BotFactoryFX

MANUAL DEL PROGRAMA

BotFactoryFX

Versión 1.34.0

Todo: búsqueda, juez, cuarentena —de tus estrategias y de un robot que te vendieron—, cartera, tus cuentas de MetaTrader 5, JForex y web3 en una sola pantalla, Dr. Quant y la exportación a cinco plataformas.

La misma Ayuda que lleva el programa dentro, capítulo a capítulo. Sale del propio código del programa, así que no puede contar nada que el programa no haga.

Dentro del programa, esta misma Ayuda está en Ayuda, con un buscador (Ctrl+K), y Ayuda → Hacer el tour recorre cada pantalla en el orden del trabajo.

BotFactoryFX 1.34.0 · 6 de noviembre de 2026

Contenido

EMPEZAR Y CONSTRUIR

Qué es BotFactoryFX 3
Inicio 3
Piloto automático 4
Construir › Manual 6
Construir › Búsqueda genética 10
Construir › Explorar 13

JUZGAR

Reality Check 15
El método 21
Instalar el robot en cada plataforma 22

DATOS, BIBLIOTECA Y CARTERA

Los datos y las temporalidades 29
Conectar MetaTrader 5 31
Conectar JForex 34
Trabajar con TradingView 36
Biblioteca 37
Cartera 38

DESPUÉS DEL BACKTEST

Cuarentena 40
Cuentas en vivo y monederos 41
Dr. Quant 43
Dr. Quant en la web, y lo que cuesta 45

AL FINAL

Preguntas frecuentes 46
Glosario de métricas 49
Atajos y consejos 49

EMPEZAR Y CONSTRUIR

Qué es BotFactoryFX

BotFactoryFX es un **laboratorio de trading algorítmico**: construye estrategias y después dedica casi todo su esfuerzo a demostrar que las que ha encontrado no valen nada. Importa tu propio histórico, escribe una estrategia a mano o deja que la búsqueda genética la encuentre, pásala por el Reality Check, junta en una cartera lo que sobreviva y exporta el resultado como MQL4, MQL5, JForex, un cBot de cTrader, Pine Script o un bot de Python que opera en Hyperliquid. Tu trabajo se guarda en esta máquina de una sesión a otra.

El menú de la izquierda es el orden en que se trabaja:

1. **Inicio** — la curva que estás mirando: una cartera simulada o el último backtest, una cuarentena, o una cuenta en vivo.
2. **Datos** — el histórico de precios, su temporalidad, lo que te cobra el bróker por operarlo y la ficha del instrumento.
3. **Piloto automático** — dile en una frase lo que quieres, y busca, juzga, guarda y combina él solo.
4. **Construir** — escribe tú la estrategia (*Manual*), deja que la encuentre el buscador (*Búsqueda genética*), o criba ideas antes de casarte con una (*Explorar*).
5. **Reality Check** — si el resultado es real, sabiendo cuántas variantes se probaron para llegar a él.
6. **Biblioteca** — las estrategias que te quedaste, con sus informes, listas para descargar en cualquiera de las seis plataformas.
7. **Cuarentena** — una estrategia corre sobre velas que van llegando, antes de ver dinero.
8. **Cartera** — junta lo que pasó las dos pruebas, el juez y la cuarentena, y las simula a la vez.
9. **En vivo** — las cuentas que operan, con sus operaciones de verdad sobre el gráfico.

Inicio

Pantalla de llegada. Enseña los indicadores de tu cartera hipotética (o los del último backtest, si no hay cartera), una curva de capital y una lista de las estrategias que guardaste hace poco.

- **Curva de capital**: la de una cuenta en vivo si tienes alguna; si no, la de una cuarentena; si no, la simulada — la última simulación de cartera, o el último backtest. Los botones de encima cambian de una a otra; ver *Cuentas en vivo y monederos*.
- **Por dónde seguir**: atajos a Datos, Construir, Reality Check y Cartera, en el orden del menú.
- **Estrategias recientes**: las 5 últimas que guardaste.
- **La Cuarentena de un vistazo**: lo que tienes en cuarentena, junto — robots, operaciones cerradas y cuántas ganadas, el resultado de todas en porcentaje de la cuenta de cada una (lo que habría hecho el mismo capital repartido entre ellas), el veredicto de cada una, su curva conjunta y la lista entera, de la mejor a la peor. *Ver* elige todas, una carpeta, un activo, una temporalidad o algunas una a una, y se acuerda. Los robots que ya operan en una cuenta no se suman: lo suyo es dinero que hizo una cuenta, no una simulación.

Piloto automático

Sección propia, antes de Construir. Di en una frase lo que quieres —el instrumento, la temporalidad, el estilo, cuánto drawdown aguantas— y hace el resto: busca, mide, elige, juzga, guarda y combina. Los pasos son las operaciones de verdad, no una narración de ellas: la misma búsqueda genética de Construir, el mismo Reality Check, la misma simulación de cartera.

La frase se lee con código, no con un modelo, así que construir no necesita ningún proveedor de IA y no puede inventarse una temporalidad que no mencionaste. Dos órdenes más también se leen con código: «juzga» y el nombre de una estrategia de la Biblioteca corre el Reality Check sobre su propio histórico, y «lleva» y su nombre «a la cartera» la marca en la selección de Cartera — cada una detrás de una ficha que dice exactamente qué va a correr, y nada corre sin tu clic. El piloto construye estrategias nuevas y nada más: una orden sobre una que ya tienes —«optimiza el periodo del RSI», «cambia el stop loss»— no es ninguna de ésas, y tampoco llega al modelo. La contesta Dr. Quant él mismo, diciendo que desde esta conversación no ejecuta nada y dónde se hace de verdad: el periodo de un indicador se cambia en Construir › Manual, el stop y el objetivo en Gestión monetaria, y después Backtest. Todo lo demás se toma como una pregunta, y las preguntas van al modelo de Dr. Quant, que sí necesita proveedor.

También vale una orden corta sin verbo —«S&P 500 diario, reversión y tendencia»—, siempre que nombre un instrumento y un estilo o una temporalidad y no sea una pregunta; lo que preguntas sigue yendo a Dr. Quant.

Antes de empezar

Enseña lo que ha entendido y, bajo «No lo dijiste, así que he elegido», todo lo que ha tenido que decidir por ti, cada cosa con su motivo. Cualquiera de ellas se cambia escribiendo otra frase; «Empezar» lo pone en marcha y «Todavía no» lo deja. Si algo hace imposible el encargo —no hay histórico cargado, un instrumento que no está cargado, una temporalidad más fina que las velas cargadas— dice qué, y no ofrece «Empezar».

El histórico y la vela se eligen en la tira de arriba: un encargo que no nombre otro instrumento corre sobre ése, aunque haya otro histórico del mismo símbolo con más velas o con justo la temporalidad pedida. Un encargo que nombre otro instrumento corre sobre la serie de ése, no sobre la que tengas abierta: carga su histórico y, si la temporalidad que pediste es más gruesa que sus velas, las reagrupa. Todo lo que hace el propio piloto usa esa misma serie: el Reality Check, el spread y la comisión de ese instrumento, y la cartera que combina. El resumen de antes de empezar dice sobre qué fichero va a correr. La pantalla de Cartera hace lo mismo: allí cada estrategia se simula sobre su propio histórico, con su spread y su comisión, y la que no lo tenga cargado se queda fuera y con su nombre escrito, en vez de correrse sobre las velas de otro instrumento.

«Ya probado aquí», sobre la serie con la que va a trabajar, es el diario de veredictos hablando: cuántos juicios lleva ese mercado, la mejor palabra y cuándo fue el último. Lo escribe solo quien juzga —la pantalla del juez, el lote, el Piloto, la Rueda— y Dr. Quant también lo lee, para que lo ya rechazado no vuelva como idea nueva. Si ahí no se juzgó nada, la línea no sale.

El resumen dice además contra qué podrá comparar el resultado el Reality Check: qué otros instrumentos hay cargados, y que los otros relojes se construyen reagrupando esta misma serie. Si no hay un segundo instrumento cargado, lo dice antes de empezar y dice lo que cuesta: con uno solo, nada de lo que salga de la corrida puede pasar de 64, y «sólida» empieza en 75. Importar uno son segundos; enterarse después cuesta la corrida entera.

Las fichas de estilo

Debajo de la caja hay una fila de fichas de estilo: «de cualquier tipo» y las siete familias. Una ficha escribe su palabra en tu frase, donde la puedes ver, mover o borrar: el encargo sigue siendo una sola frase. Y una ficha está encendida porque la frase nombra ese estilo, no porque se pulsara, así que borrar la palabra la apaga.

Pulsa varias y se juntan en la frase, y la búsqueda mezcla esas familias, igual que marcar varias fichas en Construir › Búsqueda genética. Escrito a mano funciona igual: los estilos que enumeras —«tendencia y rupturas», «acción del precio con divergencias»— se buscan todos, y el que descartas se queda fuera: «tendencia, no rupturas» busca sólo tendencia.

«De cualquier tipo» quita todas las palabras de estilo. Una frase que no nombra ningún estilo corre la búsqueda en modo aleatorio: sin familia y sin plantillas, sobre un repertorio corto — cierre, SMA, EMA, RSI y ATR.

Qué hace una vez en marcha

1. Búsqueda: ocho búsquedas genéticas, una detrás de otra, salvo que la frase pida otro número («10 búsquedas»), cada una con su semilla, sobre las familias que nombra la frase. Cada búsqueda acaba en una, dos o tres ideas según su azar, y ocho distintas encuentran lo que una sola no ve. La mitad entrena también en los instrumentos con los que comparará el Reality Check, y la otra mitad sólo en éste: en el S&P 500, entrenar con el Dow es lo que encontró las estrategias que aguantan en los dos; en bitcoin, entrenar con ether frenaba la búsqueda.
2. Backtests: cada estrategia de la última generación de cada búsqueda, medida sobre la serie entera.
3. Selección: las más fuertes de cada búsqueda —cinco, salvo que la frase pida otro número—, dejando fuera las que hacen las mismas operaciones que otra ya elegida, de esa búsqueda o de otra.
4. Reality Check: cada una, sabiendo cuántos juegos de reglas probaron entre todas las búsquedas: ocho búsquedas son un torneo ocho veces mayor, y el juez pide más evidencia en proporción. No puede saber qué elegiste en «Cómo juzga», así que la comprobación de otros instrumentos usa los cargados con más historia.
5. Su hermano: las sólidas del tramo que la búsqueda no vio se juzgan otra vez con su instrumento hermano de sujeto, desde la misma fecha. El hermano es el instrumento cargado cuyos movimientos diarios más se parecían a los de éste antes de esa fecha, con una correlación de 0,5 o más, y el stop y el objetivo viajan a la escala de su vela típica. Sólo las que siguen sólidas allí se guardan solas y se combinan; la lista enseña las dos notas. Sin hermano, no cambia nada. Sobre ruido puro, 11 de 576 búsquedas dejaron alguna sólida, y ninguna pasó esto.
6. Biblioteca: se guardan sólo las sólidas, con su veredicto —el del tramo que la búsqueda no vio, cuando lo hay —. Al terminar pregunta si quieres guardar alguna más: los botones de debajo de la lista guardan las frágiles, todas o una a una, con su veredicto y la misma cuenta de lo que probaron las búsquedas.
7. Cartera: sólo se combinan las sólidas. Como las eligió el tramo que la búsqueda no vio, la curva de la cartera en ese tramo no es del todo «no vista», y el paso lo dice. Si no sale ninguna sólida, no se construye cartera, y eso es un resultado, no un fallo.

Cerca del listón — las que no lo pasan pero se quedan cerca: 45 o más en el tramo que la búsqueda no vio, o sólidas ahí y no con su hermano. No se guardan ni se combinan solas: la lista las marca, y Mandar a cuarentena guarda una con su veredicto y abre la Cuarentena con ella ya elegida —las sólidas que pasan también llevan el botón, porque la cuarentena es su paso siguiente—. La cuarentena no cuesta nada, y es donde se ve si merecían la pena. El 45 es provisional: cada expediente apunta cómo entró su estrategia, para mover la franja con lo que haga cada grupo en vivo.

En la ficha del encargo, Probar también el procedimiento añade un séptimo paso al final: la misma búsqueda, repetida en ventanas del histórico, cada una operada sobre el tramo que no vio — la prueba que Construir › Búsqueda genética llama Probar el procedimiento. Tarda más que la propia búsqueda, y la ficha dice cuánto, así que no corre si no la marcas; su resultado sale debajo de la ejecución. Juzga cómo se encontraron las estrategias, no una de ellas, y no cambia ningún veredicto.

Se queda en esta pantalla: los seis pasos se ven aquí, y «Detener» corta la corrida en cualquier momento. Al acabar, «Ver la cartera» —o «Ver lo que ha salido», si no se construyó cartera— te lleva al resultado.

Construir › Manual

Editor visual de estrategias con bloques de reglas «Y»/«O» — la pestaña *Manual* de **Construir**, al lado de *Búsqueda genética* y *Explorar*. Trabaja sobre el histórico que elijas en la tira de arriba —el histórico y la vela—, y la estrategia toma de ahí su símbolo y su temporalidad. Para traer otro, el botón **Bajar otro en Datos** de la tira lleva a Datos. No hay muestra sintética: cada veredicto vale exactamente lo que valgan los datos que tiene detrás.

El editor tiene dos modos: **Guiado** (eliges familia y plantilla, y las reglas vienen puestas) o **Personalizado (Alchemist)** (mezclas reglas a tu aire). Lleva dentro el comprobador de redundancias en vivo. Debajo del nombre, el símbolo y la temporalidad, la tarjeta **Series de referencia** añade hasta tres mercados que las reglas leen sin operarlos —«largo en EURUSD sólo si el DXY baja»—: se eligen de la biblioteca, y cada condición tiene entonces una casilla *Serie* para leerla en una de ellas. La **gestión monetaria** va plegada al pie: el riesgo básico (stop loss, take profit, % de riesgo, posiciones máximas, lote) —el stop y el objetivo, en **pips**, en **múltiplos del ATR** o en **% del precio**: en ATR, «dos veces el ATR» es la misma idea en cualquier instrumento, que unos pips no lo son; los dos últimos se miden al decidir la entrada y quedan fijos para la operación—, el tamaño de posición (lote fijo / % de riesgo / Kelly) y la gestión avanzada (stop dinámico, punto de equilibrio, pérdida máxima diaria, operaciones máximas al día). Debajo, **Horario y salidas por reloj**, todo en horas UTC: abrir sólo si la vela de la señal empieza dentro de un horario —de 8 a 17, o de 22 a 6 cruzando la medianoche—, con cuatro botones para las sesiones — Asia, Europa, Europa y América, América— puestas en el reloj del histórico cuando se sabe; el **cierre del viernes**, que desde esa hora cierra lo abierto y no abre nada hasta el lunes —tampoco el sábado ni el domingo, que en un mercado que no cierra tienen velas—; y cerrar a una hora fija, o tras tantas velas o tantos minutos. Y el **Corte de cuenta**: si el saldo cae un tanto por ciento desde su máximo, si se encadenan tantas operaciones perdedoras o si se pierde un tanto por ciento en una semana UTC, no se abre nada más —en el backtest, hasta el final; en el robot exportado, hasta que alguien lo quite a mano: MetaTrader con `KillSwitchReset`, JForex con `killSwitchReset`, el bot de Python con `KILL_SWITCH_RESET`, y el robot lo recuerda aunque se reinicie—. A diferencia de los topes del día, no se levanta solo. Se mide sobre el saldo cerrado, y el informe dice cuándo saltó y por qué. La gestión avanzada y el reloj los simula el backtest *y además* se escriben en el robot exportado, así que el veredicto describe el robot que de verdad te descargas. Una barra que se queda al pie de la pantalla lleva **Backtest**, **Exportar…** y **Guardar en la Biblioteca**. Los dos últimos guardan la estrategia y te llevan a la Biblioteca, que es donde se exporta: **Exportar…** te abre ya su menú de descarga, con seis destinos —**MQL4**, **MQL5**, **Java (JForex)**, **C# (cTrader)**, **Pine Script** y un bot de **Python** para Hyperliquid—. Qué hacer con el fichero está en *Instalar el robot en cada plataforma*.

