

Universidad de Ingeniería y Tecnología (UTEC)

Maestría en Ciencia de Datos & IA

eToro

Análisis de Negocio y Modelado Relacional

UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA



Integrantes:

Diego Sánchez Salazar
Estuardo Oliver Campos
Nicole Arenas Lazo
Gianmarco Mejía Meléndez
Piero Palacios Bernuy
Winton Flores Quispe
Edgard Inga Froylán

Docente: Luis Chicoma

Lima, 2025

Índice general

1. Análisis de la Empresa: eToro	2
1.1. Giro de Negocio	2
1.2. Clientes	2
1.2.1. Segmentos de Clientes	2
1.2.2. Alcance Geográfico	2
1.3. Productos y Servicios	2
1.4. Canales	3
2. Modelo Entidad–Relación (ER)	4
3. Business Model Canvas de eToro	14
3.1. Canal principal de eToro	15
4. Consultas SQL para la Generación de Features en Modelos de IA	16
4.1. Planteamiento de Preguntas de Negocio	16
4.2. ¿Qué tan comprometido está el usuario en los últimos 30 días?	16
4.3. ¿Cómo es el comportamiento y la performance de trading en los últimos 90 días?	18
4.4. ¿Qué tan concentrado está el riesgo de las posiciones abiertas del usuario?	19
4.5. Concentración de Riesgo	20
4.6. Influencia Social y Copy Trading	20
5. Referencias Bibliográficas	21

Capítulo 1

Análisis de la Empresa: eToro

1.1. Giro de Negocio

eToro es una plataforma de corretaje fintech que opera en el espacio del trading e inversión en línea (multi-activos) con un componente de red social denominado “social trading”. Combina funciones de bróker electrónico con elementos de *matchmaking* social, permitiendo que los usuarios copien estrategias de otros inversores. Su modelo de ingresos se basa en comisiones, *spreads* y tarifas adicionales (retiros, conversiones, inactividad, tarifas nocturnas), además de los diferenciales entre precios de compra y venta.

1.2. Clientes

1.2.1. Segmentos de Clientes

Los principales clientes de eToro son inversionistas minoristas, principiantes o intermedios que valoran la posibilidad de copiar estrategias, buscan una experiencia digital intuitiva y operan con cuentas financiadas. Aunque en algunos mercados también tienen presencia institucional, su foco principal es el cliente minorista.

1.2.2. Alcance Geográfico

Opera en más de 100 países, concentrando sus ingresos en Europa y Reino Unido, con expansión en Asia-Pacífico y América. En 2024 sus ingresos fueron de USD 931 millones, principalmente por el trading de criptomonedas y acciones.

1.3. Productos y Servicios

- **Activos:** Acciones, criptomonedas, ETFs, divisas, materias primas e índices.

- **Trading social:** CopyTrader y Smart Portfolios.
- **Intereses:** Rendimiento sobre saldo no invertido.
- **Cuentas y servicios:** Conversión de divisas, depósitos, retiros y soporte.
- **Recompensas:** Programas para inversores populares.

1.4. Canales

Los principales canales de interacción incluyen:

- Plataforma web.
- Aplicación móvil (iOS y Android).
- Centro de ayuda y atención al cliente.
- Marketing digital, contenido educativo y redes sociales.
- Comunidad social interna.

Capítulo 2

Modelo Entidad–Relación (ER)

A continuación se muestra el esquema ER diseñado para el canal principal (aplicación móvil). Incluye la definición del proyecto, todos los *enums* y las tablas con sus relaciones.

