

MÓDULO 1 – USUARIOS

Endpoint: GET /api/usuarios/get-usuarios

Descripción: Obtiene todos los usuarios registrados.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/usuarios/get-usuarios

Ejemplo Response:

```
{ "ok": true, "usuarios": [ { "id_usuario": 1, "nombres": "Carlos", "correo": "carlos@mail.com", "estado": 1 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: GET /api/usuarios/buscar?correo=email

Descripción: Busca un usuario mediante su correo.

Parámetros: correo* = correo del usuario

Body esperado:

No aplica.

Ejemplo Request: GET /api/usuarios/buscar?correo=ana@mail.com

Ejemplo Response:

```
{ "ok": true, "usuario": { "id_usuario": 3, "nombres": "Ana", "correo": "ana@mail.com" } }
```

Errores comunes:

400 – Falta correo. 404 – Usuario no encontrado. 500 – Error en DB.

Endpoint: GET /api/usuarios/:id

Descripción: Obtiene un usuario activo por su ID.

Parámetros: id* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: GET /api/usuarios/2

Ejemplo Response:

```
{ "ok": true, "usuario": { "id_usuario": 2, "nombres": "María", "estado": 1 } }
```

Errores comunes:

400 – ID inválido. 404 – Usuario no encontrado. 500 – Error DB.

Endpoint: POST /api/usuarios/set-usuario

Descripción: Crea un usuario nuevo mediante SP sp_insertar_usuario.

Parámetros: nombres*, correo*, creado_por*

Body esperado:

```
{ "nombres": "Pedro", "apellidos": "Ramírez", "correo": "pedro@mail.com", "salario_base": 9000, "creado_por": "admin" }
```

Ejemplo Request: POST /api/usuarios/set-usuario

Ejemplo Response:

```
{ "ok": true, "mensaje": "Usuario creado correctamente" }
```

Errores comunes:

400 – Faltan datos. 500 – Error al ejecutar SP.

Endpoint: PUT /api/usuarios/:id

Descripción: Actualiza un usuario mediante SP sp_actualizar_usuario.

Parámetros: id*, modificado_por*

Body esperado:

```
{ "nombres": "Pedro Actualizado", "apellidos": "Ramírez", "salario_base": 9500, "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/usuarios/4

Ejemplo Response:

```
{ "ok": true, "mensaje": "Usuario actualizado correctamente" }
```

Errores comunes:

400 – modificado_por faltante. 500 – Error en SP.

Endpoint: DELETE /api/usuarios/:id

Descripción: Inactiva un usuario mediante SP sp_eliminar_usuario.

Parámetros: id* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/usuarios/5

Ejemplo Response:

```
{ "ok": true, "mensaje": "Usuario eliminado correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error en SP.

MÓDULO 2 – PRESUPUESTOS

Endpoint: *GET /api/presupuestos/get-presupuestos*

Descripción: Obtiene todos los presupuestos activos.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/presupuestos/get-presupuestos

Ejemplo Response:

```
{ "ok": true, "presupuestos": [ { "id_presupuesto": 1, "nombre": "Enero 2025", "anio_inicio": 2025, "mes_inicio": 1 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: *GET /api/presupuestos/:id*

Descripción: Obtiene un presupuesto por ID.

Parámetros: id* = ID del presupuesto.

Body esperado:

No aplica.

Ejemplo Request: GET /api/presupuestos/2

Ejemplo Response:

```
{ "ok": true, "presupuesto": { "id_presupuesto": 2, "nombre": "Febrero 2025", "anio_inicio": 2025, "mes_inicio": 2 } }
```

Errores comunes:

400 – ID inválido. 404 – No encontrado. 500 – Error DB.

Endpoint: *GET /api/presupuestos/usuario/:id_usuario*

Descripción: Lista presupuestos por usuario.

Parámetros: id_usuario* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: GET /api/presupuestos/usuario/1

Ejemplo Response:

```
{ "ok": true, "presupuestos": [ { "id_presupuesto": 10, "nombre": "Diciembre 2024" }, { "id_presupuesto": 9, "nombre": "Noviembre 2024" } ] }
```

Errores comunes:

400 – id_usuario requerido. 500 – Error DB.

Endpoint: GET /api/presupuestos/usuario/:id_usuario/activo

Descripción: Obtiene el presupuesto activo del usuario.