La orden de entrada

Encima de la gestión monetaria, **Orden de entrada** dice cómo entra cuando la regla dice que sí: **a mercado**, al cierre de la vela de la señal; con una **orden stop**, a favor del movimiento —por encima para comprar—; o con una **orden límite**, esperando el retroceso. La orden descansa en el cierre de la vela de la señal o en **un nivel**: el máximo o el mínimo de N velas, una media, unas bandas, un canal, el máximo o el mínimo del día anterior. Es la ruptura de libro: *compra con una orden stop en el máximo de las últimas 20 velas, y vende con otra en el mínimo*. La **distancia** va más allá del nivel, y la **caducidad** es cuántas velas más espera después de la siguiente. Si el nivel queda del lado equivocado del precio —un stop de compra por debajo del cierre—, no se pone orden, como la rechazaría un bróker. La búsqueda genética también las escribe —con la familia Ruptura, además, en un nivel—, y en *Aptitud y restricciones* eliges con cuáles puede buscar: a mercado, stop, límite, o las que quieras mientras quede una. La límite es la que el backtest trata con más optimismo: la da por llena en cuanto el precio la toca, y en un mercado real tocar no siempre es llenarse. Los seis robots exportados las ponen igual que el backtest.

Las familias de estrategias — siete. Cada una es un repertorio de indicadores, una lista corta de periodos y un juego de plantillas, y la búsqueda genética sortea de ahí cuando la marcas. El MACD, la Bollinger y el Keltner son la excepción en lo de los periodos: cada uno empaqueta dos o tres números en uno —la EMA rápida y la lenta (y la

señal); el periodo y la desviación; la EMA y el ATR—, así que la búsqueda les sortea cada número dentro de su propio rango: el MACD con la rápida de 5 a 20, la lenta al menos el doble de la rápida y como mucho 60, y la señal de 5 a 15; la Bollinger de 10 a 50 velas y de 1,0 a 3,0 desviaciones; el Keltner con la EMA de 10 a 50 y el ATR de 5 a 20. Las reglas los enseñan desempaquetados: `Bollinger Upper(20, 2.0)` son 20 velas y 2,0 desviaciones, y es lo que usan las plantillas de Bollinger. Hay una segunda excepción, por el motivo contrario: el Parabolic SAR, los osciladores Awesome y Accelerator y las cuatro líneas de Ichimoku **no tienen periodo ninguno**. Los suyos son parte de lo que son —el SAR tiene un paso de aceleración de 0,02 y un techo de 0,2, el Awesome es la media de 5 velas menos la de 34, e Ichimoku es 9, 26 y 52—, así que la búsqueda no sortea ninguno, el editor no enseña casilla de periodo, y las reglas los escriben sin número: `Awesome Osc`, y no `Awesome Osc(14)`. Una estrategia guardada que lleve ese número dentro sigue operando exactamente igual — el motor y los seis robots ya lo ignoraban; lo que se acaba es enseñarlo.

- **Tendencia** — entra en la dirección de un movimiento ya establecido: cruces y alineación de medias móviles, con filtro de ADX o sin él. Repertorio: SMA, EMA, SMMA, LWMA, ADX, MACD con su señal y su histograma, Parabolic SAR y las cuatro líneas de Ichimoku.
- **Ruptura** — el precio se escapa de un rango o de un nivel reciente. Repertorio: el máximo, el mínimo y el cierre de la vela, el cierre anterior, el máximo y el mínimo de N velas (el canal de Donchian), el máximo, el mínimo, la apertura y el cierre del día anterior, las bandas de Bollinger y las de Keltner, la anchura de la Bollinger, el ATR, la desviación típica y las velas interiores y exteriores.
- **Reversión** — apuesta a que un precio estirado vuelve. Repertorio: RSI, Stochastic, CCI, Williams %R, DeMarker, MFI, RVI, las bandas de Bollinger y su línea central, SMA y EMA, y los patrones de envolvente y de pin bar. Dos de sus plantillas son para mercados laterales y lo dicen: *Range Fade (Bollinger + ADX gate)* compra la banda de abajo y vende la de arriba, y *Range Fade (Stochastic + ADX gate)* va contra el Stochastic cuando vuelve de 20 y de 80 — las dos sólo mientras el ADX(14) esté por debajo de 20. La puerta es la mitad de cada una: ir contra un nivel paga en un mercado lateral y pierde en cada vela de una tendencia, y un ADX por debajo de 20 es lo que parece un mercado lateral escrito como número.
- **Acción del precio** — la estructura de la vela, y ningún oscilador. Repertorio: apertura, máximo, mínimo, cierre, cuerpo, rango, el cierre anterior, el máximo, el mínimo y el cierre del día anterior, y los patrones de vela — envolvente, pin bar, vela interior y exterior, doji. Los patrones son proporciones del propio rango de la vela, así que la misma regla vale en un par de divisas, en el oro o en un perpetuo.
- **Impulso** — la velocidad a la que cambia el precio. Repertorio: ROC, MACD con su señal y su histograma, los osciladores Awesome y Accelerator, TRIX, RSI y EMA.
- **Volatilidad** — expansión y contracción. Repertorio: ATR, desviación típica, anchura de la Bollinger, bandas de Keltner, volatilidad de Chaikin, rango y cuerpo.
- **Divergencia** — el precio y su oscilador discrepan. Repertorio: el cierre, el máximo y el mínimo, y tres osciladores — RSI, MACD y CCI. Se elige *uno* de ellos: que dos osciladores discrepen entre sí no dice nada del precio. Plantillas: RSI (14, 5 velas atrás), CCI (14, 13 velas atrás), MACD (12/26, 8 velas atrás), y una divergencia *oculta* sobre el RSI (14, 8 velas atrás), que es el espejo de la clásica y significa lo contrario — una continuación, no un giro.

> **Una divergencia es siempre entera.** Una condición es media divergencia. Una entera necesita, en el mismo bloque de entrada y unidas con «Y»: una condición que compare el precio —cierre, apertura, máximo o mínimo— consigo mismo unas velas atrás; una condición que compare el oscilador consigo mismo *el mismo número de velas atrás*; y las dos apuntando a lados contrarios — el precio más alto y el oscilador más bajo, o al revés. Con «O» el bloque se dispara con cualquiera de las dos mitades por su cuenta; con dos distancias distintas no hay discrepancia, sino dos frases sobre dos tramos de mercado; y con la misma dirección es impulso confirmado, lo contrario de lo que la familia promete. Cualquier otra cosa dentro del bloque —un filtro de nivel como «y el RSI venía de sobreventa»— está bien. La búsqueda genética no entrega nunca media divergencia: a lo que críe con media, se la completa.

> **Por qué no hay una octava familia.** La clasificación vieja tenía una llamada Rango / Carry / Grid. Operar en rango es lo que ya hace Reversión. Un grid necesita varias posiciones abiertas a la vez, y el motor sostiene una —sólo abre cuando está plano—, así que ofrecerlo sería prometer algo que no corre. El carry es un coste, el tipo de financiación, no una señal: no hay ninguna regla que escribir sobre él. Lo que merecía la pena guardar es el filtro de régimen, y está en las dos plantillas de rango de ahí arriba.

Estrategias que escribió otro

Debajo de las familias y las plantillas del modo **Guiado**, un estante con dieciséis puntos de partida más. Dieciséis fichas sacadas de fuentes reales —libros (Murphy, Elder, Schwager, Connors), artículos, vídeos y publicaciones—, cada una con su autor, su cita, su año, el instrumento y la temporalidad para los que se escribió, en qué se supone que está su ventaja, su propia lista de avisos y lo que dijo de ella el propio Reality Check de BotFactoryFX en el S&P 500 diario y en EURUSD diario y de cuatro horas. Se ordenan por ese veredicto: de las quince con que empezó el estante ninguna salió Solid —la mejor, el filtro ADX de Murphy en el S&P 500, 71 Fragile—, y la del retroceso con RSI de 2 periodos de Connors y Alvarez, añadida después, sí: 88 en el S&P 500 diario. Es lo que dijo el juez entonces sobre esa historia, no una promesa: júzgala otra vez sobre la tuya. Se buscan por título, autor, cita o palabra suelta, se filtran por fuente o por familia, y se ordenan también por estructura —una lista de stop, objetivo y riesgo, no un backtest—, título o fuente. Sólo las familias que tienen fichas de verdad salen en los filtros, así que ningún filtro puede dejar la lista vacía.

Usar estas reglas mete las reglas, la salida y el riesgo de la ficha en la estrategia que estás editando, le deja su símbolo, su temporalidad y su saldo inicial, y la pantalla no se mueve: las reglas caen en el editor de abajo. Son reglas nuevas, así que la cuenta de variantes que llevaran las anteriores se tira en vez de heredarse — ver *Reality Check*.

Descríbela con palabras

Encima de las series de referencia, una caja para escribir la idea como la dirías —«compra cuando el RSI de 14 cruce 30 hacia arriba y el precio esté sobre la media de 200; stop a 2 ATR y objetivo a 4»— y **Escribirla con la IA**. Dr. Quant, con tu clave, la monta con los indicadores y operadores del programa y nada más, y las reglas caen en el editor de abajo, sobre la misma estrategia —se puede deshacer— y con su símbolo, su temporalidad y su saldo. Lo que la idea pida y el programa no tenga —otro indicador, un filtro de noticias, un segundo instrumento— lo monta lo más parecido posible y te lo dice debajo, una frase por cosa; nunca en silencio. Se lee como un JSON importado: lo que no sea del modelo del programa no entra. No se prueba sola: la lees y pulsas **Backtest**. Y **cada versión que pruebes cuenta como un intento para el juez** —con un backtest o en el Reality Check, una vez por versión; retocar un número a mano no suma—, porque pedirle a una IA una versión tras otra hasta que una sale bien es buscar, aunque sea a mano. Una sola versión probada no lleva cuenta, como una estrategia escrita por ti. No está en la edición de navegador.

La nota de 0 a 100 no dice nada del dinero. Comprueba que el riesgo no sea temerario: que haya stop, que haya objetivo, que el beneficio sobre riesgo sea razonable, que el riesgo por operación sea conservador, y que aquello no sea una martingala ni un grid sin stop. Es una comprobación de sensatez estructural, y por eso casi todas las fichas seleccionadas sacan nota alta — las escribió gente que le pone stop a las cosas. Si alguna tiene ventaja sobre tu histórico es para lo que están *Explorar* y, después, el Reality Check.

En este estante nada se ordena por lo que ganó, a propósito. Ordenar propuestas por su resultado sobre el histórico que da la casualidad de que tienes cargado es una máquina de sobreajuste: te daría la que mejor encaja en tus datos, que es la definición del problema que el Reality Check existe para cazar. Y nada de lo que hay en él se inventó de la nada — combinar indicadores al azar y puntuarlos es lo que hace la búsqueda genética, muchísimo mejor y con sus muestras separadas.

Cada condición tiene la forma `indicador OP operando`. Operadores: >, <, ≥, ≤, =, cruza hacia arriba y cruza hacia abajo — y sube, baja, gira al alza y gira a la baja, que sólo leen el lado izquierdo.

Además de los indicadores, una condición puede leer el reloj: la **Hora** a la que empieza la vela, de 0 a 23, y el **Día de la semana**, del 1 el lunes al 7 el domingo, en el reloj del histórico. «La hora no está por debajo de 8» y «la hora está por debajo de 17» es una ventana; «el día no está por encima de 4», de lunes a jueves. Una ventana que cruce la medianoche no cabe en un bloque «Y» —son dos condiciones unidas por «O»—: para ésa está el horario de la gestión monetaria. Los seis robots exportados los leen igual que el backtest.

Comprobador de redundancias en vivo. La pestaña Construir busca condiciones que se solapan: indicadores repetidos, periodos casi iguales (por ejemplo SMA(20) y SMA(22)), comparaciones de un indicador consigo mismo (SMA(20) > SMA(20), que es falsa siempre) o RSI y ROC mezclados (los dos son osciladores de impulso). Las condiciones señaladas se pintan en ámbar.

Mover un número a mano

Aquí es donde se escriben los números de una estrategia, y el editor los cubre todos: el **periodo de un indicador** se cambia en la regla que lo usa, al lado del propio indicador, y el **stop y el objetivo** en *Gestión monetaria*, plegada al pie de Construir. Cambia uno y pulsa Backtest.

> **Tres indicadores guardan varios números en uno.** MACD, Bollinger y Keltner se escriben empaquetados —1226 es un MACD de 12 y 26; 2020 son unas Bollinger de 20 velas y 2,0 desviaciones — y el editor tiene UNA casilla para ese número, con sus partes escritas debajo. El Reality Check, en cambio, nombra las partes por separado («MACD(12, 26) · EMA rápida»), así que para mover la que te señala tienes que componer el número empaquetado tú: rápida × 100 + lenta, periodo × 100 + desviación × 10.

Nadie te va a decir qué valor escribir, y es a propósito. Hasta el 25-09-2026 hubo una pantalla de *Optimizar* que barría un número y ofrecía el centro de la banda en la que aguantaba; se retiró porque esa banda se mide sobre el mismo trozo con el que se construyó la estrategia, y un valor elegido ahí es un valor elegido a toro pasado. Lo que sí puedes averiguar es cuánto aguanta el número que elegiste sobre histórico que no vio: juzga la estrategia, y la fila de **sensibilidad** lo desglosa número a número.

Construir › Búsqueda genética

Evolución genética de estrategias. Cada **individuo** es una estrategia completa. La población compite por una **aptitud** y evoluciona por selección de torneo, cruce, mutación y elitismo. Trabaja sobre el histórico que elijas en *Instrumento*, al preparar la búsqueda; una tira de arriba lo nombra, con **Bajar otro en Datos** al lado. Tres pestañas:

- **Buscar** — la búsqueda genética propiamente dicha, con la **gestión monetaria** al pie.
- **Rueda** — busca, juzga y vuelve a dar la vuelta ella sola. Más abajo.
- **Población** — todas las estrategias que salieron de las búsquedas, con sus resultados.

Cómo se trabaja:

1. Carga al menos 300 velas en **Datos**. Las series de referencia que sus reglas puedan leer se añaden aquí, en **Buscar** —y en la **Rueda**—: la tarjeta *Series de referencia*, debajo del histórico, las elige de la biblioteca. Por debajo de 1.500 avisa de que el histórico es corto.
2. En **Buscar**, elige cuánto insistir en la búsqueda —**Un vistazo**, **Equilibrado**, **A fondo** o **Profunda**—, el tramo de histórico que la búsqueda puede ver, la dirección, si sólo operas en un sentido, y las **Familias** que puede escribir (más abajo).
3. Mira la caja de **Calculado por ti**: los rangos del stop y del objetivo, el coste de ida y vuelta y cuánto lo supera el objetivo más pequeño, el número mínimo de operaciones, cuántos backtests va a correr la búsqueda y aproximadamente cuánto tarda eso.
4. Pulsa **Arrancar la búsqueda**.
5. Trae el mejor resultado con **Importar al editor**. Llega a Construir con la cuenta de juegos de reglas que la búsqueda probó, que es lo que se le dice al Reality Check cuando lo juzga. Luego pon el riesgo en la sección de **Gestión monetaria**, al pie de *Buscar*. El código sale de la **Biblioteca**, una vez guardada.

Evitar mi cartera, al lado de *Sembrar con la actual*: encendido, antes de arrancar se corren las estrategias de tu última cartera guardada, cada una sobre su histórico, y cada candidata se compara con cada una igual que las compara la pantalla de *Cartera*. La que se parezca 0,7 o más a alguna pierde: cae por debajo de cualquier otra, y cuanto más se parece, más abajo. No se descarta sin mirarla: se mide y cuenta como probada, así que le sube el listón al juez como las demás. Al terminar se dice cuántas cayeron por eso. Sin cartera guardada, o sin el histórico de ninguna de sus estrategias, no arranca y lo dice.

