

Plan de Pruebas en Postman - Kontably

Descripción del Documento

El presente documento define el plan de pruebas generales para el Sistema Contable de la Fundación Niños con destino en las manos de Dios de, diseñado para ser ejecutado en Postman. El plan abarca todas las funcionalidades del sistema y proporciona una guía detallada para las pruebas de API.

Versión del Documento: 1.0.0

Proyecto: 27837_G5_ADS - Kontably

Índices de las pruebas

1. Validar autenticación de los diferentes usuarios (Tesorera y Coordinadora)
 2. Verificar operaciones CRUD para donaciones y para los egresos
 3. Probar gestión de usuarios
 4. Validar registro de egresos y de ingresos
 5. Confirmar generación de reportes de ingresos, egresos y balances
 6. Evaluar funcionalidades de exportación
 7. Probar gráficos interactivos
-

Estructura de Colecciones en Postman

1. Autenticación (Auth)

1.1 Login de Usuario

- **Endpoint:** POST /api/auth/login
- **Descripción:** Autenticación de usuarios en el sistema
- **Headers:** Content-Type: application/json

Caso de Prueba - AUT-001: Login Exitoso (Tesorera)

```
{  
    "endpoint": "/api/auth/login",  
    "method": "POST",  
    "body": {  
        "username": "tesorera",  
        "password": "12345"  
    },  
    "expected": {  
        "status": 200,  
        "response": {  
            "success": true,  
            "message": "Login successful"  
        }  
    }  
}
```

```

        "token": "<jwt_token>",
        "user": {
            "username": "tesorera",
            "role": "Tesorera"
        }
    }
}

```

Caso de Prueba - AUT-002: Login Exitoso (Coordinadora)

```

{
    "endpoint": "/api/auth/login",
    "method": "POST",
    "body": {
        "username": "coordinadora",
        "password": "admin"
    },
    "expected": {
        "status": 200,
        "response": {
            "success": true,
            "token": "<jwt_token>",
            "user": {
                "username": "coordinadora",
                "role": "Coordinadora"
            }
        }
    }
}

```

Caso de Prueba - AUT-003: Login con Credenciales Inválidas

```

{
    "endpoint": "/api/auth/login",
    "method": "POST",
    "body": {
        "username": "usuario_invalido",
        "password": "password_incorrecto"
    },
    "expected": {
        "status": 401,
        "response": {
            "success": false,
            "message": "Credenciales inválidas"
        }
    }
}

```

Caso de Prueba - AUT-004: Login con Campos Vacíos

```
{  
    "endpoint": "/api/auth/login",  
    "method": "POST",  
    "body": {  
        "username": "",  
        "password": ""  
    },  
    "expected": {  
        "status": 400,  
        "response": {  
            "success": false,  
            "message": "Los campos username y password son obligatorios"  
        }  
    }  
}
```

1.2 Validación de Token

- **Endpoint:** GET /api/auth/validate
- **Descripción:** Validar token JWT vigente

Caso de Prueba - AUT-005: Token Válido

```
{  
    "endpoint": "/api/auth/validate",  
    "method": "GET",  
    "headers": {  
        "Authorization": "Bearer <valid_token>"  
    },  
    "expected": {  
        "status": 200,  
        "response": {  
            "valid": true,  
            "user": {  
                "username": "tesorera",  
                "role": "Tesorera"  
            }  
        }  
    }  
}
```

Caso de Prueba - AUT-006: Token Inválido o Expirado

```
{  
    "endpoint": "/api/auth/validate",  
    "method": "GET",  
}
```

```

    "headers": {
        "Authorization": "Bearer <invalid_token>"
    },
    "expected": {
        "status": 401,
        "response": {
            "valid": false,
            "message": "Token inválido o expirado"
        }
    }
}

```

2. Donaciones

2.1 Registro de Donaciones

- **Endpoint:** POST /api/donaciones
- **Descripción:** Registrar una nueva donación