Parámetros: id_usuario* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: GET /api/presupuestos/usuario/1/activo

Ejemplo Response:

```
{ "ok": true, "presupuesto_activo": { "id_presupuesto": 10, "nombre": "Diciembre 2024" } }
```

Errores comunes:

404 – Usuario no tiene presupuestos. 500 – Error.

Endpoint: POST /api/presupuestos/set-presupuestos

Descripción: Crea un presupuesto mediante SP.

Parámetros: id_usuario*, nombre*, anio_inicio*, mes_inicio*, creado_por*

Body esperado:

```
{ "id_usuario": 1, "nombre": "Enero 2025", "anio_inicio": 2025, "mes_inicio": 1, "anio_fin": null, "mes_fin": null, "creado_por": "admin" }
```

Ejemplo Request: POST /api/presupuestos/set-presupuestos

Ejemplo Response:

```
{ "ok": true, "mensaje": "Presupuesto creado correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error en SP.

Endpoint: PUT /api/presupuestos/:id

Descripción: Actualiza un presupuesto mediante SP.

Parámetros: id* = ID del presupuesto.

Body esperado:

```
{ "nombre": "Presupuesto actualizado", "anio_inicio": 2025, "mes_inicio": 1, "anio_fin": 2025, "mes_fin": 12, "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/presupuestos/4

Ejemplo Response:

```
{ "ok": true, "mensaje": "Presupuesto actualizado correctamente" }
```

Errores comunes:

400 – modificado_por faltante. 500 – Error en SP.

Endpoint: DELETE /api/presupuestos/:id

Descripción: Elimina un presupuesto y sus detalles asociados.

Parámetros: id* = ID del presupuesto.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/presupuestos/5

Ejemplo Response:

```
{ "ok": true, "mensaje": "Presupuesto eliminado correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error al eliminar.

MÓDULO 3 – CATEGORÍAS

Endpoint: GET /api/categorias/get-categorias

Descripción: Obtiene todas las categorías ordenadas por presentación.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/categorias/get-categorias

Ejemplo Response:

```
{ "ok": true, "categorias": [ { "id_categoria": 1, "nombre": "Transporte", "estado": 1 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: GET /api/categorias/:id

Descripción: Obtiene una categoría por su ID.

Parámetros: id* = ID de la categoría.

Body esperado:

No aplica.

Ejemplo Request: GET /api/categorias/3

Ejemplo Response:

```
{ "ok": true, "categoria": { "id_categoria": 3, "nombre": "Alimentos", "estado": 1 } }
```

Errores comunes:

400 – ID inválido. 404 – No encontrada. 500 – Error DB.

Endpoint: POST /api/categorias/set-categorias

Descripción: Crea una categoría mediante SP sp_insertar_categoria.

Parámetros: nombre*, descripcion*, creado_por*

Body esperado:

```
{ "nombre": "Servicios", "descripcion": "Gastos de servicios del hogar", "creado_por": "admin" }
```

Ejemplo Request: POST /api/categorias/set-categorias

Ejemplo Response:

```
{ "ok": true, "mensaje": "Categoría creada correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error en SP.

Endpoint: PUT /api/categorias/:id

Descripción: Actualiza una categoría mediante SP sp_actualizar_categoria.

Parámetros: id*, modificado_por*

Body esperado:

```
{ "nombre": "Servicios Públicos", "descripcion": "Gastos de agua, luz e internet", "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/categorias/4

Ejemplo Response:

```
{ "ok": true, "mensaje": "Categoría actualizada correctamente" }
```

Errores comunes:

400 – modificado_por faltante. 500 – Error SP.

Endpoint: DELETE /api/categorias/:id

Descripción: Elimina una categoría mediante SP sp_eliminar_categoria.

Parámetros: id* = ID de la categoría.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/categorias/5

Ejemplo Response:

```
{ "ok": true, "mensaje": "Categoría eliminada correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error en SP.

MÓDULO 4 – SUBCATEGORÍAS

Endpoint: *GET /api/subcategorias/get-subcategorias*

Descripción: Obtiene todas las subcategorias registradas.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/subcategorias/get-subcategorias

Ejemplo Response:

```
{ "ok": true, "subcategorias": [ { "id_subcategoria": 1, "nombre": "Gasolina", "id_categoria": 2, "estado": 1 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: *GET /api/subcategorias/categorias/:id_categoria*

Descripción: Obtiene todas las subcategorias pertenecientes a una categoría.

Parámetros: id_categoria* = ID de la categoría.

Body esperado:

No aplica.