Listing 2.1: Modelo ER completo: proyecto, enums y tablas

```
1 Project etoro_mobile_app { database_type: "PostgreSQL"}
2
3 Enum device_platform {
4     ios
5     android}
6
7 Enum order_side {
8     buy
9     sell}
10
11 Enum order_type {
12     market
13     limit
14     stop
15     stop_limit}
16
17 Enum time_in_force {
18     day
19     gtc
20     ioc
21     fok}
22
23 Enum order_status {
24     pending
25     placed
26     partial_filled
27     filled
28     canceled
29     rejected}
```

```

30
31 Enum position_status {
32     open
33     closed}
34
35 Enum kyc_status {
36     pending
37     approved
38     rejected
39     resubmission_required}
40
41 Enum risk_level {
42     low
43     medium
44     high
45     very_high}
46
47 Enum payment_method {
48     card
49     bank_transfer
50     ewallet
51     crypto}
52
53 Enum payment_status {
54     initiated
55     settled
56     failed
57     reversed}
58
59 Enum notification_channel {
60     in_app
61     push}
62
63 Enum event_source {
64     app_ui
65     background
66     push_open
67     deep_link
68     sdk}
69
70 Enum instrument_type {
71     equity
72     etf
73     crypto
74     forex
75     commodity
76     index}

```

```

77
78 Table users {
79     id uuid [pk]
80     email varchar [unique, not null]
81     phone varchar
82     display_name varchar
83     country_code varchar(2)
84     created_at timestamptz [not null]
85     status varchar(20) // active, suspended, closed}
86
87 Table user_auth {
88     id uuid [pk]
89     user_id uuid [not null, ref: > users.id]
90     provider varchar(50) // password, apple, google, etc.
91     provider_uid varchar(200)
92     last_login_at timestamptz
93     mfa_enabled bool}
94
95 Table kyc_profiles {
96     id uuid [pk]
97     user_id uuid [not null, ref: > users.id]
98     status kyc_status [not null, default: 'pending']
99     document_type varchar(30) // id_card, passport, driver_license
100    document_country varchar(2)
101    submitted_at timestamptz
102    reviewed_at timestamptz}
103
104 Table risk_assessments {
105     id uuid [pk]
106     user_id uuid [not null, ref: > users.id]
107     level risk_level [not null]
108     questionnaire_version varchar(20)
109     score int
110     assessed_at timestamptz [not null]}
111
112 Table regulatory_consent {
113     id uuid [pk]
114     user_id uuid [not null, ref: > users.id]
115     consent_code varchar(50) // tos, privacy, marketing, pds, etc.
116     accepted bool [not null, default: true]
117     accepted_at timestamptz [not null]
118     locale varchar(10)}
119
120 Table devices {
121     id uuid [pk]
122     user_id uuid [not null, ref: > users.id]
123     platform device_platform [not null]

```

```

124     os_version varchar(40)
125     app_version varchar(20)
126     device_model varchar(80)
127     installed_at timestamptz
128     last_seen_at timestamptz}
129
130 Table push_tokens {
131     id uuid [pk]
132     device_id uuid [not null, ref: > devices.id]
133     token varchar(300) [not null]
134     provider varchar(20) // apns, fcm
135     valid bool [default: true]
136     created_at timestamptz
137     invalidated_at timestamptz}
138
139 Table app_sessions {
140     id uuid [pk]
141     user_id uuid [not null, ref: > users.id]
142     device_id uuid [not null, ref: > devices.id]
143     started_at timestamptz [not null]
144     ended_at timestamptz
145     city varchar(100)
146     ip inet
147     is_foreground bool}
148
149 Table app_events {
150     id uuid [pk]
151     session_id uuid [not null, ref: > app_sessions.id]
152     user_id uuid [ref: > users.id]
153     device_id uuid [ref: > devices.id]
154     event_name varchar(100) [not null] // e.g., 'PlaceOrder_Tap'
155     event_source event_source [not null, default: 'app_ui']
156     event_ts timestamptz [not null]
157     screen varchar(100)
158     metadata jsonb}
159
160 Table notifications {
161     id uuid [pk]
162     user_id uuid [not null, ref: > users.id]
163     channel notification_channel [not null]
164     title varchar(140)
165     body text
166     created_at timestamptz [not null]
167     delivered_at timestamptz
168     opened_at timestamptz
169     deeplink varchar(300)}
170

