo": "Nuevo ticket asignado",
    "descripcion": "Se le ha asignado el ticket #457",
    "fecha_creacion": "2025-09-26T14:30:00Z"
  }
]
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta incluye un arreglo con las notificaciones asociadas al usuario.
 - Si el usuario no tiene notificaciones, se retorna una lista vacía sin error.
-

- CP-19
- **Caso de Prueba:** Solicitud inválida al consultar notificaciones
- **Endpoint:** localhost:8080/api/usuarios/{idUsuario}/notificaciones
- **Método HTTP:** GET
- **Descripción:** Verificar que el sistema responda con error cuando se realiza una solicitud con un idUsuario invalido.
- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Solicitud inválida.",
  "detalles": "Los parámetros enviados para consultar notificaciones son incorrectos."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta incluye un mensaje de error claro indicando que la solicitud es inválida.
 - No se retorna información de notificaciones.

-
- CP-20
 - **Caso de Prueba:** Usuario no encontrado al obtener notificaciones
 - **Endpoint:** localhost:8080/api/usuarios/{idUsuario}/notificaciones
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema responda adecuadamente cuando no exista el usuario solicitado.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 404 Not Found
 - **Respuesta esperada:**

```
{
  "mensaje": "Usuario no encontrado."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta contiene un mensaje claro indicando que no se encontró el usuario con el idUsuario especificado.

-
- CP-21
 - **Caso de Prueba:** Error del servidor al obtener notificaciones
 - **Endpoint:** localhost:8080/api/usuarios/{idUsuario}/notificaciones
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema responda de forma controlada cuando ocurre un error interno inesperado al intentar recuperar las notificaciones de un usuario.
 - **Cuerpo de la petición:** No aplica.

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-22
 - **Caso de Prueba:** Modificar contraseña y datos personales
 - **Endpoint:** localhost:8080/api/usuarios/{idUserio}
 - **Método HTTP:** PATCH
 - **Descripción:** Verificar que el usuario pueda actualizar correctamente sus datos personales como email personal y su contraseña mediante una solicitud parcial de actualización.
 - **Cuerpo de la petición:**

```
{
  "email_personal": "nuevo.correo@example.com",
  "contrasena": "NuevaContraseña123"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 200 OK
- **Respuesta esperada:**

```
{
  "id": 123,
  "email_personal": "nuevo.correo@example.com"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta incluye el objeto usuario actualizado con los nuevos valores (sin mostrar la contraseña en texto plano).

-
- CP-23
 - **Caso de Prueba:** Error 400 - Datos inválidos al modificar contraseña o perfil
 - **Endpoint:** localhost:8080/api/usuarios/{idUserio}
 - **Método HTTP:** PATCH

- **Descripción:** Validar que el sistema rechace la actualización cuando los datos enviados en la solicitud son inválidos o no cumplen con las validaciones establecidas (ejemplo: email con formato incorrecto, contraseña demasiado corta, campos vacíos).
- **Cuerpo de la petición:**

```
{
  "email_personal": "nuevo.correo@example.com",
  "contrasena": "NuevaContraseña123"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Datos inválidos.",
  "detalles": [
    "El correo electrónico no es válido",
    "La contraseña debe tener mínimo 8 caracteres"
  ]
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta incluye un mensaje claro indicando qué campos no cumplen con las validaciones (ejemplo: “El correo electrónico no es válido” o “La contraseña debe tener mínimo 8 caracteres”).

-
- CP-24
 - **Caso de Prueba:** Error 404 - Usuario no encontrado al modificar datos
 - **Endpoint:** localhost:8080/api/usuarios/{idUserio}
 - **Método HTTP:** PATCH
 - **Descripción:** Verificar que el sistema devuelva un error cuando se intente actualizar los datos de un usuario que no existe en el sistema.
 - **Cuerpo de la petición:**

```
{
  "email_personal": "nuevo.correo@example.com",
  "contrasena": "NuevaContraseña123"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{
  "mensaje": "Usuario no encontrado."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta incluye un mensaje indicando que el usuario no existe o no fue encontrado.

-
- CP-25
 - **Caso de Prueba:** Error 500 - Error del servidor al actualizar información del usuario
 - **Endpoint:** localhost:8080/api/usuarios/{idUserio}
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que el sistema maneje correctamente un fallo interno inesperado al intentar actualizar los datos de un usuario.
 - **Cuerpo de la petición:**

