

1. Catálogo de Entidades y su Visibilidad

1.1 Producto Financiero y Configuración

Entidad	API (Endpoints)	Batch
ProductoFinanciero	CRUD completo	—
ProductoCliente	CRUD + cambiar estado	—
CargoProducto	CRUD completo	Lectura (generar CargoCuotaDetalle)
PrelacionPago	CRUD completo	Lectura (aplicar pagos)
ConfiguracionTasa	CRUD completo	Lectura (devengar intereses)
TramoTasaEscalonada	CRUD completo	Lectura
TasaReferencia	CRUD + actualizar tasa	Lectura
HistorialTasaReferencia	Lectura (auditoría)	Crear (al cambiar tasa)
ConfiguracionMora	CRUD completo	Lectura (calcular mora)
NivelMoraConfig	CRUD completo	Lectura
ConfiguracionCapitalizacion	CRUD completo	Lectura (capitalizar)
CalendarioFeriado	CRUD completo	Lectura

1.2 Cuentas

Entidad	API (Endpoints)	Batch
Cuenta	CRUD + consultar saldos + movimientos	Actualizar saldos
PlazoFijo	Crear, renovar, liquidar, consultar	Procesar vencimiento
CuentaAportacion	CRUD + consultar	—
BloqueoSaldo	Crear, liberar, consultar	—

1.3 Créditos

Entidad	API (Endpoints)	Batch
CuentaCredito	Crear (desembolso), consultar, ajustar condiciones	Actualizar saldos y mora
TablaAmortizacion	Consultar, regenerar (prepago/refinanciamiento/cambio fecha)	Generar al desembolso
CuotaDetalle	Consultar (plan de pagos y estado)	Actualizar con pagos y mora
CargoCuotaDetalle	Consultar (desglose por cuota)	Generar al crear tabla
CambioFechaPago	Crear, consultar historial	Recalcular cuotas
Prepago	Registrar, consultar historial	Regenerar tabla
Refinanciamiento	Ejecutar, consultar historial	Regenerar tabla, limpiar mora

1.4 Línea de Crédito

Entidad	API (Endpoints)	Batch
CuentaLineaCredito	Crear, consultar, ajustar límite	Actualizar saldos
DisposicionLineaCredito	Registrar, consultar historial	Actualizar saldo utilizado
CorteLineaCredito	Consultar estados de cuenta	Generar en corte mensual

1.5 Transacciones y Pagos

Entidad	API (Endpoints)	Batch
Transaccion	Consultar, reversar	Crear (batch genera transacciones)
AplicacionPago	Consultar (desglose de pagos)	Crear (al aplicar pago)
PrelacionPago	CRUD completo	Lectura
Prepago	Registrar, consultar	Regenerar tabla

1.6 Exclusivamente Privado (Solo Motor Batch)

Entidad	API	Batch
DiaOperativo	Consultar estado, abrir/cerrar	Leer estado para procesar
DevengamientoInteres	Consultar (auditoría/reportes)	Crear diariamente
HistorialMora	Consultar (auditoría/reportes)	Crear al cambiar nivel
ProcesoBatch	Consultar (monitoreo)	Crear y actualizar

Nota: Estas 4 entidades SÍ tienen endpoints de consulta (el usuario necesita ver qué pasó en el batch, auditar devengamiento, ver historial de mora). Lo que NO tienen es endpoint de escritura — nadie crea un DevengamientoInteres manualmente, eso solo lo hace el motor.

2. Diseño de Endpoints API

2.1 Productos

GET	/api/productos	→ Lista productos (filtros:
GET	/api/productos/{id}	→ Detalle completo (condici
POST	/api/productos	→ Crear producto financiero
PUT	/api/productos/{id}	→ Editar producto
PATCH	/api/productos/estado/{id}	→ Activar/desactivar
POST	/api/productos/simular-credito/{id}	→ Simula tabla de amortizac
	Body: { monto, plazo, tasa, diaPago, metodo, esCuotaBalon?, porcentajeBalon	
	Response: { cuotaMensual, totalIntereses, tablaProyectada[], montoBalon? }	

2.2 Cargos por Producto

GET	/api/productos/cargos/{id}	→ Lista cargos del producto
POST	/api/productos/cargos/{id}	→ Agregar cargo
PUT	/api/productos/{id}/cargos/{cargoId}	→ Editar cargo
DELETE	/api/productos/{id}/cargos/{cargoId}	→ Eliminar cargo

