

# Un robo-trader simple.

Construcción y evaluación de su desempeño.

Dr. Martín Lozano <https://mlozanoqf.github.io/>

05 de enero de 2026, 06:54 p.m.

	Fundamental	Intermedio	Especializado
Finanzas	✗	✓	✗
Estadística	✗	✓	✗
R	✗	✓	✗

## **1 Introducción.**

- Este documento ilustra la construcción, implementación y evaluación de un robo-trader sencillo para AAPL.
- La sencillez radica en que la señal de trading se basa únicamente en cruces del precio de cierre con un promedio móvil simple de 3 días SMA3, generando órdenes buy/sell.
- El desempeño se compara contra benchmarks y contra estrategias alternativas.
- Se reportan ganancia simple, número de operaciones y se documenta el proceso mediante gráficos y bitácoras de operaciones.

## 2 Paquetes.

```
1 library(tidyquant)
2 library(dplyr)
3 library(lubridate)
4 library(ggplot2)
5 library(tidyr)
6 library(purrr)
7 library(knitr)
```

### 3 Inicialización 1.

Funciones que vamos a usar en todo el documento.

```
1 raw_prices <- tq_get("AAPL", from = "2024-11-01", to = "2025-12-31") |>
2   arrange(date)
3
4 prices_ma3 <- raw_prices |>
5   tq_mutate(select = close, mutate_fun = SMA, n = 3, col_rename = "ma3")
6
7 start_date <- prices_ma3 |>
8   filter(date >= as.Date("2025-01-02"), !is.na(ma3)) |>
9   slice_head(n = 1) |>
10  pull(date)
11
12 final_date <- prices_ma3 |>
13   filter(date <= as.Date("2025-12-19")) |>
14   slice_tail(n = 1) |>
15   pull(date)
16
17 rt_data <- prices_ma3 |>
18   filter(date >= start_date - days(1), date <= final_date) |>
19   mutate(sign_diff = sign(close - ma3),
20         lag_sign = lag(sign_diff),
21         cross = !is.na(sign_diff) & !is.na(lag_sign) & sign_diff != lag_sign,
22         signal = case_when(cross & sign_diff > lag_sign ~ "buy",
23                           cross & sign_diff < lag_sign ~ "sell", TRUE ~ "hold"))
24
25 signals_rt <- rt_data |>
26   filter(signal %in% c("buy", "sell"), date >= start_date) |>
27   transmute(date, signal, price = close) |>
28   arrange(date)
29
30 signals_eval <- bind_rows(signals_rt,
31   tibble(date = final_date, signal = "final_close",
32         price = rt_data$close[rt_data$date == final_date][1])) |>
33   arrange(date)
34
35 signal_outcomes <- signals_eval |>
36   mutate(next_price = lead(price), next_signal = lead(signal)) |>
37   filter(signal %in% c("buy", "sell")) |>
38   mutate(trade_pnl = if_else(signal == "buy", next_price - price, price - next_price),
39         correct = trade_pnl > 0)
40
41 signals_omniscient <- signal_outcomes |> filter(correct) |>
42   select(date, signal, price)
43 signals_dumb <- signal_outcomes |> filter(!correct) |>
44   select(date, signal, price)
```

## 4 Inicialización 2.

Funciones que vamos a usar en todo el documento.

```
1 # Helpers
2 simulate_pnl <- function(signals_df, final_price) {
3   pnl <- 0; position <- "flat"; entry <- NA_real_
4   for (i in seq_len(nrow(signals_df))) {
5     sig <- signals_df$signal[i]; px <- signals_df$price[i]
6     if (position == "flat") {
7       position <- if (sig == "buy") "long" else "short"; entry <- px
8     } else if (sig == "buy" && position == "short") {
9       pnl <- pnl + (entry - px); position <- "long"; entry <- px
10    } else if (sig == "sell" && position == "long") {
11      pnl <- pnl + (px - entry); position <- "short"; entry <- px
12    } else if (sig == position) {
13      pnl <- pnl + if (position == "long") (px - entry) else (entry - px)
14      entry <- px
15    }
16  }
17  if (!is.na(final_price) && position != "flat") {
18    pnl <- pnl + if (position == "long") (final_price - entry) else (entry - final_price)
19  }
20
21 make_bitacora <- function(signals_df, final_date, final_price) {
22   signals_eval <- bind_rows(signals_df |> arrange(date),
23                             tibble(date = final_date, signal = "final_close",
24                                   price = final_price)) |>
25   arrange(date)
26   position <- "flat"; entry <- NA_real_; pnl_acum <- 0; logs <- list()
27   for (i in seq_len(nrow(signals_eval))) {
28     sig <- signals_eval$signal[i]; px <- signals_eval$price[i]
29     if (position == "flat") {
30       if (sig %in% c("buy", "sell")) {
31         flujo <- if (sig == "buy") -px else px
32         logs[[length(logs) + 1]] <- tibble(date = signals_eval$date[i], 'acción' = sig,
33                                             price = px, flujo = flujo,
34                                             PnL = 0, PnL_acum = pnl_acum)
35         position <- if (sig == "buy") "long" else "short"; entry <- px
36       }
37     } else if (sig %in% c("buy", "sell", "final_close")) {
38       pnl <- if (position == "long") px - entry else entry - px
39       flujo <- if (sig == "buy") -px else if (sig == "sell") px else 0
40       pnl_acum <- pnl_acum + pnl
41       logs[[length(logs) + 1]] <- tibble(date = signals_eval$date[i], 'acción' = sig,
42                                             price = px, flujo = flujo,
43                                             PnL = pnl, PnL_acum = pnl_acum)
44     if (sig == "final_close") { position <- "flat"; entry <- NA_real_ }
45     else { position <- if (sig == "buy") "long" else "short"; entry <- px }
46   }
47 }
48 bind_rows(logs)
49 }
50
51 show_head_tail <- \df, n = 5 bind_rows(slice_head(df, n = n), slice_tail(df, n = n))
```