```
{
  "email_personal": "nuevo.correo@example.com",
  "contrasena": "NuevaContraseña123"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-26
 - **Caso de Prueba:** Ver tickets asignados al agente
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=enProgreso&estado=pendiente
 - **Método HTTP:** GET
 - **Descripción:** Validar que el agente pueda consultar la lista de tickets que le han sido asignados, filtrando por estados, programado, en progreso y pendiente.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**


```
[
  {
    "id": 101,
    "titulo": "Falla en sistema de correo",
    "estado": "enProgreso"
  },
  {
    "id": 102,
    "titulo": "Solicitud de acceso VPN",
    "estado": "pendiente"
  }
]
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta incluye una lista de tickets asignados al agente.
 - Solo se muestran los tickets con estado programado, enProgreso o pendiente.

-
- CP-27
 - **Caso de Prueba:** Solicitud inválida al consultar tickets asignados
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=enProgreso&estado=pendiente
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema devuelva un error 400 Bad Request cuando se envían Cuerpo de la petición inválidos o mal formados al consultar los tickets de un agente.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 400 Bad Request
 - **Respuesta esperada:**

```
{
  "error": "Solicitud inválida.",
  "detalles": "Los parámetros enviados para consultar tickets asignados son incorrectos."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - El mensaje de error indica que parámetros de la petición son inválidos.
 - No se retorna ninguna lista de tickets.

-
- CP-28
 - **Caso de Prueba:** No se encontró al agente
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=enProgreso&estado=pendiente
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema responda correctamente cuando el agente no es encontrado

- **Cuerpo de la petición:** No aplica
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{  
  "mensaje": "No se encontro este agente"  
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - El mensaje indica que no se encontró un agente con ese id.
 - No se retorna ninguna lista de tickets.

-
- CP-29
 - **Caso de Prueba:** Error del servidor al consultar tickets asignados
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=enProgreso&estado=pendiente
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema maneje correctamente un error interno al intentar obtener los tickets asignados a un agente.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{  
  "error": "Error interno del servidor. Intente nuevamente más tarde."  
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-30
 - **Caso de Prueba:** Conseguir historial completo de tickets
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=cerrado
 - **Método HTTP:** GET
 - **Descripción:** Validar que el agente pueda consultar el historial completo de tickets que ya han sido cerrados.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**

```
[
  {
    "id": 103,
    "titulo": "Ticket cerrado #103",
    "estado": "cerrado"
  },
  {
    "id": 104,
    "titulo": "Ticket cerrado #104",
    "estado": "cerrado"
  }
]
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta incluye una lista de tickets cerrados asociados al agente.

-
- CP-31
 - **Caso de Prueba:** Parámetros inválidos al consultar historial de tickets
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=cerrado
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema devuelva error 400 Bad Request cuando se envíen parámetros con mal formato
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 400 Bad Request
 - **Respuesta esperada:**

```
{
  "error": "Solicitud inválida.",
  "detalles": "Los parámetros enviados para consultar historial de tickets son incorrectos."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - El mensaje de error indica que los parámetros de la petición son inválidos.
 - No se retorna ningún listado de tickets.

-
- CP-32
 - **Caso de Prueba:** Agente no encontrado al buscar historial de tickets
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=cerrado
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema responda correctamente cuando no existe el agente
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer

- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{  
  "mensaje": "No se encontro este agente"  
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - El mensaje indica que no se encontró el agente.
 - No se retorna ningún listado de tickets.

