

Guía de Testing en GraphQL Playground - Agente de Ventas API

Grupo 3 - Backend Multi-Agente de Comercio Inteligente

Documento para probar todos los flujos usando la interfaz web de GraphQL Playground.

Índice

1. [Acceder al Playground](#)
 2. [Configurar Autenticación \(HTTP Headers\)](#)
 3. [Login Inicial \(REST\)](#)
 4. [Queries - Productos](#)
 5. [Queries - Usuarios](#)
 6. [Queries - Órdenes](#)
 7. [Mutations - Registro de Usuarios](#)
 8. [Mutations - Órdenes](#)
 9. [Semantic Search \(Chat con Alex\)](#)
 10. [Reconocimiento de Imágenes \(Agente 2\)](#)
 11. [Flujos Completos Paso a Paso](#)
 12. [Todas las Queries en una](#)
-

1. Acceder al Playground

Abre tu navegador y ve a:

```
http://localhost:8000/graphql
```

Verás la interfaz de **GraphQL Playground** dividida en:

- **Izquierda:** Editor de queries (donde escribes)
 - **Derecha:** Documentación/schema (explorable)
 - **Abajo:** Resultados/Response
-

2. Configurar Autenticación (HTTP Headers)

Después de hacer login (ver sección 3), necesitas configurar el token JWT en los headers.

Paso 1: Abrir el panel de Headers

En la parte inferior izquierda del Playground, haz clic en **"HTTP HEADERS"** (o "Headers").

Paso 2: Pegar el token

Copia este JSON y pégalo en el panel de Headers (reemplaza **TU_TOKEN_AQUI** con el token real):

```
{
  "Authorization": "Bearer TU_TOKEN_AQUI"
}
```

Ejemplo real:

```
{
  "Authorization": "Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEwZWUwNTgxLWFiOTYtNGNkOC04NTA1LWI4NWUwMDIxYTkyZCIsInVzZXJuYW1lIjoieY2xpZW50ZV9kZW1vIiwiaW1haWwiOiJkZW1vQHRlc3QuY29tIiwicm9sZSI6MiwiZXhwIjoxNzcwMzIxODM4fQ.ervW70YQL_qISj9h5H8AjnFQbve87VmDsnA44ZVEA1Q"
}
```

⚠ **Importante:** El token debe ir sin comillas simples alrededor de la variable. Copia el token exacto que te devuelve el login.

3. Login Inicial (REST)

El login es el único endpoint que **no** está en GraphQL (es REST). Úsalo para obtener tu token.

Usando curl (una sola vez):

```
curl -X POST http://localhost:8000/auth/login \
-H "Content-Type: application/json" \
-d '{
  "username": "cliente_demo",
  "password": "demo123"
}'
```

Respuesta:

Copia el valor de **access_token** y úsalo en los Headers del Playground (ver sección 2).

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEwZWUwNTgxLWFiOTYtNGNkOC04NTA1LWI4NWUwMDIxYTkyZCIsInVzZXJuYW1lIjoieY2xpZW50ZV9kZW1vIiwiaW1haWwiOiJkZW1vQHRlc3QuY29tIiwicm9sZSI6MiwiZXhwIjoxNzcwMzIxODM4fQ.ervW70YQL_qISj9h5H8AjnFQbve87VmDsnA44ZVEA1Q",
  "token_type": "bearer"
}
```

4. Queries - Productos

4.1 Listar productos (con límite)

Query:

```
query ListProducts {  
  listProducts(limit: 10) {  
    id  
    productName  
    unitCost  
    quantityAvailable  
    stockStatus  
    warehouseLocation  
    shelfLocation  
    batchNumber  
  }  
}
```


4.2 Obtener producto por ID

Query:

```
query GetProductById($productId: UUID!) {  
  getProductById(id: $productId) {  
    id  
    productName  
    unitCost  
    quantityAvailable  
    stockStatus  
    warehouseLocation  
    shelfLocation  
    batchNumber  
  }  
}
```

Variables:

```
{  
  "productId": "550e8400-e29b-41d4-a716-446655440000"  
}
```

 **Tip:** Copia un ID real de la query `listProducts` anterior.

5. Queries - Usuarios

5.1 Obtener mi perfil

Query:

```
query GetCurrentUser {
  getCurrentUser {
    id
    username
    email
    fullName
    role
    isActive
    createdAt
  }
}
```

5.2 Obtener usuario por ID

Query:

```
query GetUserById($userId: UUID!) {
  getUserById(id: $userId) {
    id
    username
    email
    fullName
    role
    isActive
    createdAt
  }
}
```

Variables:

```
{
  "userId": "10ee0581-ab96-4cd8-8505-b85a0021a91d"
}
```

6. Queries - Órdenes

6.1 Obtener mis órdenes

Query:

```
query GetMyOrders {
  getMyOrders(limit: 10, offset: 0) {
    id
```

```
    status
    totalAmount
    itemCount
    createdAt
  }
}
```

6.2 Obtener orden por ID (con detalles completos)

Query:

```
query GetOrderById($orderId: UUID!) {
  getOrderById(id: $orderId) {
    id
    userId
    status
    subtotal
    totalAmount
    taxAmount
    shippingCost
    discountAmount
    shippingAddress
    shippingCity
    shippingState
    shippingCountry
    notes
    sessionId
    createdAt
    updatedAt
    details {
      id
      productId
      productName
      quantity
      unitPrice
      subtotal
    }
  }
}
```

Variables:

```
{
  "orderId": "550e8400-e29b-41d4-a716-446655440000"
}
```

6.3 Órdenes recientes (solo admin)

Query:

```
query GetRecentOrders {
  getRecentOrders(limit: 10, statusFilter: "CONFIRMED") {
    id
    status
    totalAmount
    itemCount
    createdAt
  }
}
```

7. Mutations - Registro de Usuarios

7.1 Crear nuevo usuario (Registro)

Mutation:

```
mutation CreateUser {
  createUser(input: {
    username: "nuevouser123",
    email: "nuevo123@example.com",
    password: "password123",
    fullName: "Usuario Nuevo",
    role: 2
  }) {
    success
    accessToken
    tokenType
    user {
      id
      username
      email
      fullName
      role
      isActive
      createdAt
    }
    error
  }
}
```

7.2 Actualizar mi perfil

Mutation:

```
mutation UpdateUser {
  updateUser(input: {
    fullName: "Nombre Actualizado",
    email: "nuevoemail@example.com"
  }) {
    id
    username
    email
    fullName
    role
    isActive
    createdAt
  }
}
```

7.3 Cambiar contraseña

Mutation:

```
mutation ChangePassword {
  changePassword(input: {
    oldPassword: "password123",
    newPassword: "nuevapassword123"
  })
}
```

8. Mutations - Órdenes

8.1 Crear orden manualmente

Mutation:

```
mutation CreateOrder {
  createOrder(input: {
    details: [
      { productId: "550e8400-e29b-41d4-a716-446655440001", quantity: 2 },
      { productId: "550e8400-e29b-41d4-a716-446655440002", quantity: 1 }
    ],
    shippingAddress: "Calle Principal 123, Cuenca",
    shippingCity: "Cuenca",
    shippingState: "Azuay",
    shippingCountry: "Ecuador",
    notes: "Pedido de prueba desde Playground",
    sessionId: "playground-session-001"
  }) {
    success
    order {
```

```
    id
    userId
    status
    subtotal
    totalAmount
    taxAmount
    shippingCost
    discountAmount
    shippingAddress
    createdAt
    details {
      id
      productId
      productName
      quantity
      unitPrice
      subtotal
    }
  }
  message
  error
}
```

💡 **Tip:** Reemplaza los `productId` con IDs reales obtenidos de `listProducts`.

8.2 Cancelar orden

Mutation:

```
mutation CancelOrder {
  cancelOrder(orderId: "550e8400-e29b-41d4-a716-446655440000", reason:
"Cambio de opinión") {
    success
    order {
      id
      status
      totalAmount
    }
    message
    error
  }
}
```

9. Semantic Search (Chat con Alex)

9.1 Búsqueda básica

Query:


```
query ChatWithAlex {  
  semanticSearch(  
    query: "Busco zapatillas Nike para correr",  
    sessionId: "session-test-001"  
  ) {  
    answer  
    query  
    error  
  }  
}
```

9.2 Estilo Cuencano

Query:

```
query ChatCuencano {  
  semanticSearch(  
    query: "Ayayay ve, busco unos Nike full buenos",  
    sessionId: "session-cuencano-001"  
  ) {  
    answer  
    query  
    error  
  }  
}
```

9.3 Estilo Juvenil

Query:

```
query ChatJuvenil {  
  semanticSearch(  
    query: "Che bro, mostrame algo copado para correr",  
    sessionId: "session-juvenil-001"  
  ) {  
    answer  
    query  
    error  
  }  
}
```

9.4 Estilo Formal

Query:

```
query ChatFormal {
  semanticSearch(
    query: "Buenos días, quisiera consultar por zapatillas deportivas",
    sessionId: "session-formal-001"
  ) {
    answer
    query
    error
  }
}
```

9.5 Pregunta RAG (Políticas/FAQs)

Query:

```
query ChatRAG {
  semanticSearch(
    query: "¿Cuál es la política de devoluciones?",
    sessionId: "session-rag-001"
  ) {
    answer
    query
    error
  }
}
```

9.6 Objeción de precio (usar misma sesión)

Query:

```
query ChatObjecion {
  semanticSearch(
    query: "Están muy caros",
    sessionId: "session-test-001"
  ) {
    answer
    query
    error
  }
}
```

9.7 Confirmar compra (misma sesión)

Query:

```
query ChatCompra {
  semanticSearch(
    query: "Dámelos",
    sessionId: "session-test-001"
  ) {
    answer
    query
    error
  }
}
```

9.8 Proporcionar dirección (misma sesión)

Query:

```
query ChatDireccion {
  semanticSearch(
    query: "Calle Larga 123 y Benigno Malo, Cuenca",
    sessionId: "session-test-001"
  ) {
    answer
    query
    error
  }
}
```

9.9 Cancelar conversación (Stop Intent)

Query:

```
query ChatCancelar {
  semanticSearch(
    query: "Mejor no gracias",
    sessionId: "session-test-001"
  ) {
    answer
    query
    error
  }
}
```

10. Reconocimiento de Imágenes (Agente 2)

Método 1: Usando Variables

Mutation:

```
mutation RecognizeProduct($image: Upload!) {
  recognizeProductImage(image: $image) {
    success
    productName
    matches
    confidence
    error
  }
}
```

Para subir archivos en el Playground:

1. Abre el panel **"QUERY VARIABLES"** (abajo a la izquierda)
2. Haz clic en **"Add file..."** o arrastra el archivo
3. Asigna el nombre de variable **image**

Método 2: Usando cURL (para archivos es más fácil)

Si el Playground no te permite subir archivos fácilmente, usa este comando:

```
curl -X POST http://localhost:8000/graphql \
  -H "Authorization: Bearer TU_TOKEN" \
  -F "operations={\"query\": \"mutation (\$image: Upload!) {
recognizeProductImage(image: \$image) { success productName matches
confidence error } }\", \"variables\": {\"image\": null}}\" \
  -F \"map={\"0\": [\"variables.image\"]}\" \
  -F \"0=@/ruta/a/tu/imagen.jpg\"
```

Verificar si Agente 2 está disponible

```
curl http://localhost:5000/health
```

11. Flujos Completos Paso a Paso

Flujo 1: Primera vez - Registro → Chat → Compra

Paso 1: Registrarte (sin token en headers)

```
mutation {
  createUser(input: {
    username: "miusuario",
    email: "mi@email.com",
    password: "mipassword",
    fullName: "Mi Nombre",
```

```
    role: 2
  }) {
    success
    accessToken
    user {
      id
      username
    }
    error
  }
}
```

Paso 2: Configurar token Copia el `accessToken` de la respuesta y pégalo en HTTP Headers (ver sección 2).

Paso 3: Ver productos

```
query {
  listProducts(limit: 5) {
    id
    productName
    unitCost
    quantityAvailable
  }
}
```

Paso 4: Iniciar chat

