Un robo-trader simple.

Construcción y evaluación de su desempeño.

Dr. Martín Lozano `https://mlozanoqf.github.io/`

20 de diciembre de 2025, 07:06 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 señal de trading se basa en cruces del precio de cierre con un promedio móvil simple de 3 días (MA3), 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)
```

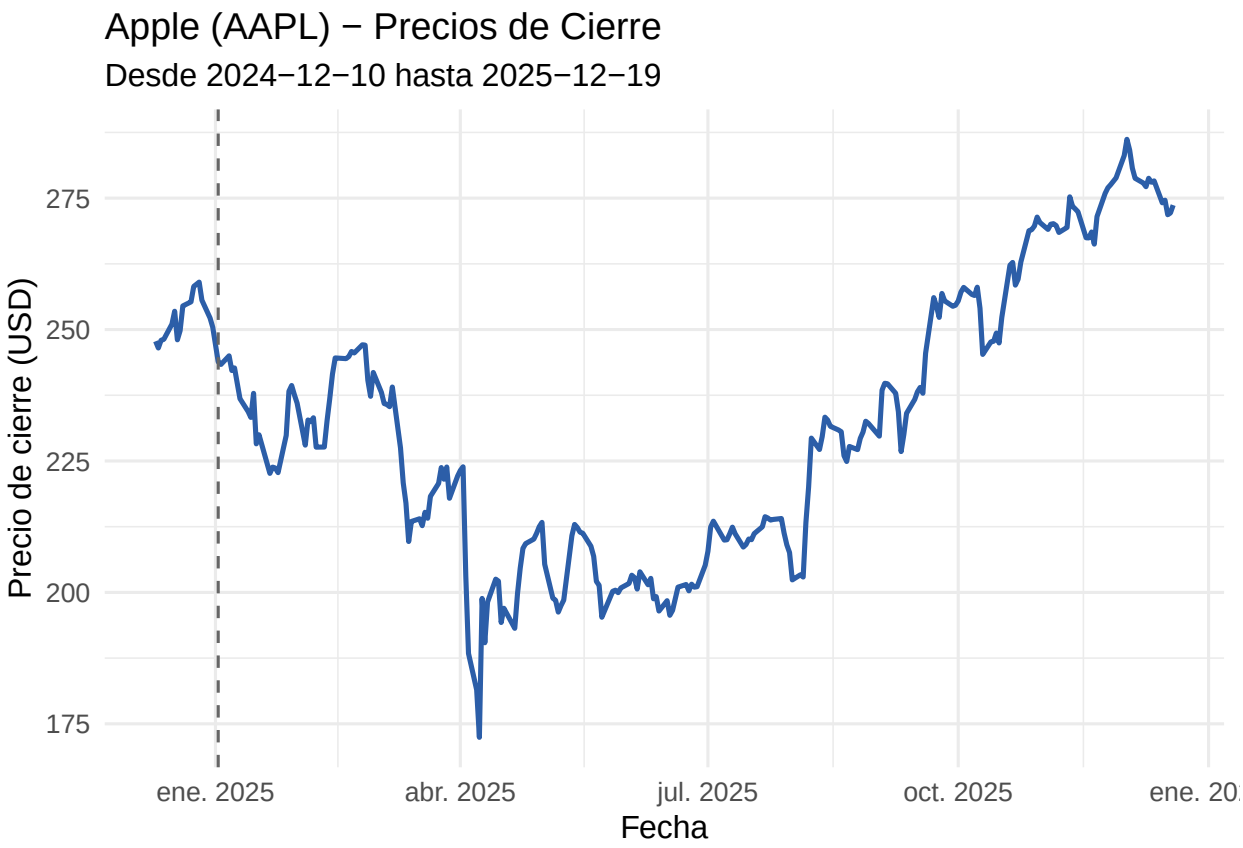
3 Inicialización.

```r
raw_prices <- tq_get("AAPL", from = "2024-11-01", to = "2025-12-31") |>
  arrange(date)

prices_ma3 <- raw_prices |>
  tq_mutate(select = close, mutate_fun = SMA, n = 3, col_rename = "ma3")

start_date <- prices_ma3 |>
  filter(date >= as.Date("2025-01-02"), !is.na(ma3)) |>
  slice_head(n = 1) |>
  pull(date)

final_date <- prices_ma3 |>
  filter(date <= as.Date("2025-12-19")) |>
  slice_tail(n = 1) |>
  pull(date)

rt_data <- prices_ma3 |>
  filter(date >= start_date - days(1), date <= final_date) |>
  mutate(sign_diff = sign(close - ma3),
         lag_sign  = lag(sign_diff),
         cross     = !is.na(sign_diff) & !is.na(lag_sign) & sign_diff != lag_sign,
         signal    = case_when(
           cross & sign_diff > lag_sign ~ "buy",
           cross & sign_diff < lag_sign ~ "sell",
           TRUE ~ "hold"))

signals_rt <- rt_data |>
  filter(signal %in% c("buy", "sell"), date >= start_date) |>
  transmute(date, signal, price = close) |>
  arrange(date)

signals_eval <- bind_rows(signals_rt,
    tibble(date = final_date, signal = "final_close", price = rt_data$close[rt_data$date ==
      ↪ final_date][1])) |>
  arrange(date)

signal_outcomes <- signals_eval |>
  mutate(next_price  = lead(price),
         next_signal = lead(signal)) |>
  filter(signal %in% c("buy", "sell")) |>
  mutate(trade_pnl = if_else(signal == "buy", next_price - price, price - next_price),
         correct   = trade_pnl > 0)

signals_omniscient <- signal_outcomes |> filter(correct) |>
  select(date, signal, price)
signals_dumb       <- signal_outcomes |> filter(!correct) |>
  select(date, signal, price)