-
- CP-33
 - **Caso de Prueba:** Error del servidor al obtener historial de tickets
 - **Endpoint:** localhost:8080/api/agentes/{idAgente}/tickets?estado=cerrado
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema gestione adecuadamente un error interno inesperado al intentar obtener el historial de tickets cerrados de un agente.
 - **Cuerpo de la petición:** No aplica
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{  
  "error": "Error interno del servidor. Intente nuevamente más tarde."  
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-34
 - **Caso de Prueba:** Crear un comentario en un ticket
 - **Endpoint:** localhost:8080/api/tickets/{ticketId}/comments
 - **Método HTTP:** POST
 - **Descripción:** Validar que un agente pueda agregar correctamente un comentario en un ticket asignado.
 - **Cuerpo de la petición:**

```
{  
  "usuario_id": 45,  
  "comentario": "Estamos revisando el problema y pronto tendrás una solución."  
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer

- **Código de Respuesta Esperado:** 201 Created
- **Respuesta esperada:**

```
{
  "id": 987,
  "ticketId": 123,
  "usuario_id": 45,
  "comentario": "Estamos revisando el problema y pronto tendrás una solución.",
  "fecha_creacion": "2025-09-29T20:00:00Z"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 201 Created.
 - Se devuelve el objeto del comentario creado con su id único.
 - El ticketId corresponde al ticket en el cual se agregó el comentario.
 - El usuario_id es el mismo que envió la petición.
 - El texto del comentario coincide con lo enviado en la solicitud.
 - Se incluye la fecha_creacion generada automáticamente por el sistema.

-
- CP-35
 - **Caso de Prueba:** Datos inválidos al crear comentario
 - **Endpoint:** localhost:8080/api/tickets/{ticketId}/comments
 - **Método HTTP:** POST
 - **Descripción:** Validar que el sistema rechace la creación de un comentario cuando los datos enviados no cumplen con el formato o campos requeridos.
 - **Cuerpo de la petición:**

```
{
  "usuario_id": 45,
  "comentario": "Estamos revisando el problema y pronto tendrás una solución."
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Los datos enviados son inválidos o faltan campos obligatorios"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta incluye un mensaje indicando que los datos son inválidos o faltan campos obligatorios.
-

- CP-36
- **Caso de Prueba:** Ticket no encontrado para comentar
- **Endpoint:** localhost:8080/api/tickets/{ticketId}/comments
- **Método HTTP:** POST
- **Descripción:** Validar que el sistema devuelva un error cuando se intente agregar un comentario a un ticket inexistente.
- **Cuerpo de la petición:**

```
{
  "usuario_id": 45,
  "comentario": "Estamos revisando el problema y pronto tendrás una solución."
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{
  "error": "El ticket no existe o no fue encontrado"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta incluye un mensaje indicando que el ticket no existe o no fue encontrado.

-
- CP-37
 - **Caso de Prueba:** Error del servidor al crear comentario en el ticket
 - **Endpoint:** localhost:8080/api/tickets/{ticketId}/comments
 - **Método HTTP:** POST
 - **Descripción:** Validar que el sistema responda adecuadamente con un error genérico cuando ocurra una falla interna al intentar crear un comentario en un ticket.
 - **Cuerpo de la petición:**

```
{
  "usuario_id": 45,
  "comentario": "Estamos revisando el problema y pronto tendrás una solución."
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-38
 - **Caso de Prueba:** Actualizar prioridad de un ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/prioridad
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que el sistema permita actualizar correctamente la prioridad de un ticket existente.
 - **Cuerpo de la petición:**

```
{
  "prioridad": "importante"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 200 OK
- **Respuesta esperada:**

```
{
  "id": 123,
  "titulo": "Problema con login",
  "descripcion": "El usuario no puede iniciar sesión",
  "prioridad": "importante",
  "estado": "abierto",
  "fecha_creacion": "2025-09-28T14:30:00Z"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta incluye la información del ticket actualizado con la nueva prioridad.

-
- CP-39
 - **Caso de Prueba:** Datos inválidos al actualizar prioridad del ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/prioridad
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que el sistema rechace la actualización de la prioridad de un ticket cuando los datos enviados no son válidos.
 - **Cuerpo de la petición:**

```
{
  "prioridad": "importante"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "El valor de la prioridad no es válido"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta incluye un mensaje indicando que el valor de la prioridad o el id del ticket no es válido.