Filtros de hora y día, en *Aptitud y restricciones*, junto a *Stop y objetivo en ATR*: encendido, alrededor de una de cada ocho condiciones que sortea la búsqueda es de reloj —«la vela empieza a las 8 o después», «de lunes a jueves»—, sobre la hora y el día de la semana de la vela en el reloj del histórico (ver *Construir › Manual*). Nunca «a las 13 en punto», que en un H4 no se cumple nunca. Viene apagado, como las piezas de hora y día de StrategyQuant X: un filtro de reloj es la forma más barata de encajar el ruido de un histórico —«los martes a las 14:00 ganaba»—, y cada uno que prueba es una variante más que el juez cuenta.

> **Sobreajuste.** Un porcentaje de aciertos por encima del 85 % con pocas operaciones suele ser ruido. La búsqueda ya exige un número mínimo de operaciones calculado a partir de tu histórico —entre 20 y 120— y se guarda el último tramo sin mirarlo. Juzga lo que encuentre en el Reality Check en vez de fiarte de su aptitud: la aptitud es lo que la búsqueda optimizó, así que lo primero de la lista es el resultado con más papeletas de estar sobreajustado.

Lo que puede distinguir el tramo del juez. Debajo de los tramos, en *Lo que puede ver la búsqueda*, el programa dice cuántas operaciones necesita el tramo del juez para distinguir un factor de beneficio de 1,2 de uno de 1,0 —

unas 750, con ganadoras y perdedoras del mismo tamaño— y uno de 1,5 —unas 150—, y cada cuántas velas tendría que operar una estrategia para llegar ahí con el tramo que has colocado. Por debajo, una ventaja modesta y una casualidad se parecen demasiado para que ningún juez las separe: es mejor saberlo antes de gastar la búsqueda.

Probar el procedimiento

Al pie de *Buscar*, antes de la gestión monetaria. El walk-forward del Reality Check reajusta el stop y el objetivo de *una* estrategia; éste repite la **búsqueda entera**, con los ajustes de arriba, una vez por ventana: busca en un tramo del histórico y opera lo que encontró en el tramo siguiente, que la búsqueda nunca vio. Es lo que harías tú operando en serio —buscar, operar un tiempo, volver a buscar— hecho sobre el pasado. **Rodante**: el tramo de búsqueda tiene siempre el mismo largo, cuatro veces el de fuera, y avanza. **Anclado**: empieza siempre al principio y crece. De tres a ocho ventanas; cinco de fábrica.

Por ventana dice lo que lo encontrado ganó dentro y fuera, por año, y la **eficiencia**: lo de fuera entre lo de dentro. El procedimiento **se sostiene** si la eficiencia mediana llega al 50 %, si ganan fuera al menos seis de cada diez ventanas y si todas las operaciones de fuera juntas dan un PF por encima de 1. Con menos de tres ventanas que operen fuera no dice nada. Juzga los ajustes, no una estrategia, y no guarda nada; cuesta tantas búsquedas como ventanas.

La misma prueba se ofrece al final del piloto, marcándola en su ficha, y debajo del veredicto del Reality Check para una estrategia que salió de una búsqueda, que repite la búsqueda que la encontró con los ajustes con que corrió.

Las familias, y el cóctel

Las siete familias —Tendencia, Ruptura, Reversión, Acción del precio, Impulso, Volatilidad y Divergencia, contadas en *Construir › Manual*— son una fila de fichas en la pestaña *Buscar*, siempre a la vista. Son lo más fuerte que se dice ahí: la búsqueda saca de las familias que marques sus indicadores, sus periodos y las plantillas con las que siembra alrededor de la mitad de su primera generación. Marca una y la línea de debajo describe la familia. Desmárcalas todas y la búsqueda no puede arrancar: sin familia no hay de dónde sortear.

Marca varias y se mezclan — desde una hasta las siete. Se unen los repertorios de indicadores, los de periodos y las plantillas, y la siembra sortea primero una familia y luego una de sus plantillas, así que cada familia marcada se lleva la misma parte de la mitad sembrada por muchas o pocas plantillas que tenga escritas. Así sale una híbrida: una ruptura con filtro de tendencia es una estrategia que ninguna de las dos familias sabe escribir sola, porque los niveles del día anterior no están en el repertorio de tendencia y el ADX no está en el de ruptura. Mezclar hace posibles las híbridas, no obligatorias — la búsqueda se queda con lo que mejor puntúe, y una estrategia que use los indicadores de una sola familia es tan bienvenida como una que los mezcle. La línea de debajo de las fichas dice cuánto creció el repertorio y lo que eso cuesta: un repertorio más ancho prueba más variantes distintas, y el Reality Check exige más evidencia fuera de muestra cuantas más candidatas se hayan probado.

Una divergencia es siempre entera, sola o mezclada. En cuanto una estrategia empieza a escribir una, la búsqueda se la completa (ver *Divergencia*, en *Construir › Manual*) en vez de dejar pasar media: media divergencia opera más a menudo que una entera, y sin esto la búsqueda se quedaba con la media. En una mezcla, una estrategia que no lleve nada de la familia de divergencia se queda exactamente como está. Marcar Divergencia sube además *Condiciones máx. / bloque* a dos si estaba en una, porque una divergencia entera necesita dos.

El esfuerzo **Profunda** ensancha el repertorio de indicadores más allá del de las familias marcadas; la siembra sigue saliendo de sus plantillas. En *Ajustes avanzados* las mismas fichas están al lado de un interruptor **Guiado** / **Aleatorio**: el aleatorio no se ciñe a ninguna familia y no siembra con plantillas.

> **Antes de las fichas, la familia no llegaba a la búsqueda.** En versiones anteriores la familia sólo se podía elegir dentro de *Ajustes avanzados*, y una búsqueda arrancada desde el panel de preparación corría siempre la de por omisión, Tendencia, se hubiera elegido allí lo que se hubiera elegido. Una estrategia encontrada así salía del repertorio de Tendencia.

La rueda

La tercera pestaña. Hace el bucle que harías tú a mano — buscar, leer el veredicto, cambiar algo y volver a buscar. Cada vuelta coge el siguiente histórico que hayas marcado y las familias que hayas marcado —de una en una o todas juntas, ver más abajo; sin ninguna marcada valen las siete—, corre una búsqueda genética y juzga el mejor resultado con las trece comprobaciones del Reality Check.

Las series de referencia también entran. La misma tarjeta *Series de referencia* de *Buscar* está en la pestaña de la rueda, y cada vuelta busca y juzga con las series que haya registradas al arrancar: las reglas que encuentre pueden leerlas, y el juez las mira como en el Reality Check —si una estrategia depende de verdad de otra serie, la comprobación en otros instrumentos queda no disponible, con su peso repartido—.

Una familia por vuelta, o todas juntas. Lo decide el interruptor de al lado de las fichas de familia. Rotando —lo de por omisión— cada vuelta es una búsqueda estrecha, y sabes de qué familia salió lo que encuentre. Mezclando corre el cóctel entero en cada vuelta: cada vuelta es más ancha y puede escribir reglas que ninguna familia sola escribe, porque los repertorios de indicadores se unen. Lo que cambia entonces de una vuelta a otra es el histórico y la semilla, no la forma de la estrategia. Ninguna de las dos es mejor; la búsqueda más ancha prueba más variantes distintas, y el Reality Check exige más evidencia fuera de muestra cuantas más se hayan probado.

Rotando, el histórico y la familia avanzan juntos, un paso cada uno por vuelta, así que la pareja se repite al llegar al mínimo común múltiplo de las dos cuentas — históricos × familias sólo cuando no comparten divisor. Con las siete marcadas no lo comparten salvo que marques un múltiplo de siete históricos —siete es primo—, así que cada histórico acaba viendo todas las familias; lo que pasa es que hacen falta históricos × 7 vueltas para llegar ahí, que con tres históricos o más queda por encima del límite de 20 de por omisión. Marca siete históricos, o catorce, y cada uno verá siempre una sola familia. Marca menos familias y el ciclo mide las que hayas marcado, y ese número sí puede compartir divisor con el de históricos: cuatro familias con dos históricos se repite cada cuatro vueltas, y cada histórico sólo ve dos de ellas. Todas las familias que marcaste se prueban igual; lo que no está garantizado es probar cada una en cada instrumento. Marca menos históricos si quieres eso dentro de una sola corrida.

Para en la primera estrategia que salga *sólida* — o también *frágil*, si desmarcas **Parar sólo por «sólida»**, que pone la meta donde el ruido llega. Si no, para al llegar al límite de vueltas (20 de por omisión) o cuando haya pasado una hora. La hora sólo se mira antes de empezar otra vuelta, así que la que está en marcha termina siempre: sobre un histórico grande, o con esfuerzo *Profunda*, esa vuelta sola puede tardar mucho más que la hora.

Los juegos de reglas de cada vuelta se suman a una sola cuenta corrida, y al veredicto de cada vuelta se le da la cuenta de hasta ahí. Cuantos más juegos de reglas se hayan probado, más probable es que el mejor lo sea por suerte, así que más alto tiene que puntuar una estrategia en las comprobaciones que miran fuera de su propio histórico para que su nota se dé por buena. La cuenta viaja con la estrategia: importada a Construir desde *Población*, o mandada desde ahí al Reality Check, una candidata de la rueda lleva lo que probó la rueda *entera* — más de lo que se le dijo al veredicto de su propia vuelta, si vinieron vueltas después, porque sacarla de toda la cosecha es sacarla de todo lo que la rueda probó. Así que volver a juzgarla no afloja esa exigencia; si acaso la aprieta. Guardada en la Biblioteca directamente desde Población, se lleva también su cuenta, que queda guardada con ella.

Todo lo que juzga va a *Población*, marcado *De la rueda*, sea cual sea su veredicto; las estrategias que hacen exactamente las mismas operaciones comparten una sola ficha. Se guardan sólo mientras la aplicación esté

abierta, y nada llega solo a la Biblioteca. **Limpiar**, en la pestaña de la rueda, o arrancar otra corrida, las quita.

> **Parar sin encontrar nada es un resultado — cuando encontrar algo era posible.** Significa que ninguna vuelta dio una estrategia que el Reality Check llame *sólida* (o *frágil*, si desmarcaste la casilla). El resumen da la mejor nota a la que se llegó, y lo que quedó en *frágil* sigue en Población. Si no se probó ni un solo juego de reglas —todos los históricos marcados demasiado cortos, demasiado planos, demasiado grandes, o con costes mayores que su movimiento típico—, lo dice, y eso no dice nada del mercado.

> **Marca al menos dos símbolos distintos.** Rota sobre los históricos que marques, no sobre temporalidades, y la comprobación en otros instrumentos necesita otro símbolo: dos temporalidades del mismo instrumento cuentan como un solo mercado. Con un solo símbolo marcado esa comprobación saca cero, nada puede salir *sólida* y, con la casilla marcada, la rueda no puede encontrar nada — así que tampoco te va a decir nada del mercado.

Construir › Explorar

La tercera pestaña de **Construir**, y la única de las tres que no termina en una estrategia. Es donde se mira una idea antes de casarse con ella: un backtest que se rehace según mueves un número.

Laboratorio — qué cambia cuando cambia un número

La estrategia que tengas abierta en Construir, dibujada sobre el histórico que tengas cargado: las velas con cada entrada y cada salida marcadas, la curva de capital debajo, la lista entera de operaciones y las métricas al lado. Mover cualquier mando de la derecha rehace el backtest sobre la serie entera —con la horquilla, la comisión y el deslizamiento del propio instrumento, y con su ficha—, así que las cifras de aquí son las mismas que daría el resto del programa para la misma estrategia.

Las reglas se quedan donde están. Lo que los mandos mueven es todo lo de alrededor: la vela sobre la que se lee la estrategia, el stop y el objetivo con el beneficio sobre riesgo calculado al lado, cuánto apostar (lote fijo, un porcentaje de la cuenta, o Kelly), el riesgo por operación, el lote, las posiciones abiertas máximas, el saldo inicial, el stop dinámico en pips o en ATR, el punto de equilibrio y los dos topes diarios. Para cambiar una condición hace falta *Manual*, y **Enviar a Manual** lleva ahí la estrategia con lo que hayas dejado puesto.

Se rehace más o menos una décima de segundo después de que sueltes el mando. **Auto** lo apaga, y **Ejecutar** lo hace cuando tú digas. El backtest corre fuera del hilo principal, así que arrastrar un deslizador no congela la pantalla, y una pasada lenta de hace tres cambios no puede caer encima de los números que estás mirando ahora.

> **La vela es un mando más, y es el que más efecto tiene de todos.** Una EMA de 20 velas son veinte horas en H1 y veinte minutos en M1 — no es la misma estrategia medida más fino, es otra estrategia.

Lo que el Laboratorio no hace

No barre un número, no dice si alguno se asienta en una meseta, y nada de lo que enseña llega nunca al juez. Lo que queda aquí es la pregunta, al lado de los mandos donde se mueven a mano esos mismos números. El barrido vivió aquí, luego se fue a una pantalla propia, y esa pantalla se retiró el 25-09-2026: medir la forma de un número sobre el tramo de entrenamiento no demuestra nada. El **Reality Check** hace la misma pregunta donde cuenta, sobre el tramo retenido, y su fila de sensibilidad desglosa la respuesta número a número.

Tampoco guarda nada. Desde aquí no llega nada a la Biblioteca, no se juzga nada, y un backtest no es un veredicto — es justo la respuesta de la que el Reality Check existe para dudar.

Moverse por el gráfico

El gráfico del Laboratorio es el mismo que sale en **En vivo** y en **Cuarentena**, y se maneja igual en los tres. Tiene la serie entera, que pueden ser decenas de miles de velas. Para recorrerla:

- **Arrastra** el gráfico a izquierda o a derecha.
- **Mayús + rueda** desplaza por el tiempo. La rueda sola hace zoom.
- Los **botones** de encima del gráfico: una pantalla atrás, una pantalla adelante, saltar a la vela más antigua, saltar a la última, acercar, alejar y encajar la serie entera.
- Las **flechas del teclado**, una vez has hecho clic en el gráfico — ← y → desplazan, Re Pág y Av Pág mueven una pantalla entera, Inicio y Fin van a los extremos, y + y − hacen zoom.

Las fechas que tienes en pantalla se escriben al lado de los botones, junto a la temporalidad, así que siempre sabes qué estás mirando.

Velas, barras, línea o área

Las velas japonesas son lo que sale por defecto y la vista recomendada — son la única de las cuatro que enseña a la vez la apertura, el cierre y los dos extremos de cada vela. Las otras tres están a un clic, encima del gráfico.

JUZGAR

Reality Check

La pantalla que da nombre al producto. No pregunta si una estrategia ganó dinero —eso ya lo contestó el backtest—: pregunta **si esa respuesta es real**. Cuatro pestañas arriba: una estrategia, varias a la vez, una que te ha mandado alguien, y cómo juzga.

Una estrategia mía

Trece comprobaciones sobre la estrategia elegida en la barra de arriba: la que está abierta en Construir, o cualquiera de la Biblioteca, para que una búsqueda que dejó cuarenta estrategias guardadas se pueda juzgar sin abrirlas una a una. Corren fuera del hilo principal porque trece comprobaciones sobre un histórico de verdad son bastante más de cien backtests — unos 150 con los valores de fábrica.