```



```

171 Table attribution_installs {
172     id uuid [pk]
173     device_id uuid [not null, ref: > devices.id]
174     network varchar(60)
175     campaign varchar(120)
176     adgroup varchar(120)
177     click_ts timestamptz
178     install_ts timestamptz}
179
180 Table currencies {
181     code varchar(3) [pk] // ISO
182     name varchar(30)
183     symbol varchar(5)}
184
185 Table exchange_rates {
186     id uuid [pk]
187     base_currency varchar(3) [not null, ref: > currencies.code]
188     quote_currency varchar(3) [not null, ref: > currencies.code]
189     rate numeric(18,8) [not null]
190     as_of timestamptz [not null]}
191
192 Table accounts {
193     id uuid [pk]
194     user_id uuid [not null, ref: > users.id]
195     base_currency varchar(3) [not null, ref: > currencies.code]
196     opened_at timestamptz [not null]
197     is_margin_enabled bool [default: false]
198     status varchar(20) // active, restricted, closed}
199
200 Table account_balances {
201     id uuid [pk]
202     account_id uuid [not null, ref: > accounts.id]
203     currency varchar(3) [not null, ref: > currencies.code]
204     cash_available numeric(20,2) [not null, default: 0]
205     cash_locked numeric(20,2) [not null, default: 0]
206     updated_at timestamptz [not null]}
207
208 Table ledger_entries {
209     id uuid [pk]
210     account_id uuid [not null, ref: > accounts.id]
211     currency varchar(3) [not null, ref: > currencies.code]
212     amount numeric(20,2) [not null] // +credit, -debit
213     type varchar(40) // trade_fill, deposit, withdrawal, fee,
        fx_conversion, interest
214     reference_id uuid
215     created_at timestamptz [not null]}
216

```

```

217 Table payments {
218     id uuid [pk]
219     user_id uuid [not null, ref: > users.id]
220     method payment_method [not null]
221     status payment_status [not null]
222     currency varchar(3) [not null, ref: > currencies.code]
223     amount numeric(20,2) [not null]
224     provider varchar(40)
225     created_at timestamptz [not null]
226     settled_at timestamptz
227     failure_reason varchar(200)}
228
229 Table deposits {
230     id uuid [pk]
231     account_id uuid [not null, ref: > accounts.id]
232     payment_id uuid [not null, ref: > payments.id]
233     amount numeric(20,2) [not null]
234     created_at timestamptz [not null]}
235
236 Table withdrawals {
237     id uuid [pk]
238     account_id uuid [not null, ref: > accounts.id]
239     payment_id uuid [not null, ref: > payments.id]
240     amount numeric(20,2) [not null]
241     fee numeric(20,2) [default: 0]
242     created_at timestamptz [not null]
243 }
244
245 Table instruments {
246     id uuid [pk]
247     symbol varchar(30) [not null] // e.g., AAPL, BTC
248     name varchar(120)
249     type instrument_type [not null]
250     quote_currency varchar(3) [ref: > currencies.code]
251     is_tradable bool [default: true]}
252
253 Table instrument_prices {
254     id uuid [pk]
255     instrument_id uuid [not null, ref: > instruments.id]
256     price numeric(20,8) [not null]
257     as_of timestamptz [not null]
258     source varchar(40)}
259
260 Table orders {
261     id uuid [pk]
262     account_id uuid [not null, ref: > accounts.id]
263     instrument_id uuid [not null, ref: > instruments.id]