# Helpers
simulate_pnl <- function(signals_df, final_price) {
  pnl <- 0; position <- "flat"; entry <- NA_real_
  for (i in seq_len(nrow(signals_df))) {
    sig <- signals_df$signal[i]; px <- signals_df$price[i]
    if (position == "flat") {
      position <- if (sig == "buy") "long" else "short"; entry <- px
```

```
      } else if (sig == "buy" && position == "short") {
        pnl <- pnl + (entry - px); position <- "long"; entry <- px
      } else if (sig == "sell" && position == "long") {
        pnl <- pnl + (px - entry); position <- "short"; entry <- px
      } else if (sig == position) {
        pnl <- pnl + if (position == "long") (px - entry) else (entry - px)
        entry <- px
      }
    }
    if (!is.na(final_price) && position != "flat") {
      pnl <- pnl + if (position == "long") (final_price - entry) else (entry - final_price)}
    pnl}

make_bitacora <- function(signals_df, final_date, final_price) {
  signals_eval <- bind_rows(signals_df |> arrange(date),
                            tibble(date = final_date, signal = "final_close", price =
                             ↪  final_price)) |>
      arrange(date)
    position <- "flat"; entry <- NA_real_; pnl_acum <- 0; logs <- list()
    for (i in seq_len(nrow(signals_eval))) {
      sig <- signals_eval$signal[i]; px <- signals_eval$price[i]
      if (position == "flat") {
        if (sig %in% c("buy", "sell")) {
          flujo <- if (sig == "buy") -px else px
          logs[[length(logs) + 1]] <- tibble(date = signals_eval$date[i], 'acción' = sig,
                                             price = px, flujo = flujo,
                                             PnL = 0, PnL_acum = pnl_acum)
          position <- if (sig == "buy") "long" else "short"; entry <- px
        }
      } else if (sig %in% c("buy", "sell", "final_close")) {
        pnl <- if (position == "long") px - entry else entry - px
        flujo <- if (sig == "buy") -px else if (sig == "sell") px else 0
        pnl_acum <- pnl_acum + pnl
        logs[[length(logs) + 1]] <- tibble(date = signals_eval$date[i], 'acción' = sig,
                                           price = px, flujo = flujo,
                                           PnL = pnl, PnL_acum = pnl_acum)
        if (sig == "final_close") { position <- "flat"; entry <- NA_real_ }
        else { position <- if (sig == "buy") "long" else "short"; entry <- px }
      }
    }
    bind_rows(logs)
  }

show_head_tail <- \(df, n = 5) bind_rows(slice_head(df, n = n), slice_tail(df, n = n))
```

4 Serie de tiempo.

```r
last15_2024 <- prices_ma3 |>
  filter(year(date) == 2024) |>
  arrange(date) |>
  tail(15)

start_date_plot <- min(last15_2024$date)
end_date   <- as.Date("2025-12-19")

aapl_window <- prices_ma3 |>
  filter(date >= start_date_plot, date <= end_date)

ggplot(aapl_window, aes(x = date, y = close)) +
  geom_line(color = "#2C5EA8", linewidth = 0.9) +
  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +
  labs(title = "Apple (AAPL) - Precios de Cierre",
       subtitle = paste("Desde", start_date_plot, "hasta", end_date),
       x = "Fecha", y = "Precio de cierre (USD)") +
  theme_minimal(base_size = 12)
```



5 Benchmarks.

```r
first_2025 <- prices_ma3 |> filter(date >= as.Date("2025-01-01")) |>
  slice_head(n = 1)
buy_hold_buy_date <- first_2025$date
buy_hold_buy_px   <- first_2025$close

sell_12dec <- prices_ma3 |> filter(date <= as.Date("2025-12-12")) |>
  slice_tail(n = 1)
buy_hold_sell_date <- sell_12dec$date
buy_hold_sell_px   <- sell_12dec$close

min_row <- prices_ma3 |>
  filter(date >= as.Date("2025-01-01"), date <= as.Date("2025-12-19")) |>
  slice_min(close, n = 1, with_ties = FALSE)
min_date <- min_row$date
min_px   <- min_row$close

final_row <- prices_ma3 |> filter(date <= as.Date("2025-12-19")) |>
  slice_tail(n = 1)
final_date <- final_row$date
final_px   <- final_row$close

buy_hold_gain <- buy_hold_sell_px - buy_hold_buy_px
trampa_gain   <- (buy_hold_buy_px - min_px) + (final_px - min_px)

strategies <- tibble(estrategia = c("Buy & Hold", "Trampa"),
  ganancia_simple = c(buy_hold_gain, trampa_gain))