1.3 Prelación de Pagos

GET	/api/productos/prelacion/{id}	→ Lista prelación con orden
POST	/api/productos/prelacion/{id}	→ Agregar cargo a prelación
PUT	/api/productos/{id}/prelacion/{preId}	→ Editar orden/cargo
DELETE	/api/productos/{id}/prelacion/{preId}	→ Eliminar de prelación

1.4 Configuración de Tareas

1.4 Configuración de Tasas

GET	/api/productos/tasa/{id}	→ Config de tasa del produc
POST	/api/productos/tasa/{id}	→ Crear configuración
PUT	/api/productos/{id}/tasa/{tasaId}	→ Editar
GET	/api/productos/{id}/tasa/{tasaId}/tramos	→ Tramos escalonados
POST	/api/productos/{id}/tasa/{tasaId}/tramos	→ Agregar tramo
PUT	/api/productos/{id}/tasa/{tasaId}/tramos/{tId}	→ Editar tramo
DELETE	/api/productos/{id}/tasa/{tasaId}/tramos/{tId}	→ Eliminar tramo
GET	/api/tasas-referencia	→ Lista tasas de referencia
POST	/api/tasas-referencia	→ Crear tasa de referencia
PUT	/api/tasas-referencia/{id}	→ Actualizar tasa (genera h
GET	/api/tasas-referencia/{id}/historial	→ Historial de cambios

1.5 Configuración de Mora

GET	/api/productos/mora/{id}	→ Config de mora del produc
POST	/api/productos/mora/{id}	→ Crear configuración
PUT	/api/productos/{id}/mora/{moraId}	→ Editar
GET	/api/productos/{id}/mora/{moraId}/niveles	→ Niveles de escalamiento
POST	/api/productos/{id}/mora/{moraId}/niveles	→ Agregar nivel
PUT	/api/productos/{id}/mora/{moraId}/niveles/{nId}	→ Editar nivel
DELETE	/api/productos/{id}/mora/{moraId}/niveles/{nId}	→ Eliminar nivel

1.6 Configuración de Capitalización

GET	/api/productos/capitalizacion/{id}	→ Config de capitalización
POST	/api/productos/capitalizacion/{id}	→ Crear
PUT	/api/productos/{id}/capitalizacion/{capId}	→ Editar

1.7 Producto-Cliente

GET	/api/producto-cliente	→ Lista (filtros: clienteId
GET	/api/producto-cliente/{id}	→ Detalle
POST	/api/producto-cliente	→ Crear relación
PATCH	/api/producto-cliente/{id}/estado	→ Cambiar estado

1.8 Cuentas

GET	/api/cuentas	→ Lista cuentas (filtros: c
GET	/api/cuentas/{id}	→ Detalle completo: saldos,

GET	/api/cuentas/{id}/saldo	→ Solo saldos (liviano)
GET	/api/cuentas/{id}/movimientos	→ Transacciones paginadas
	Query: ?desde=2026-01-01&hasta=2026-02-01&tipo=PagoCuota&pagina=1&porPagina	
GET	/api/cuentas/{id}/bloqueos	→ Bloqueos activos
POST	/api/cuentas/{id}/bloquear	→ Bloquear saldo
	Body: { monto, tipoBloqueo, motivo, numeroDocumento }	
POST	/api/cuentas/{id}/desbloquear	→ Liberar bloqueo
	Body: { bloqueoId, motivo }	