```

```

264     side order_side [not null]
265     type order_type [not null]
266     tif time_in_force [not null, default: 'gtc']
267     quantity numeric(28,10) [not null]
268     limit_price numeric(20,8)
269     stop_price numeric(20,8)
270     status order_status [not null, default: 'pending']
271     placed_at timestamptz [not null]
272     placed_via varchar(20) // mobile, web, api}
273
274 Table order_fills {
275     id uuid [pk]
276     order_id uuid [not null, ref: > orders.id]
277     fill_ts timestamptz [not null]
278     quantity numeric(28,10) [not null]
279     price numeric(20,8) [not null]
280     fee numeric(20,8) [default: 0]}
281
282 Table positions {
283     id uuid [pk]
284     account_id uuid [not null, ref: > accounts.id]
285     instrument_id uuid [not null, ref: > instruments.id]
286     status position_status [not null, default: 'open']
287     quantity numeric(28,10) [not null]
288     avg_price numeric(20,8) [not null]
289     opened_at timestamptz [not null]
290     closed_at timestamptz}
291
292 Table portfolios {
293     id uuid [pk]
294     user_id uuid [not null, ref: > users.id]
295     name varchar(80)
296     created_at timestamptz [not null]
297     type varchar(20) // user, smart}
298
299 Table portfolio_positions {
300     id uuid [pk]
301     portfolio_id uuid [not null, ref: > portfolios.id]
302     position_id uuid [not null, ref: > positions.id]
303     weight numeric(9,6)
304 }
305
306 Table smart_portfolios {
307     id uuid [pk]
308     name varchar(120) [not null]
309     description text
310     base_currency varchar(3) [not null, ref: > currencies.code]

```

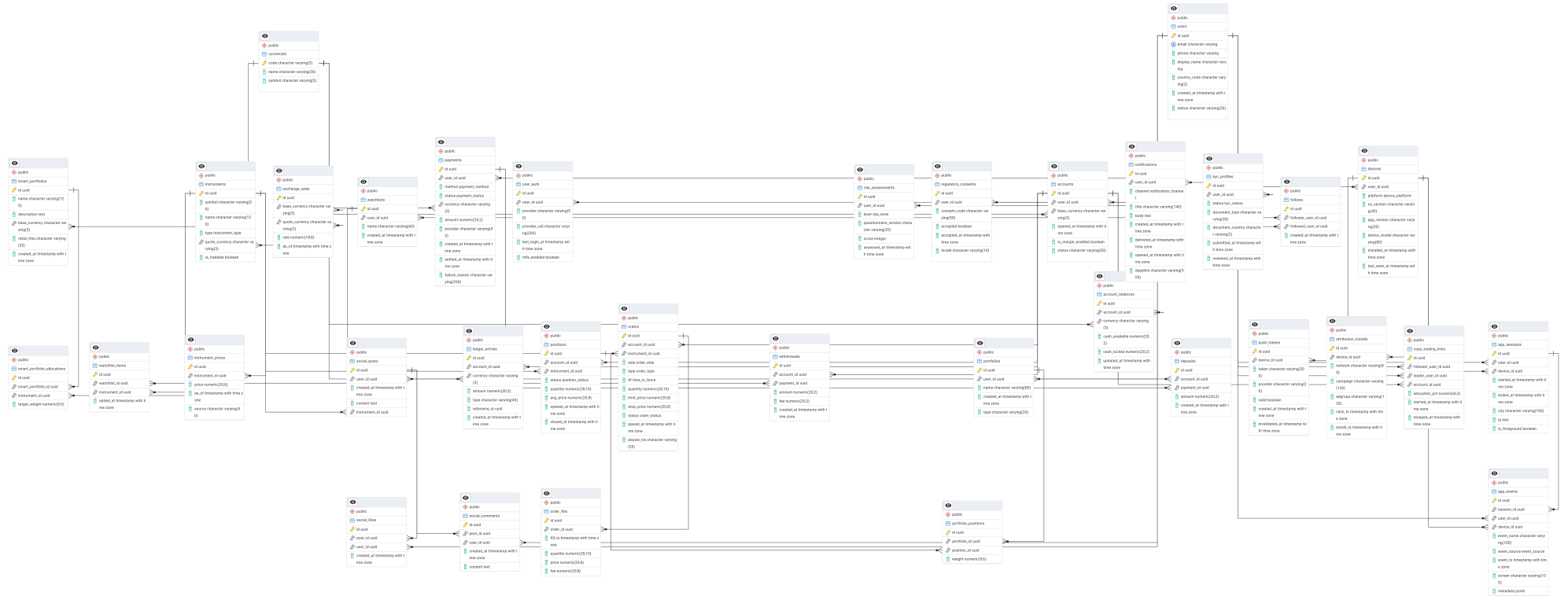
```

311     rebal_freq varchar(20) // monthly, quarterly
312     created_at timestamptz [not null]}
313
314 Table smart_portfolio_allocations {
315     id uuid [pk]
316     smart_portfolio_id uuid [not null, ref: > smart_portfolios.id]
317     instrument_id uuid [not null, ref: > instruments.id]
318     target_weight numeric(9,6) [not null]}
319
320 Table copy_trading_links {
321     id uuid [pk]
322     follower_user_id uuid [not null, ref: > users.id]
323     leader_user_id uuid [not null, ref: > users.id]
324     account_id uuid [not null, ref: > accounts.id]
325     allocation_pct numeric(6,3) [not null]
326     started_at timestamptz [not null]
327     stopped_at timestamptz}
328
329 Table watchlists {
330     id uuid [pk]
331     user_id uuid [not null, ref: > users.id]
332     name varchar(60)
333     created_at timestamptz [not null]}
334
335 Table watchlist_items {
336     id uuid [pk]
337     watchlist_id uuid [not null, ref: > watchlists.id]
338     instrument_id uuid [not null, ref: > instruments.id]
339     added_at timestamptz [not null]}
340
341 Table social_posts {
342     id uuid [pk]
343     user_id uuid [not null, ref: > users.id]
344     created_at timestamptz [not null]
345     content text
346     instrument_id uuid [ref: > instruments.id]}
347
348 Table social_comments {
349     id uuid [pk]
350     post_id uuid [not null, ref: > social_posts.id]
351     user_id uuid [not null, ref: > users.id]
352     created_at timestamptz [not null]
353     content text}
354
355 Table social_likes {
356     id uuid [pk]
357     post_id uuid [not null, ref: > social_posts.id]