strategies
```

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

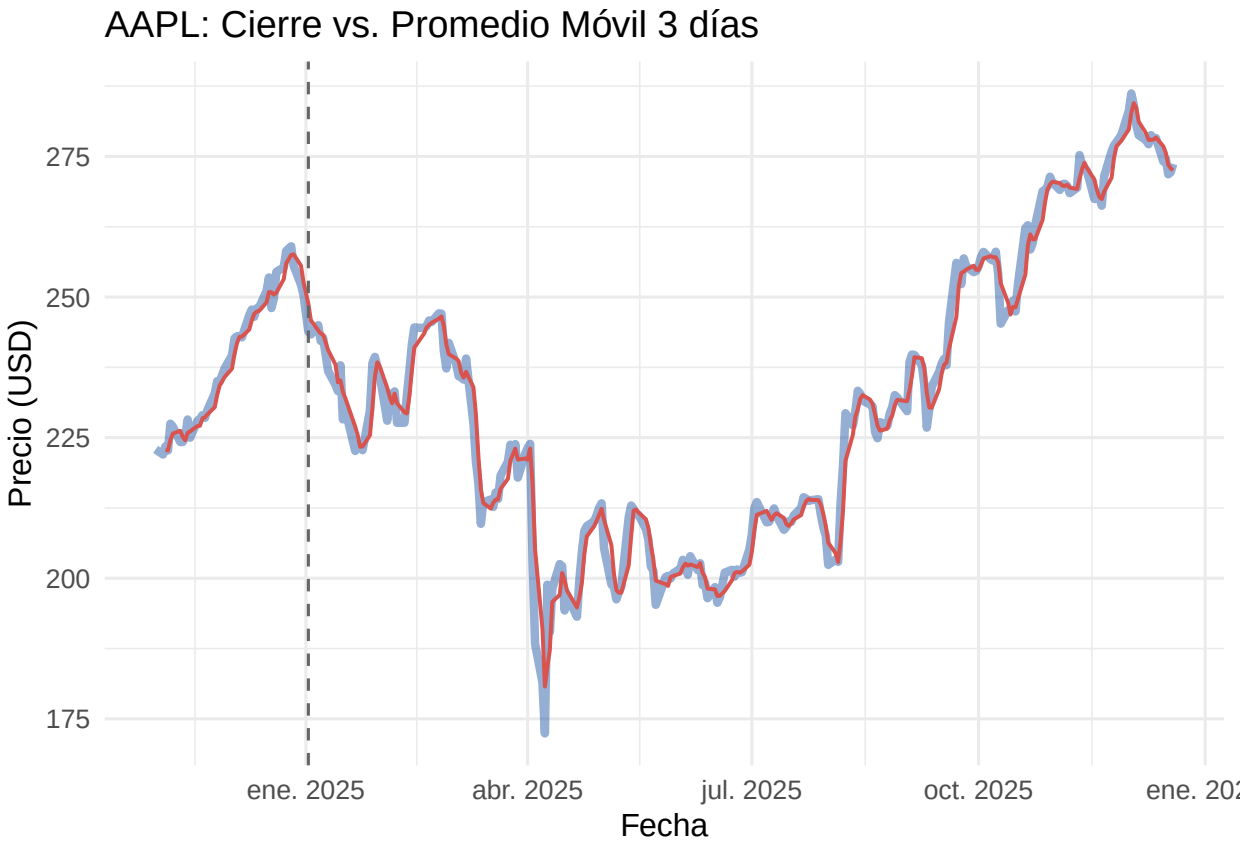
```r
P0   <- buy_hold_buy_px
Pmin <- min_px
Pend <- final_px

tramo_corto <- P0 - Pmin
tramo_largo <- Pend - Pmin
gan_trampa  <- tramo_corto + tramo_largo