## 5 Serie de tiempo.

El escenario en el que evaluaremos al robo-trader.

```
1 last15_2024 <- prices_ma3 |>
2   filter(year(date) == 2024) |>
3   arrange(date) |>
4   tail(15)
5
6 start_date_plot <- min(last15_2024$date)
7 end_date <- as.Date("2025-12-19")
8
9 aapl_window <- prices_ma3 |>
10  filter(date >= start_date_plot, date <= end_date)
11
12 ggplot(aapl_window, aes(x = date, y = close)) +
13   geom_line(color = "blue", linewidth = 0.9) +
14   geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +
15   labs(title = "Apple (AAPL) – Precios de cierre.",
16        subtitle = paste("Desde", start_date_plot, "hasta", end_date),
17        x = "Fecha", y = "Precio de cierre (USD)") +
18   theme_minimal(base_size = 12)
```



## 6 Benchmarks.

Hay dos benchmarks naturales que surgen. El primero es buy and hold, que sería una estrategia que no requiere el seguimiento de un criterio inteligente. El segundo es uno *tramposo* en donde vemos el futuro, vendemos en corto, realizamos la ganancia en el precio más bajo, compramos y vendemos al final, lo que daría una ganancia alta, posiblemente una ganancia máxima.

```
1 first_2025 <- prices_ma3 |> filter(date >= as.Date("2025-01-01")) |>
2   slice_head(n = 1)
3 buy_hold_buy_date <- first_2025$date
4 buy_hold_buy_px <- first_2025$close
5
6 sell_12dec <- prices_ma3 |> filter(date <= as.Date("2025-12-12")) |>
7   slice_tail(n = 1)
8 buy_hold_sell_date <- sell_12dec$date
9 buy_hold_sell_px <- sell_12dec$close
10
11 min_row <- prices_ma3 |>
12   filter(date >= as.Date("2025-01-01"), date <= as.Date("2025-12-19")) |>
13   slice_min(close, n = 1, with_ties = FALSE)
14 min_date <- min_row$date
15 min_px <- min_row$close
16
17 final_row <- prices_ma3 |> filter(date <= as.Date("2025-12-19")) |>
18   slice_tail(n = 1)
19 final_date <- final_row$date
20 final_px <- final_row$close
21
22 buy_hold_gain <- buy_hold_sell_px - buy_hold_buy_px
23 trampa_gain <- (buy_hold_buy_px - min_px) + (final_px - min_px)
24
25 strategies <- tibble(estrategia = c("Buy & hold", "Trampa"),
26   ganancia_simple = c(buy_hold_gain, trampa_gain))
27
28 strategies
```

```
## # A tibble: 2 x 2
##   estrategia    ganancia_simple
##   <chr>          <dbl>
## 1 Buy & hold     34.4
## 2 Trampa        173.
```

## 7 La trampa.

Este es el detalle de la estrategia tramposa.

```
1 P0    <- buy_hold_buy_px
2 Pmin <- min_px
3 Pend <- final_px
4
5 tramo_corto <- P0 - Pmin
6 tramo_largo <- Pend - Pmin
7 gan_trampa  <- tramo_corto + tramo_largo
8
9 tibble(P0, Pmin, Pend,
10      tramo_corto = tramo_corto,
11      tramo_largo = tramo_largo,
12      ganancia_trampa = gan_trampa) |>
13  mutate(across(everything(), ~ round(.x, 2))) |>
14  kable()
```

P0	Pmin	Pend	tramo_corto	tramo_largo	ganancia_trampa
243.85	172.42	273.67	71.43	101.25	172.68

Precios relevantes:  $P_0 = 243.85$  (cierre inicial),  $P_{min} = 172.42$  (mínimo anual),  $P_{end} = 273.67$  (cierre final).

Descripción de la estrategia: Abres corto al primer cierre de 2025, vendiendo a 243.85 y comprando a 172.42, el tramo corto gana  $243.85 - 172.42 = 71.43$ . Compras inmediatamente en el mínimo de 172.42 y vendes al cierre final de 273.67, el tramo largo gana  $273.67 - 172.42 = 101.25$ .

La ganancia total de la estrategia trampa es:  $71.43 + 101.25 = 172.68$ .

## 8 ¿Cómo decide el robo-trader?

El robo-trader usa un promedio móvil simple SMA de 3 días  $\text{SMA3}_t$  sobre el precio de cierre  $P_t$ :  $\text{SMA3}_t = \frac{1}{3}(P_t + P_{t-1} + P_{t-2})$ , definido cuando existen los tres cierres necesarios; por ello, los primeros 2 días resultan en NA.

Definimos  $d_t = P_t - \text{SMA3}_t$ .

La  $\text{SMA3}_t$  funciona como un nivel de referencia de corto plazo. El signo de  $d_t$  indica en qué lado está el precio respecto a ese nivel:

- Si  $d_t > 0$ , el precio está por encima de su promedio reciente. Se puede interpretar como un sesgo alcista de corto plazo.
- Si  $d_t < 0$ , el precio está por debajo de su promedio reciente. Se puede interpretar como un sesgo bajista de corto plazo.

El robot opera únicamente cuando detecta un *cambio de régimen*: es decir, cuando el precio pasa de estar consistentemente por debajo a estar por encima (o viceversa). Por eso se usa el cruce (cambio de signo) y no solo la condición “estar arriba/abajo”.

- Señal de compra en  $t$ : ocurre si  $d_{t-1} < 0$  y  $d_t > 0$ . El precio cruza desde abajo hacia arriba la SMA3. Representa un cambio de sesgo a alcista.
- Señal de venta en  $t$ : ocurre si  $d_{t-1} > 0$  y  $d_t < 0$ . El precio cruza desde arriba hacia abajo la SMA3. Representa un cambio de sesgo a bajista.
- Si no hay cruce (incluyendo el caso  $d_t = 0$ ), no hay señal y se mantiene “hold”.