Ejemplo Request: GET /api/subcategorias/categorias/2

Ejemplo Response:

```
{ "ok": true, "subcategorias": [ { "id_subcategoria": 1, "nombre": "Gasolina" }, { "id_subcategoria": 2, "nombre": "Uber" } ] }
```

Errores comunes:

400 – ID inválido. 500 – Error en DB.

Endpoint: *GET /api/subcategorias/:id*

Descripción: Obtiene una subcategoría específica por ID.

Parámetros: id* = ID de la subcategoría.

Body esperado:

No aplica.

Ejemplo Request: GET /api/subcategorias/3

Ejemplo Response:

```
{ "ok": true, "subcategoria": { "id_subcategoria": 3, "nombre": "Peaje", "id_categoria": 2 } }
```

Errores comunes:

400 – ID inválido. 404 – No encontrada. 500 – Error DB.

Endpoint: POST /api/subcategorias

Descripción: Crea una subcategoría mediante SP sp_insertar_subcategoria.

Parámetros: nombre*, id_categoria*, creado_por*

Body esperado:

```
{ "nombre": "Repuestos", "id_categoria": 2, "creado_por": "admin" }
```

Ejemplo Request: POST /api/subcategorias

Ejemplo Response:

```
{ "ok": true, "mensaje": "Subcategoría creada correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error en SP.

Endpoint: PUT /api/subcategorias/:id

Descripción: Actualiza una subcategoría mediante SP sp_actualizar_subcategoria.

Parámetros: id*, modificado_por*

Body esperado:

```
{ "nombre": "Repuestos Vehiculares", "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/subcategorias/5

Ejemplo Response:

```
{ "ok": true, "mensaje": "Subcategoría actualizada correctamente" }
```

Errores comunes:

400 – modificado_por faltante. 500 – Error en SP.

Endpoint: DELETE /api/subcategorias/:id

Descripción: Elimina una subcategoría mediante SP sp_eliminar_subcategoria.

Parámetros: id* = ID de la subcategoría.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/subcategorias/6

Ejemplo Response:

```
{ "ok": true, "mensaje": "Subcategoría eliminada correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error en SP. 409 – No se puede eliminar porque tiene registros asociados.

MÓDULO 5 – PRESUPUESTO DETALLE

Endpoint: *GET /api/presupuesto-detalle/get*

Descripción: Obtiene todos los registros de detalle de presupuesto.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/presupuesto-detalle/get

Ejemplo Response:

```
{ "ok": true, "detalles": [ { "id_detalle": 1, "id_presupuesto": 3, "id_subcategoria": 5, "monto_asignado": 1500 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: *GET /api/presupuestos/:id/detalle*

Descripción: Obtiene todos los detalles pertenecientes a un presupuesto.

Parámetros: id* = ID del presupuesto.

Body esperado:

No aplica.

Ejemplo Request: GET /api/presupuestos/3/detalle

Ejemplo Response:

```
{ "ok": true, "detalles": [ { "id_detalle": 4, "subcategoria": "Gasolina", "monto_asignado": 1200 } ] }
```

Errores comunes:

400 – ID inválido. 500 – Error al consultar DB.

Endpoint: *POST /api/presupuesto-detalle*

Descripción: Crea un detalle de presupuesto vía SP sp_insertar_presupuesto_detalle.

Parámetros: id_presupuesto*, id_subcategoria*, monto_asignado*, creado_por*

Body esperado:

```
{ "id_presupuesto": 3, "id_subcategoria": 5, "monto_asignado": 1500, "creado_por": "admin" }
```