```

```
358     user_id uuid [not null, ref: > users.id]
359     created_at timestamptz [not null]}
360
361 Table follows {
362     id uuid [pk]
363     follower_user_id uuid [not null, ref: > users.id]
364     followed_user_id uuid [not null, ref: > users.id]
365     created_at timestamptz [not null] }
```

Figura 2.1: Gráfico 1. Modelo de Datos de eToro ERP



Fuente: Elaboración propia.

Capítulo 3

Business Model Canvas de eToro

Cuadro 3.1: Matriz: Elemento vs. Detalle para eToro

Elemento	Detalle para eToro
Segmentos de Clientes	Inversores minoristas, desde principiantes hasta avanzados.
Propuesta de Valor	Plataforma multi-activo y social trading con interfaz intuitiva y comunidad global.
Canales	Aplicación móvil, plataforma web, soporte, redes sociales y marketing digital.
Relación con Clientes	Comunidad integrada, soporte directo y programas de recompensa.
Fuentes de Ingresos	Spreads, tarifas, comisiones y conversiones de divisas.
Recursos Clave	Plataforma tecnológica, red de usuarios y licencias regulatorias.
Actividades Clave	Desarrollo tecnológico, cumplimiento normativo y marketing estratégico.
Socios Clave	Proveedores de liquidez, instituciones financieras y organismos reguladores.
Estructura de Costos	Infraestructura tecnológica, soporte operativo, pagos y marketing.

Nota: Este esquema coincide con descripciones públicas de su modelo de negocio (por ejemplo, plataformas tipo *CanvasBusinessModel* analizan estos mismos componentes para eToro).