Qué preguntan: cuánto se degrada el resultado fuera de muestra; si le gana a ese mismo histórico no visto con la dirección de cada vela sorteada —el mismo tamaño y el mismo sitio, subiendo o bajando al azar alrededor de su tendencia de fondo: así las rachas de volatilidad se quedan y lo único que desaparece es la dirección—; si aguanta reajustar y caminar hacia delante; si las mismas reglas funcionan en **otros instrumentos** —los que elijas en *Cómo juzga*, y si no, los que estén cargados con más histórico una vez reagrupados a su temporalidad, con sus stops en pips llevados a la vela típica de cada uno, porque un pip no mide lo mismo en todos—; si funcionan en **otras temporalidades**, más lentas y más rápidas; cuánto le afectan sus propios parámetros; si aguanta que ese tramo empiece en otro sitio con los precios agitados —cuarenta veces, contando su peor décima—; el remuestreo de Monte Carlo —de tres maneras: operación a operación, por bloques de operaciones seguidas, que respeta las rachas, y saltando una de cada diez; cuenta la peor—, y si sin su 5 % de mejores operaciones seguiría ganando, porque si no vive de un puñado y su nota se parte por la mitad; si hay operaciones suficientes para decir algo; la consistencia entre periodos; qué pinta tiene el drawdown contando las posiciones abiertas y no sólo las cerradas; si la resolución de vela esconde algo; y cuánto coste de bróker puede absorber la ventaja antes de desaparecer.

> **Nueve de las trece están para medir lo que una búsqueda no pudo ajustar**, porque una comprobación hecha sobre las mismas velas a las que se ajustó la búsqueda no distingue una ventaja real de una ajustada, por lista que sea la prueba. Siete corren sobre el tramo del juez: el final del histórico que ni la búsqueda ni la selección vieron. Las otras dos se salen de ese histórico: la de otros instrumentos corre sobre otros ficheros —contando sólo las operaciones que abren en el tramo del juez, porque dos mercados que se mueven juntos comparten los años a los que se ajustó la estrategia—, y la de otras temporalidades corta los precios en velas distintas. La consistencia por periodos, la tolerancia al coste y la resolución de vela miran el fichero entero sobre sus propias velas: entre las tres, nueve puntos de los cien. El drawdown también lo mira, pero desde el 28-09-2026 no entra en la nota: mide cómo se gestiona el riesgo, no si hay ventaja, y se enseña como aviso.

> **La ventana de la búsqueda tiene tres tramos, y el juez sólo mira el último**. De fábrica, la búsqueda aprende con el primer 60 %, Población y la rueda eligen con el 20 % siguiente —la selección—, y el último 20 % no lo ve nadie hasta que juzga el juez. El piloto ya no elige con la selección: juzga a todas sus finalistas en el último tramo, y la selección se queda sin mirar, de separación entre lo que aprendió y lo que se juzga. Se cambió el 17-10-2026, después de medir en el duelo con StrategyQuant que esa puerta elegía peor que la media y tiraba justo las que el juez veía sólidas. Una estrategia que sale de una búsqueda lleva consigo desde dónde tiene que juzgarse, y el juez corta ahí, muevas o no su deslizador. Lo que el juez no puede saber es lo que hiciste por fuera: si encontraste la estrategia en otro fichero o retocaste sus reglas a mano contra el gráfico entero, ya ha visto ese tramo, y esas ocho comprobaciones están corrigiendo un examen que pudo copiar. Juzga una estrategia sobre histórico con el que no se construyó.

Hasta dónde mueve un número la comprobación de sensibilidad tiene tope, y para el stop y el objetivo lo pone el instrumento y no una cifra fija de pips: el 4 % del precio de la propia serie, dicho en pips de ESE instrumento, o los 1.000 y 2.000 pips de siempre cuando son mayores. En divisas no muerde nunca, porque el precio entero de un par son diez o veinte mil pips; en el oro o en un perpetuo es la diferencia entre probar vecinos alrededor del stop que tiene la estrategia y probarlos alrededor de uno recortado. Una estrategia cuyo stop ya pasa los dos techos trae el suyo. Los mismos topes acotan la rejilla del walk-forward, así que las dos comprobaciones mueven un número la misma distancia.

En qué otras temporalidades. De fábrica, el salto habitual hacia abajo y el habitual hacia arriba en la escalera de siempre —M1, M5, M15, H1, H4, D1, W1—: de H1, M15 y H4; de H4, H1 y D1; de M15, M5 y H1. En la pestaña *Cómo juzga*, junto a los instrumentos de contraste, puedes elegir otras, M30 y H2 incluidas. Elígelas antes de juzgar: lo que elijas queda escrito en el informe y en el veredicto guardado, y la ficha dice en qué relojes se probó —quitar relojes hasta que la estrategia apruebe es ajustar el veredicto, y así se ve—. Si las reglas leen otra serie, un reloj que cambie su relación con ese filtro no se prueba, y se dice.

Las temporalidades más rápidas piden datos más finos. Las más lentas se reconstruyen con las velas que estás mirando: de H1 sale H4. Las más rápidas no se pueden inventar —una hora de velas no contiene sus cuartos de hora—, así que sólo se prueban cuando el fichero que importaste es más fino y la temporalidad pedida sale de él en velas enteras: importa M5 y mira H1, y la comprobación mira M15. En los ajustes, las que no se pueden construir salen apagadas; y cuando una de las elegidas no se pudo construir, la comprobación lo dice y dice qué importar

El veredicto

Una nota sobre 100 y una palabra: **Sólida** desde 75, **Frágil** desde 45, **Débil** desde 35, **No creíble** por debajo — o **No hay pruebas suficientes**, sin nota ninguna, cuando hay demasiado pocas operaciones para juzgar. La acompañan dos cosas que importan tanto como el número: las comprobaciones que no pudieron correr, en gris en la lista y cada una con su motivo; y un recuadro de razones para leer las cifras como indicativas y no como exactas. Aprobar con tres comprobaciones sin hacer no es lo mismo que aprobar.

La nota es la media ponderada de las comprobaciones que corrieron, y después se le puede poner **techo**. Cuando se le pone, el informe lo dice y da la nota que habría tenido sin él.

- **No se probó nada fuera de este histórico.** Sólo dos comprobaciones preguntan si la ventaja existe fuera de la serie sobre la que se construyó: otros instrumentos y otras temporalidades. Todo lo demás vuelve a medir esa única serie, y una serie se puede ajustar. Sin ninguna de las dos, el veredicto no pasa de *débil*; con una, no pasa de *frágil*. *Sólida* pide las dos.
- **Demasiado poca nota se apoya en histórico no visto.** Cuando menos del 70 % del peso que contó sale de las nueve comprobaciones que miden sobre histórico que la estrategia nunca vio, la nota no pasa de 70; por debajo del 55 %, no pasa de 50.
- **La búsqueda que la encontró fue grande.** Al juez se le dice cuántas variantes se probaron para llegar a la estrategia, y cuantas más fueron, más probable es que la mejor lo sea por suerte, así que más alto tiene que puntuar en las comprobaciones que miran fuera de su propio histórico. La rueda y el piloto se lo dicen para sus propios veredictos. En esta pantalla se lo dicen para la estrategia abierta en Construir cuando salió de una búsqueda genética —importada de la búsqueda genética o de *Población*, donde una candidata de la rueda trae lo que probó la rueda entera—; para una estrategia elegida de la Biblioteca, cuando sus reglas de entrada y de salida, sus ajustes de riesgo y su gestión son los mismos que los de la estrategia abierta en Construir que lleva cuenta —guardarla no la lava—; y, en *Varias a la vez*, para cada estrategia mandada desde Población. Una línea encima del veredicto dice de cuántas variantes le hablaron al juez. La cuenta además se **guarda con la estrategia**: al guardarla en la Biblioteca —desde Construir o desde Población— y al guardarse un veredicto que la llevaba. Sólo sube, así que rejuzgarla otro día, o abrirla en Construir, no la pierde: el juez usa la mayor entre la de la sesión y la guardada. Y elegir en la Biblioteca con un **filtro por cifras** puesto también se cobra: la

elegida pasa a llevar la suma de las cuentas de todas las que el filtro comparó. Una estrategia escrita a mano, sin nada de eso, no lleva ninguna, y este techo no se le aplica. Una que escribió la IA desde *Descríbela con palabras*, en Manual, lleva una por cada versión que probaste —con un backtest o aquí—, y la línea de encima del veredicto lo dice así: buscar a mano con una IA sigue siendo buscar.

La cuenta sale siempre de una búsqueda genética, y el informe lo dice: se probaron ese número de *juegos de reglas distintos*, y ésta fue la mejor. Hasta el 25-09-2026 también podía salir de un barrido en una pantalla de *Optimizar*, que medía ese número de variantes de los mismos números; esa pantalla se retiró, así que hay una sola procedencia y una sola frase. Siempre que hay cuenta, el informe dice además qué nota saca una búsqueda de ese tamaño sobre precios construidos para no tener ninguna ventaja dentro, y así el número se puede leer contra la suerte.

Con la cuenta vienen además dos cifras de la estadística, que informan y no puntúan. El **Sharpe deflactado** es la probabilidad de que el Sharpe de la estrategia, en el tramo donde se eligió, supere al que daría la suerte a la mejor de tantas pruebas. Separa bien el ruido de una ventaja, pero sólo mira ese tramo: una ventaja que dejó de funcionar después, o un defecto de los datos, puntúa igual que una de verdad. La **probabilidad de sobreajuste** (PBO) la mide la búsqueda: parte su histórico en dos mitades de todas las maneras posibles y cuenta cuántas veces la mejor de una mitad quedó por debajo de la mediana en la otra. Cerca del 50 %, elegir la mejor fue elegir al azar; pero un valor bajo no es una ventaja: sobre precios sin ninguna sale en torno al 17 %, porque con costes las reglas que operan poco pierden menos en cualquier mitad y la mejor sigue arriba sin ganar nada.

La cuenta vive en esta sesión, no en el disco. Una estrategia guardada en la Biblioteca se sigue juzgando con ella mientras sus reglas sean aquellas a las que la cuenta pertenece: guardarla no la lava. Si la editas después de guardarla, la copia guardada deja de coincidir y se juzga sin cuenta; la que tienes abierta en Construir la conserva de todos modos. Cerrar el programa la pierde entera.

No cargar un segundo instrumento cuesta puntos. Mientras no haya un segundo símbolo cargado en velas al menos tan finas como las de la estrategia —se reagrupan a su temporalidad, y tienen que salir 200—, la comprobación de otros instrumentos puntúa cero: es una prueba que no se hizo, y dejarla fuera no puede hacer que una estrategia parezca mejor que si se hubiera hecho. Impórtalo en la pantalla de Datos. Las velas más gruesas no valen: las finas se reagrupan en gruesas, y al revés no hay nada que reagrupar.

Reglas que leen otra serie. Una condición puede leer una segunda serie: «comprar en H1 sólo si el diario sube». Una estrategia así no se puede llevar a otro instrumento tal cual: hacerlo obliga a decidir en qué se convierte allí esa serie diaria, y esa decisión cambiaría la estrategia que se está juzgando. Así que para ella esa comprobación queda *no disponible* en vez de cero, y cargar más instrumentos no lo cambia. La de otras temporalidades sí corre, pero se salta cualquier reloj que cambiaría cómo encaja el filtro con lo que filtra; si el filtro va en la misma temporalidad que la estrategia, no queda ningún reloj y esa comprobación también queda no disponible. El techo de arriba sigue en pie: sin pruebas de fuera de este instrumento no hay *sólida*, sea cual sea el motivo. Un filtro que no cambia ni una operación sobre este histórico no es un filtro: la comprobación de otros instrumentos juzga la estrategia como las reglas sin él, y la de otras temporalidades no se salta ningún reloj por su culpa — sigue corriendo la estrategia tal como está escrita, porque en un reloj más rápido ese mismo filtro puede empezar a decidir.

> Nada entre lo que dividir. Ocho de las trece comparan factores de beneficio —ganancia bruta entre pérdida bruta—, así que un tramo en el que la estrategia no perdió ni una sola operación las deja sin denominador. Eso no es un resultado perfecto: casi siempre quiere decir un stop que ese tramo no llegó a tocar, y una muestra que todavía no se ha encontrado con el riesgo que está corriendo. Una comprobación así dice «no medible» en vez de una cifra, no cuenta como prueba ni a favor ni en contra, y su peso se reparte entre las que sí se pudieron medir. Suspender sí puede: cuando las versiones agitadas o reajustadas no ganan dinero, esa respuesta está medida y vale. Lo que nunca puede es aprobar. Los techos de arriba siguen aplicándose, así que una estrategia que no ha perdido nunca no puede llegar a «sólida»: sin una comparación medida fuera de su propio histórico, esa palabra no está disponible. Una fila dice esas mismas dos palabras y sí cuenta: la del tamaño de muestra. Lo que pregunta es si el resultado medio se distingue de cero, así que cuando todas las operaciones del tramo no visto devolvieron lo mismo, la dispersión que falta es la respuesta — y esa fila puntúa cero. La lista lo dice debajo de ella.

Debajo del veredicto

¿Esto es una estrategia? Las trece comprobaciones dan por hecho que lo que se midió es una estrategia. Esto pregunta si lo es: si una sola operación sostiene todo el resultado, si un sentido paga lo que pierde el otro, si está dentro del mercado casi todo el rato, si casi todas las operaciones salen por el mismo sitio, y si casi nunca pierde — que sobre suficientes operaciones es mucho más veces un fallo que una ventaja. Corre solo, sobre las mismas velas y los mismos costes que el veredicto, y avisa cuando no hay nada que contar.

¿Encaja con lo que ya tienes? Corre tu última cartera guardada dos veces —sin esta estrategia y con ella, con el mismo dinero— y dice si el conjunto es mejor. Meter una estrategia le quita capital a las demás, así que tiene que ganarse la parte que les quita. Compara rendimiento por unidad de drawdown, rechaza una estrategia que pierde dinero por su cuenta (una así puede alisar la curva de una cartera y costarle dinero a la cuenta igualmente), y a una casi copia de algo que ya está dentro la llama copia. Son dos carteras enteras, así que espera a que lo pulses tú.

Probar el procedimiento que la encontró. Un botón, para las estrategias que salieron de una búsqueda: repite esa búsqueda —con los ajustes con que corrió, que desde esta versión guarda cada estrategia encontrada por una búsqueda— en ventanas de este histórico, cada una operada sobre el tramo que no vio. Es la misma prueba que Probar el procedimiento en Construir › Búsqueda genética y el último paso opcional del piloto. Pregunta si la forma en que se encontró la estrategia encuentra cosas que se sostienen, no si ésta se sostiene, así que informa y no cambia la nota. Cuesta varias búsquedas, y por eso espera a que se la pidan. Una escrita a mano, o encontrada antes de que el programa guardara cómo, no tiene procedimiento que repetir, y el panel lo dice.

Cuándo funciona y cuándo no. El factor de beneficio y el resultado por operación en cada régimen del mercado — volatilidad baja, media o alta según el ATR(14) frente al precio; en tendencia o sin ella según el ADX(14), con el corte en 25; la sesión; el año—, en los dos tramos del juez: donde se eligió la estrategia y el que la búsqueda no vio. El régimen de cada operación se lee en la vela anterior a la entrada, que es la última que un robot puede mirar, y los tercios de la volatilidad salen sólo del primer tramo, así que el tramo retenido no ajusta nada. Si un régimen perdió en los dos tramos, con al menos doce operaciones en cada uno y una pérdida que pesa, lo dice como una regla —«apágala cuando el ADX(14) esté por debajo de 25»— y cuánto habría quedado fuera de muestra sin esas operaciones. Si pierden todos, no propone ninguno: entonces no es el régimen, es la estrategia. Las sesiones van en UTC cuando se sabe el reloj del histórico —MetaTrader sella las velas en la hora de su servidor— y si no, en bloques de seis horas del propio fichero. No mueve la nota.

Fuera de muestra, con su margen. Debajo de las filas del veredicto, el factor de beneficio y el resultado medio del tramo que la búsqueda no vio, cada uno con el margen en que cae el 90 % de las veces si se sortean de nuevo sus operaciones: un 1,34 sobre treinta operaciones cabe entre algo que pierde y algo que gana mucho, y uno sobre tres mil no. Y la potencia de esa muestra: cuántas veces distinguiría un factor de beneficio de 1,2 de uno de 1,0, y

cuántas operaciones harían falta para hacerlo ocho de cada diez —con ganadoras y perdedoras del tamaño de las suyas—. Con treinta operaciones rara vez pasa del 15 %: es la razón de que el veredicto pida tantas pruebas antes de decir «sólida». No mueve la nota.