tibble(P0, Pmin, Pend,
       tramo_corto = tramo_corto,
       tramo_largo = tramo_largo,
       ganancia_trampa = gan_trampa) |>
  mutate(across(everything(), ~ round(.x, 2))) |>
  kable()
```

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

- Trampa: P0 = 243.85 (cierre inicial), Pmin = 172.42 (mínimo anual), Pend = 273.67 (cierre final). El tramo corto gana $243.85 - 172.42 = 71.43$; el tramo largo gana $273.67 - 172.42 = 101.25$. La ganancia total de Trampa es $71.43 + 101.25 = 172.68$.

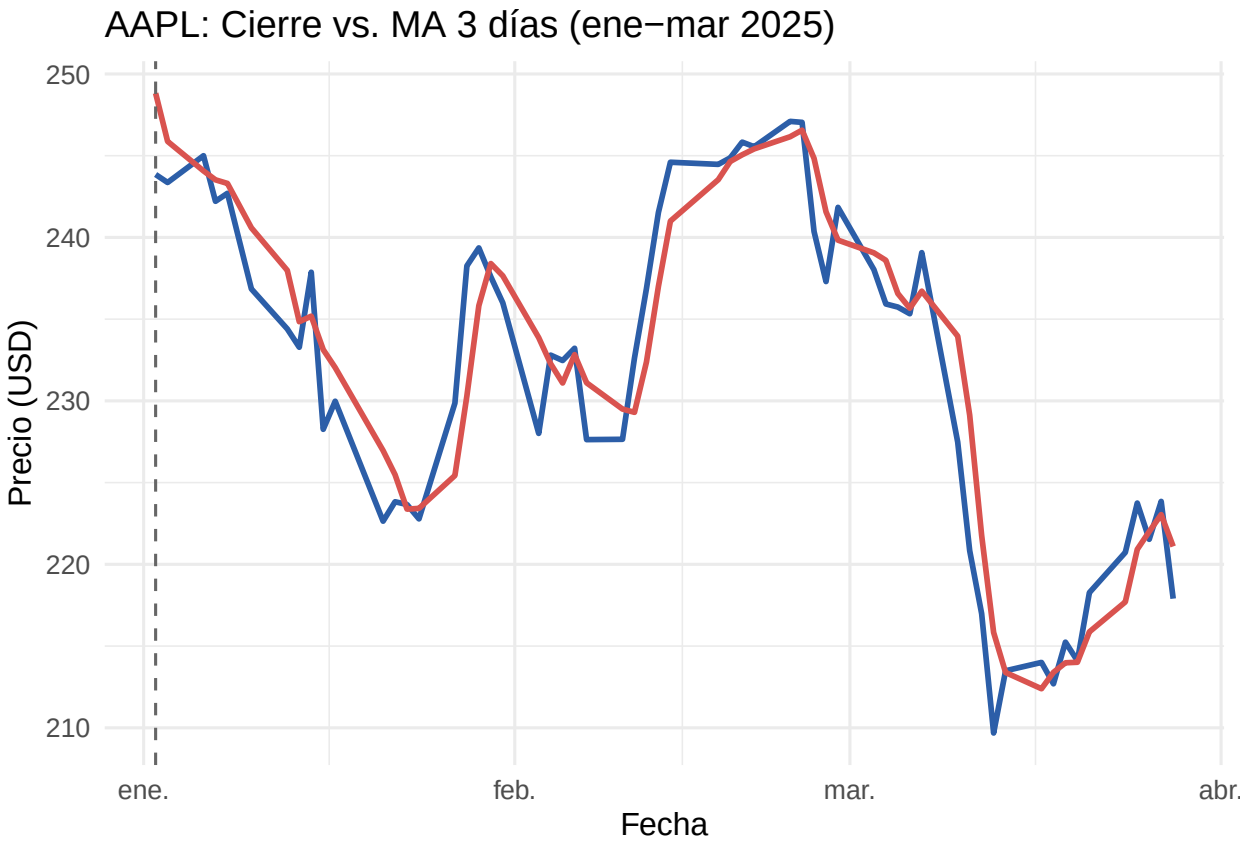
6 Promedio móvil.

```r
ggplot(prices_ma3, aes(date)) +
  geom_line(aes(y = close), color = "#2C5EA8", linewidth = 1.5, alpha = 0.5) +
  geom_line(aes(y = ma3), color = "#D9534F", linewidth = 0.7, na.rm = TRUE) +
  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +
  labs(title = "AAPL: Cierre vs. Promedio Móvil 3 días",
       x = "Fecha", y = "Precio (USD)") +
  theme_minimal(base_size = 12)
```



7 Promedio móvil, zoom.

```r
aapl_ma_short <- prices_ma3 |>
  filter(date >= as.Date("2025-01-01"), date <= as.Date("2025-03-30"))

ggplot(aapl_ma_short, aes(date)) +
  geom_line(aes(y = close), color = "#2C5EA8", linewidth = 1) +
  geom_line(aes(y = ma3), color = "#D9534F", linewidth = 1, na.rm = TRUE) +
  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray40") +
  labs(title = "AAPL: Cierre vs. MA 3 días (ene-mar 2025)",
       x = "Fecha", y = "Precio (USD)") +
  theme_minimal(base_size = 12)
```



8 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 (short + long): 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 (MA3): opera cada cruce cierre vs SMA 3d; 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: aplica la regla de cruces pero solo ejecuta las señales que resultan ganadoras; si salen dos señales iguales seguidas, se cierra y reabre al nuevo precio, por eso el PnL refleja esos tramos y está bien calculado aun con sell-sell o buy-buy.
- Robo-dumb: igual lógica pero solo con señales perdedoras; también cierra y reabre en señales iguales consecutivas, de modo que el PnL recoge correctamente esos tramos negativos.

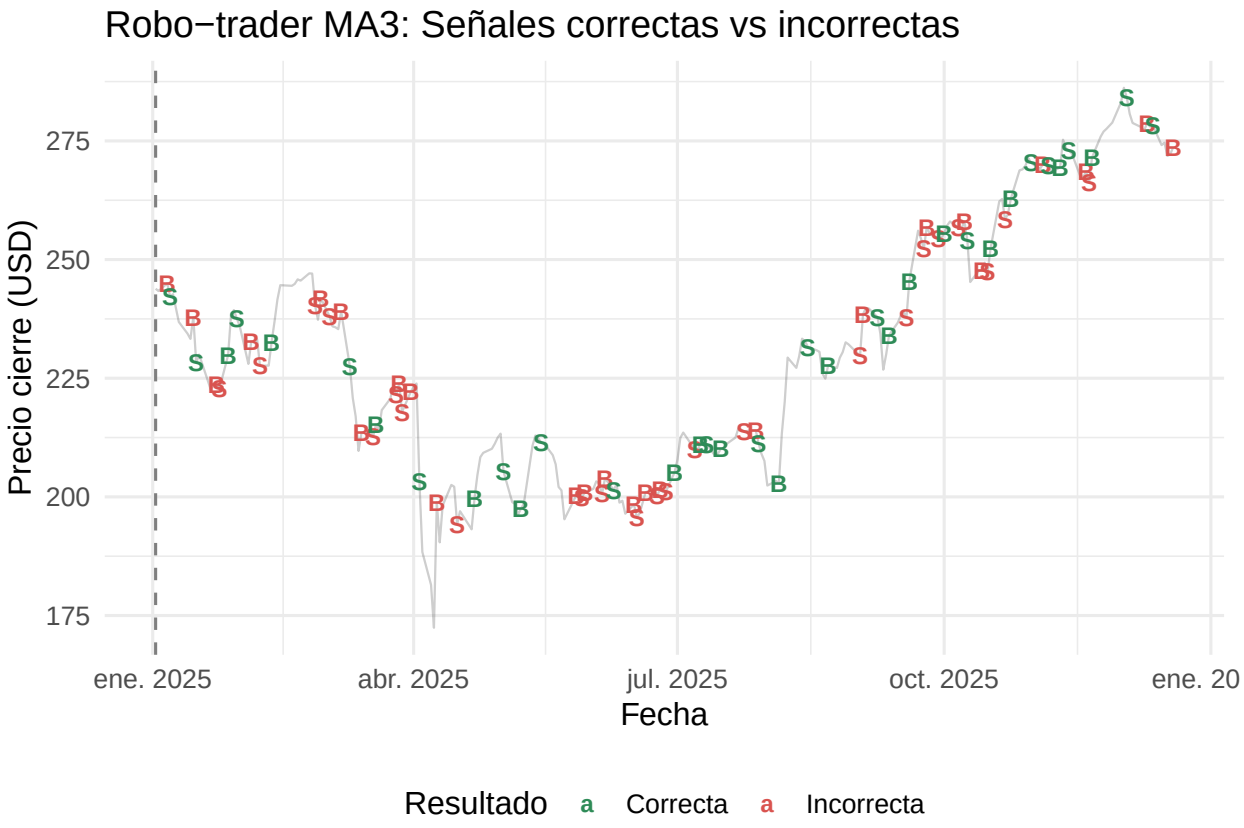
9 Desempeño de las 6 estrategias.