3.1. Canal principal de eToro

Aunque eToro emplea múltiples canales para interactuar con los clientes (aplicación móvil, plataforma web, marketing digital, contenido educativo y redes sociales), el canal principal de eToro es claramente su **plataforma digital** (aplicación + plataforma web) como medio de entrega de sus servicios de inversión y *trading*. (Referencia 30)

Razonamientos:

- Todo el servicio de *trading*, *copy trading* y gestión de activos ocurre dentro de su plataforma digital (web o app).
- El cliente principal de eToro llega, inicia sesión, realiza operaciones y utiliza funciones sociales directamente en dicha plataforma.
- Otros canales (marketing, educación, atención al cliente) son complementarios para atraer, retener y apoyar al usuario, pero la plataforma es donde ocurre el “producto”.
- En su propio sitio web promocionan su “app de inversión fácil de usar” y el acceso a más de 7,000 activos como su oferta central.
- En descripciones del funcionamiento de eToro, se menciona que su red de distribución es principalmente digital, alcanzando una audiencia global a través de la plataforma en línea y el marketing estratégico. (Referencia 31)

Capítulo 4

Consultas SQL para la Generación de Features en Modelos de IA

4.1. Planteamiento de Preguntas de Negocio

A continuación se plantean y resuelven cinco preguntas de negocio orientadas a la creación de *features* relevantes para el desarrollo de modelos de inteligencia artificial en el contexto de eToro. Estas consultas se apoyan en datos de comportamiento, trading, riesgo y dinámica social, permitiendo una caracterización robusta del usuario y su interacción con la plataforma.

4.2. ¿Qué tan comprometido está el usuario en los últimos 30 días?

Esta pregunta analiza qué tan comprometido se mantiene el usuario en los últimos 30 días, empleando indicadores como *sessions_30d*, *active_days_30d*, *events_30d* y *push_open_rate_30d*. Estas variables permiten estimar el nivel de *engagement* o retención activa.

Un usuario con mayor número de sesiones, más días activos y una alta tasa de apertura de notificaciones tiende a presentar una baja probabilidad de abandono, mientras que una disminución en estos indicadores es señal temprana de desinterés o riesgo de *churn*. En modelos de IA, estos valores son clave para la predicción de abandono, la estimación del *Customer Lifetime Value* (CLV) y la personalización de campañas de fidelización.

Script SQL:

Listing 4.1: Compromiso del usuario en los últimos 30 días

```
1 WITH time_window AS (  
2   SELECT now()::timestampz AS as_of, (now() - interval '30 days')::  
       timestampz AS since
```

```

3 ),
4 sessions AS (
5     SELECT s.user_id,
6           COUNT(*) AS sessions_30d,
7           COUNT(DISTINCT date_trunc('day', s.started_at)) AS
            active_days_30d
8     FROM app_sessions s, time_window w
9     WHERE s.started_at >= w.since
10    GROUP BY s.user_id
11 ),
12 events AS (
13     SELECT e.user_id,
14           COUNT(*) AS events_30d
15     FROM app_events e, time_window w
16     WHERE e.event_ts >= w.since
17    GROUP BY e.user_id
18 ),
19 pushes AS (
20     SELECT n.user_id,
21           COUNT(*) FILTER (WHERE n.channel = 'push') AS pushes_sent_30d,
22           COUNT(*) FILTER (WHERE n.channel = 'push' AND n.opened_at IS
                NOT NULL) AS pushes_opened_30d
23     FROM notifications n, time_window w
24     WHERE n.created_at >= w.since
25    GROUP BY n.user_id
26 )
27 SELECT u.id AS user_id,
28        COALESCE(s.sessions_30d,0) AS sessions_30d,
29        COALESCE(s.active_days_30d,0) AS active_days_30d,
30        COALESCE(e.events_30d,0) AS events_30d,
31        COALESCE(p.pushes_sent_30d,0) AS pushes_sent_30d,
32        COALESCE(p.pushes_opened_30d,0) AS pushes_opened_30d,
33        CASE WHEN COALESCE(p.pushes_sent_30d,0) = 0 THEN 0.0
34              ELSE p.pushes_opened_30d::numeric / p.pushes_sent_30d::
                numeric END AS push_open_rate_30d
35 FROM users u
36 LEFT JOIN sessions s ON s.user_id = u.id
37 LEFT JOIN events e   ON e.user_id = u.id
38 LEFT JOIN pushes p   ON p.user_id = u.id;

```

4.3. ¿Cómo es el comportamiento y la performance de trading en los últimos 90 días?