## 9 Supuestos operativos del robo-trader.

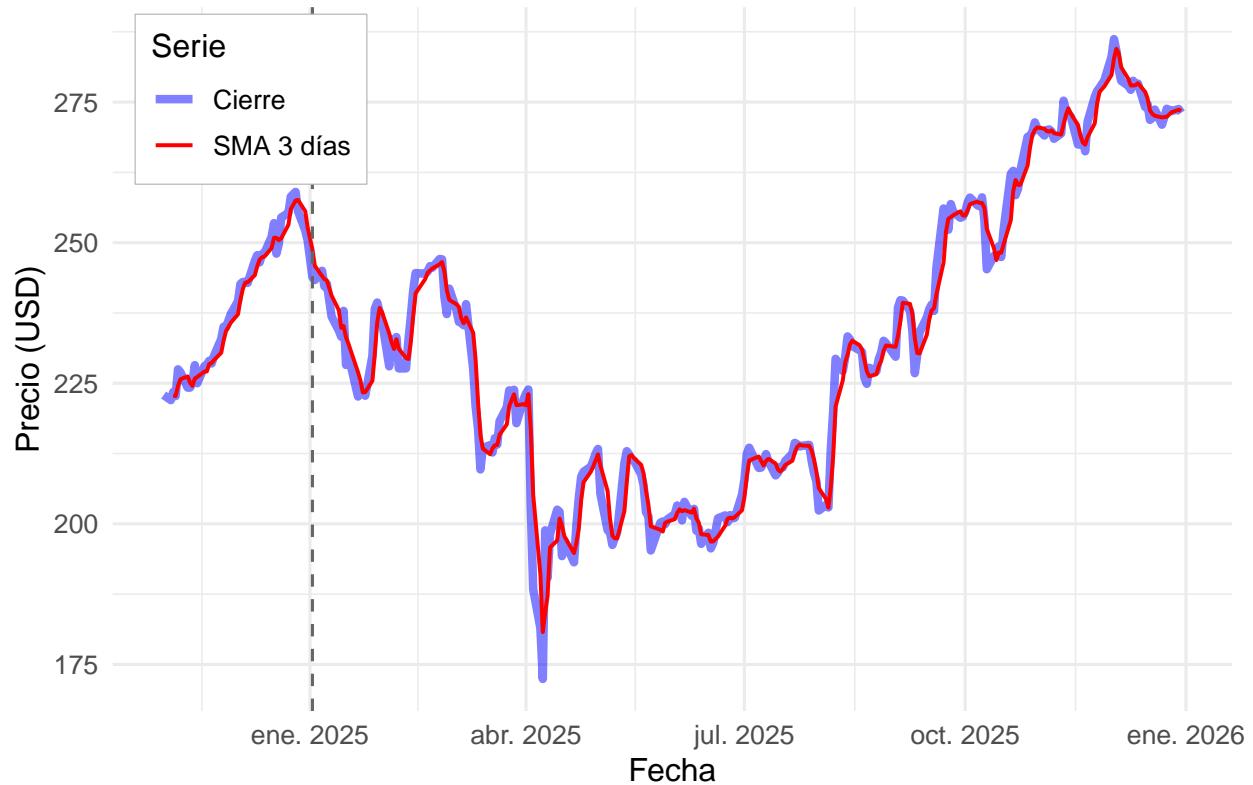
- Se evalúa la señal una vez por día.
- Las operaciones se asumen ejecutadas al precio de cierre del día de la señal.
- El robot siempre mantiene una posición. Un **sell** cierra un largo y abre un corto; un **buy** cierra un corto (buy-to-cover) y abre un largo. No existe estado “cash” entre señales.
- Se ignoran comisiones, bid-ask spread, slippage y costos de préstamo/financiamiento.
- No hay gestión de riesgo, los cambios de posición ocurren únicamente por cruces de SMA3.
- Al término del periodo se fuerza un **final\_close** para cerrar cualquier posición abierta al último precio de cierre.

## 10 Visualización del promedio móvil.

El promedio móvil parece adelantarse a los precios de cierre.

```
1 ggplot(prices_ma3, aes(date)) +
2   geom_line(aes(y = close, color = "Cierre"), linewidth = 1.5, alpha = 0.5) +
3   geom_line(aes(y = ma3, color = "SMA 3 días"), linewidth = 0.7, na.rm = TRUE) +
4   geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +
5   scale_color_manual(name = "Serie",
6     values = c("Cierre" = "blue", "SMA 3 días" = "red")) +
7   labs(title = "AAPL: Cierre vs. Promedio Móvil 3 días.",
8     x = "Fecha", y = "Precio (USD)") +
9   theme_minimal(base_size = 12) +
10  theme(legend.position = c(0.02, 0.99), legend.justification = c(0, 1),
11    legend.background = element_rect(fill = "white", color = "gray70", linewidth = 0.2),
12    legend.key = element_rect(fill = "white", color = NA))
```

AAPL: Cierre vs. Promedio Móvil 3 días.