```r
set.seed(123)
prices <- prices_ma3

P0 <- prices |> filter(date >= as.Date("2025-01-01")) |> slice_head(n = 1) |>
  pull(close)
bh_sell_px <- prices |> filter(date <= as.Date("2025-12-12")) |> slice_tail(n = 1) |>
  pull(close)
Pmin <- prices |> filter(date >= as.Date("2025-01-01"), date <= final_date) |>
  slice_min(close, n = 1, with_ties = FALSE) |> pull(close)
Pend <- prices |> filter(date == final_date) |> pull(close)

buy_hold_gain <- bh_sell_px - P0
trampa_gain   <- (P0 - Pmin) + (Pend - Pmin)

trading_days <- prices |>
  filter(date >= start_date, date <= final_date) |>
  pull(date)

random_dates <- sort(sample(trading_days, size = 83, replace = FALSE))
start_side <- sample(c("buy", "sell"), 1)
random_signals_vec <- if (start_side == "buy") rep(c("buy", "sell"), length.out = 83) else
  ↪  rep(c("sell", "buy"), length.out = 83)

signals_rand <- tibble(date = random_dates,
                       signal = random_signals_vec) |>
  left_join(prices |> select(date, close), by = "date") |>
  rename(price = close) |>
  arrange(date)

# Reuso de signal_outcomes ya calculado arriba
robo_trader_gain      <- simulate_pnl(signals_rt, final_price = Pend)
random_gain           <- simulate_pnl(signals_rand, final_price = Pend)
robo_omnisciente_gain <- simulate_pnl(signals_omniscient, final_price = Pend)
robo_dumb_gain        <- simulate_pnl(signals_dumb, final_price = Pend)

rt_trades   <- nrow(signals_rt)
rand_trades <- nrow(signals_rand)
omn_trades  <- nrow(signals_omniscient)
dumb_trades <- nrow(signals_dumb)

strategies <- tibble(estrategia = c("Buy & Hold", "Trampa", "Robo-trader",
                                    "Aleatoria", "Robo-omnisciente",
                                    "Robo-dumb"),
trades     = c(NA_integer_, NA_integer_, rt_trades, rand_trades, omn_trades,
               dumb_trades),
ganancia_simple = c(buy_hold_gain, trampa_gain, robo_trader_gain, random_gain,
                    robo_omnisciente_gain, robo_dumb_gain),
ganancia_por_trade = ganancia_simple / trades)

strategies_tbl <- strategies |>
  mutate(ganancia_simple      = round(ganancia_simple, 2),
         ganancia_por_trade   = round(ganancia_por_trade, 2))

