

## ## Catálogo de Herramientas MCP

### ### SECCIÓN 1: GESTIÓN DE PACIENTES

#### #### 1.1 searchPatients

- **\*\*Descripción\*\***: Buscar pacientes por nombre, documento o teléfono

- **\*\*Parámetros\*\***:

- `q` (string, requerido): Término de búsqueda

- `limit` (number, opcional): Máximo resultados (1-100, default: 20)

- **\*\*Flujo\*\***:

1. Recibe término de búsqueda
2. Ejecuta LIKE query en campos: name, document, phone
3. Retorna lista de pacientes con información básica

- **\*\*Ejemplo de uso\*\***:

```
`` `json
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/call",
  "params": {
    "name": "searchPatients",
    "arguments": {
      "q": "Juan",
      "limit": 10
    }
  }
}
`` `
```

#### #### 1.2 getPatient

- **\*\*Descripción\*\***: Obtener información detallada de un paciente por ID

- **\*\*Parámetros\*\***:

- `patient\_id` (number, requerido): ID del paciente

- **\*\*Flujo\*\***:

1. Busca paciente por ID
2. Retorna toda la información del paciente incluyendo campos extendidos
3. Error si no existe

- **\*\*Campos retornados\*\***: Todos los campos de la tabla patients

#### #### 1.3 createPatient

- **\*\*Descripción\*\***: Crear nuevo paciente con campos extendidos obligatorios

- **\*\*Parámetros obligatorios\*\***:

- `document` (string): Documento de identidad
- `document\_type\_id` (number): ID del tipo de documento
- `name` (string): Nombre completo
- `phone` (string): Teléfono principal
- `email` (string): Email
- `birth\_date` (string): Fecha nacimiento YYYY-MM-DD
- `gender` (string): Masculino|Femenino|Otro|No especificado
- `address` (string): Dirección
- `municipality\_id` (number): ID del municipio
- `insurance\_eps\_id` (number): ID de la EPS
- `insurance\_affiliation\_type` (string):

Contributivo|Subsidiado|Vinculado|Particular|Otro

- `blood\_group\_id` (number): ID del grupo sanguíneo
- `population\_group\_id` (number): ID del grupo poblacional
- `education\_level\_id` (number): ID del nivel educativo
- `marital\_status\_id` (number): ID del estado civil
- `estrato` (number): Estrato socioeconómico (0-6)

- **\*\*Parámetros opcionales\*\***:

- `phone\_alt` (string): Teléfono alternativo
- `has\_disability` (boolean): Tiene discapacidad
- `disability\_type\_id` (number): ID del tipo de discapacidad
- `notes` (string): Notas adicionales

- **\*\*Flujo\*\***:

1. Valida campos obligatorios
2. Inserta en tabla patients
3. Retorna paciente creado con ID asignado

#### #### 1.4 updatePatient

- **\*\*Descripción\*\***: Actualizar información de un paciente existente

- **\*\*Parámetros\*\***:

- `patient\_id` (number, requerido): ID del paciente
- Cualquier campo del paciente para actualizar

- **\*\*Flujo\*\***:

1. Valida que exista el paciente

2. Actualiza solo los campos proporcionados
3. Retorna información actualizada

## ### SECCIÓN 2: GESTIÓN DE CITAS

### #### 2.1 getAppointments

- **\*\*Descripción\*\***: Obtener citas por fecha específica con filtros
- **\*\*Parámetros\*\***:
  - `date` (string, opcional): Fecha YYYY-MM-DD
  - `status` (string, opcional): Pendiente|Confirmada|Completada|Cancelada
  - `patient\_id` (number, opcional): Filtrar por paciente
  - `doctor\_id` (number, opcional): Filtrar por médico
- **\*\*Flujo\*\***:
  1. Construye query con filtros WHERE
  2. JOIN con tablas: patients, doctors, specialties, locations
  3. Retorna lista de citas con información completa

### #### 2.2 createAppointment

- **\*\*Descripción\*\***: Crear nueva cita médica
- **\*\*Parámetros requeridos\*\***:
  - `patient\_id` (number): ID del paciente
  - `doctor\_id` (number): ID del médico
  - `specialty\_id` (number): ID de la especialidad
  - `location\_id` (number): ID de la sede
  - `scheduled\_at` (string): Fecha y hora YYYY-MM-DD HH:MM:SS
- **\*\*Parámetros opcionales\*\***:
  - `duration\_minutes` (number): Duración en minutos (default: 30)
  - `appointment\_type` (string): Presencial|Telemedicina
  - `reason` (string): Motivo de la cita
- **\*\*Flujo\*\***:
  1. Valida existencia de patient, doctor, specialty, location
  2. Verifica disponibilidad en fecha/hora
  3. Inserta cita con estado 'Pendiente'
  4. Retorna cita creada

### #### 2.3 updateAppointmentStatus

- **\*\*Descripción\*\***: Actualizar estado de una cita
- **\*\*Parámetros\*\***:

- `appointment\_id` (number, requerido): ID de la cita
- `status` (string, requerido): Nuevo estado
- `notes` (string, opcional): Notas adicionales
- `cancellation\_reason` (string, opcional): Razón si se cancela
- **\*\*Flujo\*\***:
  1. Busca cita por ID
  2. Actualiza estado y campos adicionales
  3. Registra timestamp de cambio

### ### SECCIÓN 3: GESTIÓN DE MÉDICOS

#### #### 3.1 getDoctors

- **\*\*Descripción\*\***: Listar médicos con especialidades y ubicaciones
- **\*\*Parámetros\*\***:
  - `active\_only` (boolean): Solo médicos activos (default: true)
  - `specialty\_id` (number, opcional): Filtrar por especialidad
  - `location\_id` (number, opcional): Filtrar por ubicación
- **\*\*Flujo\*\***:
  1. JOIN con doctor\_specialties y doctor\_locations
  2. Agrupa especialidades y ubicaciones por doctor
  3. Aplica filtros según parámetros

#### #### 3.2 createDoctor

- **\*\*Descripción\*\***: Crear nuevo médico en el sistema
- **\*\*Parámetros requeridos\*\***:
  - `name` (string): Nombre completo
  - `license\_number` (string): Número de licencia médica
- **\*\*Parámetros opcionales\*\***:
  - `email` (string): Email
  - `phone` (string): Teléfono
  - `specialties` (array): IDs de especialidades
  - `locations` (array): IDs de ubicaciones
- **\*\*Flujo\*\***:
  1. Inserta doctor
  2. Asocia especialidades en doctor\_specialties
  3. Asocia ubicaciones en doctor\_locations

### ### SECCIÓN 4: ESPECIALIDADES Y UBICACIONES

#### #### 4.1 getSpecialties

- **\*\*Descripción\*\***: Listar especialidades médicas
- **\*\*Parámetros\*\***:
  - `active\_only` (boolean): Solo activas (default: true)

#### #### 4.2 createSpecialty

- **\*\*Descripción\*\***: Crear nueva especialidad médica
- **\*\*Parámetros\*\***:
  - `name` (string, requerido): Nombre
  - `description` (string, opcional): Descripción
  - `default\_duration\_minutes` (number): Duración default (default: 30)

#### #### 4.3 getLocations

- **\*\*Descripción\*\***: Listar sedes/ubicaciones disponibles
- **\*\*Parámetros\*\***:
  - `active\_only` (boolean): Solo activas (default: true)

#### #### 4.4 createLocation

- **\*\*Descripción\*\***: Crear nueva sede/ubicación
- **\*\*Parámetros\*\***:
  - `name` (string, requerido): Nombre de la sede
  - `address` (string, opcional): Dirección
  - `phone` (string, opcional): Teléfono
  - `type` (string): Tipo (default: 'Sucursal')
  - `capacity` (number): Capacidad (default: 0)

### ### SECCIÓN 5: TABLAS LOOKUP

#### #### 5.1 getDocumentTypes

- **\*\*Descripción\*\***: Obtener tipos de documento
- **\*\*Retorna\*\***: Lista de tipos: CC, CE, TI, PS, NIT, Otro

#### #### 5.2 getBloodGroups

- **\*\*Descripción\*\***: Obtener grupos sanguíneos
- **\*\*Retorna\*\***: A+, A-, B+, B-, AB+, AB-, O+, O-

#### #### 5.3 getEducationLevels

- **\*\*Descripción\*\***: Obtener niveles educativos
- **\*\*Retorna\*\***: Sin educación, Primaria, Secundaria, Media, Tecnológica, Posgrado, Otro

#### #### 5.4 getMaritalStatuses

- **\*\*Descripción\*\***: Obtener estados civiles
- **\*\*Retorna\*\***: Soltero, Casado, Unión libre, Separado, Viudo, Otro

#### #### 5.5 getPopulationGroups

- **\*\*Descripción\*\***: Obtener grupos poblacionales
- **\*\*Retorna\*\***: General, Indígena, Afrodescendiente, ROM, Víctimas, etc.

#### #### 5.6 getDisabilityTypes

- **\*\*Descripción\*\***: Obtener tipos de discapacidad
- **\*\*Retorna\*\***: Visual, Auditiva, Motor, Cognitiva, Psicosocial, Otra

#### #### 5.7 getMunicipalities

- **\*\*Descripción\*\***: Obtener municipios
- **\*\*Parámetros\*\***:
  - `zone\_id` (number, opcional): Filtrar por zona
- **\*\*Retorna\*\***: Lista con información de zona

#### #### 5.8 getZones

- **\*\*Descripción\*\***: Obtener zonas geográficas
- **\*\*Retorna\*\***: Zona de Socorro, Zona San Gil

#### #### 5.9 getEPS

- **\*\*Descripción\*\***: Obtener lista de EPS
- **\*\*Retorna\*\***: Lista de entidades prestadoras de salud

### ### SECCIÓN 6: CONSULTAS ESPECIALES

#### #### 6.1 getDaySummary

- **\*\*Descripción\*\***: Resumen completo del día con estadísticas
- **\*\*Parámetros\*\***:
  - `date` (string, opcional): Fecha (default: hoy)
- **\*\*Retorna\*\***: Estadísticas de citas, pacientes atendidos, etc.