Esta consulta busca comprender el comportamiento y desempeño del trading durante los últimos 90 días, extrayendo variables como `trade_count_90d`, `win_rate_90d`, `avg_trade_notional_usd_90d`, `realized_pnl_usd_90d`, `instruments_traded_90d` y `currencies_traded_90d`.

Estas métricas permiten distinguir estilos y rendimientos de traders (activos, pasivos, rentables o con pérdidas) y facilitan la creación de modelos de clasificación de inversores, detección de anomalías y recomendaciones personalizadas.

Script SQL:

Listing 4.2: Performance de trading en los últimos 90 días

```
1 WITH time_frame AS (  
2     SELECT now()::timestampz AS as_of, (now() - interval '90 days')::  
3         timestampz AS since  
4 ),  
5 latest_fx AS (  
6     SELECT c.code AS currency,  
7         COALESCE((SELECT er.rate FROM exchange_rates er  
8             WHERE er.base_currency = c.code AND er.quote_currency  
9                 = 'USD'  
10            ORDER BY er.as_of DESC LIMIT 1), 1.0)::numeric AS  
11         to_usd  
12 FROM currencies c  
13 ),  
14 fills AS (  
15     SELECT o.account_id, a.user_id, o.side, f.quantity, f.price, i.  
16         quote_currency, f.fill_ts, i.symbol  
17 FROM orders o  
18 JOIN order_fills f ON f.order_id = o.id  
19 JOIN accounts a ON a.id = o.account_id  
20 JOIN instruments i ON i.id = o.instrument_id, time_frame w  
21 WHERE f.fill_ts >= w.since  
22 ),  
23 per_user AS (  
24     SELECT f.user_id,  
25         COUNT(*) AS trade_count_90d,  
26         COUNT(*) FILTER (WHERE (CASE WHEN f.side='sell' THEN f.quantity  
27             *f.price ELSE -f.quantity*f.price END) > 0) AS  
28             winning_trades_90d,  
29         AVG(ABS(f.quantity*f.price)*COALESCE(fx.to_usd,1.0)) AS  
30             avg_trade_notional_usd_90d,  
31         COUNT(DISTINCT f.quote_currency) AS currencies_traded_90d  
32 FROM fills f  
33 LEFT JOIN latest_fx fx ON fx.currency = f.quote_currency
```

```

27     GROUP BY f.user_id
28 ),
29 realized AS (
30     SELECT f.user_id,
31           SUM((CASE WHEN f.side='sell' THEN f.quantity*f.price ELSE -f.
32                quantity*f.price END)*COALESCE(fx.to_usd,1.0)) AS
33                realized_pnl_usd_90d,
34           COUNT(DISTINCT f.symbol) AS instruments_traded_90d
35     FROM fills f
36     LEFT JOIN latest_fx fx ON fx.currency = f.quote_currency
37     GROUP BY f.user_id
38 )
39 SELECT u.id AS user_id,
40        COALESCE(p.trade_count_90d,0) AS trade_count_90d,
41        CASE WHEN COALESCE(p.trade_count_90d,0)=0 THEN 0.0 ELSE p.
42            winning_trades_90d::numeric / p.trade_count_90d::numeric END
43        AS win_rate_90d,
44        COALESCE(p.avg_trade_notional_usd_90d,0) AS
45            avg_trade_notional_usd_90d,
46        COALESCE(r.realized_pnl_usd_90d,0) AS realized_pnl_usd_90d,
47        COALESCE(r.instruments_traded_90d,0) AS instruments_traded_90d,
48        COALESCE(p.currencies_traded_90d,0) AS currencies_traded_90d
49 FROM users u
50 LEFT JOIN per_user p ON p.user_id = u.id
51 LEFT JOIN realized r ON r.user_id = u.id;

```

4.4. ¿Qué tan concentrado está el riesgo de las posiciones abiertas del usuario?

