

PRUEBAS ENDPOINTS - Documento 4: Estado del Sistema y Health Checks

Descripción

Este documento contiene pruebas para los endpoints de **monitoreo, estado del sistema e información general**.

1. GET /health

Health check básico del servidor

1.1 Prueba Exitosa - Health Check Simple

Request:

```
GET /health
```

Response (200 OK):

```
{
  "status": "OK",
  "timestamp": "2025-11-21T10:55:00.000Z"
}
```

Validaciones esperadas:

- ✓ HTTP 200 OK
- ✓ Status siempre es "OK" si el servidor responde
- ✓ Timestamp válido en ISO 8601

Notas:

- Este endpoint es ideal para monitoreo con probes (Kubernetes, Docker, etc.)
 - No requiere autenticación
 - Debe tener tiempo de respuesta < 100ms
-

2. GET /info

Información del sistema y configuración

2.1 Prueba Exitosa - Info Completo (Modo SIMULADO)

Request:

```
GET /info
```

Response (200 OK) - Modo SIMULADO:

```
{
  "success": true,
  "application": {
    "name": "Hacienda API",
    "version": "1.0.0",
    "description": "API REST para facturación electrónica con Costa Rica Hacienda"
  },
  "environment": {
    "nodeEnv": "development",
    "port": 3000,
    "logLevel": "info"
  },
  "mode": {
    "current": "SIMULATED",
    "description": "Modo de simulación - Sin conexión real a Hacienda",
    "atvConfigured": false,
    "certificatesAvailable": false
  },
  "database": {
    "type": "filesystem",
    "invoicesDir": "./invoices",
    "consecutiveFile": "./src/data/consecutivo.json"
  },
  "features": {
    "emitirFactura": true,
    "validarFactura": true,
    "enviarAHacienda": false,
    "almacenamiento": true,
    "bilingual": true
  },
  "status": {
    "healthy": true,
    "uptime": 245.123,
    "totalRequests": 47,
    "totalInvoices": 3
  },
  "endpoints": {
    "facturas": {
      "emitir": "POST /api/facturas/emitir",
      "validar": "POST /api/facturas/validar",
      "enviar": "POST /api/facturas/enviar",
      "listar": "GET /api/facturas",
      "obtener": "GET /api/facturas/:consecutivo"
    },
    "englishInvoices": {
      "emit": "POST /api/english-invoices/emit",
      "validate": "POST /api/english-invoices/validate",
      "list": "GET /api/english-invoices"
    }
  },
  "system": {
    "health": "GET /health",
```

```
    "info": "GET /info"
  },
  "documentation": {
    "apiGuide": "docs/API_USER_GUIDE.md",
    "technicalManual": "docs/MANUAL_TECNICO.md",
    "testingGuide": "docs/TESTING.md",
    "payloads": "docs/COMPLETE_PAYLOADS.md"
  },
  "timestamp": "2025-11-21T10:55:00.000Z"
}
```

Validaciones esperadas:

- ☒ HTTP 200 OK
- ☒ Mode es "SIMULATED"
- ☒ Información completa del sistema
- ☒ Endpoints documentados
- ☒ Features apropiados para modo SIMULADO

2.2 Prueba Exitosa - Info Completo (Modo REAL)

Request (con certificados configurados):

```
GET /info
```

Response (200 OK) - Modo REAL:

```
{
  "success": true,
  "application": {
    "name": "Hacienda API",
    "version": "1.0.0",
    "description": "API REST para facturación electrónica con Costa Rica Hacienda"
  },
  "environment": {
    "nodeEnv": "production",
    "port": 3000,
    "logLevel": "warn"
  },
  "mode": {
    "current": "REAL",
    "description": "Modo producción - Conexión real con Hacienda",
    "atvConfigured": true,
    "certificatesAvailable": true,
    "clientId": "3101234567-XYZ123"
  },
  "database": {
    "type": "filesystem",
    "invoicesDir": "./invoices",
    "consecutiveFile": "./src/data/consecutivo.json"
  }
}
```

```

},
"features": {
  "emitirFactura": true,
  "validarFactura": true,
  "enviarAHacienda": true,
  "almacenamiento": true,
  "bilingual": true
},
"status": {
  "healthy": true,
  "uptime": 1245.567,
  "totalRequests": 127,
  "totalInvoices": 45,
  "sentToHacienda": 43
},
"haciendaConnection": {
  "status": "CONNECTED",
  "lastCheck": "2025-11-21T10:54:30.000Z",
  "responseTime": "125ms",
  "validationEndpoint": "/api/factura/validar",
  "emissionEndpoint": "/api/factura/emitir"
},
"endpoints": {
  "facturas": {
    "emitir": "POST /api/facturas/emitir",
    "validar": "POST /api/facturas/validar",
    "enviar": "POST /api/facturas/enviar",
    "listar": "GET /api/facturas",
    "obtener": "GET /api/facturas/:consecutivo"
  }
},
"timestamp": "2025-11-21T10:55:00.000Z"
}