Caso de Prueba - DON-001: Registrar Donación Exitosa

```

{
    "endpoint": "/api/donaciones",
    "method": "POST",
    "headers": {
        "Authorization": "Bearer <valid_token>",
        "Content-Type": "application/json"
    },
    "body": {
        "donante": "Empresa XYZ",
        "tipo": "Dinero",
        "cantidad": 100,
        "valorUnitario": 50.00
    },
    "expected": {
        "status": 201,
        "response": {
            "success": true,
            "message": "Donación registrada exitosamente",
            "data": {
                "id": "<generated_id>",
                "donante": "Empresa XYZ",
                "tipo": "Dinero",
                "cantidad": 100,
                "valorUnitario": 50.00,
                "subtotal": 5000.00
            }
        }
    }
}

```

```
        }
    }
}
```

Caso de Prueba - DON-002: Registrar Donación con Campos Vacíos

```
{
  "endpoint": "/api/donaciones",
  "method": "POST",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "body": {
    "donante": "",
    "tipo": "",
    "cantidad": null,
    "valorUnitario": null
  },
  "expected": {
    "status": 400,
    "response": {
      "success": false,
      "message": "Todos los campos son obligatorios"
    }
  }
}
```

Caso de Prueba - DON-003: Registrar Donación con Valores Inválidos

```
{
  "endpoint": "/api/donaciones",
  "method": "POST",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "body": {
    "donante": "Donante Test",
    "tipo": "Alimentos",
    "cantidad": -5,
    "valorUnitario": -10.00
  },
  "expected": {
    "status": 400,
    "response": {
      "success": false,
      "message": "Cantidad y valor unitario deben ser números positivos"
    }
  }
}
```

```
    }
}
```

2.2 Listar Donaciones

- **Endpoint:** GET /api/donaciones
- **Descripción:** Obtener todas las donaciones registradas

Caso de Prueba - DON-004: Listar Todas las Donaciones

```
{
  "endpoint": "/api/donaciones",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "data": [
        {
          "id": "<id_1>",
          "donante": "Empresa XYZ",
          "tipo": "Dinero",
          "cantidad": 100,
          "valorUnitario": 50.00,
          "subtotal": 5000.00
        }
      ],
      "total": 1
    }
  }
}
```

Caso de Prueba - DON-005: Listar Donaciones sin Auth

```
{
  "endpoint": "/api/donaciones",
  "method": "GET",
  "expected": {
    "status": 401,
    "response": {
      "success": false,
      "message": "Token de autenticación requerido"
    }
  }
}
```

2.3 Obtener Donación por ID

- **Endpoint:** GET /api/donaciones/{id}
- **Descripción:** Obtener detalles de una donación específica

Caso de Prueba - DON-006: Obtener Donación Existente

```
{  
    "endpoint": "/api/donaciones/<existing_id>",  
    "method": "GET",  
    "headers": {  
        "Authorization": "Bearer <valid_token>"  
    },  
    "expected": {  
        "status": 200,  
        "response": {  
            "success": true,  
            "data": {  
                "id": "<existing_id>",  
                "donante": "Empresa XYZ",  
                "tipo": "Dinero",  
                "cantidad": 100,  
                "valorUnitario": 50.00,  
                "subtotal": 5000.00  
            }  
        }  
    }  
}
```

Caso de Prueba - DON-007: Obtener Donación No Existente

```
{  
    "endpoint": "/api/donaciones/non_existent_id",  
    "method": "GET",  
    "headers": {  
        "Authorization": "Bearer <valid_token>"  
    },  
    "expected": {  
        "status": 404,  
        "response": {  
            "success": false,  
            "message": "Donación no encontrada"  
        }  
    }  
}
```

2.4 Actualizar Donación

- **Endpoint:** PUT /api/donaciones/{id}
- **Descripción:** Actualizar información de una donación

