

Pruebas Postman - Sistema de Subastas

Configuración Base

Base URL del API Gateway: `http://localhost:8000`

API 1.1: Servicio de Autenticación y Gestión de Cuenta

1. Registro de Usuario

Endpoint: `POST http://localhost:8000/api/v1/auth/register`

Headers:

Content-Type: application/json

Body (JSON):

```
json

{
  "username": "participante1",
  "email": "participante1@espe.edu.ec",
  "password": "password123",
  "firstName": "Juan",
  "lastName": "Pérez",
  "phoneNumber": "0987654321"
}
```

2. Login de Usuario

Endpoint: `POST http://localhost:8000/api/v1/auth/login`

Headers:

Content-Type: application/json

Body (JSON):

```
json

{
  "email": "participante1@espe.edu.ec",
  "password": "password123"
}
```

Response esperada:

```
json

{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "username": "participante1",
  "email": "participante1@espe.edu.ec",
  "expiresIn": 86400
}
```

3. Refresh Token

Endpoint: POST http://localhost:8000/api/v1/auth/refreshToken

Headers:

Content-Type: application/json

Body (JSON):

```
json

{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

4. Cambiar Contraseña Propia

Endpoint: PUT http://localhost:8000/api/v1/users/me/password

Headers:

Content-Type: application/json
Authorization: Bearer {{accessToken}}

Body (JSON):

```
json

{
  "oldPassword": "password123",
  "newPassword": "newPassword456"
}
```

API 1.2: Servicio de Administración de Usuarios

1. Listar Usuarios (Solo Administrador)

Endpoint: GET `http://localhost:8000/api/v1/admin/users?page=0&limit=10`

Headers:

Authorization: Bearer {{adminToken}}

2. Obtener Usuario por ID

Endpoint: GET `http://localhost:8000/api/v1/admin/users/1`

Headers:

Authorization: Bearer {{adminToken}}

3. Actualizar Usuario

Endpoint: PUT `http://localhost:8000/api/v1/admin/users/1`

Headers:

Content-Type: application/json

Authorization: Bearer {{adminToken}}

Body (JSON):

```
json

{
  "role": "MODERADOR",
  "isActive": true
}
```

4. Eliminar Usuario

Endpoint: DELETE `http://localhost:8000/api/v1/admin/users/1`

Headers:

Authorization: Bearer {{adminToken}}

API 2: Servicio de Subastas (Moderador)

Gestión de Productos

1. Crear Producto

Endpoint: POST `http://localhost:8000/api/v1/productos`

Headers:

Content-Type: application/json
Authorization: Bearer {{moderadorToken}}

Body (JSON):

```
json
{
  "nombre": "Laptop Gamer RTX 4090",
  "descripcion": "Laptop gaming de alta gama con RTX 4090",
  "categoria": "electronica",
  "precioBase": 1500.00,
  "imagenes": [
    "https://example.com/laptop1.jpg",
    "https://example.com/laptop2.jpg"
  ]
}
```

2. Listar Mis Productos

Endpoint: GET `http://localhost:8000/api/v1/productos?page=0&size=10`

Headers:

Authorization: Bearer {{moderadorToken}}

3. Obtener Producto por ID

Endpoint: GET `http://localhost:8000/api/v1/productos/1`

Headers:

Authorization: Bearer {{moderadorToken}}

4. Actualizar Producto

Endpoint: PUT `http://localhost:8000/api/v1/productos/1`

Headers:

Content-Type: application/json

Authorization: Bearer {{moderadorToken}}

Body (JSON):

```
json

{
  "nombre": "Laptop Gamer RTX 4090 - Actualizada",
  "descripcion": "Laptop gaming de alta gama con RTX 4090 - Versión actualizada",
  "categoria": "electronica",
  "precioBase": 1600.00,
  "imagenes": [
    "https://example.com/laptop_updated1.jpg"
  ]
}
```

5. Eliminar Producto

Endpoint: DELETE http://localhost:8000/api/v1/productos/1

Headers:

Authorization: Bearer {{moderadorToken}}

Gestión de Subastas

1. Crear Subasta

Endpoint: POST http://localhost:8000/api/v1/subastas

Headers:

Content-Type: application/json

Authorization: Bearer {{moderadorToken}}

Body (JSON):

json