1.9 Créditos

GET	/api/creditos/{cuentaId}	→ Detalle completo del crédito
	Response: condiciones, saldos, estado mora, días mora, método, balón, etc.	
GET	/api/creditos/{cuentaId}/plan-pagos	→ Tabla de amortización vigente
GET	/api/creditos/{cuentaId}/cuota-actual	→ Cuota vigente con desglose
GET	/api/creditos/{cuentaId}/resumen-deuda	→ Total adeudado a la fecha
GET	/api/creditos/{cuentaId}/historial-mora	→ Historial de cambios de mora
GET	/api/creditos/{cuentaId}/historial-tablas	→ Versiones de tabla de amortización
GET	/api/creditos/{cuentaId}/cambios-fecha	→ Historial de cambios de fecha
GET	/api/creditos/{cuentaId}/prepagos	→ Historial de prepagos
GET	/api/creditos/{cuentaId}/refinanciamientos	→ Historial de refinanciamientos
POST	/api/creditos/desembolsar	→ Desembolsar crédito
	Body: { productoClienteId, monto, plazo, tasa, diaPago, metodoAmortizacion, frecuenciaPago, baseDias, esCuotaBalon?, tipoCuotaBalon?, porcentajeBalon?, tasaMoratoria, diasGracia, tipoCalculoMora }	
	Acción: Crea Cuenta + CuentaCredito + TablaAmortizacion + CuotaDetalle[] +	
	Response: { cuentaId, numeroCuenta, tablaAmortizacionId, cuotas[] }	
POST	/api/creditos/{cuentaId}/pagar	→ Aplicar pago de cuota
	Body: { monto, referencia, canal }	
	Response: { transaccionId, desglose[], cuotasPagadas[], nuevoSaldoCapital,	
POST	/api/creditos/{cuentaId}/prepagar	→ Pago anticipado
	Body: { monto, tipoPrepago, canal }	
	Response: { transaccionId, nuevoSaldo, nuevaCuota?, cuotasRestantes?, nueva	
POST	/api/creditos/{cuentaId}/cambiar-dia-pago	→ Cambiar día de pago
	Body: { nuevoDiaPago, motivo, aplicaDesdeCuota }	
	Acción: Registra CambioFechaPago + recalcula fechas cuotas + nueva TablaAmo	
POST	/api/creditos/{cuentaId}/refinanciar	→ Refinanciar crédito
	Body: { nuevaTasa, nuevoPlazo, nuevoDiaPago, nuevoMetodo, nuevaFrecuencia,	
	Acción: Snapshot estado actual + nueva tabla + limpia mora	
	Response: { refinanciamientoId, nuevaTablaId, nuevaCuota, cuotas[] }	

POST	/api/creditos/{cuentaId}/condonar	→ Condonar mora/interés/cargos
	Body: { tipo: "mora" "interes" "cargos", monto?, motivo }	
POST	/api/creditos/{cuentaId}/reversar-pago	→ Reversar un pago aplicado
	Body: { transaccionId, motivo }	
	Acción: Crea transacción de reverso + deshace aplicación	

1.10 Ahorros

GET	/api/ahorros/{cuentaId}	→ Detalle: saldo, interés a
GET	/api/ahorros/{cuentaId}/intereses-devengados	→ Detalle diario de intere
POST	/api/ahorros/abrir-cuenta	→ Abrir cuenta de ahorro vi
	Body: { productoClienteId, depositoInicial? }	
POST	/api/ahorros/{cuentaId}/depositar	→ Registrar depósito
	Body: { monto, referencia, canal }	
POST	/api/ahorros/{cuentaId}/retirar	→ Registrar retiro
	Body: { monto, referencia, canal }	
POST	/api/ahorros/{cuentaId}/transferir	→ Transferencia entre cuent
	Body: { cuentaDestinoId, monto, concepto }	

1.11 Plazo Fijo (CDP)

GET	/api/cdp/{cuentaId}	→ Detalle: monto, plazo, ta
GET	/api/cdp/{cuentaId}/proyeccion	→ Proyección de intereses a
POST	/api/cdp/constituir	→ Constituir nuevo CDP
	Body: { productoClienteId, monto, plazoEnDias, renovacionAutomatica, accion	
POST	/api/cdp/{cuentaId}/renovar	→ Renovar manualmente
POST	/api/cdp/{cuentaId}/liquidar	→ Liquidar anticipadamente
POST	/api/cdp/{cuentaId}/cambiar-renovacion	→ Cambiar política de renov
	Body: { renovacionAutomatica, accionAlVencimiento }	

1.12 Aportaciones (Cooperativas)

GET	/api/aportaciones/{cuentaId}	→ Detalle: saldo, mínimo, e
POST	/api/aportaciones/abrir	→ Abrir cuenta de aportació
POST	/api/aportaciones/{cuentaId}/aportar	→ Registrar aportación
	Body: { monto, referencia }	
POST	/api/aportaciones/{cuentaId}/retirar	→ Retiro (solo al salir de
	Body: { monto, motivo }	

