🍲 Olla del Barrio - Sistema de Pagos y Anti-Bypass

Marketplace de comidas caseras AMBA con sistema de pagos integrado y protección anti-bypass.

📋 Características Principales

💰 Sistema de Pagos

- **Comisión variable**: 8-12% según loyalty tier (SOLO sobre comida, NO sobre envío)
- **Pagos semanales automáticos**: Cálculo y transferencia a CBU de productores
- **Integración Mercado Pago**: Webhooks automáticos para aprobación de pagos
- **Reportes financieros**: Dashboards para productores y admin

🔒 Sistema Anti-Bypass (CRÍTICO)

Información Oculta Hasta Pago

1. **Dirección exacta**: Solo visible después de pago + 30min antes del retiro
2. **Teléfono real**: Reemplazado por proxy de Twilio hasta pago
3. **Foto fachada**: Versión con blur hasta pago aprobado
4. **Chat interno obligatorio**: Con detección automática de números

Protecciones Implementadas

```sql
-- Triggers automáticos
- detect_phone_numbers() → Censura números en chat
- update_order_on_verified_payment() → Revela info solo si pagó
- create_manufacturer_transfer_on_verified() → Transferencia automática

-- RLS Policies
- Dirección completa: NULL hasta pago
- Teléfono: Solo proxy visible
- Chat: Números bloqueados automáticamente
```

Sistema de Reportes

- Repartidores reportan bypass → **Bono $500** si se confirma
- Productores penalizados → **Comisión sube a 15%**
- 3 reportes confirmados → **Suspensión cuenta**

📊 Comisiones y Loyalty Tiers

Tier	Comisión	Requisito
0 - Normal	12%	Default
1 - Bronce	10%	50 ventas sin bypass
2 - Plata	8%	200 ventas sin bypass
Penalización	15%	Bypass confirmado

IMPORTANTE: La comisión se calcula SOLO sobre `subtotal_cents` (comida), NUNCA sobre `envio_cents`.

🗂️ Estructura de Tablas

Tablas Principales

```
profiles → Usuarios base (cliente/productor/repartidor/admin)
producers → Productores con datos de contacto ocultos
dishes → Catálogo de platos
orders → Órdenes con anti-bypass integrado
order_items → Items de cada orden
payments → Pagos de Mercado Pago
payouts → Pagos semanales a productores
```

Tablas Anti-Bypass

```
chat_messages → Chat con censura automática
bypass_reports → Reportes de intentos de bypass
producer_bank_accounts → CBU/Alias para pagos
```

🚀 Flujo de Pago Completo

1. Cliente Hace Pedido

typescript

```
const result = await supabase.rpc('create_order', {
  p_client_id: userId,
  p_producer_id: producerId,
  p_items: [
    { dish_id: 'xxx', cantidad: 2 },
    { dish_id: 'yyy', cantidad: 1 }
  ],
  p_delivery_type: 'retiro',
  p_direccion_aproximada: 'Zona Palermo, cerca de Plaza Serrano',
  p_horario_retiro_desde: '2024-01-15 13:00:00'
});

// Resultado:
{
  order_id: 'abc-123',
  total_cents: 500000, // $5000
  comision_cents: 60000, // 12% de subtotal
  subtotal_cents: 500000,
  envio_cents: 0
}
```

2. Cliente Ve Datos Para Transferir

typescript

```
// En este punto, el cliente ve:
- Dirección aproximada: "Zona Palermo" ❌ NO exacta
- Teléfono: OCULTO ❌
- Foto fachada: CON BLUR ❌

// Solo puede proceder a pagar con Mercado Pago
```

3. Cliente Paga con Mercado Pago

typescript

```
// Webhook automático → FastAPI
POST /webhooks/mercadopago
{
  "type": "payment",
  "action": "payment.updated",
  "data": { "id": "123456789" }
}

// FastAPI llama a Supabase RPC
supabase.rpc('process_payment_approval', {
  p_mp_payment_id: '123456789',
  p_webhook_data: {...}
});
```

4. Sistema Revela Información

typescript

```
// Automáticamente después del pago:
update orders set
  info_revelada_at = now(),
  info_expira_at = horario_retiro_hasta + interval '1 hour',
  status = 'pagado'
where id = order_id;

// AHORA el cliente puede llamar:
const contactInfo = await supabase.rpc('reveal_contact_info', {
  p_order_id: orderId,
  p_user_id: userId
});

// Resultado:
{
  producer: {
    nombre: "Las Milanesas de Rosa",
    direccion: "Av. Corrientes 1234, Palermo", ✅ EXACTA
    telefono: "+5491134567890", ✅ PROXY TWILIO
    foto_fachada: "url-sin-blur.jpg" ✅ SIN BLUR
  },
  expira_en: 7200 // 2 horas
}
```

5. Comisión Retenida Automáticamente

sql

```sql
-- Trigger ejecutado automáticamente:
insert into manufacturer_transfers (
  manufacturer_id,
  order_id,
  monto_bruto, -- $5000
  comision_cents, -- $600 (12%)
  monto_neto -- $4400 (88%)
) values (...);
```