-
- CP-40
 - **Caso de Prueba:** Ticket no encontrado para actualizar prioridad
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/prioridad
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que el sistema devuelva un error cuando se intenta actualizar la prioridad de un ticket inexistente.
 - **Cuerpo de la petición:**

```
{
  "prioridad": "importante"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{
  "error": "El ticket no existe"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta incluye un mensaje indicando que el ticket no existe.

-
- CP-41
 - **Caso de Prueba:** Error del servidor al actualizar prioridad del ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/prioridad
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que el sistema maneje adecuadamente un error interno al intentar actualizar la prioridad de un ticket.
 - **Cuerpo de la petición:**


```
{
  "prioridad": "importante"
}
```

- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-42
 - **Caso de Prueba:** Cerrar un ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/cerrar
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que un ticket existente pueda ser cerrado correctamente, cambiando su estado a cerrado.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**

```
{
  "id": 123,
  "titulo": "Problema con login",
  "descripcion": "El usuario no puede iniciar sesión",
  "prioridad": "alta",
  "estado": "cerrado",
  "fecha_creacion": "2025-09-28T14:30:00Z",
  "fecha_cierre": "2025-09-29T20:00:00Z"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 200 OK.
 - La respuesta incluye el ticket actualizado con estado cerrado.

-
- CP-43
 - **Caso de Prueba:** Solicitud inválida al cerrar el ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/cerrar
 - **Método HTTP:** PATCH

- **Descripción:** Validar que, si la solicitud enviada para cerrar un ticket contiene datos inválidos o mal formateados, el sistema la rechace.
- **Cuerpo de la petición:** No aplica.
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Solicitud inválida para cerrar el ticket"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 400 Bad Request.
 - La respuesta incluye un mensaje indicando que la solicitud es inválida.

-
- CP-44
 - **Caso de Prueba:** Ticket no encontrado para cerrar
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/cerrar
 - **Método HTTP:** PATCH
 - **Descripción:** Validar que, si se intenta cerrar un ticket inexistente o cuyo ID no está registrado en el sistema, la operación falle con el código adecuado.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 404 Not Found
 - **Respuesta esperada:**

```
{
  "error": "El ticket no fue encontrado"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 404 Not Found.
 - La respuesta incluye un mensaje que indique que el ticket no fue encontrado.

-
- CP-45
 - **Caso de Prueba:** Error del servidor al cerrar el ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}/cerrar
 - **Método HTTP:** PATCH
 - **Descripción:** Validar el comportamiento del sistema cuando ocurre un error interno al intentar cerrar un ticket existente.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.
-

- CP-46
- **Caso de Prueba:** Reasignar tickets a los agentes
- **Endpoint:** localhost:8080/api/tickets/{idTicket}
- **Método HTTP:** POST
- **Descripción:** Validar que el administrador pueda reasignar un ticket a un agente disponible.
- **Cuerpo de la petición:**

```
{
  "agenteId": 123
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 201 Created
- **Respuesta esperada:**

```
{
  "id": 123,
  "titulo": "Problema con login",
  "descripcion": "El usuario no puede iniciar sesión",
  "agenteId": 456,
  "estado": "asignado",
  "fecha_creacion": "2025-09-28T14:30:00Z"
}
```

- **Prueba Exitosa si:**
 - El ticket es reasignado correctamente al agente.
 - La respuesta incluye información del ticket con el nuevo agentId asignado.
-

- CP-47
- **Caso de Prueba:** Datos inválidos al reasignar ticket a un agente
- **Endpoint:** localhost:8080/api/tickets/{idTicket}
- **Método HTTP:** POST
- **Descripción:** Validar que el sistema rechace la reasignación si los datos enviados no son válidos.
- **Cuerpo de la petición:**

```
{
  "agenteId": 123
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Los datos enviados son inválidos"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve el código 400.
 - La respuesta incluye un mensaje indicando que los datos enviados son inválidos.

-
- CP-48
 - **Caso de Prueba:** Ticket o agente no encontrado para reasignación
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}
 - **Método HTTP:** POST
 - **Descripción:** Validar que el sistema notifique cuando el ticket o el agente especificado no existan.
 - **Cuerpo de la petición:**

```
{
  "agenteId": 123
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 404 Not Found
- **Respuesta esperada:**

```
{
  "error": "No se encontró el ticket o el agente solicitado"
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código 404.
 - El mensaje de error indica que no se encontró el ticket o el agente solicitado.

-
- CP-49
 - **Caso de Prueba:** Error del servidor al reasignar ticket
 - **Endpoint:** localhost:8080/api/tickets/{idTicket}
 - **Método HTTP:** POST
 - **Descripción:** Validar que el sistema maneje correctamente errores internos al intentar reasignar un ticket.

- **Cuerpo de la petición:**