```
query {
  semanticSearch(query: "Busco zapatillas Nike", sessionId: "mi-sesion-001") {
    answer
    error
  }
}
```

Paso 5: Seguir conversación (misma sesión)

```
query {
  semanticSearch(query: "Me gustaron los Pegasus", sessionId: "mi-sesion-001") {
    answer
    error
  }
}
```

Paso 6: Confirmar compra

```
query {  
  semanticSearch(query: "Dámelos", sessionId: "mi-sesion-001") {  
    answer  
    error  
  }  
}
```

Paso 7: Dar dirección

```
query {  
  semanticSearch(query: "Calle Principal 123, Cuenca", sessionId: "mi-sesion-001") {  
    answer  
    error  
  }  
}
```

Paso 8: Ver tu orden creada

```
query {  
  getMyOrders(limit: 5) {  
    id  
    status  
    totalAmount  
    itemCount  
    createdAt  
  }  
}
```

Flujo 2: Usuario Indeciso (Vago)**Paso 1: Búsqueda**

```
query {  
  semanticSearch(query: "Quiero zapatillas", sessionId: "indeciso-001") {  
    answer  
  }  
}
```

Paso 2: No decide

```
query {
  semanticSearch(query: "No sé", sessionId: "indeciso-001") {
    answer
  }
}
```

Paso 3: Sigue sin decidirse (el agente debe recomendar bestsellers)

```
query {
  semanticSearch(query: "Cualquiera", sessionId: "indeciso-001") {
    answer
  }
}
```

Flujo 3: Cancelación de Compra

Paso 1: Buscar

```
query {
  semanticSearch(query: "Quiero Adidas", sessionId: "cancel-001") {
    answer
  }
}
```

Paso 2: Mostrar interés

```
query {
  semanticSearch(query: "Me gustan", sessionId: "cancel-001") {
    answer
  }
}
```

Paso 3: Cancelar

```
query {
  semanticSearch(query: "Mejor no gracias", sessionId: "cancel-001") {
    answer
  }
}
```

12. Todas las Queries en una

Si quieres explorar todo el schema, aquí tienes una mega-query de prueba:

```
query ExplorarTodo($productId: UUID, $userId: UUID, $orderId: UUID) {
  # Productos
  productos: listProducts(limit: 3) {
    id
    productName
    unitCost
    quantityAvailable
  }

  # Usuario actual
  yo: getCurrentUser {
    id
    username
    email
    fullName
  }

  # Mis órdenes
  misOrdenes: getMyOrders(limit: 3) {
    id
    status
    totalAmount
    itemCount
  }
}
```

Y las variables (opcionales):

```
{
  "productId": "00000000-0000-0000-0000-000000000000",
  "userId": "00000000-0000-0000-0000-000000000000",
  "orderId": "00000000-0000-0000-0000-000000000000"
}
```

Resumen Rápido

Acción	Tipo	Requiere Auth
Login	REST	No
Registro	Mutation	No
Listar productos	Query	Opcional
Chat con Alex	Query	Sí
Crear orden	Mutation	Sí

Acción	Tipo	Requiere Auth
Ver mis órdenes	Query	Sí
Reconocer imagen	Mutation	Sí
Ver órdenes (admin)	Query	Sí (role=1)



Troubleshooting

"Debes iniciar sesion para usar el chat"

- Verifica que tengas el token en HTTP Headers
- Verifica que no tengas comillas simples alrededor del token
- Recuerda que el formato es: "Bearer eyJhbG..."

"Unauthorized" en cualquier query

- El token expiró (dura 30 minutos por defecto)
- Vuelve a hacer login y actualiza el header

No aparecen productos

- Verifica que la base de datos tenga datos: `listProducts`
- Si está vacía, ejecuta el script de seed/inicialización

Agente 2 no responde

- Verifica que esté corriendo: `curl http://localhost:5000/health`
- El upload de imágenes en Playground puede ser tricky, usa curl si es necesario

¡Listo para probar! 🚀

Abre `http://localhost:8000/graphql` y empieza copiando y pegando las queries de arriba.