> **Los costes salen del instrumento.** El spread, la comisión y el slippage se apuntan una vez en la pantalla de Datos, y la búsqueda, el backtest y este veredicto usan esos mismos números. Una estrategia que sólo sobrevive a coste cero no es una estrategia: cambia los costes ahí y vuelve a correr esto.

Varias a la vez

Una búsqueda devuelve las estrategias ordenadas por el fitness para el que se criaron, lo que pone arriba a la más sobreajustada. Esta pestaña juzga una lista corta entera y la reordena por veredicto. Manda estrategias aquí desde *Construir › Población*: las que pasen tu listón, o de una en una; cada envío sustituye la lista.

Juzgarlas todas carga los instrumentos de contraste una vez y corre las trece comprobaciones sobre cada estrategia, varias a la vez, con los mismos costes, los mismos instrumentos, el mismo reparto y los mismos ajustes de medición que la comprobación de una sola. El resultado sale ordenado de mejor a peor: primero las que tienen nota, por nota, y a igualdad gana la que tiene menos comprobaciones sin hacer; después las que no tienen pruebas suficientes; y al final las que fallaron del todo. Abre una fila para leer su informe entero. **Parar** conserva y ordena lo que ya hubiera terminado.

Cada una se juzga con su propia cuenta. Aquí se suelen juzgar las ganadoras de una búsqueda genética, así que es donde más importa la cuenta: una estrategia de la búsqueda genética trae el número de juegos de reglas que probó, y una candidata de la rueda, el que probó la rueda entera. Ninguna toma prestada la cifra de su vecina. La cabecera de la lista lo dice: un número cuando todas comparten el mismo; cuando no, el rango, y cuántas no llevan ninguna.

> **Todas se juzgan sobre los datos cargados ahora.** Una estrategia construida sobre otro símbolo u otra temporalidad se juzga igualmente sobre estas velas; la pestaña lo avisa encima, con cuántas son y cuáles.

Una de fuera

Alguien te ha mandado un robot, o una captura de un backtest que pinta demasiado bien. Adjunta aquí su fichero: el **código** (.mq4, .mq5, .java, .pine) y/o el **informe** que exportó su plataforma. Sacas lo que se pueda leer del código —símbolo, temporalidad, lote, stop y objetivo cuando están escritos como asignaciones llanas, los inputs de MQL y las líneas `#property`, los campos `@Configurable` de JForex— y lo que dice el informe, recalculado desde su propia lista de operaciones en vez de copiado de su página de resumen. El código se lee por patrones, así que «no lo declara» puede querer decir «no lo he encontrado»: los inputs de Pine, por ejemplo, no se leen.

Si adjuntaste su informe y se pudieron leer operaciones, pulsa **Ejecutar la mía sobre este histórico**: la estrategia elegida en la barra de arriba se mide sobre tu histórico cargado y a los costes de tu bróker, y las dos caen en una misma tabla. Con sólo el código no hay contra qué comparar, y no sale ninguna tabla.

> **El código se lee, nunca se ejecuta.** BotFactoryFX no puede correr MQL, Java ni Pine. Las dos columnas son dos preguntas distintas —su plataforma sobre su histórico, la tuya sobre el tuyo— y sólo se pueden comparar si las dos cubrieron el mismo periodo y el mismo instrumento. Cuando no, lee las columnas y olvídate de la diferencia.

Lo que dicen sus operaciones. Con su lista de operaciones, sin correr sus reglas, se juzga lo que se puede: si sus cuentas cuadran con sus propias operaciones, cómo quedan rehechas con tu dinero y tus costes (o, si su fichero los trae, con los de su prueba) y, si el histórico cargado es del mismo activo, si sus operaciones caben en tus velas:

precios que no existieron, horas sin mercado, entradas demasiado buenas. Con el de otro activo no se ponen sobre tus velas, porque sobre las de otro un informe honrado parecería un fraude. Lo que no se puede con una lista sola se dice: fuera de muestra, mejor que el azar, las vecinas, los otros instrumentos… todo lo que vuelve a correr las reglas.

Sus reglas, traducidas. Un `.sqx` de StrategyQuant X trae también las reglas, y se traducen a una estrategia del motor *sólo si casan exactas*: el mismo indicador con la misma fórmula, la misma comparación, la misma forma de entrar y de salir. Una sola pieza que no casa y no hay traducción, con cada pieza, su sitio y su motivo: juzgar una parecida sería juzgar otra con el nombre de ésta. Hoy casan pocas, y se dice por qué: el ATR de SQX es el de Wilder y el del motor la media simple; SQX mueve el stop dinámico con la apertura de cada vela; sus órdenes en espera se renuevan con cada señal; su viernes cierra a la hora y minuto exactos. Cuando casan, se ven sus reglas en palabras, lo que se ejecuta distinto en algún caso, se pueden **correr en tu histórico y comparar** operación a operación con las suyas —con el desfase de reloj entre sus datos y los tuyos, si lo hay— y **juzgarla con las trece comprobaciones**. SQX no escribe en el fichero cuántas estrategias generó para dar con ésa: si lo sabes, pon la cifra, y el juez la tendrá en cuenta y sólo usará las velas posteriores a su prueba; sin ella, la juzga como la única que se probó, que es la lectura más blanda.

Cómo juzga

El juez, abierto para que lo mires: las trece comprobaciones, lo que vale cada una y lo que mide.

El reparto. En cuanto hay informe, mueve cualquiera de los pesos y aparece aquí una segunda cifra, al lado de la nota de fábrica: lo que saca el informe con *tu* reparto. La siguiente pasada la escribe también en el propio informe. No cambia nunca la nota oficial ni su palabra, y es a propósito: un *sólida* comprado apagando las comprobaciones que te suspenden se podría exportar, o enseñar a alguien, como el veredicto. Auditar al juez, no comprarlo. Tu cifra no lleva techo nunca y la oficial puede llevarlo, así que una diferencia entre las dos puede ser un techo y no tu reparto.

Qué se mide: la partición entrenamiento / examen (0,7), las ventanas por periodo (6), los sorteos contra el azar (50, y más cuando el tramo del juez es corto) y los remuestreos de Monte Carlo (1.000). Estos mueven el experimento, así que *sí* cambian el veredicto, y entran en vigor cuando pulsas **Medir otra vez**.

Contrastar contra. Elige los instrumentos que usa la comprobación de otros instrumentos. Tú sabes cuáles van juntos —el oro con la plata, el euro-dólar con la libra-dólar— y el programa no lo puede sacar de una lista de ficheros. Sólo se ofrecen los instrumentos que se pueden poner en la temporalidad de la estrategia: los de sus mismas velas o más finas, reagrupados, y con 200 como mínimo al terminar. Si no eliges ninguno, usa los que tengan más histórico.

Un reparto y unos ajustes se pueden guardar con un nombre y aplicarse luego. Lo guardado se queda sólo en esta máquina.

Un nombre propio

Cuando el juez da una estrategia por **Sólida**, una tarjeta debajo del informe le ofrece un nombre propio, sacado de un tema que va con su familia: ríos para la tendencia, montañas para la ruptura, mitos del regreso para la reversión a la media, nombres de la física para el impulso, vientos para la volatilidad, de la poesía para la acción del precio y de la música para la divergencia. El primero que ofrece es el que nació —o se coronó— más cerca del día en que pasó, y si es ese mismo día, lo dice. *Otros tres* trae más, puedes escribir el tuyo, o *No, gracias*. El nombre va delante del técnico, que se queda debajo porque es el que dice qué hace la estrategia: en la Biblioteca y su búsqueda, en la Cuarentena y en Inicio. Dos estrategias de tu Biblioteca no comparten nombre; cuando se agota un tema, el nombre lleva el día en que pasó («Danubio del 7 de noviembre»). Una Frágil se queda sólo con su nombre técnico.

El método

Todo en este programa sigue un método: **la eliminación**. Un backtest casi siempre sale bonito, y una búsqueda que mide miles de variantes garantiza que algunas salgan maravillosas por pura suerte. Así que el trabajo nunca es coleccionar ganadoras: es destruir impostoras lo antes y lo más barato posible, y fiarse sólo de lo que sobrevivió a todo lo que intentó matarlo. Cuando una búsqueda tira el 99 % de lo que crió, eso no es desperdicio: es el método funcionando.

Eliminar, no coleccionar

Cada etapa existe para matar lo que dejó pasar la anterior. La búsqueda cría y mide a granel; la selección se queda con un puñado; el Reality Check intenta demostrar que cada superviviente no vale nada; la Biblioteca apunta el veredicto junto a la estrategia, para que no se olvide por lo bajo; la Cuarentena vigila en vivo la distancia entre la promesa y la cuenta; y la Cartera sólo combina lo que aguantó. El error caro es rescatar una curva de capital bonita de la etapa que acaba de condenarla: la curva era la prueba del expediente, el veredicto es el juicio.

La evidencia tiene precio, y el juez lo cobra: cuantas más variantes probó una búsqueda, más tiene que demostrar la mejor de ellas sobre histórico que nadie vio — ser la mejor de tres mil es, sobre todo, una propiedad del número tres mil. Por eso la cuenta de candidatas viaja con la estrategia, no se pone a cero por cambiar de fichero, y se gasta: nunca se lava.

El embudo: las dudas baratas primero

Ordena las dudas por lo que cuesta resolverlas. Un backtest cuesta segundos. Las trece comprobaciones del juez cuestan minutos. Tu propia atención cuesta horas. La Cuarentena cuesta semanas de calendario — pero ningún dinero. Una cuenta real es el único paso que puede costar dinero, así que va el último, y pequeño. El embudo es el menú leído de arriba abajo: medir a granel, juzgar la lista corta, poner en cuarentena lo que pasó, y sólo entonces arriesgar algo. Nunca gastes un paso caro en lo que uno más barato podía haber matado — y nunca te saltes el barato porque la curva se vea clarísima: clarísima es exactamente como se ve la suerte.

Las señales del sobreajuste

Sobreajustar es leer el ruido y llamarlo regla. Sus señales:

- **Demasiado bueno para el juego que se juega**: rentabilidades muy por encima de lo que ese mercado paga a las estrategias que sobreviven en él.
- **Se derrumba fuera de muestra**: brillante en las velas donde se crió, corriente en el tramo que nadie vio.
- **Un pico afilado en vez de una meseta**: mueve un paso cualquier parámetro y el resultado se deshace — el número estaba afinado al ruido, no a un comportamiento.
- **Sólo funciona en un sitio**: un instrumento, una temporalidad; las mismas reglas en cualquier otra parte no ganan nada.
- **Vive de un puñado de operaciones**: quítale su mejor 5 % y no queda nada.
- **Fue la mejor de miles**, y se le pide que demuestre lo mismo que a la única que se probó jamás.
- **Se juzgó hasta que aprobó**: retocar y volver a juzgar sobre las mismas velas es ajustar el veredicto, no la estrategia — el diario de veredictos existe para que eso se vea.

> **El juez sólo puede vigilar lo que vio.** Su tramo reservado protege de lo que la búsqueda hizo dentro de este programa. Si afinaste las reglas a mano mirando el gráfico entero, o trajiste la estrategia ya ajustada de otro sitio, ese tramo es evidencia quemada: las comprobaciones están puntuando un trabajo que la estrategia pudo copiar. Juzga sobre histórico en el que no se construyó, y nunca repitas una prueba sobre datos quemados.

Nada de esto dice nunca que una estrategia sea buena. El método sólo dice una cosa con confianza: *ésta falló*. Lo que no consiga matar, te está permitido arriesgarlo — pequeño, vigilado, y sabiendo que sobrevivir al embudo es el principio de la confianza, no el final de la duda.

Instalar el robot en cada plataforma

El menú de descarga de una estrategia en la **Biblioteca** produce un fichero terminado para seis destinos; **Exportar...**, en Construir, guarda la estrategia y te abre ese menú. Ponerlo a funcionar es distinto en cada uno, y casi todas las formas de fallar son silenciosas: el fichero compila, no pasa nada, y ningún error dice por qué. Cada trampa de las de abajo es una en la que la gente se atasca de verdad.

> **Antes de nada: el robot no es el backtest.** Lo que simuló el backtest, el fichero lo lleva: las reglas, leídas sólo sobre velas cerradas; la entrada —a mercado, o una orden STOP o LIMIT esperando tantas velas como esperó el backtest—; el stop y el objetivo; el break-even y el stop dinámico; las salidas por tiempo; los dos topes diarios, contados sobre el día UTC como los cuenta el backtest; el tamaño —lote fijo, porcentaje de riesgo o Kelly—; y una posición cada vez. Cada fichero dice en su cabecera dónde su plataforma no puede hacer exactamente lo que hizo el backtest. Lo que ningún fichero puede llevar es todo lo que la simulación tuvo que suponer: el spread real de tu bróker en el momento de la ejecución, el slippage en una vela rápida, las ejecuciones parciales, los requotes, un spread ensanchado por una noticia, y un hueco que salta por encima de un stop o de una orden en espera, que el bróker ejecuta más allá del nivel mientras el backtest la ejecutaba en el nivel. El código es además mínimo a propósito: sin filtro de noticias y sin filtro de sesión. Pruébalo primero en una cuenta demo, en todas las plataformas, sin excepción.

Exporta con el histórico de la estrategia abierto. El fichero se escribe con la ficha del instrumento de la serie abierta en la pantalla de Datos —el tamaño del pip, las unidades de un lote, el apalancamiento con el que el backtest limitó cada operación—, así que cada distancia y cada tamaño significan lo que significaban cuando se midieron. Con otro instrumento abierto, los ficheros de MetaTrader, JForex y Python se escriben con la ficha de ése; el script de Pine se da cuenta, la deja fuera y saca la suya del gráfico.

Stops en pips salidos de una búsqueda. Cuando la estrategia salió del Piloto o de una búsqueda y su stop y su objetivo van en pips, el menú de descarga empieza por **Sus stops**: **Adaptarlos al ATR**, lo recomendado y lo marcado de partida, o **Mantenerlos fijos**. Adaptado, el fichero lleva el stop como tantas veces el ATR(14) —las que lo dejan exactamente igual de ancho que el juzgado en la última vela que juzgó el Reality Check—, y desde ahí crece y se encoge con la volatilidad en vez de quedarse fijo mientras el mercado cambia de escala. Medido en ocho mercados, las estrategias que el Reality Check no rechazó batieron después al azar un 75 % de las veces en vez de un 63 %. El objetivo y un trailing en pips se adaptan igual; el break-even y el desplazamiento de una orden en espera se quedan en pips. A una estrategia hecha a mano no se le pregunta: el fichero conserva sus pips.

MetaTrader 4 — el .mq4

1. En MetaTrader, **File** → **Open Data Folder** (Archivo → Abrir carpeta de datos). Éste es el paso que la gente falla: la carpeta que buscas no es donde está instalado el programa, y sólo MetaTrader sabe dónde está.

2. Copia el fichero en `MQL4/Experts/`.
3. Pulsa `F4` para abrir MetaEditor, abre el fichero y pulsa `F7` para compilar. Tiene que decir *0 errors*; un aviso da igual.
4. De vuelta en el terminal, pulsa con el botón derecho en el Navigator (`Ctrl+N`) y elige **Refresh**. El robot aparece bajo **Expert Advisors**.
5. Arrástralo a un gráfico del mismo símbolo y la misma temporalidad para los que se construyó. En el diálogo, la pestaña **Common** tiene **Allow live trading**: márcala.
6. Enciende **AutoTrading** en la barra de herramientas. Tiene que estar en verde.

Una **cara sonriente** en la esquina de arriba a la derecha del gráfico quiere decir que está corriendo. Una cara triste quiere decir que AutoTrading está apagado, o que a ese gráfico no se le permitió operar en vivo. MT4 corre **un robot por gráfico**: para operar dos, abre dos gráficos.