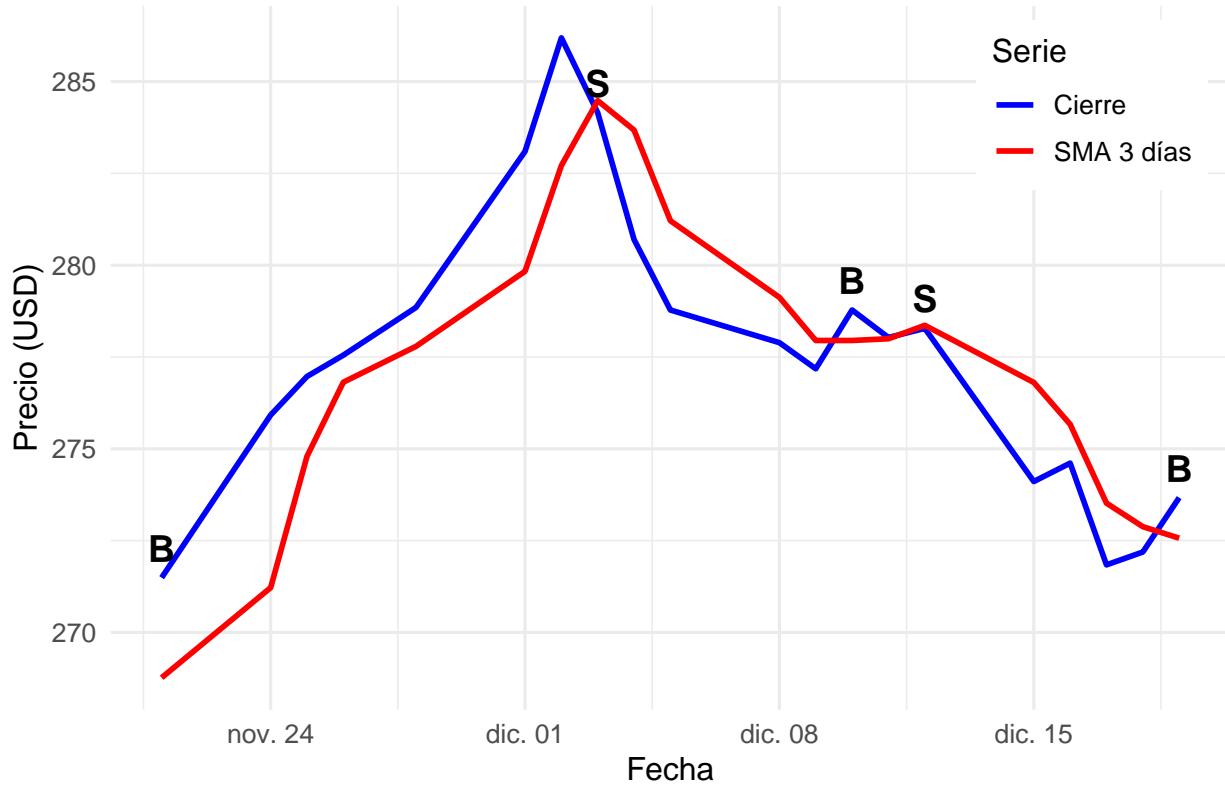


## 11 Promedio móvil, zoom.

Un acercamiento para visualizar mejor el papel del promedio móvil.

```
1 # Ventana del zoom
2 start_zoom <- as.Date("2025-11-21")
3 end_zoom   <- as.Date("2025-12-19")
4
5 # Señales buy/sell por cruce precio vs SMA3
6 signals_short <- prices_ma3 |>
7   mutate(sign_diff = sign(close - ma3), lag_sign  = lag(sign_diff),
8         cross = !is.na(sign_diff) & !is.na(lag_sign) & sign_diff != lag_sign,
9         signal = case_when(cross & sign_diff > lag_sign ~ "buy",
10                           cross & sign_diff < lag_sign ~ "sell", TRUE ~ NA_character_)) |>
11   filter(date >= start_zoom, date <= end_zoom, !is.na(signal)) |>
12   transmute(date, signal, price = close)
13
14 # Serie y etiquetas B/S
15 aapl_ma_short <- prices_ma3 |>
16   filter(date >= start_zoom, date <= end_zoom)
17
18 ggplot(aapl_ma_short, aes(date)) +
19   geom_line(aes(y = close, color = "Cierre"), linewidth = 1) +
20   geom_line(aes(y = ma3, color = "SMA 3 días"), linewidth = 1, na.rm = TRUE) +
21   geom_text(data = signals_short,
22             aes(y = price, label = if_else(signal == "buy", "B", "S")),
23             vjust = -0.6, fontface = "bold", color = "black", size = 5) +
24   scale_color_manual(name = "Serie",
25                       values = c("Cierre" = "blue", "SMA 3 días" = "red")) +
26   labs(title = "AAPL: Cierre vs. SMA 3 días. Cinco señales.",
27         x = "Fecha", y = "Precio (USD)") +
28   theme_minimal(base_size = 12) +
29   theme(legend.position = c(0.98, 0.98), legend.justification = c(1, 1),
30         legend.background = element_rect(fill = "white", color = NA),
31         legend.key = element_rect(fill = "white", color = NA))
```

### AAPL: Cierre vs. SMA 3 días. Cinco señales.



- **21-nov:** buy a 271.49. Se abre posición larga. PnL realizado en este punto: 0. PnL acumulado previo: 24.89.
- **03-dic:** sell a 284.15. Se cierra el largo y, en esquema always-in, se abre corto. PnL del tramo largo:  $284.15 - 271.49 = 12.66$ . PnL acumulado:  $24.89 + 12.66 = 37.55$ .
- **10-dic:** buy a 278.78. Se cierra el corto abierto el 03-dic y se abre largo. PnL del tramo corto:  $284.15 - 278.78 = 5.37$ . PnL acumulado:  $37.55 + 5.37 = 42.92$ .
- **12-dic:** sell a 278.28. Se cierra el largo abierto el 10-dic y se abre corto. PnL del tramo largo:  $278.28 - 278.78 = -0.50$ . PnL acumulado:  $42.92 - 0.50 = 42.42$ .
- **19-dic:** buy a 273.67. Se cierra el corto abierto el 12-dic y se abre largo. PnL del tramo corto:  $278.28 - 273.67 = 4.61$ . PnL acumulado:  $42.42 + 4.61 = 47.03$ .

## 12 Descripción de 6 estrategias.

- **Buy & hold:** compras AAPL al primer cierre de 2025 y vendes el 12/dic/2025; PnL = diferencia entre esos cierres, sin trades intermedios.
- **Trampa:** abres corto al primer cierre de 2025, cubres y giras a largo en el mínimo anual (hasta 19/dic/2025) y cierras largo al final; capturas la caída y luego el rebote.
- **Robo-trader:** opera cada cruce cierre vs SMA3; cambia de lado en cada señal y liquida al cierre final; PnL suma cada tramo entre señales consecutivas.
- **Aleatoria:** 83 señales buy/sell alternadas en fechas al azar; PnL con la misma regla de cerrar y reabrir en cada señal, cierre forzado al final.
- **Robo-omnisciente (benchmark ex-post):** genera señales con la regla de cruces, pero ejecuta únicamente aquellas cuyo tramo posterior resulta con PnL positivo. Es un *oráculo* no implementable en tiempo real porque filtra operaciones usando información futura. Si aparecen señales del mismo tipo de forma consecutiva (buy-buy o sell-sell), se fuerza un cierre y reapertura al nuevo precio para que el PnL contabilice el tramo intermedio entre ambas ejecuciones.
- **Robo-dumb (benchmark ex-post):** usa la misma regla de cruces, pero ejecuta únicamente aquellas entradas cuyo tramo posterior resulta con PnL negativo. También es no implementable en tiempo real por depender de información futura. Ante señales consecutivas del mismo tipo, aplica el mismo mecanismo de cierre y reapertura para que el PnL refleje el tramo entre precios y capture correctamente los tramos desfavorables.