strategies_tbl |>
  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

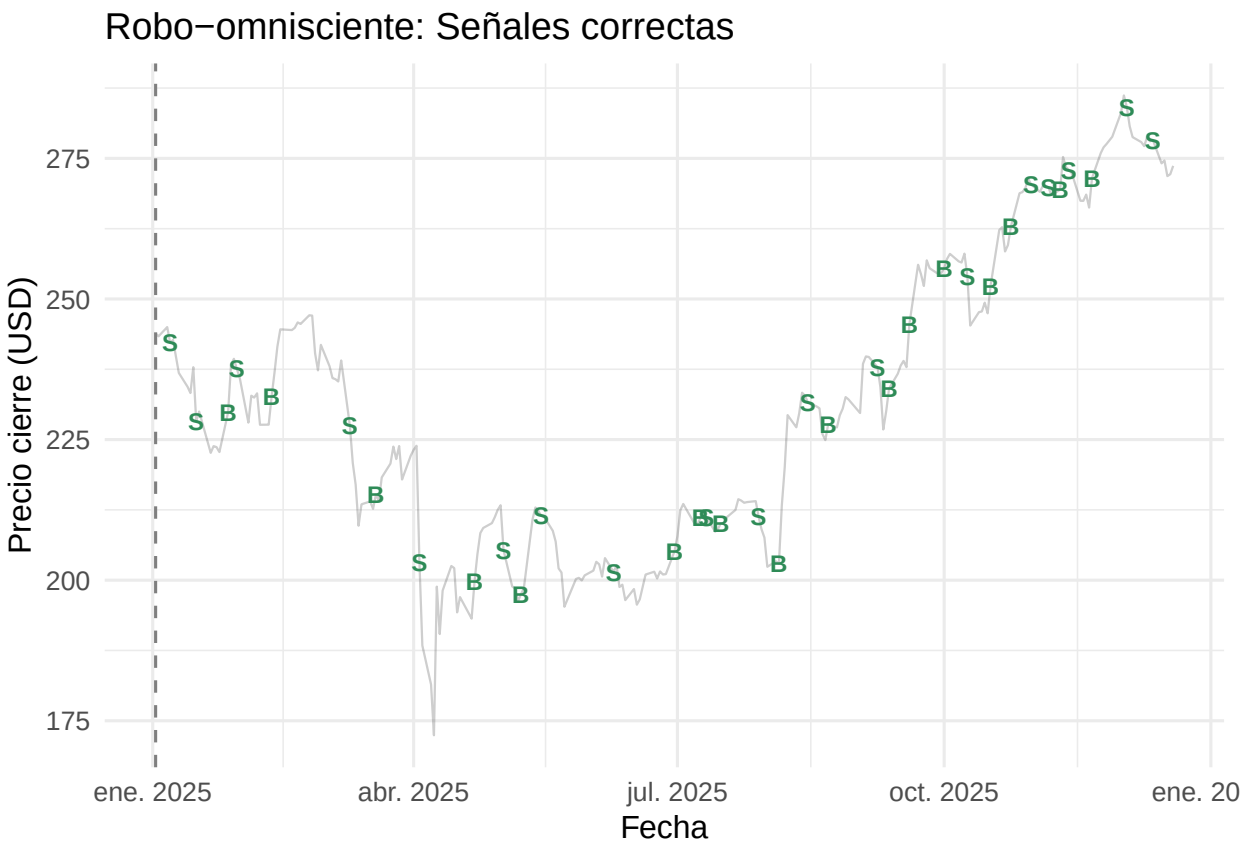
10 Visualización del Robo-trader.

```r
signals_plot <- signals_rt |>
  left_join(signal_outcomes |> select(date, signal, correct),
            by = c("date", "signal")) |>
  mutate(correcto = if_else(correct, "Correcta", "Incorrecta"))

ggplot(rt_data, aes(date, close)) +
  geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
  geom_text(data = signals_plot,
            aes(y = price, label = if_else(signal == "buy", "B", "S"),
                color = correcto),fontface = "bold", size = 3) +
  scale_color_manual(values = c(Correcta = "#2E8B57", Incorrecta = "#D9534F"), name =
    ↪  "Resultado") +
  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
  labs(title = "Robo-trader MA3: Señales correctas vs incorrectas",
       x = "Fecha", y = "Precio cierre (USD)") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "bottom")
```