Ejemplo Request: POST /api/presupuesto-detalle

Ejemplo Response:

```
{ "ok": true, "mensaje": "Detalle creado correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error en SP.

Endpoint: PUT /api/presupuesto-detalle/:id

Descripción: Actualiza un registro de detalle usando sp_actualizar_presupuesto_detalle.

Parámetros: id* = ID del detalle.

Body esperado:

{ "monto_asignado": 1800, "modificado_por": "admin" }

Ejemplo Request: PUT /api/presupuesto-detalle/4

Ejemplo Response:

{ "ok": true, "mensaje": "Detalle actualizado correctamente" }

Errores comunes:

400 – modificado_por faltante. 500 – Error en SP.

Endpoint: DELETE /api/presupuesto-detalle/presupuesto/:id_presupuesto

Descripción: Elimina TODOS los detalles asociados a un presupuesto.

Parámetros: id_presupuesto* = ID del presupuesto.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/presupuesto-detalle/presupuesto/3

Ejemplo Response:

{ "ok": true, "mensaje": "Detalles eliminados correctamente" }

Errores comunes:

400 – ID inválido. 500 – Error al eliminar registros.

Endpoint: DELETE /api/presupuesto-detalle/:id

Descripción: Elimina un detalle específico mediante SP sp_eliminar_presupuesto_detalle.

Parámetros: id* = ID del detalle.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/presupuesto-detalle/4

Ejemplo Response:

{ "ok": true, "mensaje": "Detalle eliminado correctamente" }

Errores comunes:

400 – ID inválido. 500 – Error eliminando detalle.

MÓDULO 6 – OBLIGACIONES FIJAS

Endpoint: *GET /api/obligaciones/get*

Descripción: Obtiene todas las obligaciones fijas registradas.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/obligaciones/get

Ejemplo Response:

```
{ "ok": true, "obligaciones": [ { "id_obligacion": 1, "id_usuario": 3, "nombre": "Renta", "monto": 5000, "dia_pago": 1 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: *GET /api/obligaciones/usuario/:id_usuario*

Descripción: Obtiene las obligaciones fijas de un usuario.

Parámetros: id_usuario* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: GET /api/obligaciones/usuario/3

Ejemplo Response:

```
{ "ok": true, "obligaciones": [ { "id_obligacion": 1, "nombre": "Renta", "monto": 5000 } ] }
```

Errores comunes:

400 – id_usuario inválido. 500 – Error DB.

Endpoint: *GET /api/obligaciones/:id_obligacion*

Descripción: Obtiene una obligación fija por ID.

Parámetros: id_obligacion* = ID de la obligación.

Body esperado:

No aplica.

Ejemplo Request: GET /api/obligaciones/5

Ejemplo Response:

```
{ "ok": true, "obligacion": { "id_obligacion": 5, "nombre": "Internet", "monto": 850 } }
```

Errores comunes:

400 – ID inválido. 404 – No encontrada. 500 – Error.

Endpoint: POST /api/obligaciones/set

Descripción: Crea una obligación fija mediante SP sp_insertar_obligacion.

Parámetros: id_usuario*, nombre*, monto*, dia_pago*, creado_por*

Body esperado:

```
{ "id_usuario": 3, "nombre": "Seguro de auto", "monto": 1200, "dia_pago": 15, "creado_por": "admin" }
```

Ejemplo Request: POST /api/obligaciones/set

Ejemplo Response:

```
{ "ok": true, "mensaje": "Obligación creada correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error al ejecutar SP.

Endpoint: PUT /api/obligaciones/:id_obligacion

Descripción: Actualiza una obligación mediante SP sp_actualizar_obligacion.

Parámetros: id_obligacion*, modificado_por*

Body esperado:

```
{ "nombre": "Seguro de auto actualizado", "monto": 1300, "dia_pago": 20, "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/obligaciones/5

Ejemplo Response:

```
{ "ok": true, "mensaje": "Obligación actualizada correctamente" }
```

Errores comunes:

400 – modificado_por faltante. 500 – Error en SP.

Endpoint: DELETE /api/obligaciones/:id_obligacion

Descripción: Inactiva una obligación mediante SP sp_eliminar_obligacion.

Parámetros: id_obligacion* = ID de la obligación.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/obligaciones/7

Ejemplo Response:

```
{ "ok": true, "mensaje": "Obligación eliminada correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error en SP.

MÓDULO 7 – TRANSACCIONES

Endpoint: *GET /api/transacciones/get*

Descripción: Obtiene todas las transacciones del sistema.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/transacciones/get

Ejemplo Response:

```
{ "ok": true, "transacciones": [ { "id_transaccion": 1, "id_usuario": 3, "tipo": "gasto", "monto": 300 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: *GET /api/transacciones/usuario/:id_usuario*

Descripción: Obtiene todas las transacciones de un usuario.

Parámetros: id_usuario* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: GET /api/transacciones/usuario/3

Ejemplo Response:

```
{ "ok": true, "transacciones": [ { "id_transaccion": 1, "tipo": 'gasto', "monto": 300 } ] }
```

Errores comunes:

400 – id_usuario inválido. 500 – Error al consultar DB.

Endpoint: *GET /api/transacciones/presupuesto/:id_presupuesto*

Descripción: Obtiene transacciones asociadas a un presupuesto.