6. Pago Semanal a Productor

python

```
# Cron semanal (GitHub Actions o manual)
POST /admin/payouts/generate
{
  "semana_inicio": "2024-01-01",
  "semana_fin": "2024-01-07"
}

# Resultado:
{
  "payouts_created": 15,
  "total_ventas": 500000,
  "total_comisiones": 60000,
  "total_neto": 440000
}
```

🛡️ Seguridad RLS

Información Sensible NUNCA Expuesta

sql

```
-- ❌ El cliente NUNCA puede hacer:
select direccion_completa from producers; -- RLS lo bloquea

-- ❌ El productor NUNCA puede ver antes del pago:
select telefono_cliente from orders where id = 'xxx'; -- NULL hasta pago

-- ✅ Solo después del pago Y usando RPC:
select * from reveal_contact_info('order-id', 'user-id'); -- ✅ OK
```

Bypass Detection

sql

```
-- Trigger automático en CADA mensaje de chat:
create trigger detect_phone_numbers_trigger
 before insert on chat_messages
 for each row execute function detect_phone_numbers();

-- Resultado:
{
 mensaje_original: "Llamame al 11-1234-5678",
 mensaje_censurado: "Llamame al •••••• (usá el chat de la app)",
 contiene_numero: true,
 bloqueado: true
}
```

🧮 APIs Principales

Supabase RPCs

typescript

```
// Crear orden con reserva de stock
create_order(...)

// Revelar contacto SOLO si pagó
reveal_contact_info(order_id, user_id)

// Calcular comisión según tier
calculate_commission(producer_id, subtotal_cents)

// Reportar bypass
report_bypass(order_id, reportado_por, tipo, evidencia)

// Confirmar bypass (admin)
confirm_bypass_report(report_id, admin_id, confirmado)

// Generar payouts semanales
generate_weekly_payouts(semana_inicio, semana_fin)

// Buscar productores cercanos
search_nearby_producers(lat, lng, max_distance_km)
```

FastAPI Endpoints

python

```
# Webhook Mercado Pago
POST /webhooks/mercadopago

# Sincronizar Notion → Supabase
POST /sync/notion

# Generar payouts (admin)
POST /admin/payouts/generate
POST /admin/payouts/process

# Reportes
GET /admin/reports/weekly?semana_inicio=...&semana_fin=...
GET /admin/reports/bypass?status=reportado
```

🔧 Configuración

Variables de Entorno

bash

```
# Supabase
SUPABASE_URL=https://xxx.supabase.co
SUPABASE_ANON_KEY=eyJxxx...
SUPABASE_SERVICE_ROLE_KEY=eyJxxx...

# Mercado Pago
MP_ACCESS_TOKEN=APP_USR-xxx
MP_WEBHOOK_SECRET=xxx

# Notion (opcional)
NOTION_TOKEN=secret_xxx
NOTION_DATABASE_ID=xxx

# Twilio (proxy de teléfono)
TWILIO_ACCOUNT_SID=ACxxx
TWILIO_AUTH_TOKEN=xxx
TWILIO_PHONE_NUMBER=+5491123456789
```

Ejecución

bash

```
# Migración inicial
psql -h db.xxx.supabase.co -U postgres < schema.sql
psql -h db.xxx.supabase.co -U postgres < rpc_functions.sql
psql -h db.xxx.supabase.co -U postgres < rls_policies.sql
psql -h db.xxx.supabase.co -U postgres < seed.sql

# FastAPI (Railway/Render)
pip install -r requirements.txt
uvicorn main:app --host 0.0.0.0 --port 8000

# Next.js (Vercel)
npm install
npm run build
npm start
```

📊 Dashboards

Dashboard Productor

- ✅ Ventas del día/semana/mes
- ✅ Comisiones retenidas
- ✅ Próximo pago estimado
- ✅ Loyalty tier actual
- ✅ Gestión de menú (Notion sync)
- ✅ Reportes de bypass (si alguno)

Dashboard Admin

- ✅ Comprobantes pendientes verificación
- ✅ Reportes de bypass para investigar
- ✅ Payouts pendientes de proceso
- ✅ Estadísticas generales
- ✅ Gestión de usuarios/penalizaciones

Dashboard Cliente

- ✅ Historial de pedidos
- ✅ Favoritos y suscripciones
- ✅ Chat con productor (filtrado)
- ✅ Seguimiento en tiempo real

🎯 Métricas de Éxito Anti-Bypass

```sql
-- % de transacciones dentro de la plataforma
select
  count(*) filter (where status = 'entregado') as completadas,
  count(*) filter (where exists (
    select 1 from bypass_reports br
    where br.order_id = orders.id
    and br.status = 'confirmado'
  )) as bypass_confirmados,
  (1 - count(*) filter (where exists (...)) / count(*)::float) * 100 as tasa_exito
from orders;

-- Resultado esperado: >95% de tasa de éxito
```

📞 Soporte

- **Reportar bypass**: Desde dashboard de repartidor/admin
- **Problemas técnicos**: admin@olladelbarrio.com
- **WhatsApp soporte**: +54 9 11 1234-5678

Desarrollado con ❤️ para conectar abuelas con el barrio