

Documentación de API - SmartRehabBar

Base URL

Desarrollo Local:

`http://localhost:3000`

Heroku (Producción):

`https://smartrehabbar-demo-7f620514b4ed.herokuapp.com/`

Endpoints

1. Pacientes

1.1 Listar Pacientes

GET /api/patients

Respuesta exitosa (200):

```
{
  "success": true,
  "data": [
    {
      "id": 1,
      "name": "Juan Pérez",
      "createdAt": "2025-10-03T10:30:00.000Z"
    },
    {
      "id": 2,
      "name": "María García",

```

```
      "createdAt": "2025-10-03T11:00:00.000Z"
    }
  ]
}
```

1.2 Crear Paciente

POST /api/patients
Content-Type: application/json

Body:

```
{
  "name": "Juan Pérez"
}
```

Respuesta exitosa (201):

```
{
  "success": true,
  "data": {
    "id": 1,
    "name": "Juan Pérez",
    "createdAt": "2025-10-03T10:30:00.000Z"
  }
}
```

Errores:

- 400 : Nombre vacío o inválido
- 500 : Error del servidor

1.3 Obtener Paciente

GET /api/patients/:id

Respuesta exitosa (200):

```
{
  "success": true,
  "data": {
    "id": 1,
    "name": "Juan Pérez",
    "createdAt": "2025-10-03T10:30:00.000Z",
    "sessions": [
      {
        "id": 1,
        "startTime": "2025-10-03T10:35:00.000Z",
        "endTime": "2025-10-03T10:45:00.000Z"
      }
    ]
  }
}
```

Errores:

- 404 : Paciente no encontrado

2. Sesiones

2.1 Crear Sesión (Iniciar)

POST /api/sessions
Content-Type: application/json

Body:

```
{
  "patientId": 1
}
```

Respuesta exitosa (201):

```
{
  "success": true,
  "data": {
    "id": 1,
    "patientId": 1,
  }
}
```

```
    "startTime": "2025-10-03T10:35:00.000Z",
    "endTime": null,
    "notes": null
  }
}
```

Errores:

- 400 : patientId inválido o faltante
- 404 : Paciente no encontrado
- 409 : Ya existe una sesión activa para este paciente

2.2 Finalizar Sesión

PATCH /api/sessions/:id
Content-Type: application/json

Body (opcional):

```
{
  "notes": "Sesión completada exitosamente"
}
```

Respuesta exitosa (200):

```
{
  "success": true,
  "data": {
    "id": 1,
    "patientId": 1,
    "startTime": "2025-10-03T10:35:00.000Z",
    "endTime": "2025-10-03T10:45:00.000Z",
    "notes": "Sesión completada exitosamente",
    "statistics": {
      "totalSteps": 24,
      "duration": 600,
      "averageBalance": {
        "left": 48.5,
        "right": 51.5
      },
      "maxImbalance": 15.2
    }
  }
}
```

```
}  
}  
}
```

Errores:

- 404 : Sesión no encontrada
- 400 : Sesión ya finalizada

2.3 Obtener Sesión

GET /api/sessions/:id

Respuesta exitosa (200):

```
{  
  "success": true,  
  "data": {  
    "id": 1,  
    "patientId": 1,  
    "patient": {  
      "id": 1,  
      "name": "Juan Pérez"  
    },  
    "startTime": "2025-10-03T10:35:00.000Z",  
    "endTime": "2025-10-03T10:45:00.000Z",  
    "notes": null,  
    "measurementCount": 24  
  }  
}
```

3. Mediciones

3.1 Registrar Medición (Pie Izquierdo)

POST /api/measurements/left
Content-Type: application/json

Body:

```
{
  "sessionId": 1,
  "weight": 65.5,
  "duration": 850,
  "timestamp": "2025-10-03T10:36:15.234Z"
}
```

Respuesta exitosa (201):

```
{
  "success": true,
  "data": {
    "id": 1,
    "sessionId": 1,
    "foot": "left",
    "weight": 65.5,
    "duration": 850,
    "timestamp": "2025-10-03T10:36:15.234Z",
    "pairedMeasurementId": null,
    "status": "pending"
  }
}
```

Nota: El servidor intentará sincronizar esta medición con una del pie derecho. Si encuentra una pareja dentro de la ventana de 3 segundos, emitirá un evento Socket.IO `measurement:new` con ambas mediciones.

Errores:

- 400 : Datos inválidos (peso fuera de rango, sessionId faltante, etc.)
- 404 : Sesión no encontrada
- 409 : Sesión ya finalizada

3.2 Registrar Medición (Pie Derecho)

POST /api/measurements/right
 Content-Type: application/json

Body y respuesta: Idénticos a 3.1, pero con `"foot": "right"`

3.3 Obtener Mediciones de una Sesión

GET /api/sessions/:id/measurements

Query Parameters (opcionales):

- limit : Número máximo de resultados (default: 100)
- offset : Offset para paginación (default: 0)
- paired : Filtrar solo mediciones emparejadas (true / false)

Ejemplo:

GET /api/sessions/1/measurements?limit=20&paired=true

Respuesta exitosa (200):

```
{
  "success": true,
  "data": {
    "measurements": [
      {
        "id": 1,
        "foot": "left",
        "weight": 65.5,
        "duration": 850,
        "timestamp": "2025-10-03T10:36:15.234Z",
        "pairedMeasurementId": 2
      },
      {
        "id": 2,
        "foot": "right",
        "weight": 68.2,
        "duration": 820,
        "timestamp": "2025-10-03T10:36:15.456Z",
        "pairedMeasurementId": 1
      }
    ],
    "total": 24,
    "limit": 20,
    "offset": 0
  }
}
```

Eventos Socket.IO

Conexión

Cliente se conecta:

```
import io from 'socket.io-client';

const socket = io('http://localhost:3000');

socket.on('connect', () => {
  console.log('Conectado:', socket.id);
});
```

Eventos del Servidor → Cliente

1. measurement:new

Emitido cuando se registra una nueva medición (emparejada o individual).

Payload (medición emparejada):

```
{
  "type": "paired",
  "sessionId": 1,
  "left": {
    "id": 1,
    "weight": 65.5,
    "duration": 850,
    "timestamp": "2025-10-03T10:36:15.234Z"
  },
  "right": {
    "id": 2,
    "weight": 68.2,
    "duration": 820,
    "timestamp": "2025-10-03T10:36:15.456Z"
  },
  "balance": {
    "leftPercent": 48.9,
    "rightPercent": 51.1,
    "difference": 2.2,
    "status": "good"
  }
}
```



```
}  
}
```

Payload (medición individual - timeout):

```
{  
  "type": "unpaired",  
  "sessionId": 1,  
  "foot": "left",  
  "measurement": {  
    "id": 3,  
    "weight": 64.8,  
    "duration": 900,  
    "timestamp": "2025-10-03T10:36:20.123Z"  
  }  
}
```

Balance Status:

- "good" : diferencia < 10%
- "warning" : diferencia 10-20%
- "critical" : diferencia > 20%

2. session:started

Emitido cuando se inicia una nueva sesión.

Payload:

```
{  
  "sessionId": 1,  
  "patientId": 1,  
  "patientName": "Juan Pérez",  
  "startTime": "2025-10-03T10:35:00.000Z"  
}
```

3. session:ended

Emitido cuando se finaliza una sesión.

Payload:

```
{
  "sessionId": 1,
  "endTime": "2025-10-03T10:45:00.000Z",
  "statistics": {
    "totalSteps": 24,
    "duration": 600,
    "averageBalance": {
      "left": 48.5,
      "right": 51.5
    }
  }
}
```

Códigos de Estado HTTP

Código	Significado	Uso
200	OK	Solicitud exitosa (GET, PATCH)
201	Created	Recurso creado exitosamente (POST)
400	Bad Request	Datos inválidos o faltantes
404	Not Found	Recurso no encontrado
409	Conflict	Conflicto (ej: sesión ya activa)
500	Internal Server Error	Error del servidor

Formato de Errores

Todos los errores siguen este formato:

```
{
  "success": false,
  "error": {
    "code": "INVALID_INPUT",
    "message": "El nombre del paciente es requerido",
    "details": {
```

```
    "field": "name",  
    "value": ""  
  }  
}  
}
```

Códigos de Error Comunes:

- INVALID_INPUT : Datos de entrada inválidos
- NOT_FOUND : Recurso no encontrado
- CONFLICT : Conflicto de estado
- DATABASE_ERROR : Error de base de datos
- INTERNAL_ERROR : Error interno del servidor

Validaciones

Pacientes

- name : String, 1-100 caracteres, requerido

Sesiones

- patientId : Integer, debe existir, requerido
- notes : String, 0-500 caracteres, opcional

Mediciones

- sessionId : Integer, debe existir y estar activa, requerido
- weight : Float, 1-300 kg, requerido
- duration : Integer, 100-5000 ms, opcional
- timestamp : ISO 8601 string, requerido

Ejemplos de Uso

Flujo Completo con cURL

```
# 1. Crear paciente
curl -X POST http://localhost:3000/api/patients \
  -H "Content-Type: application/json" \
  -d '{"name": "Juan Pérez"}'

# 2. Iniciar sesión
curl -X POST http://localhost:3000/api/sessions \
  -H "Content-Type: application/json" \
  -d '{"patientId": 1}'

# 3. Registrar medición pie izquierdo
curl -X POST http://localhost:3000/api/measurements/left \
  -H "Content-Type: application/json" \
  -d '{
    "sessionId": 1,
    "weight": 65.5,
    "duration": 850,
    "timestamp": "2025-10-03T10:36:15.234Z"
  }'

# 4. Registrar medición pie derecho
curl -X POST http://localhost:3000/api/measurements/right \
  -H "Content-Type: application/json" \
  -d '{
    "sessionId": 1,
    "weight": 68.2,
    "duration": 820,
    "timestamp": "2025-10-03T10:36:15.456Z"
  }'

# 5. Finalizar sesión
curl -X PATCH http://localhost:3000/api/sessions/1 \
  -H "Content-Type: application/json" \
  -d '{"notes": "Sesión completada"}'

# 6. Obtener mediciones
curl http://localhost:3000/api/sessions/1/measurements
```

Documento creado: 2025-10-03

Versión: 1.0

Autor: Equipo SmartRehabBar