Caso de Prueba - DON-008: Actualizar Donación Exitosa

```
{
  "endpoint": "/api/donaciones/<existing_id>",
  "method": "PUT",
  "headers": {
    "Authorization": "Bearer <valid_token>",
    "Content-Type": "application/json"
  },
  "body": {
    "donante": "Empresa XYZ Actualizada",
    "tipo": "Dinero",
    "cantidad": 150,
    "valorUnitario": 60.00
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "message": "Donación actualizada exitosamente",
      "data": {
        "id": "<existing_id>",
        "donante": "Empresa XYZ Actualizada",
        "tipo": "Dinero",
        "cantidad": 150,
        "valorUnitario": 60.00,
        "subtotal": 9000.00
      }
    }
  }
}
```

2.5 Eliminar Donación

- **Endpoint:** DELETE /api/donaciones/{id}
- **Descripción:** Eliminar una donación del sistema

Caso de Prueba - DON-009: Eliminar Donación Exitosa

```
{
  "endpoint": "/api/donaciones/<existing_id>",
  "method": "DELETE",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
}
```

```

    "expected": {
        "status": 200,
        "response": {
            "success": true,
            "message": "Donación eliminada exitosamente"
        }
    }
}

```

Caso de Prueba - DON-010: Eliminar Donación No Existente

```

{
    "endpoint": "/api/donaciones/non_existent_id",
    "method": "DELETE",
    "headers": {
        "Authorization": "Bearer <valid_token>"
    },
    "expected": {
        "status": 404,
        "response": {
            "success": false,
            "message": "Donación no encontrada"
        }
    }
}

```

3. Gestión de Usuarios

3.1 Listar Usuarios

- **Endpoint:** GET /api/usuarios
- **Descripción:** Obtener lista de usuarios del sistema

Caso de Prueba - USR-001: Listar Usuarios (Coordinadora)

```

{
    "endpoint": "/api/usuarios",
    "method": "GET",
    "headers": {
        "Authorization": "Bearer <coordinadora_token>"
    },
    "expected": {
        "status": 200,
        "response": {
            "success": true,
            "data": [
                {

```

```

        "id": "<id_1>",
        "username": "tesorera",
        "role": "Tesorera"
    },
    {
        "id": "<id_2>",
        "username": "coordinadora",
        "role": "Coordinadora"
    }
]
}
}

```

Caso de Prueba - USR-002: Listar Usuarios (Tesorera - Sin Permiso)

```

{
    "endpoint": "/api/usuarios",
    "method": "GET",
    "headers": {
        "Authorization": "Bearer <tesorera_token>"
    },
    "expected": {
        "status": 403,
        "response": {
            "success": false,
            "message": "No tiene permisos para realizar esta acción"
        }
    }
}

```

3.2 Crear Usuario

- **Endpoint:** POST /api/usuarios
- **Descripción:** Registrar un nuevo usuario en el sistema

Caso de Prueba - USR-003: Crear Usuario Exitoso

```

{
    "endpoint": "/api/usuarios",
    "method": "POST",
    "headers": {
        "Authorization": "Bearer <coordinadora_token>",
        "Content-Type": "application/json"
    },
    "body": {
        "username": "nuevo_usuario",
        "password": "password123",
    }
}

```

```

        "role": "Tesorera"
    },
    "expected": {
        "status": 201,
        "response": {
            "success": true,
            "message": "Usuario creado exitosamente",
            "data": {
                "id": "<generated_id>",
                "username": "nuevo_usuario",
                "role": "Tesorera"
            }
        }
    }
}

```

Caso de Prueba - USR-004: Crear Usuario con Username Existente

```

{
    "endpoint": "/api/usuarios",
    "method": "POST",
    "headers": {
        "Authorization": "Bearer <coordinadora_token>"
    },
    "body": {
        "username": "tesorera",
        "password": "password123",
        "role": "Tesorera"
    },
    "expected": {
        "status": 400,
        "response": {
            "success": false,
            "message": "El nombre de usuario ya existe"
        }
    }
}

```

3.3 Actualizar Usuario

- **Endpoint:** PUT /api/usuarios/{id}
- **Descripción:** Actualizar información de un usuario

Caso de Prueba - USR-005: Actualizar Usuario Exitoso

```

{
    "endpoint": "/api/usuarios/<existing_user_id>",
    "method": "PUT",