Parámetros: id_presupuesto* = ID del presupuesto.

Body esperado:

No aplica.

Ejemplo Request: GET /api/transacciones/presupuesto/4

Ejemplo Response:

```
{ "ok": true, "transacciones": [ { "id_transaccion": 10, "monto": 500, "tipo": "gasto" } ] }
```

Errores comunes:

400 – ID inválido. 500 – Error en DB.

Endpoint: GET /api/transacciones/:id_transaccion

Descripción: Obtiene una transacción mediante su ID.

Parámetros: id_transaccion* = ID de la transacción.

Body esperado:

No aplica.

Ejemplo Request: GET /api/transacciones/12

Ejemplo Response:

```
{ "ok": true, "transaccion": { "id_transaccion": 12, "monto": 100, "tipo": "ingreso" } }
```

Errores comunes:

400 – ID inválido. 404 – No encontrada. 500 – Error.

Endpoint: POST /api/transacciones

Descripción: Crea una transacción mediante SP sp_insertar_transaccion.

Parámetros: id_usuario*, id_presupuesto*, id_subcategoria*, tipo*, monto*, creado_por*

Body esperado:

```
{ "id_usuario": 3, "id_presupuesto": 4, "id_subcategoria": 7, "tipo": "gasto", "monto": 250, "creado_por": "admin" }
```

Ejemplo Request: POST /api/transacciones

Ejemplo Response:

```
{ "ok": true, "mensaje": "Transacción ingresada correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error en SP.

Endpoint: PUT /api/transacciones/:id_transaccion

Descripción: Actualiza una transacción existente mediante SP sp_actualizar_transaccion.

Parámetros: id_transaccion*, modificado_por*

Body esperado:

```
{ "tipo": "gasto", "monto": 350, "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/transacciones/8

Ejemplo Response:

```
{ "ok": true, "mensaje": "Transacción actualizada correctamente" }
```

Errores comunes:

400 – modificado_por requerido. 500 – Error en SP.

Endpoint: DELETE /api/transacciones/:id_transaccion

Descripción: Elimina una transacción mediante SP sp_eliminar_transaccion.

Parámetros: id_transaccion* = ID de la transacción.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/transacciones/9

Ejemplo Response:

```
{ "ok": true, "mensaje": "Transacción eliminada correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error en SP.

Endpoint: POST /api/transacciones/registrar

Descripción: Registra una transacción COMPLETA mediante SP sp_registrar_transaccion_completa.

Parámetros: id_usuario*, id_presupuesto*, id_subcategoria*, tipo*, monto*, descripcion*, creado_por*

Body esperado:

```
{ "id_usuario": 3, "id_presupuesto": 4, "id_subcategoria": 7, "tipo": "gasto", "monto": 200, "descripcion": "Supermercado", "creado_por": "admin" }
```

Ejemplo Request: POST /api/transacciones/registrar

Ejemplo Response:

```
{ "ok": true, "mensaje": "Transacción registrada correctamente" }
```

Errores comunes:

400 – Datos incompletos. 500 – Error en SP lógico.

MÓDULO 8 – METAS DE AHORRO

Endpoint: GET /api/metast

Descripción: Obtiene todas las metas de ahorro registradas.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: GET /api/metast

Ejemplo Response:

```
{ "ok": true, "metas": [ { "id_meta": 1, "id_usuario": 3, "nombre": "Ahorro para viaje", "monto_objetivo": 10000, "monto_actual": 2500 } ] }
```

Errores comunes:

500 – Error al consultar DB2.

Endpoint: GET /api/metast/usuario/:id_usuario

Descripción: Obtiene las metas de ahorro de un usuario específico.

Parámetros: id_usuario* = ID del usuario.

Body esperado:

No aplica.

Ejemplo Request: GET /api/metast/usuario/3

Ejemplo Response:

```
{ "ok": true, "metas": [ { "id_meta": 1, "nombre": "Viaje", "monto_objetivo": 10000 } ] }
```

Errores comunes:

400 – id_usuario inválido. 500 – Error DB.

Endpoint: GET /api/metast/:id_meta

Descripción: Obtiene una meta de ahorro por su ID.

Parámetros: id_meta* = ID de la meta.

Body esperado:

No aplica.

Ejemplo Request: GET /api/metast/5

Ejemplo Response:

```
{ "ok": true, "meta": { "id_meta": 5, "nombre": "Emergencias", "monto_objetivo": 5000, "monto_actual": 1200 } }
```

Errores comunes:

400 – ID inválido. 404 – Meta no encontrada. 500 – Error.

Endpoint: POST /api/metast

Descripción: Crea una nueva meta de ahorro mediante SP sp_insertar_meta.

Parámetros: id_usuario*, nombre*, monto_objetivo*, creado_por*

Body esperado:

```
{ "id_usuario": 3, "nombre": "Compra de laptop", "monto_objetivo": 15000, "creado_por": "admin" }
```

Ejemplo Request: POST /api/metast

Ejemplo Response:

```
{ "ok": true, "mensaje": "Meta creada correctamente" }
```

Errores comunes:

400 – Datos faltantes. 500 – Error en SP.

Endpoint: PUT /api/metast/id_meta

Descripción: Actualiza una meta existente mediante SP sp_actualizar_meta.

Parámetros: id_meta*, modificado_por*

Body esperado:

```
{ "nombre": "Laptop nueva", "monto_objetivo": 16000, "modificado_por": "admin" }
```

Ejemplo Request: PUT /api/metast/4

Ejemplo Response:

```
{ "ok": true, "mensaje": "Meta actualizada correctamente" }
```

Errores comunes:

400 – modificado_por faltante. 500 – Error en SP.

Endpoint: DELETE /api/metast/id_meta

Descripción: Elimina una meta mediante SP sp_eliminar_meta.

Parámetros: id_meta* = ID de la meta.

Body esperado:

No aplica.

Ejemplo Request: DELETE /api/metast/7

Ejemplo Response:

```
{ "ok": true, "mensaje": "Meta eliminada correctamente" }
```

Errores comunes:

400 – ID inválido. 500 – Error en SP.

Endpoint: POST /api/metast/recalcular

Descripción: Recalcula TODAS las metas de ahorro usando SP sp_actualizar_todas_metas_ahorro.

Parámetros: No requiere parámetros.

Body esperado:

No aplica.

Ejemplo Request: POST /api/metast/recalcular

Ejemplo Response:

```
{ "ok": true, "mensaje": "Metas recalculadas correctamente" }
```

Errores comunes:

500 – Error en SP lógico.

MÓDULO 9 – REPORTES

Endpoint: GET /api/reportes/gastos-por-categoria

Descripción: Retorna el total de gastos agrupados por categoría.

Parámetros: id_usuario* = ID del usuario, anio*, mes*

Body esperado:

No aplica.

Ejemplo Request: GET /api/reportes/gastos-por-categoria?id_usuario=3&anio;=2025&mes;=1

Ejemplo Response:

```
{ "ok": true, "reporte": [ { "categoria": "Alimentos", "total_gastado": 3200 }, { "categoria": "Transporte",  
"total_gastado": 1800 } ] }
```

Errores comunes:

400 – Faltan parámetros. 500 – Error en DB.

Endpoint: GET /api/reportes/balance-mensual

Descripción: Retorna balance mensual del usuario (ingresos, gastos, ahorro).

Parámetros: id_usuario*, anio*, mes*

Body esperado:

No aplica.

Ejemplo Request: GET /api/reportes/balance-mensual?id_usuario=3&anio;=2025&mes;=1

Ejemplo Response:

```
{ "ok": true, "balance": { "ingresos": 8000, "gastos": 4500, "ahorro": 3500 } }
```

Errores comunes:

400 – Parámetros incompletos. 500 – Error DB.

Endpoint: GET /api/reportes/monto-ejecutado

Descripción: Obtiene el monto ejecutado para una subcategoría.

Parámetros: id_presupuesto*, id_subcategoria*

Body esperado:

No aplica.

Ejemplo Request: GET /api/reportes/monto-ejecutado?id_presupuesto=4&id_subcategoria=7

Ejemplo Response:

```
{ "ok": true, "monto_ejecutado": 950 }
```

Errores comunes:

400 – Faltan parámetros. 500 – Error en DB.

Endpoint: *GET /api/porcentaje-ejecucion*

Descripción: Calcula el porcentaje de ejecución (gasto vs asignado).

Parámetros: id_presupuesto*, id_subcategoria*

Body esperado:

No aplica.

Ejemplo Request: GET /api/porcentaje-ejecucion?id_presupuesto=4&id_subcategoria=7

Ejemplo Response:

```
{ "ok": true, "porcentaje": 63.33 }
```

Errores comunes:

400 – Falta id_presupuesto o id_subcategoria. 500 – Error en cálculo.