MetaTrader 5 — el .mq5

Los mismos seis pasos, con `MQL5/Experts/` y el botón de la barra de herramientas que aquí se llama **Algo Trading**. Tres diferencias que conviene saber:

- **Si compila y no opera nunca, lee la pestaña Experts** del Toolbox (`Ctrl+T`). MQL5 crea los handles de los indicadores en `OnInit`, y si uno falla el robot devuelve `INIT_FAILED` y se para antes de su primer tick. Ese fallo se escribe ahí y en ningún otro sitio.
- **Netting o hedging.** El robot lleva una posición cada vez, como hizo el backtest, y la reconoce por su `MagicNumber`. En una cuenta netting todas las operaciones de un símbolo son una sola posición, así que otro robot o una operación tuya a mano sobre el mismo símbolo se funden con la suya. Deja ese símbolo para este robot, o usa una cuenta hedging. El tipo de cuenta lo dice el terminal, debajo del nombre de la cuenta.
- **Usa el Strategy Tester** (`Ctrl+R`) antes que el gráfico en vivo. Es la forma más rápida de ver si el robot llega a operar, y no cuesta nada.

MetaTrader 4 y 5 — los ajustes que añade el fichero

- **`BacktestPipSize`** — cuando el fichero se escribió con una ficha del instrumento, cada distancia en pips (el stop, el objetivo, la orden de entrada, el break-even, el stop dinámico) se cuenta en el pip del instrumento que se midió, no en los dígitos con los que tu bróker cotiza el símbolo. Comprueba que tu bróker lo cotiza a la misma escala que el histórico que mediste.
- **`BacktestLeverage`** — el tamaño nunca pasa de lo que la cuenta pueda sostener de margen al apalancamiento con el que el backtest limitó cada operación, o al de la propia cuenta si es menor. El valor del pip lo pone tu bróker, y el tamaño se ajusta a los saltos de lote de tu bróker; por debajo de su lote mínimo la operación no se toma, y el registro de Experts dice por qué.
- **`ServerUtcOffsetHours`** — aparece cuando la estrategia tiene topes diarios, salidas por tiempo, o lee los niveles del día anterior. El backtest contó esos días y esas horas en UTC; MetaTrader sella sus velas con el reloj del servidor. Déjalo en 0 si el histórico que mediste salió del MetaTrader de este mismo bróker; ponle el desfase del servidor respecto a UTC —2 o 3 en la mayoría de brókeres, y cambia con el horario de verano— si ese histórico se cortó en UTC.
- **La hora del servidor** — con cierre a una hora, topes del día o niveles del día anterior, el robot necesita `ServerUtcOffsetHours`: lo que su servidor va por delante de UTC, menos lo que iba el reloj del histórico con que lo mediste (cero si el histórico vino del MetaTrader de ese mismo bróker). Fuera del Probador de Estrategias el robot mide el huso de su servidor en cada vela nueva, lo escribe en el diario de Expertos y salta una alerta si el ajuste no cuadra con el reloj del histórico, que el fichero lleva escrito arriba cuando se sabe.
- **Reinicios** — la última vela sobre la que decidió el robot, y con topes diarios el día que va contando, se guardan en las variables globales del terminal (`F3`), así que un reinicio ni decide dos veces la misma vela ni se olvida

del día. Dos copias del robot sobre el mismo símbolo y el mismo periodo de gráfico necesitan `MagicNumber` distintos.

Dukascopy JForex — el .java

1. En la plataforma JForex, abre el panel **Strategies**, elige **Open** y busca el fichero. La aplicación de escritorio puede dejarlo antes en la carpeta de estrategias de tu JForex, en Linux y en Windows: **Enviarla a la carpeta de la plataforma → JForex**, en el menú de descarga de la Biblioteca.
2. Compílalo dentro de la plataforma. La clase implementa `IStrategy`, con la lógica en `onBar`.
3. Pulsa con el botón derecho en la estrategia y pulsa **Start**. Los campos `@Configurable` salen en el diálogo de arranque: el instrumento y el periodo del gráfico, el stop y el objetivo, el lote o el riesgo, y los que use la estrategia de la orden de entrada, las salidas por tiempo y la gestión. De los que son parte de la medición, el fichero avisa de que cambiarlos es operar una estrategia que nadie midió. El pip, el tamaño del lote y el apalancamiento del instrumento medido van escritos en el código, no se ofrecen como ajustes.

La exportación a JForex se niega a lo que JForex no puede operar como se midió: el tamaño por riesgo o por Kelly fuera de los pares de divisas, y cualquier mercado que no sea un par de divisas, oro, plata, bitcoin o ether, porque JForex los cuenta en contratos de un tamaño que el fichero no puede saber. Fuera de los pares de divisas necesita además la ficha del instrumento, porque el pip y el lote no se deducen del símbolo: exportado sin histórico abierto, el fichero lleva sólo la negativa.

> **Java exige que el nombre del fichero y el de la clase coincidan.** Renombrar `MyRobot.java` a `test.java` hace que deje de compilar, con un error que habla de una clase pública y no del nombre. Renombra también la clase, o deja el fichero en paz.

Pásalo primero por el **Historical Tester**: viene con la plataforma y usa los ticks de la propia Dukascopy, que es de verdad una segunda opinión y no la misma que te dieron aquí.

En vivo o en una cuenta demo, la estrategia lleva un diario y un latido en la carpeta de ficheros de JForex, dentro de `BotFactoryFX`, para que *En vivo* la ponga al lado de su backtest decisión a decisión; en el Historical Tester no escribe nada. Ejecútala en tu ordenador: una estrategia que corre en los servidores remotos de Dukascopy los escribe allí.

cTrader — el cBot de C#

1. En cTrader, abre **Algo** (en versiones anteriores, **Automate**) y crea un cBot nuevo. El nombre que le pongas es el que verás en la lista.
2. Borra todo lo que trae la plantilla y pega el fichero entero.
3. Pulsa **Build**. Tiene que terminar sin errores.
4. Añade una instancia en el símbolo y la temporalidad con que se midió la estrategia, revisa sus ajustes y arráncala.

Calcula los indicadores él mismo. No usa la biblioteca de indicadores de cTrader: cada uno va portado del motor de BotFactoryFX con la misma aritmética, así que cada regla lee el número que leyó el backtest. Lee sólo velas cerradas y corre en UTC: la hora, el día de la semana y los niveles del día anterior son los de UTC, como en el backtest.

Los ajustes salen en grupos: el tamaño, los stops, la entrada, el día, el reloj, el corte de cuenta y las otras series, sólo los que use la estrategia. Lleva escritos el pip y el lote del instrumento medido: si en tu bróker el lote es de otro número de unidades, no opera y lo dice en el registro (`Check the instrument` lo apaga a sabiendas). En una temporalidad distinta de la medida se para al arrancar. Reconoce sus operaciones por su **Label**, así que dos estrategias en la misma cuenta no se tocan.

Lo que cuenta sobrevive a un reinicio. Su capital —el saldo al arrancar más lo que hacen sus propias operaciones, como lo cuenta el backtest—, los topes del día y el corte de cuenta se guardan en el almacén local de la instancia, y al volver a arrancar cuenta las operaciones que cerraron mientras estaba parado. **Start over** lo olvida y empieza desde el saldo. En el probador de cTrader siempre empieza de cero.

> **El corte de cuenta no se quita solo.** Cuando salta, el cBot no abre nada más —lo abierto sigue hasta su stop, su objetivo o su salida— hasta que pongas **Start over** y lo vuelvas a arrancar. Es a propósito: es lo que hace el backtest.

Pásalo antes por el **backtesting de cTrader** con datos de ticks: son los precios de tu bróker, una segunda opinión de verdad. BotFactoryFX todavía no lee cuentas de cTrader —no hay conector—, así que *Cuarentena* y *En vivo* no lo siguen.

`TradingView — el script de Pine`

1. Abre un gráfico del símbolo que toca, y pon la temporalidad en la que se construyó la estrategia.
2. Abre el **Pine Editor** en el panel de abajo, y luego **Open → New blank strategy**.
3. Selecciona todo lo que haya en el editor y pega el script encima. Guárdalo con un nombre.
4. **Add to chart**. La pestaña **Strategy Tester** de abajo lo ejecuta.

> **El script opera el gráfico en el que esté puesto.** En el par o la temporalidad equivocados sigue funcionando, sigue dando cifras, y describe una estrategia que nadie construyó. Es, con diferencia, la forma más común de que un resultado de Pine acabe sin significar nada.

Dos cosas más hacen que sus números no cuadren con los de aquí. **El script no pone comisión, slippage, spread ni funding**, y dice por qué en su cabecera: la exportación no lleva los costes con los que se midió el backtest, y cualquier cifra que escribiera ahí sería inventada. Así que sus resultados se separan de los del backtest justo en esos costes. Poner *Commission* y *Slippage* en las *Properties* del script a lo que te cobra tu bróker acerca los totales, no las operaciones: TradingView cotiza un solo precio, así que ningún ajuste mueve los niveles un spread; su slippage mueve además las órdenes a mercado, que el backtest ejecutaba al cierre; y para el funding no tiene ajuste ninguno. Y el feed de TradingView para un símbolo no es el de tu bróker: otras sesiones, otros cierres, otros huecos.

Lo que el script sí lleva: la entrada al cierre de la vela de la señal, como entró el backtest —sus órdenes a mercado se ejecutan a ese cierre—; una entrada STOP o LIMIT que espera y se cancela tras el mismo número de velas; los dos topes diarios, contados sobre el día UTC; y el pip, el lote y el apalancamiento del instrumento que mediste, cuando el histórico abierto al exportar es el de la estrategia — si no, los saca del gráfico, y el script dice cómo.

TradingView no manda órdenes. Nada de lo que añadas a un gráfico opera una cuenta. Para actuar hacen falta una alerta, un webhook y un puente que convierta la alerta en una orden en un bróker: tres piezas más, y aquí no se da ninguna.

`Hyperliquid — el bot de Python`

El único destino que se conecta él solo a un sitio donde se opera, y el único donde una equivocación cuesta dinero sin una plataforma de por medio. Para correr en seco le basta **Python 3.8 o posterior**, y nada más.

1. **Córrelo en seco.** `python3 tubot.py`. Trae velas de verdad, imprime la orden que habría mandado, y no manda nada. Sin clave, sin instalar nada, sin cuenta.
2. **Léelo.** Los indicadores están dentro del fichero, portados de este motor: ése es todo el sentido de este destino. Lo que calcula es lo que calculó el backtest.

3. **Déjalo en seco unos días** y compara sus señales con las del backtest. Es el paso que todo el mundo se salta y el único que caza un desajuste antes de que salga caro.
4. **Crea una API wallet** en Hyperliquid, en el apartado **API**. Puede operar pero **no puede retirar fondos**. Esa propiedad es la diferencia entre una equivocación cara y una total: no uses la clave de tu cuenta.
5. **Exporta las credenciales** e instala el SDK:
   `export HL_ACCOUNT_ADDRESS=0x…`
   `export HL_SECRET_KEY=0x…`
   `pip install hyperliquid-python-sdk eth-account`
6. `BFX_LIVE=1 python3 tubot.py` opera en la **red de pruebas**, donde el dinero no es real. Añadir `BFX_MAINNET=1` es lo que le pone dinero de verdad detrás.

Gestiona la operación como la gestionó el backtest: el stop y el objetivo se mandan a Hyperliquid **con la entrada**, como órdenes de disparo que sólo reducen posición, así que se ejecutan en el tick que las toca y se ejecutan igual aunque este programa se muera. Una entrada STOP o LIMIT se queda esperando en Hyperliquid con su stop y su objetivo enganchados, y se cancela tras tantas velas como esperó el backtest. El break-even, el stop dinámico, las salidas por tiempo, el tope de pérdida diaria y el tope de operaciones por día funcionan todos. Lleva **una posición cada vez**, porque es lo que midió el backtest, y si lo reinicias con una operación abierta la recoge en vez de abrir otra encima.

> **Lee esto antes de poner BFX_MAINNET=1.**
>
> - **Mira `PIP_SIZE` al principio del fichero.** Cada distancia de la estrategia se cuenta en pips. Cuando el histórico que mediste traía su ficha del instrumento, `PIP_SIZE` es el pip de esa ficha, y el bot lo conserva aunque el precio haya cruzado una potencia de diez desde entonces, porque un stop de 30 pips tiene que quedarse donde se midió. Sólo cuando no había ficha, y el símbolo no es ni un par de divisas ni un contrato conocido, es una suposición; entonces el bot la compara con el precio en vivo en su primera vela y **se niega a operar en real** si no cuadran, y te dice qué número poner.
> - **Calcula el tamaño de cada operación como lo calculó el backtest** —el lote fijo, el porcentaje de riesgo o Kelly— con las unidades por lote del instrumento medido, y con el tope que el apalancamiento del backtest podía sostener de margen, o el del propio sitio si es menor. Hyperliquid lo ajusta después al salto de tamaño de la moneda, y una orden a la que eso le cambie el tamaño lo dice.
>
> No te cobra nada por spread, comisión ni funding, porque no puede saberlos: su acumulado es bruto, y tanto el guardián de la pérdida diaria como Kelly leen ese número bruto.

Se despierta dos segundos después de que cierre cada vela, la lee y actúa, y nunca decide sobre la vela en curso: ten el reloj del ordenador en hora. Si no estaba mirando cuando cerraron velas (dormido, sin red, parado), las recorre en orden al volver y gestiona la operación como la habría gestionado; pero abre operación sólo en la vela que acaba de cerrar, y sólo dentro de los dos minutos siguientes a su cierre, nunca con una señal cuyo momento ya pasó. En un servidor, córrelo con `screen`, `tmux` o una unidad de systemd. Una operación abierta sobrevive a que pares el bot: su stop y su objetivo están esperando en Hyperliquid, no en el proceso — con una excepción que el fichero nombra: una orden de entrada que Hyperliquid no aceptara con el stop y el objetivo enganchados se queda sola, y una operación que se abra con el bot parado no tiene stop hasta que el bot vuelva a correr. Lo dice cuando lo paras.

> **La clave no va nunca dentro del fichero.** Se lee del entorno a propósito. No la pegues en el script, y no subas a ningún sitio un fichero que la lleve: una estrategia está hecha para compartirse, y una clave dentro se comparte con ella.

Cuando la estrategia lee otras series

Una condición puede leer una serie distinta del gráfico que opera: «comprar en H1 sólo si el GBPUSD diario está por encima de su media». El fichero le pide esa serie a la plataforma él mismo, y la lee como la leyó el backtest: en cada vela del gráfico, la última vela de la otra serie que hubiera *cerrado* para entonces, nunca la que se está formando. Al descargarlo, la Biblioteca avisa de que el fichero lee otras series, y la cabecera del fichero enumera lo que lee y lo que hay que comprobar.