```

```

    "headers": {
        "Authorization": "Bearer <coordinadora_token>",
        "Content-Type": "application/json"
    },
    "body": {
        "username": "usuario_actualizado",
        "role": "Coordinadora"
    },
    "expected": {
        "status": 200,
        "response": {
            "success": true,
            "message": "Usuario actualizado exitosamente"
        }
    }
}

```

3.4 Eliminar Usuario

- **Endpoint:** DELETE /api/usuarios/{id}
- **Descripción:** Eliminar un usuario del sistema

Caso de Prueba - USR-006: Eliminar Usuario Exitoso

```

{
    "endpoint": "/api/usuarios/<existing_user_id>",
    "method": "DELETE",
    "headers": {
        "Authorization": "Bearer <coordinadora_token>"
    },
    "expected": {
        "status": 200,
        "response": {
            "success": true,
            "message": "Usuario eliminado exitosamente"
        }
    }
}

```

Caso de Prueba - USR-007: Eliminar Último Administrador

```

{
    "endpoint": "/api/usuarios/<last_admin_id>",
    "method": "DELETE",
    "headers": {
        "Authorization": "Bearer <coordinadora_token>"
    },
    "expected": {

```

```

    "status": 400,
    "response": {
        "success": false,
        "message": "No se puede eliminar el último administrador"
    }
}

```

4. Egresos

4.1 Registrar Egreso

- **Endpoint:** POST /api/egresos
- **Descripción:** Registrar un nuevo egreso

Caso de Prueba - EGR-001: Registrar Egreso Exitoso

```

{
    "endpoint": "/api/egresos",
    "method": "POST",
    "headers": {
        "Authorization": "Bearer <tesorera_token>",
        "Content-Type": "application/json"
    },
    "body": {
        "descripcion": "Compra de suministros",
        "tipo": "Operativo",
        "monto": 150.00,
        "fecha": "2024-01-15"
    },
    "expected": {
        "status": 201,
        "response": {
            "success": true,
            "message": "Egreso registrado exitosamente",
            "data": {
                "id": "<generated_id>",
                "descripcion": "Compra de suministros",
                "tipo": "Operativo",
                "monto": 150.00,
                "fecha": "2024-01-15"
            }
        }
    }
}

```

Caso de Prueba - EGR-002: Registrar Egreso con Monto Negativo

```
{  
    "endpoint": "/api/egresos",  
    "method": "POST",  
    "headers": {  
        "Authorization": "Bearer <tesorera_token>"  
    },  
    "body": {  
        "descripcion": "Test",  
        "tipo": "Operativo",  
        "monto": -100.00,  
        "fecha": "2024-01-15"  
    },  
    "expected": {  
        "status": 400,  
        "response": {  
            "success": false,  
            "message": "El monto debe ser un valor positivo"  
        }  
    }  
}
```

4.2 Listar Egresos

- **Endpoint:** GET /api/egresos
- **Descripción:** Obtener lista de egresos

Caso de Prueba - EGR-003: Listar Egresos con Filtros

```
{  
    "endpoint": "/api/egresos?fecha_inicio=2024-01-01&fecha_fin=2024-01-31",  
    "method": "GET",  
    "headers": {  
        "Authorization": "Bearer <tesorera_token>"  
    },  
    "expected": {  
        "status": 200,  
        "response": {  
            "success": true,  
            "data": [],  
            "total": 0  
        }  
    }  
}
```

4.3 Categorías de Egreso

- **Endpoint:** GET /api/egresos/categorias
- **Descripción:** Obtener categorías disponibles de egresos

Caso de Prueba - EGR-004: Listar Categorías

```
{
  "endpoint": "/api/egresos/categorias",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "data": [
        "Operativo",
        "Administrativo",
        "Emergencia",
        "Otro"
      ]
    }
  }
}
```

5. Reportes

5.1 Reporte de Ingresos

- **Endpoint:** GET /api/reportes/ingresos
- **Descripción:** Generar reporte de ingresos por período