*Al filtrar señales ex-post (quedándonos solo con las ganadoras/perdedoras), es posible que en la secuencia filtrada aparezcan dos señales consecutivas del mismo tipo (buy-buy o sell-sell), aunque en la secuencia original de cruces eso no ocurra. En esos casos, se aplica la convención de ‘cierre y reapertura’ al nuevo precio para contabilizar el tramo intermedio.*

## 13 Desempeño de las 6 estrategias.

```
1 set.seed(123)
2 prices <- prices_ma3
3
4 P0 <- prices |> filter(date >= as.Date("2025-01-01")) |> slice_head(n = 1) |> pull(close)
5 bh_sell_px <- prices |> filter(date <= as.Date("2025-12-12")) |> slice_tail(n = 1) |>
6   pull(close)
7 Pmin <- prices |> filter(date >= as.Date("2025-01-01"), date <= final_date) |>
8   slice_min(close, n = 1, with_ties = FALSE) |> pull(close)
9 Pend <- prices |> filter(date == final_date) |> pull(close)
10
11 buy_hold_gain <- bh_sell_px - P0
12 trampa_gain <- (P0 - Pmin) + (Pend - Pmin)
13
14 trading_days <- prices |>
15   filter(date >= start_date, date <= final_date) |>
16   pull(date)
17
18 random_dates <- sort(sample(trading_days, size = 83, replace = FALSE))
19 start_side <- sample(c("buy", "sell"), 1)
20 random_signals_vec <- if (start_side == "buy") rep(c("buy", "sell"), length.out = 83) else
21   rep(c("sell", "buy"), length.out = 83)
22
23 signals_rand <- tibble(date = random_dates, signal = random_signals_vec) |>
24   left_join(prices |> select(date, close), by = "date") |>
25   rename(price = close) |>
26   arrange(date)
27
28 # Reuso de signal_outcomes ya calculado arriba
29 robo_trader_gain <- simulate_pnl(signals_rt, final_price = Pend)
30 random_gain <- simulate_pnl(signals_rand, final_price = Pend)
31 robo_omnisciente_gain <- simulate_pnl(signals_omniscient, final_price = Pend)
32 robo_dumb_gain <- simulate_pnl(signals_dumb, final_price = Pend)
33
34 rt_trades <- nrow(signals_rt)
35 rand_trades <- nrow(signals_rand)
36 omn_trades <- nrow(signals_omniscient)
37 dumb_trades <- nrow(signals_dumb)
38
39 strategies <- tibble(estrategia = c("Buy & hold", "Trampa", "Robo-trader",
40                           "Aleatoria", "Robo-omnisciente", "Robo-dumb"),
41   trades = c(NA_integer_, NA_integer_, rt_trades, rand_trades, omn_trades, dumb_trades),
42   ganancia_simple = c(buy_hold_gain, trampa_gain, robo_trader_gain, random_gain,
43                       robo_omnisciente_gain, robo_dumb_gain),
44   ganancia_por_trade = ganancia_simple / trades)
45
46 strategies_tbl <- strategies |>
47   mutate(ganancia_simple = round(ganancia_simple, 2),
48         ganancia_por_trade = round(ganancia_por_trade, 2))
49
50 strategies_tbl |>
51   kable(digits = c(NA, 0, 2, 2))
```

estrategia	trades	ganancia_simple	ganancia_por_trade
Buy & hold	NA	34.43	NA
Trampa	NA	172.68	NA
Robo-trader	83	47.03	0.57
Aleatoria	83	22.30	0.27
Robo-omnisciente	35	189.00	5.40
Robo-dumb	48	-245.07	-5.11

## 14 Visualización del Robo-trader.

Hay señales correctas e incorrectas.

```
1 signals_plot <- signals_rt |>
2   left_join(signal_outcomes |> select(date, signal, correct),
3             by = c("date", "signal")) |>
4   mutate(correcto = if_else(correct, "Correcta", "Incorrecta"))
5
6 ggplot(rt_data, aes(date, close)) +
7   geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
8   geom_text(data = signals_plot,
9             aes(y = price, label = if_else(signal == "buy", "B", "S"),
10                  color = correcto), fontface = "bold", size = 3) +
11   scale_color_manual(values = c(Correcta = "#2E8B57", Incorrecta = "red"),
12                       name = "Resultado") +
13   geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
14   labs(title = "Robo-trader SMA3: Señales correctas vs incorrectas.",
15        x = "Fecha", y = "Precio cierre (USD)") +
16   theme_minimal(base_size = 12) +
17   theme(legend.position = "bottom")
```

Robo-trader SMA3: Señales correctas vs incorrectas.