- **Dónde se eligen.** En **Construir** —en *Manual*, debajo del nombre y el símbolo; en *Evolve*, en *Buscar*, debajo del histórico, y en la *Rueda*— y en el **Piloto automático**: la misma tarjeta *Series de referencia* → **Añadir serie de referencia**, y todas enseñan las mismas series. El Reality Check juzga con las que haya cargadas. Salen de un histórico de la biblioteca —el mismo SP500 en M1 da su H4 y su M30—; lo que no esté se baja o se importa antes en **Datos**, también un CSV. Las velas se reagrupan a la temporalidad que elijas, que sólo puede ser la del histórico o una más gruesa que la contenga entera.
- **La serie tiene que estar cargada al exportar.** Las series de referencia viven en memoria: después de reiniciar, la estrategia las sigue nombrando, pero nada dice qué mercado eran. La exportación se niega hasta que las vuelvas a añadir en *Series de referencia*, en **Construir** o en el **Piloto automático**, en vez de adivinarlo.
- **El símbolo es un ajuste; la temporalidad, no.** Cada bróker llama a los mercados a su manera — `EURUSD.a`, `GER40` frente a `DE40` —, así que cada serie tiene un símbolo que puedes cambiar: `Ref1Symbol` en MetaTrader, **Series ref1: instrument** en JForex, **Series ref1: symbol** en cTrader, los campos de *Series de referencia* en Pine, el `coin` de `REFERENCES` en Python. Su temporalidad es parte de la estrategia y es fija.
- **Las dos series tienen que ir a la misma hora.** El backtest junta la otra serie con la del gráfico por la hora de cada vela. Un fichero de MetaTrader va en la hora del servidor del bróker —dos o tres horas por delante de UTC — y uno de Dukascopy o Binance en UTC: juntos, una condición leería la otra serie horas antes de tiempo, que es leer el futuro. Al cargar una serie de referencia, BotFactoryFX mide el desfase por cómo se mueven las dos y avisa si lo hay. Cárgalas de la misma fuente.
- **Mientras no haya bastante historia de la otra serie, esas condiciones cuentan como falsas**, exactamente igual que en el backtest. Nunca decide con un valor que no pudo leer.
- **Las velas diarias y semanales dependen del reloj.** El backtest usó las velas de tu fichero. MetaTrader corta los días a la hora del servidor del bróker, TradingView a la sesión del feed, JForex y Hyperliquid en UTC, y cTrader a la hora que fije su bróker. Si eso no coincide con tu fichero, el robot lee otras velas.
- **Los probadores.** El de MetaTrader 4 no mueve los otros símbolos vela a vela; el de MetaTrader 5 necesita ticks reales u OHLC de un minuto; el de JForex necesita que cada instrumento esté añadido a la prueba; el de cTrader carga cada símbolo del histórico de tu bróker. Pruébalo hacia delante en una cuenta demo.
- **MetaTrader 4 no tiene histórico de H2, H3, H6, H8 ni H12.** Una serie en una de ésas espera hasta que la construyas en un gráfico offline. MetaTrader 5 sí las tiene.

Lo que comparten las seis

- **Los indicadores son los del motor.** Donde una plataforma calcula un indicador de otra manera que el motor — su ADX, su RVI, su línea de señal del MACD, y varios más—, el fichero lo calcula él, con la fórmula del motor, para que el robot lea la línea que leyó el backtest. El cBot de cTrader y el bot de Python van más allá: los calculan todos ellos.
- **Nada de la estrategia llega al fichero en crudo.** El texto libre se vuelve inerte, y cada símbolo, cada nombre de serie y cada número se comprueban, así que una estrategia que te manden no puede colar código en lo que exportas.
- **Los números que el motor no mide se rechazan** —una vuelta atrás negativa o con decimales, un periodo por debajo de 1— en las seis, con las mismas palabras, en vez de escribirse y operarse.

- **Cuando un destino no puede llevarse una estrategia con honradez, no ofrece fichero.** La Biblioteca dice por qué: una serie de referencia sin cargar, una temporalidad que la plataforma no tiene, las negativas de JForex por tamaño y por mercado, un gráfico en una temporalidad que cTrader no tiene, un script de Pine que se pasa de los límites de TradingView, un indicador del que el bot de Python no tiene porte.
- **Compilan.** Los ficheros de MQL4 y MQL5 se compilan con MetaEditor, el de Java con el compilador de la propia JForex y el de C# con el paquete de cTrader de Spotware, dentro de las pruebas de este programa. Y el cBot, además, se ejecuta vela a vela contra el motor: decide y opera lo mismo.

DATOS, BIBLIOTECA Y CARTERA

Los datos y las temporalidades

Todo lo del instrumento que estás probando vive en la pantalla **Datos**: el histórico de precios, la temporalidad, lo que tu bróker te cobra por operarlo y qué es el instrumento. Se pregunta una vez y se usa en todas partes.

De dónde sale el histórico

Las fuentes están en una fila, en dos grupos. Primero las que te dan el histórico hecho: tu propio **CSV** —el de tu bróker, o el que exporta el gráfico de TradingView—, luego **Dukascopy** para divisas y **Binance** para cripto. Después, pasada una raya fina, las plataformas a las que te conectas: **MetaTrader 5**, **MetaTrader 4** y **JForex**, que leen por un conector la plataforma que tengas abierta en esta máquina, y **web3**, donde pegas la dirección pública de tu monedero y BotFactoryFX te dice en qué sitio ha operado. No hace falta que sepas cuál: es la dirección la que lo dice. Lo que se pide ahí es un dato público —el que cualquiera lee en la cadena—, no una clave privada, y no se firma nada.

Encima de la fila, **¿Qué quieres bajar?** busca un activo en todas a la vez: escribe «dow», «oro», «bitcoin» o un símbolo, y te dice en qué fuentes está y desde cuándo hay historia; un clic abre su pestaña con el activo ya elegido. JForex sale según lo que ofrece tu cuenta, que el programa aprende la primera vez que la conectas: no todas las cuentas ofrecen lo mismo. MetaTrader no sale, porque lo que tiene depende de tu bróker: míralo en su pestaña.

La biblioteca de históricos

Los ficheros importados se guardan en disco de una sesión a otra — hasta **50 series**. Cargar un segundo instrumento lo añade, no sustituye al primero. Las exportaciones grandes no son problema: se leen como un flujo y se pliegan a una temporalidad manejable según van cargando, así que un fichero M1 de medio giga no tiene que caber en memoria.

Cada histórico se guarda en el formato propio del programa, que ocupa poco y se lee rápido, pero que ningún otro programa abre. Para llevártelo a otro, el botón **Exportar a CSV** de su fila lo guarda como CSV —la hora en UTC y la misma cabecera que se importa—, y un CSV exportado vuelve a entrar idéntico. Y si una estrategia ha de leer otro mercado —una serie de referencia—, se baja o se importa aquí como cualquier histórico y se elige en **Construir** o en el **Piloto automático**.

Un histórico que trae el futuro

Hay históricos de bróker en los que una hora del día —casi siempre la primera— lleva dentro el máximo y el mínimo del día entero que empieza con ella; suele ser historia rehecha a partir de velas diarias. Una regla que lea el máximo o el mínimo de esa vela —rupturas, canales, Ichimoku, estocástico, ATR— ve el día que viene, y el backtest gana lo que nadie puede ganar operando. El Reality Check no puede verlo con sus pruebas, que barajan y cortan velas enteras, así que se mira en los datos: si el histórico cargado lo trae, esta pantalla lo dice arriba con la hora y el tramo, la revisión de lo que importas lo cuenta como grave y el Reality Check lo dice antes que nada. Recorta el histórico para que empiece después de ese tramo, o bájalo de otra fuente. Lo que no lo trae no enseña nada.

Las acciones y sus desdoblamientos

Cuando una empresa desdobla sus acciones —parte cada una en varias—, el precio baja en esa misma proporción de un día para otro y nadie pierde nada. Dukascopy sirve las acciones sin corregirlo: su Apple trae el 4 por 1 de agosto de 2020 como una caída del 74 % en una hora, y un backtest sobre esa serie gana o pierde con una caída que no pasó. Por eso, en las **acciones y los ETF** que bajas de Dukascopy, y en un CSV que lleve el nombre que Dukascopy le da al activo (AAPLUSUSD), el programa los busca y los corrige solo antes de guardar: un hueco entre

un cierre y la apertura siguiente que es justo de 2, 3, 4, 5… y hasta 50 por 1 (o al revés, si las junta), tras el que el precio se queda en el nivel nuevo. Todo lo anterior se pasa a la escala nueva, como hacen los gráficos de las bolsas, y el panel de calidad de los datos lo dice con la fecha, el factor y cuántas velas cambiaron. Un salto así en un índice, una divisa, un metal o una cripto no se toca —no tienen acciones que repartir, así que es otra cosa—, ni en un CSV cuyo nombre no está en ningún catálogo, como «AAPL»: ahí la revisión de los datos sólo lo avisa, con la fecha y el factor.

Cuánto histórico guardar, y lo que cuesta

La aplicación de escritorio no tiene tope de velas. Carga lo que tu idea necesite. Lo que sí tiene es una consecuencia, y la vista previa de la importación la pone en pantalla antes de que te comprometas: dice a la vez la resolución, el número de velas y el rango de fechas, para que veas qué estás cediendo.

Y lo que se cede es justo lo que hay que entender. Un fichero se guarda con la resolución con la que lo importas, y a partir de ahí el selector de temporalidad sólo puede ofrecer ésa y las más gruesas — cuatro velas M15 hacen una H1, y nada saca las cuatro de la una. Así que la elección se hace una sola vez, al importar. Una exportación M1 de 23 años son 8,7 millones de filas; el mismo fichero te da cada minuto de los últimos meses, o cada cuatro horas de las dos décadas enteras, y las dos están a un desplegable de distancia.

Medido en esta máquina, un backtest de un cruce sencillo, para que sepas lo que compras:

- **50.000 velas** → 41 ms, 122 MB de memoria
- **100.000 velas** → 78 ms, 156 MB
- **500.000 velas** → 345 ms, 318 MB
- **1.000.000 de velas** → 783 ms, 586 MB

Es lineal — unos 0,8 microsegundos por vela, sin escalón en ninguna parte. Un backtest suelto sigue siendo rápido con cualquiera de esas cifras. Lo que crece es una *búsqueda genética*, porque corre 840 backtests: un millón de velas son unos minutos repartidos entre tus núcleos, en vez de segundos. Y la memoria es real — varios millones de velas la agotan. A partir de 100.000 la vista previa de la importación lo dice.

Elige la ventana que tu estrategia opera de verdad. Nadie hace un backtest de 23 años de velas de un minuto; una idea intradía necesita semanas de M1, y una de swing, años de H4. Un fichero grande tarda alrededor de medio minuto en leerse, sea cual sea la resolución que elijas — va en flujo, así que la aplicación no se queda parada.

La ficha del instrumento

Cada serie lleva su ficha del instrumento: el tamaño del pip, las unidades que hay en un lote y de qué lado del par está el dólar de la cuenta. La importación la deduce — del nombre, si es un par de divisas, un metal, petróleo o un índice conocido; del precio, para todo lo demás, donde un lote es una unidad; y del propio sitio cuando la fuente la declara, como hace Hyperliquid. Con ella se cuenta cada distancia de stop, cada tamaño de posición y cada cifra de un informe, y con ella se escribe el robot exportado. Cuando la ficha tiene algo que conviene saber —un cruce sin pata en dólares, cuyo valor del pip es estimado; un índice que cotiza en euros, en libras o en yenes, que este motor no convierte—, la fila de la biblioteca lo dice.

La ficha sigue al símbolo. El símbolo de cada histórico es el de la fuente de donde se bajó, y en *Lo que ya tienes cargado* se ve, no se escribe: sigue al histórico resaltado. Si alguna vez está mal —un CSV cuyo nombre no lo dice —, se corrige pulsándolo en su fila de la biblioteca y confirmando con Intro, y sólo cambia ese histórico. Cambiarlo sustituye la ficha: para un par de divisas, un metal o un índice conocido, el nombre nuevo la decide al momento; para todo lo demás se borra la vieja y se deduce otra vez de los precios, ahí mismo si es el histórico abierto y al abrirlo si no. Una ficha que llegue con el cambio —de una fuente que sabe el contrato de verdad— manda sobre las dos.

Lo que cuesta operarlo, y de dónde sale

Cada histórico nace con sus costes, y la ficha de costes dice de dónde salen. De **MetaTrader**, los de tu terminal: el diferencial de la ficha del símbolo, la comisión que te cobraron de verdad y el swap. De **Binance** y **Hyperliquid**, los del sitio: el diferencial del libro y su comisión publicada. De **Dukascopy** y de **JForex** —que es su plataforma—, el diferencial *medido en su propio feed*: al final de la descarga se bajan también las velas de compra y de venta de los dos últimos meses completos, y la resta, hora a hora de la semana, es lo que se cobra en cada entrada; y la comisión de su tabla publicada para las cuentas más pequeñas (35 $ por millón negociado y lado en divisas, 52,5 en el resto; el 0,10 % en acciones y ETF, más un mínimo por operación que el backtest no cobra). Una cuenta mayor paga menos: bájala si es tu caso. Si la medida no se puede hacer —el feed pierde un fichero—, el histórico se guarda con el preajuste y se dice.

Un **CSV** no dice de qué bróker viene, y lleva un preajuste: la cifra de una cuenta estándar en divisas, llevada a coste relativo. Contrastada contra el diferencial en bruto de Dukascopy, en la plata, el cobre, el crudo, el gas, la cripto, las acciones o los bonos se quedaba entre cinco y treinta veces corta, así que para los instrumentos medidos no baja de lo que Dukascopy cobra en bruto. Sigue sin ser tu bróker: con tus cifras, siempre mejor.

Cambiar la temporalidad reconstruye las velas

El selector de temporalidad no es una etiqueta. Elegir una más gruesa **reconstruye todas las velas** —la apertura de la primera, el cierre de la última, el máximo y el mínimo verdaderos de todo el periodo, el volumen sumado— y el gráfico, el backtest, la búsqueda genética y el código exportado se mueven todos a la vez con ella.

Una vez guardada, sólo se ofrecen la resolución propia del fichero y las más gruesas. Cuatro velas M15 hacen una H1; nada saca las cuatro de la una, así que una temporalidad más fina no se ofrece, en vez de fingirla. Las velas semanales empiezan el lunes y las mensuales el día 1, en UTC, así que dos ficheros que se solapen dan siempre las mismas velas.

Conectar MetaTrader 5

BotFactoryFX puede sacar el histórico de precios directamente de un terminal de MetaTrader 5 en marcha — todos los instrumentos que ofrezca tu bróker, cualquier temporalidad, sin exportar un solo fichero a mano. Lee el terminal y **nunca manda una orden**.

MetaTrader 4 se conecta igual, con un puente propio: lo cuenta el último apartado de este capítulo, *Conectar MetaTrader 4*.

Cómo se conecta

Por un añadido pequeño que se llama el **conector**: un *servicio* de MQL5 que corre dentro de MetaTrader, sin gráfico. Habla con esta aplicación escribiendo ficheros en la carpeta común del propio MetaTrader.

Es una elección deliberada. Una conexión de red —el servidor MCP de MetaTrader, un socket o `WebRequest`— necesita un ajuste dentro del terminal que hay que encender y recordar, y se sabe que MetaTrader lo olvida entre arranques. Los ficheros no necesitan puerto, ni lista de permitidos, ni regla del cortafuegos, ni diálogo. Y como a un *servicio* lo arranca el terminal cada vez que se abre, se instala una vez y no se vuelve a pensar en ello.

Instalarlo, una sola vez

1. Abre MetaTrader 5 y entra en tu cuenta.
2. Aquí, en **Ajustes → Plataformas de trading**, pulsa **Instalar el conector** en el panel de MetaTrader 5. También está en la pantalla **Datos**: elige **MetaTrader 5** entre las fuentes, abre **Importar desde MetaTrader 5** y pulsa

Instalar el conector.

3. Si MetaTrader estaba abierto, ciérralo y vuelve a abrirlo; si estaba cerrado, ábrelo. El conector arranca con él.

El botón lo hace todo: encuentra la carpeta de MetaTrader —también dentro del prefijo de Wine—, deja el conector en su carpeta de servicios y lo pone para arrancar con el terminal, con la misma entrada que escribe MetaTrader cuando añades un servicio a mano. A partir de ahí el terminal lo arranca cada vez que se abre, y el panel de aquí se pone verde por su cuenta. Si prefieres no cerrar MetaTrader ahora, puedes arrancarlo desde el Navegador (`Ctrl+N`): **Servicios** → **BotFactoryFX** → clic derecho → **Añadir servicio**, y luego **Iniciar**. Una sola vez: añadirlo dos veces deja dos copias, y el panel lo dice.

La carpeta del puente

El conector es un programa pequeño con nombre propio —**BotFactoryFX Bridge**, el puente— y viaja dentro de esta aplicación como una carpeta: el servicio compilado (`BotFactoryFX.ex5`), su fuente (`BotFactoryFX.mq5`, para que cualquiera pueda leer lo que hace), el icono que MetaTrader enseña en el Navegador, un `LEEME.txt` y el manual en PDF. Debajo de las tarjetas de instalar hay tres botones: **Guardar la carpeta del puente (conector, manual, icono)** copia la carpeta entera a `~/.botfactoryfx/conectores/metatrader5` y enseña la ruta con un botón para copiarla; **Abrir el manual** la guarda y abre el PDF; **Abrir la carpeta** la guarda y la abre en tu gestor de ficheros. Es para instalarlo a mano, leer el fuente o llevarse el conector a otro ordenador — el botón de instalar de arriba hace el trabajo de cada día por su cuenta.