Caso de Prueba - REP-001: Generar Reporte de Ingresos

```
{
  "endpoint": "/api/reportes/ingresos?fecha_inicio=2024-01-01&fecha_fin=2024-12-31",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "data": {
        "periodo": {
          "inicio": "2024-01-01",

```

```

        "fin": "2024-12-31"
    },
    "totalDonaciones": 0,
    "totalMonto": 0.00,
    "desglosePorTipo": []
}
}
}
}
```

5.2 Reporte de Balances

- **Endpoint:** GET /api/reportes/balances
- **Descripción:** Generar reporte de balance general

Caso de Prueba - REP-002: Generar Reporte de Balances

```
{
  "endpoint": "/api/reportes/balances?fecha_inicio=2024-01-01&fecha_fin=2024-12-31",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "data": {
        "periodo": {
          "inicio": "2024-01-01",
          "fin": "2024-12-31"
        },
        "totalIngresos": 0.00,
        "totalEgresos": 0.00,
        "balance": 0.00
      }
    }
  }
}
```

5.3 Gráficos Interactivos

- **Endpoint:** GET /api/reportes/graficos
- **Descripción:** Obtener datos para gráficos interactivos

Caso de Prueba - REP-003: Datos para Gráfico de Donaciones por Tipo

```
{
  "endpoint": "/api/reportes/graficos/donaciones-tipo",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "data": {
        "labels": ["Dinero", "Alimentos", "Medicamentos", "Ropa"],
        "datasets": [
          {
            "label": "Donaciones por Tipo",
            "data": [0, 0, 0, 0]
          }
        ]
      }
    }
  }
}
```

Caso de Prueba - REP-004: Datos para Gráfico de Ingresos vs Egresos

```
{
  "endpoint": "/api/reportes/graficos/ingresos-egresos",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response": {
      "success": true,
      "data": {
        "labels": ["Enero", "Febrero", "Marzo"],
        "datasets": [
          {
            "label": "Ingresos",
            "data": [0, 0, 0]
          },
          {
            "label": "Egresos",
            "data": [0, 0, 0]
          }
        ]
      }
    }
  }
}
```

```
    }
}
```

6. Exportación

6.1 Exportar Reporte a Excel

- **Endpoint:** GET /api/exportar/excel
- **Descripción:** Exportar reporte en formato Excel

Caso de Prueba - EXP-001: Exportar Reporte a Excel

```
{
  "endpoint": "/api/exportar/excel?tipo=ingresos&fecha_inicio=2024-01-01&fecha_fin=2024-1
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response_type": "file",
    "content_type": "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet"
  }
}
```

6.2 Exportar Reporte a PDF

- **Endpoint:** GET /api/exportar/pdf
- **Descripción:** Exportar reporte en formato PDF

Caso de Prueba - EXP-002: Exportar Reporte a PDF

```
{
  "endpoint": "/api/exportar/pdf?tipo=balance&fecha_inicio=2024-01-01&fecha_fin=2024-12-3
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response_type": "file",
    "content_type": "application/pdf"
  }
}
```

6.3 Exportar Donaciones a CSV

- **Endpoint:** GET /api/exportar/csv

- **Descripción:** Exportar donaciones en formato CSV

Caso de Prueba - EXP-003: Exportar Donaciones a CSV

```
{
  "endpoint": "/api/exportar/csv",
  "method": "GET",
  "headers": {
    "Authorization": "Bearer <valid_token>"
  },
  "expected": {
    "status": 200,
    "response_type": "file",
    "content_type": "text/csv"
  }
}
```