1.13 Línea de Crédito

GET	/api/linea-credito/{cuentaId}	→ Detalle: límite, utilizad
GET	/api/linea-credito/{cuentaId}/disposiciones	→ Historial de disposicion
GET	/api/linea-credito/{cuentaId}/cortes	→ Historial de estados de
POST	/api/linea-credito/abrir Body: { productoClienteId, limiteAprobado, tasa, diaCorte, diasParaPago, po	→ Crear línea de crédito
POST	/api/linea-credito/{cuentaId}/disponer Body: { monto, concepto }	→ Usar la línea
POST	/api/linea-credito/{cuentaId}/pagar Body: { monto, referencia, canal }	→ Pagar a la línea
PATCH	/api/linea-credito/{cuentaId}/limite Body: { nuevoLimite, motivo }	→ Ajustar límite aprobado

1.14 Transacciones

GET	/api/transacciones/{id}	→ Detalle de transacción
GET	/api/transacciones/{id}/aplicacion-pago	→ Desglose de cómo se apli
GET	/api/transacciones Query: ?cuentaId=&tipo=&desde=&hasta=&canal=&pagina=&porPagina=	→ Búsqueda global de transa

1.15 Día Operativo y Batch (Monitoreo)

GET	/api/dia-operativo/actual	→ Día contable actual y est
GET	/api/dia-operativo/historial	→ Historial de días operati
POST	/api/dia-operativo/abrir	→ Abrir nuevo día contable
POST	/api/dia-operativo/cerrar	→ Cerrar día manualmente
GET	/api/dia-operativo/pendientes	→ Días en Error que requier
GET	/api/batch/procesos	→ Lista procesos batch (fil
GET	/api/batch/procesos/{id}	→ Detalle con contadores
GET	/api/batch/procesos/{id}/transacciones	→ Transacciones generadas p

1.16 Calendario

GET	/api/calendario/feriados	→ Lista feriados (filtro: a
-----	--------------------------	-----------------------------

POST	/api/calendario/feriados	→ Crear feriado
PUT	/api/calendario/feriados/{id}	→ Editar
DELETE	/api/calendario/feriados/{id}	→ Eliminar

5. Flujos Críticos — Paso a Paso

5.1 Desembolso de Crédito

Endpoint: POST /api/creditos/desembolsar

1. Validar ProductoCliente activo
2. Validar monto: $\text{MontoMinimo} \leq \text{monto} \leq \text{MontoMaximo}$
3. Validar plazo: $\text{PlazoMinimoMeses} \leq \text{plazo} \leq \text{PlazoMaximoMeses}$
4. Validar tasa: $\text{TasaInteresAnualMinima} \leq \text{tasa} \leq \text{TasaInteresAnualMaxima}$
5. Validar DiaPago: 1-31
6. Crear Cuenta (TipoCuenta=CreditoSimple, Estado=Activa)
7. Crear CuentaCredito con todas las condiciones
8. Leer CargoProducto WHERE FrecuenciaCobro = AlDesembolso
 - Para cada cargo:
 - Si DescontarDelDesembolso = true → restar del monto neto
 - Si no → crear Transaccion de cobro aparte
9. Generar TablaAmortizacion (Version=1, MotivoRegeneracion="INICIAL")
 - Para cada cuota (1 a PlazoEnMeses):
 - a. Calcular FechaVencimiento con DiaPago:
 - SP: si DiaPago > diasDelMes → usa último día
 - SP: si DiaPago=31 → siempre último día
 - Ej: DiaPago=30, febrero 2026 → 28/02/2026
 - b. Calcular MontoCapital y MontoInteres según MetodoAmortizacion:
 - Esta dejame validar algo primero aca a lo interno
 - c. Si EsCuotaBalon y es la última cuota:
 - MontoCapital = SaldoCapital restante (incluye el balón)
 - Marcar EsCuotaBalon = true en la cuota
 - d. Leer CargoProducto WHERE FrecuenciaCobro = EnCadaCuota
 - Crear CargoCuotaDetalle por cada cargo recurrente
 - Sumar en CuotaDetalle.MontoTotalCargos
10. Crear Transaccion (tipo: Desembolso)
11. Actualizar CuentaCredito.SaldoCapital = MontoDesembolsado
12. Actualizar ProductoCliente.FechaDesembolso = hoy