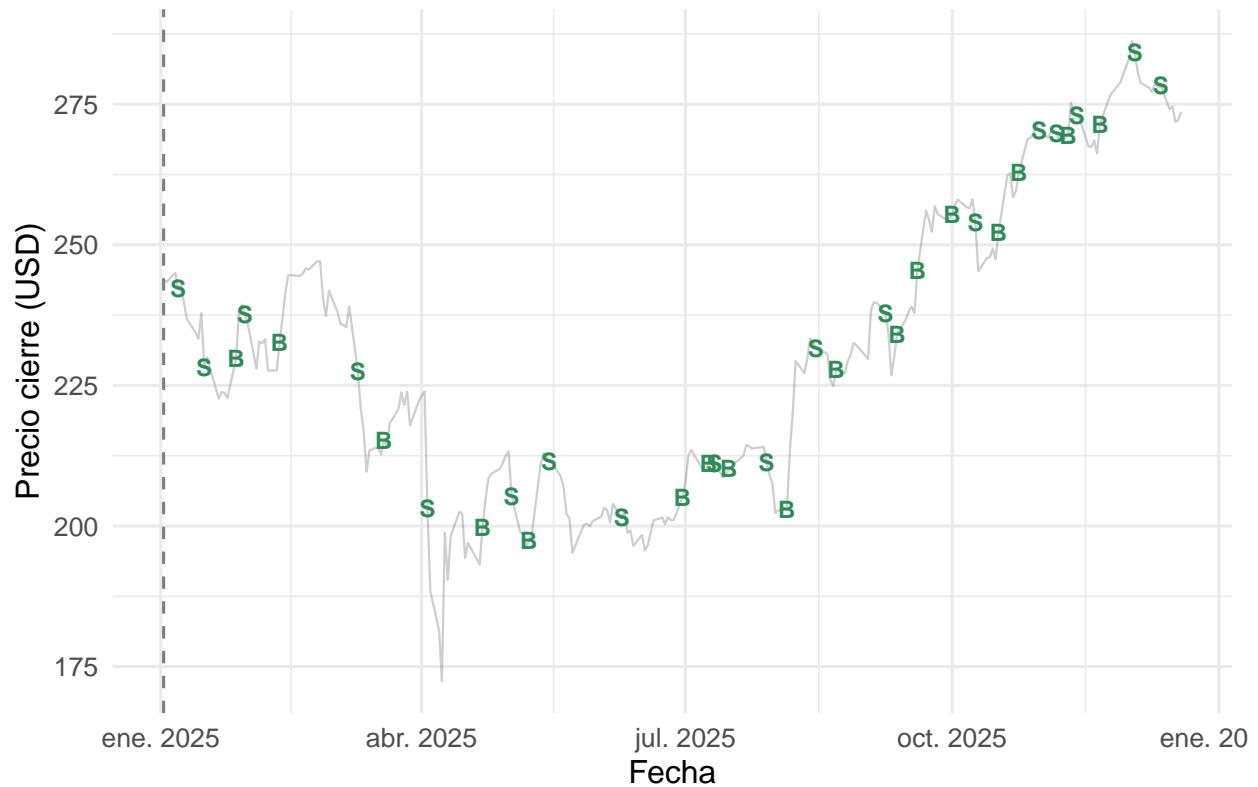


## 15 Visualización del Robo-omnisciente.

Un robot que lo sabe todo y nunca se equivoca.

```
1 signals_plot_omn <- signals_omniscient |>
2   mutate(correcto = "Correcta")
3
4 ggplot(rt_data, aes(date, close)) +
5   geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
6   geom_text(data = signals_plot_omn,
7             aes(y = price, label = if_else(signal == "buy", "B", "S"), color = correcto),
8             fontface = "bold", size = 3) +
9   scale_color_manual(values = c(Correcta = "#2E8B57"), name = "Resultado") +
10  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
11  labs(title = "Robo-omnisciente: Señales correctas.",
12        x = "Fecha", y = "Precio cierre (USD)") +
13  theme_minimal(base_size = 12) +
14  theme(legend.position = "none")
```

Robo-omnisciente: Señales correctas.

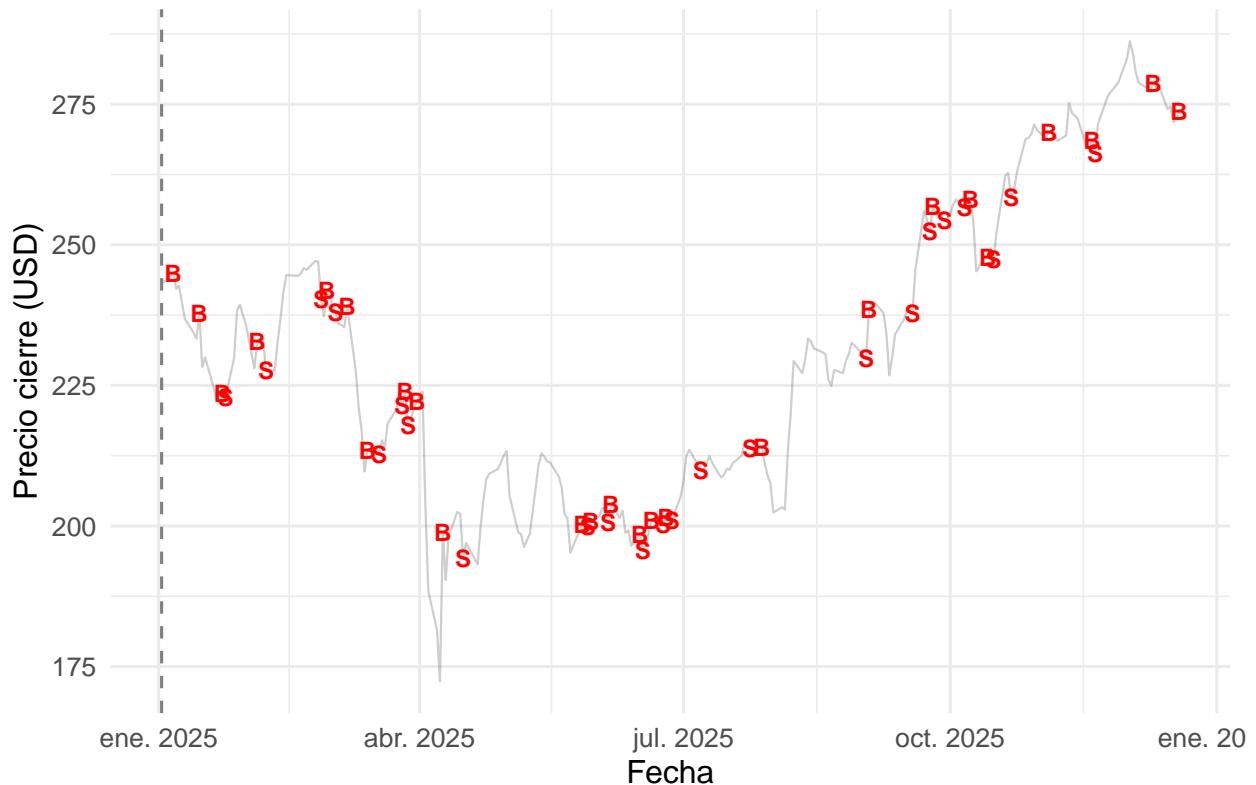


## 16 Visualización del Robo-dumb.

Un robot que lo sabe todo y siempre se equivoca.

```
1 signals_plot_dumb <- signals_dumb |>
2   mutate(correcto = "Incorrecta")
3
4 ggplot(rt_data, aes(date, close)) +
5   geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
6   geom_text(data = signals_plot_dumb,
7             aes(y = price, label = if_else(signal == "buy", "B", "S"), color = correcto),
8             fontface = "bold", size = 3) +
9   scale_color_manual(values = c(Incorrecta = "red"), name = "Resultado") +
10  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
11  labs(title = "Robo-dumb: Señales incorrectas.",
12        x = "Fecha", y = "Precio cierre (USD)") +
13  theme_minimal(base_size = 12) +
14  theme(legend.position = "none")
```

Robo-dumb: Señales incorrectas.