Matriz de Pruebas Resumida

Módulo	Caso de Prueba	Método	Endpoint
Auth	AUT-001: Login Tesorera	POST	/api/auth/loginAlta
Auth	AUT-002: Login Coordinadora	POST	/api/auth/loginAlta
Auth	AUT-003: Credenciales inválidas	POST	/api/auth/loginAlta
Auth	AUT-004: Campos vacíos	POST	/api/auth/loginAlta
Auth	AUT-005: Token válido	GET	/api/auth/validAtleta
Auth	AUT-006: Token inválido	GET	/api/auth/validAtleta
Donaciones	DON-001: Registrar donación	POST	/api/donacionesAlta
Donaciones	DON-002: Campos vacíos	POST	/api/donacionesAlta
Donaciones	DON-003: Valores inválidos	POST	/api/donacionesMedia
Donaciones	DON-004: Listar donaciones	GET	/api/donacionesAlta
Donaciones	DON-005: Listar sin autenticación	GET	/api/donacionesAlta
Donaciones	DON-006: Obtener por ID	GET	/api/donacionesM/{id}
Donaciones	DON-007: ID no existente	GET	/api/donacionesM/{id}
Donaciones	DON-008: Actualizar donación	PUT	/api/donacionesM/{id}
Donaciones	DON-009: Eliminar donación	DELETE	/api/donacionesA/{id}
Donaciones	DON-010: Eliminar no existente	DELETE	/api/donacionesM/{id}
Usuarios	USR-001: Listar usuarios (Coordinadora)	GET	/api/usuariosAlta
Usuarios	USR-002: Listar usuarios (Tesorera)	GET	/api/usuariosMedia

Módulo	Caso de Prueba	Método	Endpoint
Usuarios	USR-003: Crear usuario	POST	/api/usuariosAlta
Usuarios	USR-004: Usuario existente	POST	/api/usuariosMedia
Usuarios	USR-005: Actualizar usuario	PUT	/api/usuarios/{id}
Usuarios	USR-006: Eliminar usuario	DELETE	/api/usuarios/{id}
Usuarios	USR-007: Eliminar admin	DELETE	/api/usuarios/{id}
Egresos	EGR-001: Registrar egreso	POST	/api/egresosAlta
Egresos	EGR-002: Monto negativo	POST	/api/egresosMedia
Egresos	EGR-003: Listar con filtros	GET	/api/egresosMedia
Egresos	EGR-004: Categorías	GET	/api/egresos/categorias
Reportes	REP-001: Ingresos	GET	/api/reportes/ingresos
Reportes	REP-002: Balances	GET	/api/reportes/balances
Reportes	REP-003: Gráfico por tipo	GET	/api/reportes/graficos/donaciones-tipo
Reportes	REP-004: Gráfico Ingresos/Egresos	GET	/api/reportes/graficos/ingresos-egresos
Exportar	EXP-001: Exportar Excel	GET	/api/exportar/excel
Exportar	EXP-002: Exportar PDF	GET	/api/exportar/pdf
Exportar	EXP-003: Exportar CSV	GET	/api/exportar/csv

Variables de Entorno Postman

Crear las siguientes variables de entorno en Postman:

```
{  
    "base_url": "http://localhost:8080/api",  
    "tesorera_token": "",  
    "coordinadora_token": "",  
    "donacion_id": "",  
    "usuario_id": "",  
    "egreso_id": ""  
}
```

Flujos de Prueba Recomendados

Flujo 1: Login y Autenticación

1. AUT-001 → AUT-002 → AUT-003 → AUT-004 → AUT-005 → AUT-006

Flujo 2: Gestión Completa de Donaciones

1. AUT-001 (obtener token)
2. DON-001 (crear donación)
3. DON-004 (listar donaciones)
4. DON-006 (obtener por ID)
5. DON-008 (actualizar)
6. DON-009 (eliminar)

Flujo 3: Gestión de Usuarios (Coordinadora)

1. AUT-002 (login coordinadora)
2. USR-001 (listar usuarios)
3. USR-003 (crear usuario)
4. USR-005 (actualizar usuario)
5. USR-006 (eliminar usuario)

Flujo 4: Generación de Reportes

1. AUT-001 (obtener token)
2. REP-001 (reporte ingresos)
3. REP-002 (reporte balances)
4. REP-003 (gráfico donaciones)
5. REP-004 (gráfico ingresos/egresos)

Flujo 5: Exportación de Datos

1. AUT-001 (obtener token)
2. EXP-001 (exportar Excel)
3. EXP-002 (exportar PDF)
4. EXP-003 (exportar CSV)