```
{
  "agenteId": 123
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-50
 - **Caso de Prueba:** Crear categorías para los tickets
 - **Endpoint:** localhost:8080/api/categorias
 - **Método HTTP:** POST
 - **Descripción:** Validar que el administrador pueda crear una nueva categoría para organizar los tickets.
 - **Cuerpo de la petición:**

```
{
  "nombre": "Soporte Técnico",
  "descripcion": "Problemas relacionados con hardware y software"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 201 Created
- **Respuesta esperada:**

```
{
  "id": 10,
  "nombre": "Soporte Técnico",
  "descripcion": "Problemas relacionados con hardware y software"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 201.
 - La respuesta incluye la categoría creada con su id, nombre y descripcion.
-

- CP-51
- **Caso de Prueba:** Datos inválidos al crear categoría de ticket
- **Endpoint:** localhost:8080/api/categorias
- **Método HTTP:** POST
- **Descripción:** Validar que el sistema no permita crear una categoría con datos incompletos o inválidos.
- **Cuerpo de la petición:**

```
{
  "nombre": "Soporte Técnico",
  "descripcion": "Problemas relacionados con hardware y software"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Los datos enviados son inválidos o faltan campos obligatorios"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 400.
 - La respuesta incluye un mensaje indicando que los datos son inválidos o faltantes.
-

- CP-52
- **Caso de Prueba:** Error del servidor al crear categoría de ticket
- **Endpoint:** localhost:8080/api/categorias
- **Método HTTP:** POST
- **Descripción:** Validar que el sistema maneje correctamente errores internos al crear una categoría.
- **Cuerpo de la petición:**

```
{
  "nombre": "Soporte Técnico",
  "descripcion": "Problemas relacionados con hardware y software"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.
-

- CP-53
- **Caso de Prueba:** Crear cuentas de empleados y agentes
- **Endpoint:** localhost:8080/api/usuarios
- **Método HTTP:** POST
- **Descripción:** Validar que el administrador pueda crear cuentas de usuarios con rol de empleado o agente.
- **Cuerpo de la petición:**

```
{
  "nombre": "Carlos Gómez",
  "email": "carlos.gomez@empresa.com",
  "contrasena": "Password123",
  "rol": "AGENTE"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 201 Created
- **Respuesta esperada:**

```
{
  "id": 101,
  "nombre": "Carlos Gómez",
  "email": "carlos.gomez@empresa.com",
  "rol": "AGENTE"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 201.
 - La respuesta incluye los datos del usuario creado (id, nombre, email, rol).
-

- CP-54
- **Caso de Prueba:** Datos inválidos al crear cuenta de usuario
- **Endpoint:** localhost:8080/api/usuarios
- **Método HTTP:** POST
- **Descripción:** Validar que no se permita crear un usuario con datos incompletos o inválidos.
- **Cuerpo de la petición:**

```
{
  "nombre": "Carlos Gómez",
  "email": "carlos.gomez@empresa.com",
  "contrasena": "Password123",
  "rol": "AGENTE"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{
  "error": "Los datos enviados son inválidos o faltan campos obligatorios"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 400.
 - La respuesta incluye un mensaje que indica qué datos son inválidos o faltan.

-
- CP-55

- **Caso de Prueba:** Error del servidor al registrar cuenta de usuario
- **Endpoint:** localhost:8080/api/usuarios
- **Método HTTP:** POST
- **Descripción:** Validar que el sistema maneje errores internos al intentar crear una cuenta.
- **Cuerpo de la petición:**

```
{
  "nombre": "Carlos Gómez",
  "email": "carlos.gomez@empresa.com",
  "contrasena": "Password123",
  "rol": "AGENTE"
}
```

- **Headers:** Content-Type: application/json Authorization: Bearer
- **Código de Respuesta Esperado:** 500 Internal Server Error
- **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```


- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-56
 - **Caso de Prueba:** Conseguir todos los tickets, filtrados por estado, prioridad, agente o fecha
 - **Endpoint:** localhost:8080/api/tickets?localhost:8080/api/tickets?estado=pendiente&prioridad=urgente&agenteId=12&fecha=2025-09-25
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema permita consultar la lista de tickets aplicando filtros por estado, prioridad, agente o fecha.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**