## 17 Bitácora Robo-trader.

El robo-trader se equivoca pero a pesar de eso hace dinero.

```
1 bitacora_num <- make_bitacora(signals_rt, final_date, Pend) |>
2   mutate(price = round(price, 2),
3         flujo = round(flujo, 2),
4         PnL = round(PnL, 2),
5         PnL_acum = round(PnL_acum, 2))
6
7 bitacora_num |>
8   show_head_tail(n = 5) |>
9   kable(digits = c(NA, NA, 2, 2, 2, 2))
```

date	acción	price	flujo	PnL	PnL_acum
2025-01-06	buy	245.00	-245.00	0.00	0.00
2025-01-07	sell	242.21	242.21	-2.79	-2.79
2025-01-15	buy	237.87	-237.87	4.34	1.55
2025-01-16	sell	228.26	228.26	-9.61	-8.06
2025-01-23	buy	223.66	-223.66	4.60	-3.46
2025-12-03	sell	284.15	284.15	12.66	37.55
2025-12-10	buy	278.78	-278.78	5.37	42.92
2025-12-12	sell	278.28	278.28	-0.50	42.42
2025-12-19	buy	273.67	-273.67	4.61	47.03
2025-12-19	final_close	273.67	0.00	0.00	47.03

- **03-dic:** sell a 284.15. Se abre posición corta. PnL realizado en este punto: 0. PnL acumulado previo: 37.55.
- **10-dic:** buy a 278.78. Se cierra el corto abierto el 03-dic y se abre largo. PnL del tramo corto:  $284.15 - 278.78 = 5.37$ . PnL acumulado:  $37.55 + 5.37 = 42.92$ .
- **12-dic:** sell a 278.28. Se cierra el largo abierto el 10-dic y se abre corto. PnL del tramo largo:  $278.28 - 278.78 = -0.50$ . PnL acumulado:  $42.92 - 0.50 = 42.42$ .
- **19-dic:** buy a 273.67. Se cierra el corto abierto el 12-dic y se abre largo. PnL del tramo corto:  $278.28 - 273.67 = 4.61$ . PnL acumulado:  $42.42 + 4.61 = 47.03$ .
- **19-dic:** final\_close a 273.67. Se cierra el largo abierto el 19-dic. PnL del tramo largo:  $273.67 - 273.67 = 0.00$ . PnL acumulado final: 47.03.

## 18 Resumen Robo-trader.

Una matriz de confusión.

```
1 confusion <- signal_outcomes |>
2   mutate(outcome = if_else(correct, "correcta", "incorrecta")) |>
3   count(signal, outcome) |>
4   pivot_wider(names_from = outcome, values_from = n, values_fill = 0)
5
6 confusion
```

```
## # A tibble: 2 x 3
##   signal correcta incorrecta
##   <chr>     <int>      <int>
## 1 buy         17          25
## 2 sell        18          23
```

## 19 Bitácora estrategia aleatoria.

El número de señales aleatorias corresponde al número de señales del robo-trader.

```
1 bitacora_aleatoria_num <- make_bitacora(signals_rand, final_date, Pend) |>
2   mutate(price = round(price, 2),
3         flujo = round(flujo, 2),
4         PnL = round(PnL, 2),
5         PnL_acum = round(PnL_acum, 2))
6
7 bitacora_aleatoria_num |>
8   show_head_tail(n = 5) |>
9   kable(digits = c(NA, NA, 2, 2, 2, 2))
```

date	acción	price	flujo	PnL	PnL_acum
2025-01-07	buy	242.21	-242.21	0.00	0.00
2025-01-10	sell	236.85	236.85	-5.36	-5.36
2025-01-13	buy	234.40	-234.40	2.45	-2.91
2025-01-22	sell	223.83	223.83	-10.57	-13.48
2025-01-23	buy	223.66	-223.66	0.17	-13.31
2025-12-11	sell	278.03	278.03	-0.75	25.28
2025-12-12	buy	278.28	-278.28	-0.25	25.03
2025-12-16	sell	274.61	274.61	-3.67	21.36
2025-12-19	buy	273.67	-273.67	0.94	22.30
2025-12-19	final_close	273.67	0.00	0.00	22.30

## 20 Bitácora Robo-omnisciente.

Equivalente al modelo perfecto.

```

1 bitacora_omnisciente_num <- make_bitacora(signals_omniscient, final_date, Pend) |>
2   mutate(price = round(price, 2), flujo = round(flujo, 2),
3         PnL = round(PnL, 2), PnL_acum = round(PnL_acum, 2))
4
5 bitacora_omnisciente_num |>
6   show_head_tail(n = 5) |>
7   kable(digits = c(NA, NA, 2, 2, 2, 2))

```

date	acción	price	flujo	PnL	PnL_acum
2025-01-07	sell	242.21	242.21	0.00	0.00
2025-01-16	sell	228.26	228.26	13.95	13.95
2025-01-27	buy	229.86	-229.86	-1.60	12.35
2025-01-30	sell	237.59	237.59	7.73	20.08
2025-02-11	buy	232.62	-232.62	4.97	25.05
2025-11-13	sell	272.95	272.95	3.52	164.40
2025-11-21	buy	271.49	-271.49	1.46	165.86
2025-12-03	sell	284.15	284.15	12.66	178.52
2025-12-12	sell	278.28	278.28	5.87	184.39
2025-12-19	final_close	273.67	0.00	4.61	189.00

- **13-nov:** `sell` a 272.95. Se abre posición corta. PnL realizado en este punto: 0. PnL acumulado previo: 164.40.
- **21-nov:** `buy` a 271.49. Se cierra el corto abierto el 13-nov y se abre largo. PnL del tramo corto:  $272.95 - 271.49 = 1.46$ . PnL acumulado:  $164.40 + 1.46 = 165.86$ .
- **03-dic:** `sell` a 284.15. Se cierra el largo abierto el 21-nov y se abre corto. PnL del tramo largo:  $284.15 - 271.49 = 12.66$ . PnL acumulado:  $165.86 + 12.66 = 178.52$ .
- **12-dic:** `sell` a 278.28. Señal repetida; se cierra el corto abierto el 03-dic y se reabre corto. PnL del tramo corto:  $284.15 - 278.28 = 5.87$ . PnL acumulado:  $178.52 + 5.87 = 184.39$ .
- **19-dic:** `final_close` a 273.67. Se cierra el corto abierto el 12-dic. PnL del tramo corto:  $278.28 - 273.67 = 4.61$ . PnL acumulado final: 189.00.