5.2 Pago de Cuota con Prelación

Endpoint: POST /api/creditos/{cuentaId}/pagar

1. Obtener CuentaCredito y ProductoFinanciero
2. Leer PrelacionPago ORDER BY Orden ASC
 - Ejemplo: [InteresMoratorio(1), InteresCorriente(2), Capital(3), Seguro(4)]
3. Obtener cuotas: vencidas + cuota actual ORDER BY NumeroCuota ASC

```

3. obtener cuotas vencidas + cuota actual, ORDER BY NumeroCuota ASC
4. montoDisponible = monto del pago
5. Para cada cuota (de la más antigua a la más reciente):
   Para cada cargo en la prelación:

   Si cargo = InteresMoratorio:
     pendiente = cuota.MontoMoraGenerada - cuota.MoraPagada
     aplicar = min(montoDisponible, pendiente)
     → Crear AplicacionPago(TipoCargo=InteresMoratorio, MontoAplicado=aplicar)
     → cuota.MoraPagada += aplicar
     → montoDisponible -= aplicar

   Si cargo = InteresCorriente:
     pendiente = cuota.MontoInteres - cuota.InteresPagado
     aplicar = min(montoDisponible, pendiente)
     → Crear AplicacionPago(TipoCargo=InteresCorriente)
     → cuota.InteresPagado += aplicar
     → montoDisponible -= aplicar

   Si cargo = Capital:
     pendiente = cuota.MontoCapital - cuota.CapitalPagado
     aplicar = min(montoDisponible, pendiente)
     → Crear AplicacionPago(TipoCargo=Capital)
     → cuota.CapitalPagado += aplicar
     → CuentaCredito.SaldoCapital -= aplicar
     → montoDisponible -= aplicar

   Si cargo = Seguro (u otro cargo):
     → Recorrer CargoCuotaDetalle de esta cuota
     → Para cada cargo pendiente:
       aplicar = min(montoDisponible, cargo.MontoProyectado - cargo.MontoPagado)
       → Crear AplicacionPago
       → cargo.MontoPagado += aplicar
       → cuota.CargosPagados += aplicar
       → montoDisponible -= aplicar

   Si cuota totalmente pagada:
     → Estado = Pagada, FechaPago = hoy
   Si cuota parcialmente pagada:
     → Estado = PagadaParcial

   Si montoDisponible = 0 → break

6. Crear Transaccion (tipo: PagoCuota, SaldoAnterior, SaldoPosterior)
7. Recalcular estado de mora:
   → Si ya no hay cuotas vencidas: NivelMora = SinMora, DiasEnMora = 0
   → Si aún hay: recalcular DiasEnMora desde la cuota más antigua vencida
8. Response: { transaccionId, desglose[], cuotasPagadas[], nuevoSaldoCapital, sob

```

5.3 Cambio de Día de Pago

Endpoint: POST /api/creditos/{cuentaId}/cambiar-dia-pago

1. Validar nuevoDiaPago (1-31)
2. Validar que AplicaDesdeCuota sea una cuota pendiente
3. Registrar CambioFechaPago:
 - DiaPagoAnterior = CuentaCredito.DiaPago
 - DiaPagoNuevo = nuevoDiaPago
4. Actualizar CuentaCredito.DiaPago = nuevoDiaPago
5. Marcar TablaAmortizacion vigente como EsVigente = false
6. Crear nueva TablaAmortizacion:
 - Copiar cuotas pagadas tal cual
 - Para cuotas pendientes (>= AplicaDesdeCuota):
 - Recalcular FechaVencimiento con SP: (aca va un sp)
 - mes = mes de la cuota original
 - diasEnMes = días reales de ese mes
 - fechaVencimiento = min(nuevoDiaPago, diasEnMes) de ese mes/año
 - MontoCapital y MontoInteres NO cambian (solo fechas)
 - MotivoRegeneracion = "CAMBIO_FECHA"
 - TablaAnteriorId = tabla anterior
 - EsVigente = true
7. Vincular CambioFechaPago.NuevaTablaAmortizacionId

6. Reglas de Seguridad

Regla	Implementación
Operaciones sensibles requieren ApprovalFlow	Refinanciar, condonar, reversar, ajustar límite línea crédito
Transacciones son inmutables	No existe PUT/DELETE en transacciones. Reverso = nueva transacción
Día operativo controla operaciones	Si DiaOperativo.Estado != Abierto → rechazar operaciones financieras