```

Validaciones esperadas:

- ✓ Mode es "REAL"
- ✓ haciendaConnection presente con status "CONNECTED"
- ✓ Certificados disponibles
- ✓ Feature enviarAHacienda es true
- ✓ Información de conexión con Hacienda

2.3 Prueba - Info vs Environment Variables

Escenario 1: NODE_ENV=development, ATV sin configurar

```

{
  "mode": { "current": "SIMULATED" },
  "environment": { "nodeEnv": "development" }
}

```

Escenario 2: NODE_ENV=production, ATV configurado

```
{
  "mode": { "current": "REAL" },
  "environment": { "nodeEnv": "production" }
}
```

Escenario 3: NODE_ENV=test, ATV sin configurar

```
{
  "mode": { "current": "SIMULATED" },
  "environment": { "nodeEnv": "test" }
}
```

Validaciones esperadas:

- ☒ Mode detectado automáticamente según configuración
- ☒ Información refleja estado real del sistema

3. GET /api/facturas/status

Estado específico del sistema de facturación

3.1 Prueba Exitosa - Estado General

Request:

```
GET /api/facturas/status
```

Response (200 OK):






```
{
  "success": true,
  "status": "OK",
  "mode": "SIMULATED",
  "timestamp": "2025-11-21T10:58:00.000Z",
  "invoicing": {
    "totalEmitted": 3,
    "totalSent": 1,
    "totalPending": 2,
    "totalRejected": 0
  },
  "storage": {
    "storageDir": "./invoices",
    "diskUsage": {
      "used": "2.5MB",
      "available": "50GB",
      "percentage": 0.005
    },
    "directories": {
```

```

    "invoices": 3,
    "sent": 1,
    "failed": 0
  },
  "adapter": {
    "mode": "SIMULATED",
    "initialized": true,
    "lastOperation": "EMISSION",
    "lastOperationTime": "2025-11-21T10:32:00.000Z"
  },
  "logging": {
    "logLevel": "info",
    "logFiles": {
      "combined": "logs/combined.log",
      "error": "logs/error.log"
    }
  },
  "lastError": null,
  "systemHealth": {
    "apiResponseTime": "45ms",
    "cpuUsage": "12%",
    "memoryUsage": "78MB"
  }
}

```

Validaciones esperadas:

-  HTTP 200 OK
-  Estadísticas de facturación precisas
-  Información de almacenamiento
-  Estado del adaptador ATV
-  Health metrics

3.2 Prueba Exitosa - Estado con Error Previo

Request (después de intentar operación fallida):

```
GET /api/facturas/status
```

Response (200 OK):

```

{
  "success": true,
  "status": "PARTIAL",
  "mode": "REAL",
  "timestamp": "2025-11-21T11:00:00.000Z",
  "invoicing": {
    "totalEmitted": 5,
    "totalSent": 3,
    "totalPending": 1,

```

```

    "totalRejected": 1
  },
  "adapter": {
    "mode": "REAL",
    "initialized": true,
    "lastOperation": "SEND",
    "lastOperationTime": "2025-11-21T10:59:00.000Z",
    "lastOperationStatus": "FAILED"
  },
  "lastError": {
    "timestamp": "2025-11-21T10:59:00.000Z",
    "operation": "enviarComprobante",
    "error": "HACIENDA_TIMEOUT",
    "message": "Connection timeout with Hacienda server",
    "details": {
      "clave": "50602272020110310000100010010000000005",
      "attemptNumber": 1
    }
  },
  "warnings": [
    {
      "severity": "WARN",
      "message": "Hacienda server responded slow (2s timeout)"
    }
  ]
}

```

Validaciones esperadas:

- ☒ Status es "PARTIAL" cuando hay errores
- ☒ lastError contiene detalles del error
- ☒ Array warnings incluido
- ☒ Información de último intento fallido

3.3 Prueba - Comparación Entre Modos

Modo SIMULADO:

```

{
  "mode": "SIMULATED",
  "invoicing": {
    "totalEmitted": 3,
    "totalSent": 0,
    "totalPending": 3
  },
  "adapter": {
    "mode": "SIMULATED"
  }
}

```

Modo REAL:

```
{
  "mode": "REAL",
  "invoicing": {
    "totalEmitted": 45,
    "totalSent": 43,
    "totalPending": 1,
    "totalRejected": 1
  },
  "adapter": {
    "mode": "REAL",
    "haciendaConnection": {
      "status": "CONNECTED"
    }
  }
}
```

Validaciones esperadas:

- ☒ Modo SIMULADO: `totalSent` siempre 0
- ☒ Modo REAL: información de Hacienda incluida
- ☒ Estadísticas correlacionadas correctamente