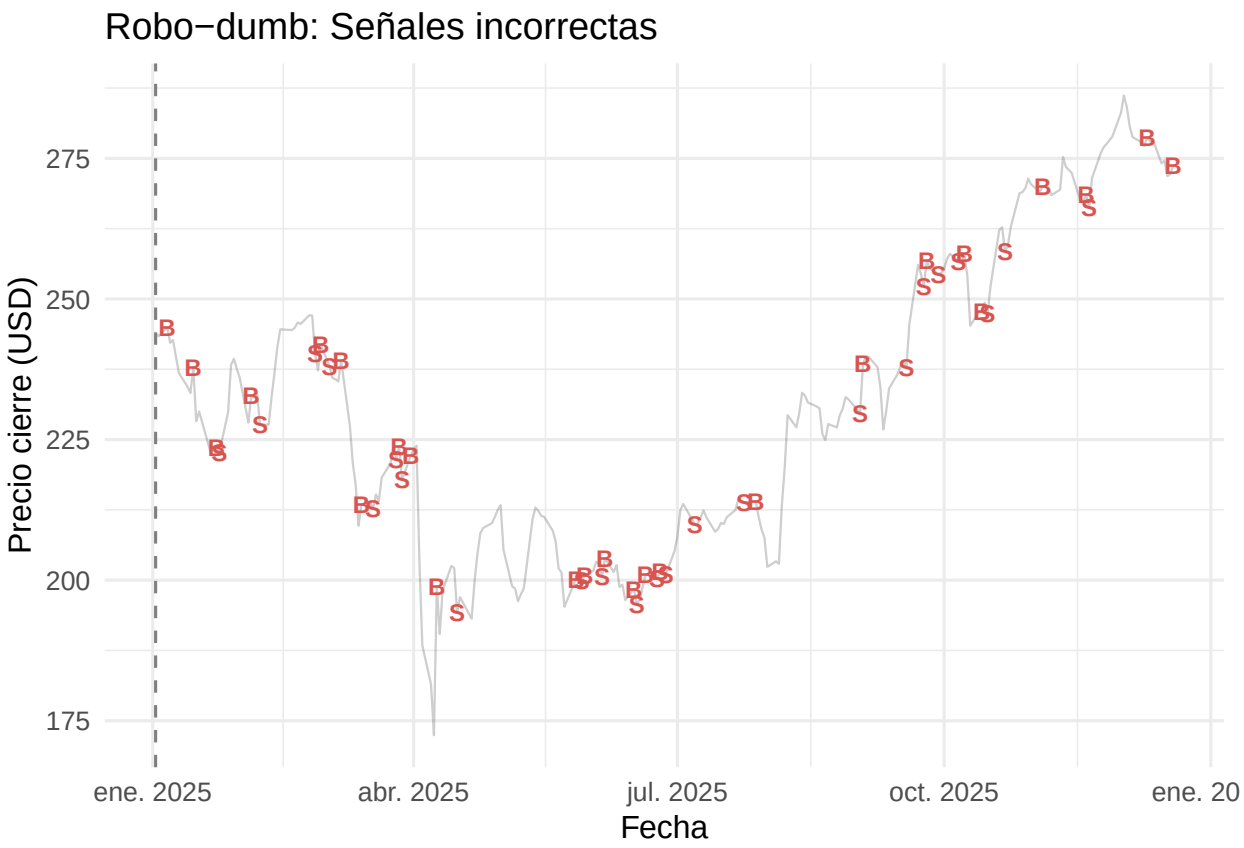
11 Visualización del Robo-omnisciente.

```
signals_plot_omn <- signals_omniscient |>
  mutate(correcto = "Correcta")

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



12 Visualización del Robo-dumb.

```r
signals_plot_dumb <- signals_dumb |>
  mutate(correcto = "Incorrecta")

ggplot(rt_data, aes(date, close)) +
  geom_line(color = "gray35", linewidth = 0.4, alpha = 0.3) +
  geom_text(data = signals_plot_dumb,
            aes(y = price, label = if_else(signal == "buy", "B", "S"), color = correcto),
            fontface = "bold", size = 3) +
  scale_color_manual(values = c(Incorrecta = "#D9534F"), name = "Resultado") +
  geom_vline(xintercept = as.Date("2025-01-02"), linetype = "dashed", color = "gray50") +
  labs(title = "Robo-dumb: Señales incorrectas",
       x = "Fecha", y = "Precio cierre (USD)") +
  theme_minimal(base_size = 12) +
  theme(legend.position = "none")
```



13 Bitácora Robo-trader.

```
bitacora_num <- make_bitacora(signals_rt, final_date, Pend) |>
  mutate(price     = round(price, 2),
         flujo     = round(flujo, 2),
         PnL       = round(PnL, 2),
         PnL_acum  = round(PnL_acum, 2))

bitacora_num |>
  show_head_tail(n = 5) |>
  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: señal sell abre un short en 284.15; ganancia 0 en la apertura, PnL acum 37.55 arrastrado de antes.
- 10-dic: señal buy cierra el short 284.15→278.78 con +5.37 y abre un long a 278.78; PnL acum 42.92.
- 12-dic: señal sell cierra el long 278.78→278.28 con -0.50 y abre un short a 278.28; PnL acum 42.42.
- 19-dic: señal buy cierra el short 278.28→273.67 con +4.61 y abre un long a 273.67; PnL acum 47.03.
- 19-dic (final_close): cierra el long 273.67→273.67 con 0; PnL final 47.03.

14 Resumen Robo-trader.

```r
confusion <- signal_outcomes |>
  mutate(outcome = if_else(correct, "correcta", "incorrecta")) |>
  count(signal, outcome) |>
  pivot_wider(names_from = outcome, values_from = n, values_fill = 0)

confusion
```

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

15 Bitácora estrategia aleatoria.

```
bitacora_aleatoria_num <- make_bitacora(signals_rand, final_date, Pend) |>
  mutate(price     = round(price, 2),
         flujo     = round(flujo, 2),
         PnL       = round(PnL, 2),
         PnL_acum  = round(PnL_acum, 2))

bitacora_aleatoria_num |>
  show_head_tail(n = 5) |>
  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

16 Bitácora Robo-omnisciente.

```
bitacora_omnisciente_num <- make_bitacora(signals_omniscient, final_date, Pend) |>
  mutate(price     = round(price, 2),
         flujo     = round(flujo, 2),
         PnL       = round(PnL, 2),
         PnL_acum  = round(PnL_acum, 2))

bitacora_omnisciente_num |>
  show_head_tail(n = 5) |>
  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: señal sell abre un short a 272.95; ganancia 0 (solo apertura), PnL acum 164.40.
- 21-nov: señal buy cierra el short 272.95→271.49 con +1.46 y abre un long a 271.49; PnL acum 165.86.
- 03-dic: señal sell cierra el long 271.49→284.15 con +12.66 y abre un short a 284.15; PnL acum 178.52.
- 12-dic: señal sell repetida cierra el short 284.15→278.28 con +5.87 y reabre short a 278.28; PnL acum 184.39.
- 19-dic (final_close): cierra el short 278.28→273.67 con +4.61; PnL final 189.00.