```
[
  {
    "id": 123,
    "titulo": "Problema con login",
    "descripcion": "El usuario no puede iniciar sesión",
    "prioridad": "alta",
    "estado": "abierto",
    "agenteId": 45,
    "fecha_creacion": "2025-09-28T14:30:00Z"
  },
  {
    "id": 124,
    "titulo": "Error en módulo de pagos",
    "descripcion": "Falla al procesar pagos",
    "prioridad": "media",
    "estado": "en progreso",
    "agenteId": 46,
    "fecha_creacion": "2025-09-28T15:00:00Z"
  }
]
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 200.
 - La respuesta incluye una lista de tickets que cumplen con los filtros aplicados.

-
- CP-57
 - **Caso de Prueba:** Filtros inválidos al consultar tabla de tickets
 - **Endpoint:** localhost:8080/api/tickets?localhost:8080/api/tickets?estado=pendiente&prioridad=urgente&agenteId=12&fecha=2025-09-25

- **Método HTTP:** GET
- **Descripción:** Validar que el sistema rechace solicitudes con filtros inválidos o valores no reconocidos.
- **Cuerpo de la petición:** No aplica.
- **Headers:** Authorization: Bearer
- **Código de Respuesta Esperado:** 400 Bad Request
- **Respuesta esperada:**

```
{  
  "error": "Filtros inválidos o valores no reconocidos"  
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 400.
 - La respuesta incluye un mensaje que indique qué filtros son inválidos.

-
- CP-58
 - **Caso de Prueba:** Error del servidor al obtener tickets filtrados
 - **Endpoint:** localhost:8080/api/tickets?localhost:8080/api/tickets?estado=pendiente&prioridad=urgente&agenteId=12&fecha=2025-09-25
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema maneje correctamente errores internos al consultar la tabla de tickets.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{  
  "error": "Error interno del servidor. Intente nuevamente más tarde."  
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-59
 - **Caso de Prueba:** Exportar informes del sistema en PDF
 - **Endpoint:** localhost:8080/api/admin/estadisticas/exportar/pdf
 - **Método HTTP:** GET
 - **Descripción:** Validar que el administrador pueda exportar correctamente los informes del sistema en formato PDF.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**

```
{
  "mensaje": "Archivo PDF generado correctamente",
  "url": "/descargas/informe_2025-09-29.pdf"
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 200.
 - El archivo PDF se descarga correctamente y contiene los datos del informe solicitado.

-
- CP-60
 - **Caso de Prueba:** Error del servidor al generar archivo PDF
 - **Endpoint:** localhost:8080/api/admin/estadisticas/exportar/pdf
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema maneje correctamente un error interno inesperado al generar el archivo PDF.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.

-
- CP-61
 - **Caso de Prueba:** Obtener métricas del sistema
 - **Endpoint:** localhost:8080/api/admin/estadisticas
 - **Método HTTP:** GET
 - **Descripción:** Validar que el administrador pueda visualizar correctamente el dashboard con métricas generales del sistema.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 200 OK
 - **Respuesta esperada:**

```
{
  "tickets_abiertos": 15,
  "tickets_cerrados": 40,
  "empleados_activos": 25,
  "agentes_activos": 5
}
```

- **Prueba Exitosa si:**
 - El sistema devuelve código 200.
 - La respuesta contiene métricas en formato JSON (ej. tickets abiertos, cerrados, usuarios activos).

-
- CP-62
 - **Caso de Prueba:** Error del servidor al buscar métricas
 - **Endpoint:** localhost:8080/api/admin/estadisticas
 - **Método HTTP:** GET
 - **Descripción:** Validar que el sistema maneje correctamente un error interno inesperado al obtener métricas.
 - **Cuerpo de la petición:** No aplica.
 - **Headers:** Authorization: Bearer
 - **Código de Respuesta Esperado:** 500 Internal Server Error
 - **Respuesta esperada:**

```
{
  "error": "Error interno del servidor. Intente nuevamente más tarde."
}
```

- **Prueba Exitosa si:**
 - El sistema responde con código HTTP 500 Internal Server Error.
 - La respuesta incluye un mensaje de error genérico sin exponer detalles.