Esta consulta evalúa la concentración de riesgo de las posiciones abiertas del usuario utilizando el *Índice de Herfindahl-Hirschman* (`hhi_open_positions`), junto con el número de instrumentos distintos (`distinct_instruments_open`) y el indicador `margin_enabled`.

Un HHI alto indica concentración de riesgo, mientras que un valor bajo refleja diversificación. Es clave para modelos de riesgo, exposición sistémica y estrategias de cobertura.

Script SQL:

Listing 4.3: Concentración de riesgo de posiciones abiertas

```

1 WITH latest_fx AS (
2     SELECT c.code AS currency,
3           COALESCE((SELECT er.rate FROM exchange_rates er
4                    WHERE er.base_currency = c.code AND er.quote_currency
5                        = 'USD'
6                    ORDER BY er.as_of DESC LIMIT 1), 1.0)::numeric AS
7           to_usd

```

```

6   FROM currencies c
7 ),
8 latest_price AS (
9   SELECT DISTINCT ON (ip.instrument_id) ip.instrument_id, ip.price
10  FROM instrument_prices ip
11  ORDER BY ip.instrument_id, ip.as_of DESC
12 ),
13 open_value AS (
14   SELECT a.user_id, p.instrument_id,
15          (COALESCE(lp.price,0)*p.quantity * COALESCE(fx.to_usd,1.0)) AS
16          position_value_usd
17  FROM positions p
18  JOIN accounts a ON a.id = p.account_id
19  LEFT JOIN instruments i ON i.id = p.instrument_id
20  LEFT JOIN latest_price lp ON lp.instrument_id = p.instrument_id
21  LEFT JOIN latest_fx fx ON fx.currency = i.quote_currency
22  WHERE p.status = 'open'
23 ),
24 weights AS (
25   SELECT user_id, instrument_id, position_value_usd,
26          position_value_usd / NULLIF(SUM(position_value_usd) OVER (
27            PARTITION BY user_id),0) AS w
28  FROM open_value
29 )
30 SELECT u.id AS user_id,
31        COALESCE(SUM(w.w*w.w) FILTER (WHERE w.user_id=u.id), 0) AS
32        hhi_open_positions,
33        COUNT(DISTINCT w.instrument_id) FILTER (WHERE w.user_id=u.id) AS
34        distinct_instruments_open,
35        EXISTS (SELECT 1 FROM accounts a WHERE a.user_id=u.id AND a.
36                is_margin_enabled) AS margin_enabled
37 FROM users u
38 LEFT JOIN weights w ON w.user_id = u.id
39 GROUP BY u.id;

```

4.5. Concentración de Riesgo

Evalúa la diversificación del portafolio mediante el índice HHI (*Herfindahl-Hirschman*).

4.6. Influencia Social y Copy Trading

Identifica líderes sociales usando métricas de seguidores, copiers, likes y comentarios.

Capítulo 5

Referencias Bibliográficas

Bibliografía

- [1] Wikipedia. *EToro*. Recuperado de: <https://en.wikipedia.org/wiki/EToro>
- [2] Business Model Zoo. *eToro Business Model Overview*. Recuperado de: <https://www.businessmodelzoo.com/exemplars/etoro/>
- [3] Routine Wealth. *How Does eToro Make Money?*. Recuperado de: <https://www.routinewealth.com/finance/how-does-etoro-make-money/>
- [4] eToro. *Trading Platforms*. Recuperado de: <https://www.etoro.com/es/trading/platforms/>
- [5] Product Mint. *eToro Business Model — How Does eToro Make Money?*. Recuperado de: <https://productmint.com/etoro-business-model-how-does-etoro-make-money/>
- [6] ElectroIQ. *eToro Statistics*. Recuperado de: <https://electroiq.com/stats/etoro-statistics/>
- [7] eToro. *Sitio Oficial*. Recuperado de: <https://www.etoro.com/es/>