4. GET /api/version

Información de versión de la API

4.1 Prueba Exitosa - Versión

Request:

```
GET /api/version
```

Response (200 OK):

```
{
  "success": true,
  "version": "1.0.0",
  "apiVersion": "1.0.0",
  "buildNumber": "20251121.001",
  "buildDate": "2025-11-21T08:30:00.000Z",
  "nodeVersion": "18.17.0",
  "expressVersion": "4.18.2",
  "environment": "development",
  "timestamp": "2025-11-21T11:00:00.000Z"
}
```

Validaciones esperadas:

- ☒ Version semántica válida (MAJOR.MINOR.PATCH)
- ☒ Información de dependencias principales

- ☒ Build metadata disponible
-

5. Pruebas de Response Headers

5.1 Validar Headers de Respuesta

Request:

```
GET /api/facturas
```

Response Headers:

```
HTTP/1.1 200 OK
X-API-Version: 1.0.0
X-System-Mode: SIMULATED
X-Node-Env: development
Content-Type: application/json; charset=utf-8
Content-Security-Policy: default-src 'self'
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

Validaciones esperadas:

- ☒ X-API-Version presente
 - ☒ X-System-Mode presente (SIMULATED o REAL)
 - ☒ X-Node-Env presente
 - ☒ Headers de seguridad (Helmet)
 - ☒ Content-Type correcto
-

6. Pruebas de Tiempos de Respuesta

6.1 Bench - Emisión de Factura

Request:

```
POST /api/facturas/emitir
Content-Type: application/json
X-Request-ID: bench-001

{...factura payload...}
```

Métrica:

- Modo SIMULADO: < 500ms
- Modo REAL: < 5s (depende de Hacienda)

Validaciones esperadas:

- ☒ Respuesta dentro del tiempo esperado
 - ☒ Consistencia entre intentos múltiples
-

6.2 Bench - Listado de Facturas

Request:

```
GET /api/facturas?limit=100
```

Métrica (con 100 facturas almacenadas):

- < 1000ms esperado
-

6.3 Bench - Validación

Request:

```
POST /api/facturas/validar
Content-Type: application/json

{
  "payload": {...factura completa...}
}
```

Métrica:

- Modo SIMULADO: < 100ms
 - Modo REAL: < 2s (validación con Hacienda)
-

7. Pruebas de Monitoreo en Vivo

7.1 Script de Monitoreo (Bash/PowerShell)

PowerShell:

```
$url = "http://localhost:3000/health"
while ($true) {
    $response = Invoke-WebRequest -Uri $url -UseBasicParsing
    $timestamp = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
    Write-Host "[$timestamp] Status: $($response.StatusCode)"
    Start-Sleep -Seconds 10
}
```

Bash:

```
#!/bin/bash
while true; do
    curl -s http://localhost:3000/health | jq .timestamp
    sleep 10
done
```

8. Pruebas de Load Testing

8.1 Simular Carga con Apache Bench

```
# 100 requests, 10 concurrent
ab -n 100 -c 10 http://localhost:3000/health

# Resultado esperado
Requests per second: >100 (en SIMULADO)
Time per request: <100ms
Failed requests: 0
```

8.2 Simular Carga con Artillery

```
# load-test.yml
config:
  target: "http://localhost:3000"
  phases:
    - duration: 60
      arrivalRate: 10

scenarios:
  - name: "Factura Workflow"
    flow:
      - post:
          url: "/api/facturas/emitir"
          json:
            emisor: {...}
            receptor: {...}
      - get:
          url: "/api/facturas"
```

Ejecutar:

```
artillery run load-test.yml
```

Resumen de Pruebas - Documento 4

Prueba	Endpoint	Método	Esperado	Estado
1.1	/health	GET	200 OK	✓
2.1	/info	GET (SIMULADO)	200 OK	✓
2.2	/info	GET (REAL)	200 OK	✓
2.3	/info	GET (comparativa)	Corrección automática	✓
3.1	/api/facturas/status	GET	200 OK	✓

3.2	/api/facturas/status	GET (con error)	200 OK (PARTIAL)	✓
3.3	/api/facturas/status	GET (comparativa)	Diferencias esperadas	✓
4.1	/api/version	GET	200 OK	✓
5.1	Headers de respuesta	-	Correctos	✓
6.1	Benchmark emisión	POST	<500ms SIMULADO	✓
6.2	Benchmark listado	GET	<1000ms	✓
6.3	Benchmark validación	POST	<100ms SIMULADO	✓
7.1	Monitoreo continuo	GET	Script funcional	✓
8.1	Apache Bench	-	>100 RPS	✓
8.2	Artillery load test	-	Sin errores	✓

Próximo documento: Pruebas de Errores y Casos Límite