Análisis Completo del Proyecto - Revisión de Módulos

Fecha: 31 de Octubre de 2025
Análisis: Revisión exhaustiva de rutas API, enums y campos del schema

🎯 Objetivo del Análisis

Revisar todos los módulos del proyecto para identificar problemas similares al encontrado en la creación de clientes:
1. Rutas API faltantes
2. Desajuste entre valores de enum en formularios vs schema de Prisma
3. Nombres de campos incorrectos en las consultas

✅ Estado de Enums - CORRECTOS

EmploymentType ✓

Schema Prisma:

```
enum EmploymentType {
  EMPLOYED
  SELF_EMPLOYED
  UNEMPLOYED
  RETIRED
  STUDENT
}
```

Uso en formularios: ✅ CORRECTO
- `/app/admin/clients/new/page.tsx` - Corregido
- `/app/admin/clients/[id]/edit/page.tsx` - Corregido

LoanType ✓

Schema Prisma:

```
enum LoanType {
  PERSONAL
  BUSINESS
  MORTGAGE
  AUTO
  EDUCATION
}
```

Uso en formularios: ✅ CORRECTO
- `/components/loans/new-loan-form.tsx`
- `/components/loans/loan-form.tsx`
- `/components/credit-applications/credit-application-form.tsx`

ApplicationStatus ✓

Schema Prisma:

```
enum ApplicationStatus {
  PENDING
  UNDER_REVIEW
  APPROVED
  REJECTED
  CANCELLED
}
```

Uso en componentes: ✅ CORRECTO
- `/components/credit-applications/credit-application-review.tsx`
- `/components/credit-applications/credit-applications-list.tsx`

LoanStatus ✓

Schema Prisma:

```
enum LoanStatus {
  ACTIVE
  PAID_OFF
  DEFAULTED
  CANCELLED
}
```

Uso en componentes: ✅ CORRECTO
- `/components/loans/loan-list.tsx`

UserStatus ✓

Schema Prisma:

```
enum UserStatus {
  ACTIVE
  INACTIVE
  SUSPENDED
}
```

Uso en componentes: ✅ CORRECTO
- `/components/admin/user-management.tsx`

PaymentStatus ✓

Schema Prisma:

```
enum PaymentStatus {
  PENDING
  COMPLETED
  FAILED
  CANCELLED
}
```

Uso: ✅ CORRECTO en todos los componentes

WhatsAppMessageType ✓

Schema Prisma:

```
enum WhatsAppMessageType {
  PAYMENT_RECEIVED
  PAYMENT_REMINDER
  LOAN_APPROVED
  LOAN_UPDATE
  MARKETING
  CUSTOM
}
```

Uso en componentes: ✅ CORRECTO
- `/components/admin/whatsapp-messages-dashboard.tsx`

✅ Estado de Campos del Schema - CORRECTOS

User Model ✓

Campos correctos usados:
- `firstName` y `lastName` (NO `name`) ✅
- Todas las rutas API usan correctamente `firstName` + `lastName`

Archivos verificados:
- `/api/clients/route.ts` ✅
- `/api/credit-applications/route.ts` ✅
- `/api/loans/route.ts` ✅

Loan Model ✓

Campos correctos usados:
- `principalAmount` (NO `loanAmount`) ✅
- `balanceRemaining` (NO `remainingBalance`) ✅

Componentes verificados:
- `/components/loans/loan-list.tsx` ✅
- `/components/loans/loan-form.tsx` ✅
- `/components/loans/loan-details.tsx` ✅
- `/components/loans/loan-detail.tsx` ✅
- `/components/loans/new-loan-form.tsx` ✅

AmortizationSchedule Model ✓

Campo `remainingBalance`: ✅ CORRECTO
- Este modelo SÍ tiene `remainingBalance` en el schema
- Uso en `/api/credit-applications/[id]/review/route.ts` es correcto

❌ PROBLEMAS ENCONTRADOS

1. Ruta API Faltante: `/api/payments/route.ts`

Problema:
- El componente `/components/payments/payment-history.tsx` hace fetch a `/api/payments`
- La ruta API no existe, solo hay subrutas:

- `/api/payments/cash/`
- `/api/payments/transactions/`

Impacto:
- El historial de pagos no funciona
- Error 404 al cargar el componente PaymentHistory

Solución requerida:
- Crear `/api/payments/route.ts` con endpoint GET
- Implementar paginación y filtros
- Control de acceso por rol

2. Posible Inconsistencia: FileCategory (NO CRÍTICO)

Schema Prisma:

```
enum FileCategory {
  IDENTITY_DOCUMENT
  INCOME_PROOF
  BANK_STATEMENT
  CONTRACT
  SIGNATURE
  PHOTO
  OTHER
}
```

Valores usados en componentes:
- `/components/files/document-manager.tsx` : usa valores custom como 'identification', 'income', 'address'
- `/components/files/file-manager.tsx` : usa valores custom como 'all', 'identification', 'income_proof'

Nota: Estos parecen ser valores de UI/filtros locales, no valores que se envían directamente a la DB. Requiere verificación adicional si hay problemas con la carga de archivos.

📊 Resumen de Rutas API

Rutas con route.ts ✅

- `/api/clients/route.ts` ✅ (recién creado)
- `/api/credit-applications/route.ts` ✅
- `/api/loans/route.ts` ✅
- `/api/notifications/route.ts` ✅
- `/api/personal-references/route.ts` ✅
- `/api/test-users/route.ts` ✅

Rutas sin route.ts principal (pero con subrutas funcionales) ⚠️

- `/api/admin/` - Solo contenedor de subrutas ✅
- `/api/files/` - Solo tiene `/api/files/[...path]` ✅
- `/api/public/` - Solo contenedor ✅
- `/api/reports/` - Solo subrutas específicas ✅
- `/api/webhooks/` - Solo webhooks específicos ✅
- `/api/whatsapp/` - Solo subrutas específicas ✅

Rutas faltantes críticas ❌

- `/api/payments/route.ts` ❌ **REQUIERE ACCIÓN**

🔍 Verificación de Campos en Consultas

Búsqueda de campos problemáticos:

```
# name: en select (debería ser firstName/lastName)
❌ 0 ocurrencias encontradas ✅

# loanAmount (debería ser principalAmount)
❌ 0 ocurrencias encontradas ✅

# remainingBalance en Loan (debería ser balanceRemaining)
✅ 1 ocurrencia en AmortizationSchedule (correcto) ✅
```

📋 Acciones Requeridas

Prioridad ALTA 🔴

1. **Crear `/api/payments/route.ts`**
   - Endpoint GET para listar pagos
   - Filtros por status, clientId, loanId
   - Paginación
   - Control de acceso por rol

Prioridad MEDIA 🟡

1. **Verificar FileCategory** (si hay problemas con subida de archivos)
   - Revisar si los valores custom causan problemas
   - Alinear con enum del schema si es necesario

Prioridad BAJA 🟢

1. **Documentación**
   - Mantener actualizada la lista de enums
   - Documentar convenciones de nomenclatura

✅ Conclusiones

Puntos Positivos:

- ✅ La mayoría de los enums están correctamente alineados
- ✅ Los campos del schema se usan correctamente en casi todo el proyecto
- ✅ El fix de `EmploymentType` en clientes fue el único desajuste de enum encontrado
- ✅ El fix de campos `name` → `firstName/lastName` está completo
- ✅ Los campos de Loan (`principalAmount` , `balanceRemaining`) se usan correctamente

Problemas Encontrados:

- ❌ Falta crear `/api/payments/route.ts` - **CRÍTICO**
- ⚠️ Posible inconsistencia en FileCategory - **NO CRÍTICO**

Estado General:

🟢 **BUENO** - Solo un problema crítico encontrado (payments route) y todo lo demás está correctamente implementado.

Próximo paso: Crear la ruta `/api/payments/route.ts` para completar la funcionalidad del historial de pagos.

Documentado por: DeepAgent
Proyecto: EscalaFin MVP - Sistema de Gestión de Préstamos