## 21 Bitácora Robo-dumb.

Equivalente al anti-modelo.

```

1 bitacora_dumb_num <- make_bitacora(signals_dumb, final_date, Pend) |>
2   mutate(price = round(price, 2), flujo = round(flujo, 2),
3         PnL = round(PnL, 2), PnL_acum = round(PnL_acum, 2))
4
5 bitacora_dumb_num |>
6   show_head_tail(n = 5) |>
7   kable(digits = c(NA, NA, 2, 2, 2, 2))

```

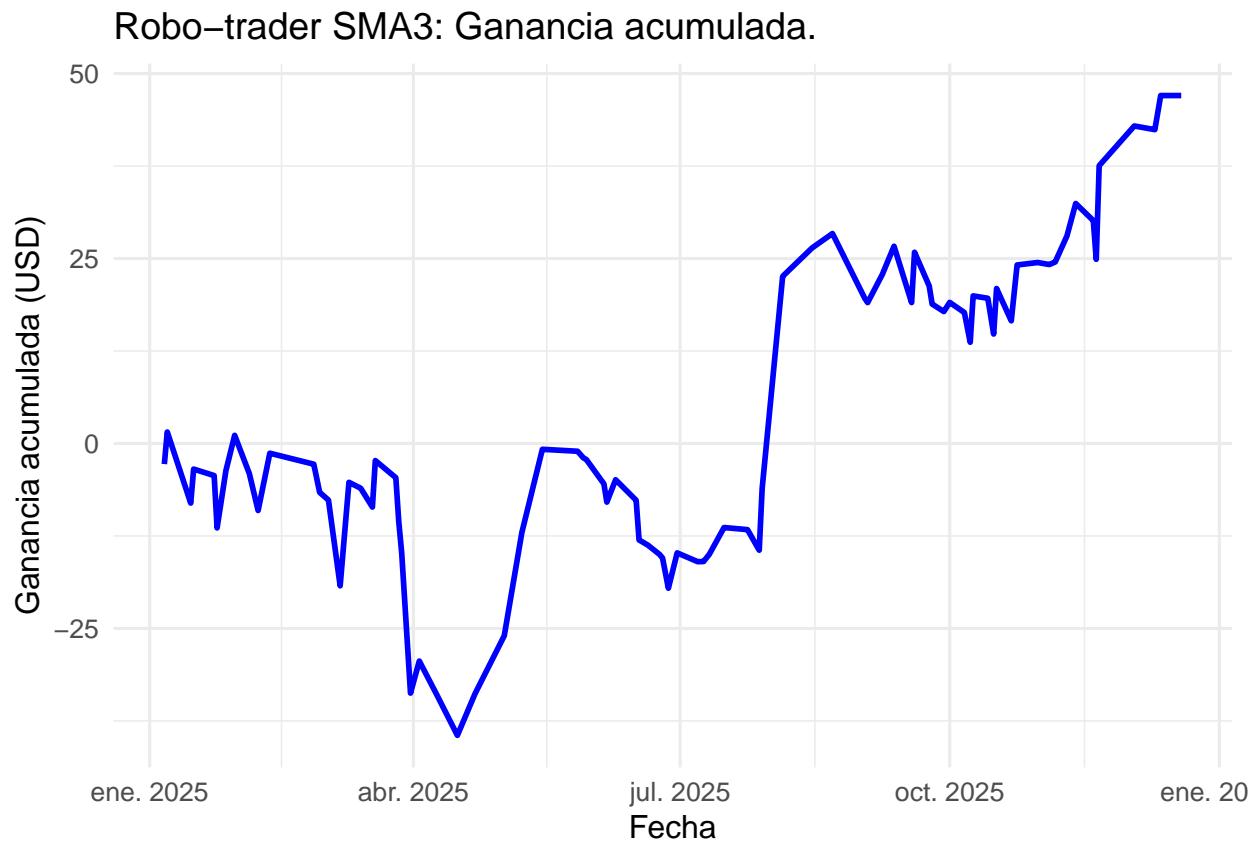
date	acción	price	flujo	PnL	PnL_acum
2025-01-06	buy	245.00	-245.00	0.00	0.00
2025-01-15	buy	237.87	-237.87	-7.13	-7.13
2025-01-23	buy	223.66	-223.66	-14.21	-21.34
2025-01-24	sell	222.78	222.78	-0.88	-22.22
2025-02-04	buy	232.80	-232.80	-10.02	-32.24
2025-11-19	buy	268.56	-268.56	-1.48	-225.12
2025-11-20	sell	266.25	266.25	-2.31	-227.43
2025-12-10	buy	278.78	-278.78	-12.53	-239.96
2025-12-19	buy	273.67	-273.67	-5.11	-245.07
2025-12-19	final_close	273.67	0.00	0.00	-245.07

- **19-nov:** buy a 268.56. Se abre posición larga. PnL realizado en este punto: 0. PnL acumulado previo: -225.12.
- **20-nov:** sell a 266.25. Se cierra el largo abierto el 19-nov y se abre corto. PnL del tramo largo:  $266.25 - 268.56 = -2.31$ . PnL acumulado:  $-225.12 - 2.31 = -227.43$ .
- **10-dic:** buy a 278.78. Se cierra el corto abierto el 20-nov y se abre largo. PnL del tramo corto:  $266.25 - 278.78 = -12.53$ . PnL acumulado:  $-227.43 - 12.53 = -239.96$ .
- **19-dic:** buy a 273.67. Señal repetida; se cierra el largo abierto el 10-dic y se reabre largo. PnL del tramo largo:  $273.67 - 278.78 = -5.11$ . PnL acumulado:  $-239.96 - 5.11 = -245.07$ .
- **19-dic:** final\_close a 273.67. Se cierra el largo abierto el 19-dic. PnL del tramo largo:  $273.67 - 273.67 = 0.00$ . PnL acumulado final: -245.07.

## 22 Ganancias Robo-trader.

Resumen gráfico.

```
1 ggplot(signal_outcomes |> mutate(PnL_acum = cumsum(replace_na(trade_pnl, 0))),  
2   aes(x = date, y = PnL_acum)) +  
3   geom_line(color = "blue", linewidth = 1) +  
4   labs(title = "Robo-trader SMA3: Ganancia acumulada.",  
5     x = "Fecha", y = "Ganancia acumulada (USD)") +  
6   theme_minimal(base_size = 12)
```



## **23 Conclusión.**

- La comparación evidencia cómo una regla mecánica SMA3 se posiciona frente a estrategias pasivas, un benchmark idealizado, ex-posts, y controles aleatorios.
- La bitácora y los gráficos permiten validar la lógica del algoritmo y la trazabilidad del PnL por operación.
- En conjunto, el ejercicio sirve como referencia metodológica para evaluar reglas de trading simplificado en términos de desempeño y frecuencia de operación.