17 Bitácora Robo-dumb.

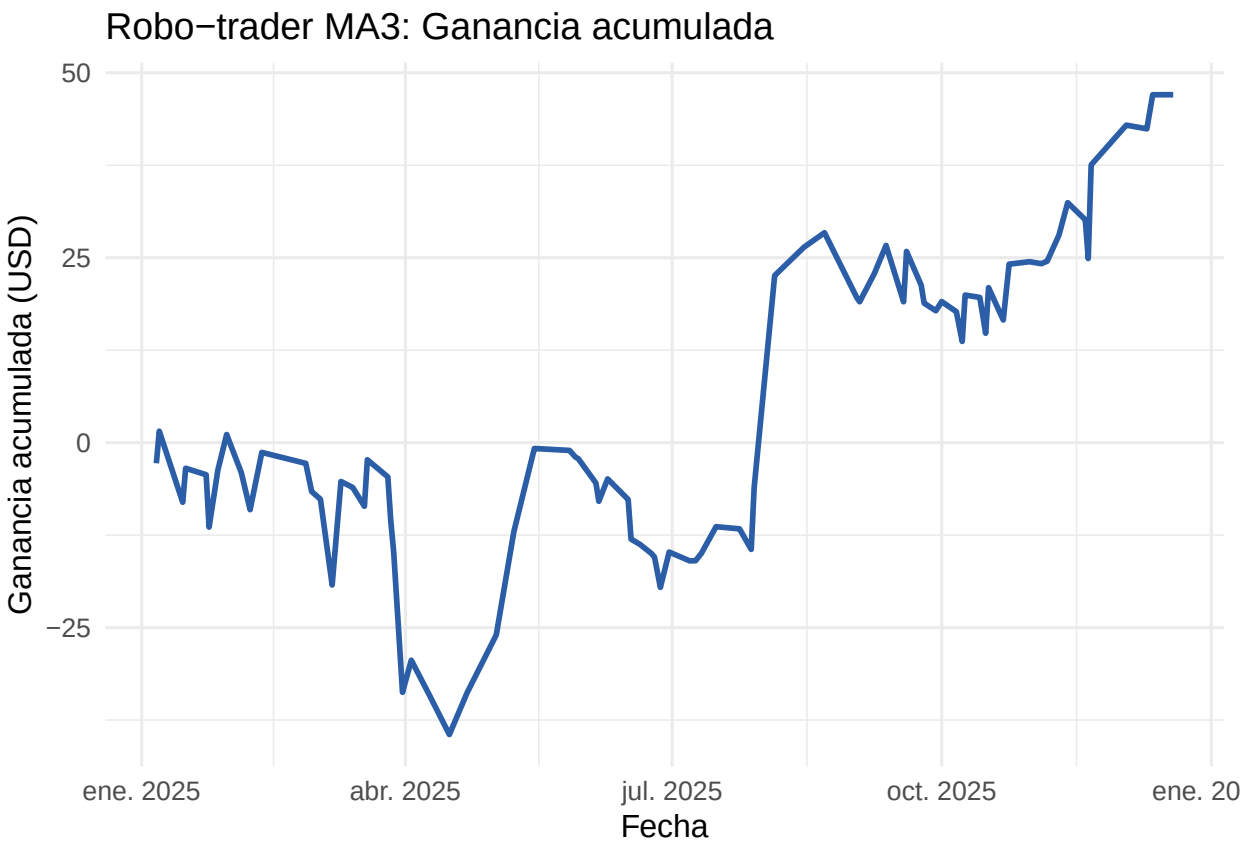
```
bitacora_dumb_num <- make_bitacora(signals_dumb, final_date, Pend) |>
  mutate(price     = round(price, 2),
         flujo     = round(flujo, 2),
         PnL       = round(PnL, 2),
         PnL_acum  = round(PnL_acum, 2))

bitacora_dumb_num |>
  show_head_tail(n = 5) |>
  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: señal buy abre un long en 268.56; ganancia 0 en la apertura, PnL acum -225.12 arrastrado de antes.
- 20-nov: señal sell cierra el long 268.56→266.25 con -2.31 y abre un short a 266.25; PnL acum -227.43.
- 10-dic: señal buy cierra el short 266.25→278.78 con -12.53 y abre un long a 278.78; PnL acum -239.96.
- 19-dic (buy repetida): cierra el long 278.78→273.67 con -5.11 y reabre long a 273.67; PnL acum -245.07.
- 19-dic (final_close): cierra el long 273.67→273.67 con 0; PnL final -245.07.

18 Ganancias Robo-trader.

```
ggplot(signal_outcomes |> mutate(PnL_acum = cumsum(replace_na(trade_pnl, 0))),
       aes(x = date, y = PnL_acum)) +
  geom_line(color = "#2C5EA8", linewidth = 1) +
  labs(title = "Robo-trader MA3: Ganancia acumulada",
       x = "Fecha", y = "Ganancia acumulada (USD)") +
  theme_minimal(base_size = 12)
```



19 Conclusión.

- La comparación evidencia cómo una regla mecánica (MA3) se posiciona frente a estrategias pasivas, un benchmark idealizado 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 en términos de desempeño y frecuencia de operación.