#### #### 6.2 getPatientHistory

- **\*\*Descripción\*\***: Historial completo de citas de un paciente
- **\*\*Parámetros\*\***:
  - `patient\_id` (number, requerido): ID del paciente
  - `limit` (number): Número máximo de registros (default: 10)

#### #### 6.3 getDoctorSchedule

- **\*\*Descripción\*\***: Agenda de un médico en fecha específica
- **\*\*Parámetros\*\***:
  - `doctor\_id` (number, requerido): ID del médico
  - `date` (string, opcional): Fecha (default: hoy)

#### #### 6.4 executeCustomQuery

- **\*\*Descripción\*\***: Ejecutar consulta SQL personalizada (solo SELECT)
- **\*\*Parámetros\*\***:
  - `query` (string, requerido): Consulta SQL SELECT
  - `params` (array, opcional): Parámetros para la consulta
- **\*\*Restricciones\*\***: Solo queries SELECT, sin múltiples statements

### ## Flujo de Configuración del Cliente MCP

#### ### 1. Configuración Inicial

```
```json
{
  "mcpServers": {
    "biosanarcall": {
      "command": "curl",
      "args": [
        "-X", "POST",
        "https://biosanarcall.site/mcp-inspector",
        "-H", "Content-Type: application/json",
        "-H", "X-API-Key: mcp-key-biosanarcall-2025"
      ],
      "transport": "http"
    }
  }
}
```
```

### ### 2. Inicialización

1. Cliente envía `initialize` request
2. Servidor responde con capabilities y serverInfo
3. Cliente envía `initialized` notification

### ### 3. Listar Herramientas

```
```json
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/list"
}
```
```

### ### 4. Ejecutar Herramienta

```
```json
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "tools/call",
  "params": {
    "name": "nombreHerramienta",
    "arguments": {
      // argumentos específicos
    }
  }
}
```
```

## ## Ejemplos Prácticos de Uso

### ### Ejemplo 1: Crear Paciente Completo

```
```json
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/call",
```



```

"params": {
  "name": "createPatient",
  "arguments": {
    "document": "1234567890",
    "document_type_id": 1,
    "name": "María García López",
    "phone": "3001234567",
    "email": "maria.garcia@email.com",
    "birth_date": "1985-03-15",
    "gender": "Femenino",
    "address": "Calle 123 #45-67",
    "municipality_id": 14,
    "insurance_eps_id": 12,
    "insurance_affiliation_type": "Contributivo",
    "blood_group_id": 3,
    "population_group_id": 1,
    "education_level_id": 4,
    "marital_status_id": 2,
    "estrato": 3,
    "has_disability": false,
    "notes": "Paciente nuevo registrado vía MCP"
  }
}
}
}
` ``

```

### ### Ejemplo 2: Buscar y Crear Cita

```

` `` `json
// 1. Buscar paciente
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/call",
  "params": {
    "name": "searchPatients",
    "arguments": {
      "q": "María García",
      "limit": 5
    }
  }
}

```

```
}  
}  
}
```

// 2. Crear cita para el paciente encontrado

```
{  
  "jsonrpc": "2.0",  
  "id": 2,  
  "method": "tools/call",  
  "params": {  
    "name": "createAppointment",  
    "arguments": {  
      "patient_id": 5,  
      "doctor_id": 1,  
      "specialty_id": 1,  
      "location_id": 1,  
      "scheduled_at": "2025-08-25 09:00:00",  
      "appointment_type": "Presencial",  
      "reason": "Consulta general"  
    }  
  }  
}  
}  
}  
````
```

### ### Ejemplo 3: Obtener Datos de Lookup

```
```json  
// Obtener tipos de documento para formulario  
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "method": "tools/call",  
  "params": {  
    "name": "getDocumentTypes",  
    "arguments": {}  
  }  
}  
}
```

// Obtener grupos sanguíneos

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "tools/call",
  "params": {
    "name": "getBloodGroups",
    "arguments": {}
  }
}
` ``
```

## **## Manejo de Errores**

### **### Códigos de Error Comunes**

- ``-32700``: Parse error (JSON malformado)
- ``-32600``: Invalid request (estructura incorrecta)
- ``-32601``: Method not found (método no existe)
- ``-32602``: Invalid params (parámetros incorrectos)
- ``-32603``: Internal error (error del servidor)

### **### Ejemplo de Error**

```
````json
{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -32603,
    "message": "Error en createPatient: Campo 'document' es requerido",
    "data": {
      "field": "document",
      "provided": null
    }
  }
}
}
` ``
```