```
{
  "productId": 1,
  "description": "Subasta de Laptop Gamer RTX 4090",
  "startingPrice": 1500.00,
  "itemStatus": "nuevo",
  "itemCategory": "electronica",
  "daysToEndTime": 1,
  "fechaInicio": "2025-07-10T14:00:00Z",
  "fechaFin": "2025-07-10T16:00:00Z",
  "precioInicial": 1500.00
}
```

2. Listar Mis Subastas

Endpoint: GET http://localhost:8000/api/v1/subastas

Headers:

Authorization: Bearer {{moderadorToken}}

3. Iniciar Subasta

Endpoint: POST http://localhost:8000/api/v1/subastas/1/start

Headers:

Authorization: Bearer {{moderadorToken}}

4. Finalizar Subasta

Endpoint: POST http://localhost:8000/api/v1/subastas/1/end

Headers:

Authorization: Bearer {{moderadorToken}}

5. Extender Subasta

Endpoint: POST http://localhost:8000/api/v1/subastas/1/extend

Headers:

Content-Type: application/json

Authorization: Bearer {{moderadorToken}}

Body (JSON):

```
json

{
  "minutes": 30
}
```

6. Entrar a Sesión de Moderación

Endpoint: POST http://localhost:8000/api/v1/subastas/1/moderationsession

Headers:

Authorization: Bearer {{moderadorToken}}

7. Salir de Sesión de Moderación

Endpoint: DELETE http://localhost:8000/api/v1/subastas/1/moderationsession

Headers:

Authorization: Bearer {{moderadorToken}}

API 3: Servicio de Pujas

1. Realizar Puja

Endpoint: POST http://localhost:8000/api/v1/subastas/1/pujas

Headers:

Content-Type: application/json

Authorization: Bearer {{participanteToken}}

Body (JSON):

json

```
{  
  "bidPrice": 1600.00  
}
```

API 4: Servicio de Consulta de Subastas Activas

1. Listar Subastas Activas

Endpoint: GET `http://localhost:8000/api/v1/subastas?page=0&limit=10&categoria=electronica`

Headers:

Authorization: Bearer {{participanteToken}}

2. Obtener Subasta por ID

Endpoint: GET `http://localhost:8000/api/v1/subastas/1`

Headers:

Authorization: Bearer {{participanteToken}}

4. Historial de Usuario

Endpoint: GET http://localhost:8000/api/v1/usuarios/me/historial?page=0&limit=10

Headers:

Authorization: Bearer {{participanteToken}}

API 5: Servicio de Notificaciones (WebSocket)

Conexión WebSocket

URL: ws://localhost:8085/ws?token={{accessToken}}

Protocolo: WebSocket

Autenticación: Token JWT en query parameter

Eventos que puedes enviar:

1. Unirse a Sala de Subasta

```
json
{
  "action": "join_auction",
  "auctionId": "1"
}
```

2. Salir de Sala de Subasta

```
json
{
  "action": "leave_auction"
}
```

Eventos que recibirás:

1. Nueva Puja

json

```
{
  "event": "nueva_puja",
  "data": {
    "subastald": "1",
    "nuevoPrecio": 1600.00,
    "nombrePujador": "participante1",
    "timestamp": "2025-07-10T14:30:00Z"
  }
}
```

2. Subasta Extendida

json

```
{
  "event": "subasta_extendida",
  "data": {
    "subastald": "1",
    "nuevaHoraFin": "2025-07-10T16:30:00Z",
    "minutosAñadidos": 30
  }
}
```

3. Moderador Ingresó

json

```
{
  "event": "moderador_ingreso",
  "data": {
    "subastald": "1",
    "nombreModerador": "moderador1",
    "horaIngreso": "2025-07-10T14:35:00Z"
  }
}
```

Variables de Entorno en Postman

Configura estas variables en tu entorno de Postman:

baseUrl: http://localhost:8000

accessToken: [Token obtenido del login]

refreshToken: [Refresh token obtenido del login]

adminToken: [Token de usuario administrador]

moderadorToken: [Token de usuario moderador]

participanteToken: [Token de usuario participante]

Flujo de Pruebas Recomendado

1. Autenticación

1. Registrar usuario participante
2. Registrar usuario moderador
3. Login con cada usuario
4. Guardar tokens en variables de entorno

2. Gestión de Productos (Moderador)

1. Crear producto
2. Listar productos
3. Obtener producto por ID
4. Actualizar producto

3. Gestión de Subastas (Moderador)

1. Crear subasta
2. Iniciar subasta
3. Listar subastas

4. Participación en Subastas (Participante)

1. Listar subastas activas
2. Obtener detalles de subasta
3. Realizar puja
4. Validar puja

5. WebSocket (Opcional)

1. Conectar a WebSocket
 2. Unirse a sala de subasta
 3. Realizar puja y observar notificaciones
-

Códigos de Estado Esperados

- **200 OK:** Operación exitosa
- **201 Created:** Recurso creado exitosamente
- **400 Bad Request:** Datos inválidos
- **401 Unauthorized:** Token inválido o ausente
- **403 Forbidden:** Permisos insuficientes
- **404 Not Found:** Recurso no encontrado
- **409 Conflict:** Conflicto de negocio (ej: puja muy baja)
- **500 Internal Server Error:** Error interno del servidor