En Linux: qué Wine

MetaTrader 5 corre en Linux con Wine, y **MetaQuotes da soporte a Wine 10.0 y superior**. Con cualquiera anterior, el terminal avisa él mismo en su propio registro, en cada arranque, de que ésa es una versión *inestable y sin soporte*. Esta aplicación lee la línea de arranque en la que el terminal dice sobre qué Wine va, y te la enseña, porque un problema del anfitrión de debajo tiene la misma cara que un problema de la aplicación de encima.

En Debian, Ubuntu y Mint, el Wine de la propia distribución suele ser demasiado viejo. El de WineHQ no:

```
sudo apt install --install-recommends winehq-stable
```

> **Dos trampas al actualizar Wine.** Las dos son silenciosas — ninguna da un error que puedas buscar.
>
> - **Cierra antes todo lo que Wine tenga en marcha**, incluidos los procesos que no ves: `wineserver -k`. La versión vieja y la nueva guardan sus sockets en carpetas distintas, así que el Wine nuevo no puede enterarse de que el viejo sigue reteniendo tu carpeta de MetaTrader. Sencillamente no abre, y no dice nada.
> - **Si quitas los paquetes del Wine viejo**, el acceso directo del menú de MetaTrader puede dejar de funcionar: Debian y Ubuntu traen un comando `wine-stable` que WineHQ no trae, y el acceso directo lo llama por ese nombre. Pulsas el icono y no pasa nada de nada. Edita el fichero `.desktop` y cambia `wine-stable` por `wine`.

Instalar MetaTrader 5 en Linux, o dejárselo a una IA

Que la conexión vaya fluida depende sobre todo de que MetaTrader esté bien puesto sobre Wine: Wine de WineHQ, un prefijo sólo para él (`~/.wine-mt5`, que esta aplicación encuentra sola) y ese prefijo en modo **Windows 10** —en Windows 7 falla de formas que no parecen de Wine—. El procedimiento verificado, paso a paso y con la comprobación de cada paso y sus cuatro trampas, viaja en la carpeta del puente como `INSTALAR-MT5-EN-LINUX.txt`.

Si prefieres delegarlo en un asistente de IA con acceso a tu terminal, dale `PROMPT-PARA-TU-IA.txt`, de la misma carpeta, o pulsa **Copiar el prompt para tu IA** en el panel de MetaTrader 5 cuando no encuentra ningún terminal. Lleva los mismos pasos y los límites que tiene que respetar: te pide permiso antes de cada `sudo`, no purga ni borra nada sin preguntar, no descarga el instalador —lo bajas tú de tu bróker—, no toca nunca tu usuario ni tu contraseña del bróker, se para y pregunta ante cualquier salida inesperada, y al final te enseña las cinco comprobaciones.

Si un instrumento vuelve vacío

MetaTrader sólo guarda velas de los gráficos que le han pedido dibujar. La primera petición de un instrumento que no ha abierto nunca le hace descargar antes el histórico — para varios años eso puede tardar un minuto o dos, y después es instantáneo. Si aun así vuelve vacío, abre ese gráfico una vez en MetaTrader y desplázate hacia atrás.

El horario del instrumento

Desde el conector 1.5.0, cada histórico de MetaTrader llega con el horario que tu bróker publica para ese símbolo: cuándo acepta órdenes y cuándo sólo da precio, día a día y en su hora de servidor, que es la misma de las velas. Lo ves en la ficha de costes. Y la auditoría de lo que llega lo usa: un día de sesión sin ninguna vela es un festivo y no un hueco —hasta tres seguidos; una semana entera ya es un agujero en los datos—, y lo que falta se cuenta contra las horas en que el bróker da precio, no contra las que el fichero suele tener. Lo que no sabe: es el horario de hoy. Si el bróker lo cambió —un índice que antes sólo cotizaba en horario de bolsa—, lo viejo puede salir con huecos que no lo son; y una víspera de fiesta que cierra antes cuenta como velas que faltan, porque MetaTrader publica la semana tipo y no el calendario de festivos. Con un conector anterior no llega, y todo sigue como antes: el horario se deduce de las propias velas.

El backtest no opera cuando tu bróker no deja. Lo que se decide al cierre de una vela —entrar, poner una orden en espera, salir por una regla o por el reloj— un robot lo manda con el primer precio de la vela siguiente, y sólo pasa si ahí se puede negociar; lo que espera en el servidor —la orden pendiente, el stop, el objetivo, el trailing— sólo se ejecuta en velas con horas de negociación. Un índice que cotiza desde la una y abre a las ocho ya no llena órdenes de madrugada; y si a las ocho el precio abre más allá del stop, el stop se llena a la apertura, como después de un fin de semana.

Después de una actualización

El programa tiene el conector al día él solo: cuando una versión nueva trae un conector nuevo, sustituye el fichero viejo en cada MetaTrader donde lo encuentra, nada más arrancar, y te lo dice. Un servicio en marcha se queda con el código con el que arrancó, así que MetaTrader sigue con el viejo hasta que se cierra y se vuelve a abrir; el panel lo dice, y cómo cambiar sin cerrarlo: en MetaTrader, Navegador → Servicios → BotFactoryFX → Detener, y luego Iniciar.

Y cuando el conector no contesta, el panel lee los ficheros del propio MetaTrader para decirte por qué: si MetaTrader está cerrado o abierto, y si al conector le falta estar instalado, estar al día, estar puesto para arrancar, o sobra una copia. Sea lo que sea, lo arregla el botón de al lado.

Conectar MetaTrader 4

MetaTrader 4 tiene su propio puente, el gemelo de éste: hace las mismas preguntas y contesta lo mismo —velas, la ficha del contrato, la cuenta, lo abierto y lo cerrado—, y habla el mismo protocolo, por eso lleva su mismo número de versión. Cambia lo que es. MQL4 no tiene servicios, y el puente es un **indicador** que va encima de un gráfico, de cualquiera: no dibuja nada, y el símbolo y la temporalidad le dan igual. Un indicador y no un experto porque, medido en un MetaTrader 4 de verdad, con el botón **AutoTrading** apagado un experto no recibe ni un evento de su temporizador y un indicador sigue funcionando; y porque MetaTrader 4 sólo deja operar a los

expertos y a los scripts: como indicador, el puente no podría mandar una orden aunque alguien la escribiera dentro. Tampoco ocupa el hueco de experto del gráfico, así que puede ir en el mismo gráfico que un robot.

Lo instala el mismo botón, **Instalar el conector**, en el panel de MetaTrader 4 de **Ajustes → Plataformas de trading**; también está en la pantalla **Datos**, eligiendo **MetaTrader 4** entre las fuentes y abriendo **Importar desde MetaTrader 4**. Copia el indicador a la carpeta `MQL4/Indicators` del terminal y, **con MetaTrader 4 cerrado**, añade a su perfil un gráfico H1 que lo lleva encima, y que MetaTrader 4 abre cada vez que arranca. Así que cierra MetaTrader 4, pulsa el botón y vuelve a abrirlo: en ese gráfico, abajo a la izquierda, una etiqueta que empieza por `BotFactoryFX Bridge` dice que el puente está en marcha y es de sólo lectura, y el panel de aquí se pone verde por su cuenta.

Con MetaTrader 4 abierto, el botón sólo copia el fichero: al cerrarse, MetaTrader 4 reescribe su perfil desde lo que tiene abierto, y un gráfico añadido por fuera se perdería. El panel lo dice, y ofrece el camino que no espera: en el **Navegador** de MetaTrader 4, arrastra **Indicadores → BotFactoryFX** a cualquier gráfico. Arranca en el acto, y MetaTrader 4 lo guarda en el perfil.

Habla en una carpeta propia dentro de la carpeta común, no en la del puente de MetaTrader 5: los dos MetaTrader la comparten en Windows, y también en un prefijo de Wine que tenga los dos, y cada puente recogería las preguntas del otro. MetaTrader 4 guarda *órdenes* y no *deals*: cada orden cerrada llega como su entrada y su salida, y sólo las cerradas. Y sólo entrega lo que enseña su pestaña **Historial de cuenta**: si llegan sólo los últimos meses, ahí, con el botón derecho, elige **Todo el historial**.

Si un histórico no está todavía en MetaTrader 4, el puente no se queda esperando: sigue contestando lo demás, vuelve a probar durante 20 segundos y, si no llega, te dice que abras ese gráfico en MetaTrader y te desplaces hacia atrás para que se descargue.

Al actualizar, el programa sustituye el indicador viejo por el nuevo en cada MetaTrader 4 donde lo encuentra, sin tocar su perfil; el que ya está en marcha sigue con su código hasta que MetaTrader 4 se cierra y se vuelve a abrir. Para quitarlo, cierra el gráfico del puente y borra el indicador de `MQL4/Indicators`. Su panel lleva los mismos tres botones de la carpeta del puente, que se guarda en `~/.botfactoryfx/conectores/metatrader4` con su manual.

Y tus robots: en la **Biblioteca**, el menú **Descargar** de una estrategia lleva, bajo **Enviarla a la carpeta de la plataforma**, **MetaTrader 4**, que deja el `.mq4` en `MQL4/Experts/BotFactoryFX`; luego se abre en MetaEditor, se compila con `F7` y se arrastra a un gráfico. Los robots MQL4 que exporta esta versión escriben su latido y su diario junto al puente de MetaTrader 4, y los de antes, junto al de MetaTrader 5: vuelve a exportarlos para verlos con su terminal.

Conectar JForex

BotFactoryFX puede sacar el histórico de precios directamente de una plataforma JForex en marcha en esta máquina — todos los que ofrezca tu cuenta, los tengas suscritos o no, cualquier temporalidad, sin exportar un fichero a mano. Lee la plataforma y **nunca manda una orden**.

Cómo se conecta

Con la misma idea que MetaTrader 5: un añadido pequeño que se llama el **puente** —*BotFactoryFX Bridge*— que corre dentro de la plataforma y habla con esta aplicación escribiendo ficheros, aquí en `~/.botfactoryfx/jforex`. Sin puerto, sin lista de permitidos, sin regla del cortafuegos. La diferencia está en qué es el añadido: JForex no tiene servicios, ejecuta *estrategias* escritas en Java. El puente viaja de dos maneras: compilado, en un `.jfx`, que es lo que JForex arranca, y en su fuente, `BotFactoryFX.java`, para que puedas leer cada línea antes de ejecutarlo. El compilado sale de ese fuente con el compilador de la propia JForex y sus

mismas opciones, y es, byte a byte, el que escribe JForex cuando lo compila; no lleva nada de tu cuenta ni de tu máquina. Atiende una petición cada vez, y por eso el panel pone en cola un segundo instrumento en vez de pedir los dos a la vez. JForex sólo da velas de lo que tienes suscrito, así que, para una descarga, el puente suscribe él el instrumento y lo quita cuando lleva un minuto sin pedirse: tu lista de JForex queda como la dejaste.

Instalarlo, una sola vez

En Linux y en Windows lo instala esta aplicación, entero y de una vez:

1. Aquí, en **Ajustes** → **Plataformas de trading**, pulsa **Instalar el puente en JForex** en el panel de JForex (es el mismo que tiene la pantalla **Datos** al elegir **JForex** entre las fuentes). La aplicación encuentra tu JForex —en Linux, por el acceso directo que deja su instalador o en `~/JForex4`; en Windows, por lo que su instalador apunta en el registro o en `C:\Program Files\JForex4`—, deja el puente en su carpeta de estrategias, compilado y con su fuente, y añade dos líneas a las opciones de JForex (`JForex4.vmoptions`, con una copia de las de antes al lado) para que lo arranque al abrirse.
2. Si JForex estaba abierto, ciérralo. Ábrelo y entra en tu cuenta: el puente arranca solo, en tu ordenador, unos segundos después (medidos: 13 en Linux, unos 20 en Windows), y el panel de aquí se pone verde por su cuenta. Desde entonces arranca cada vez que abres JForex, sin que tengas que abrir ni arrancar nada dentro.

Si tu JForex ya arranca otras estrategias al abrirse, esa configuración no se toca —arrancarían también las tuyas, y ésas pueden operar—, y el panel te dice que arranques el puente a mano.

En Windows, JForex se instala en `C:\Program Files\JForex4`, que es a la vez su carpeta de datos: ahí guarda sus estrategias, y ahí escribe esta aplicación. Está medido con el JForex para Windows sobre Wine; queda por medir en un Windows de verdad si su instalador deja escribir en esa carpeta a un usuario que no es administrador. Si no deja, el panel lo dice con la ruta: da permiso de escritura a tu usuario en esa carpeta y vuelve a pulsar el botón.

En Mac —donde la aplicación todavía no busca JForex, porque allí no está medido— o si no encuentra el tuyo, se hace a mano:

1. Abre JForex y entra en tu cuenta.
2. Aquí, pulsa **Copiar el conector**: deja el puente, compilado y con su fuente, en `~/.botfactoryfx/jforex`, y enseña la ruta del compilado, con un botón para copiarla.
3. En JForex, en la pestaña **STRATEGIES**, pulsa **Abrir** —el icono de la carpeta— y pega esa ruta en la casilla del nombre del fichero: la carpeta empieza por punto, y en Linux y en Mac el diálogo no la enseña.
4. Arráncalo en tu ordenador: **Start** → **Local Run** (con **Remote Run** JForex lo rechaza, porque pide acceso completo). Viene compilado, así que arranca sin más. Déjalo corriendo: el panel de aquí se pone verde por su cuenta.

El permiso que pide JForex

La primera vez que el puente arranca en una cuenta, JForex enseña un diálogo —**Full Access Disclaimer**— preguntando si la estrategia puede tener **acceso completo**. Acéptalo con **I Agree**. Dukascopy guarda la respuesta en su servidor, con la cuenta: con una cuenta que ya lo aceptó, JForex no pregunta nada, tampoco en otro ordenador. La estrategia lleva la anotación `@RequiresFullAccess` porque la política de JForex sólo deja a una estrategia escribir dentro de su propia carpeta de ficheros, y el puente tiene que escribir sus respuestas en `~/.botfactoryfx/jforex`, que es donde esta aplicación las lee. El permiso va de *dónde puede escribir*, no de operar: el puente no tiene ningún código que mande, cambie o cancele una orden, y puedes comprobarlo en su fuente. El manual enseña el diálogo.

La carpeta del puente

El panel lleva además tres botones: **Guardar la carpeta del puente (conector, manual, icono)** copia la carpeta entera —el puente compilado y su fuente, el icono, un `LEEME.txt` y el manual en PDF— a `~/.botfactoryfx/conectores/jforex` y enseña la ruta; **Abrir el manual** la guarda y abre el PDF; **Abrir la carpeta** la guarda y la abre en tu gestor de ficheros.

Después de una actualización

Una estrategia en marcha se queda con el código con el que arrancó. Así que una versión nueva de BotFactoryFX trae un puente nuevo, pero el que JForex está corriendo sigue siendo el viejo. Si lo instaló esta aplicación, el panel, en cuanto lo abres, deja el nuevo en la carpeta de estrategias de JForex él solo y te lo dice: entra en marcha la próxima vez que abras JForex. A mano: cópialo otra vez, para la estrategia y vuelve a arrancarla. El panel compara la versión que corre con la que él lleva, y te lo dice.

Si algo no contesta

- **No contesta ninguna plataforma JForex**: ¿está corriendo la estrategia? Instalada por esta aplicación, arranca al abrir JForex y entrar en tu cuenta, no antes: si JForex ya estaba abierto al instalarla, ciérralo y vuelve a abrirlo. Su línea de consola dice `BotFactoryFX Bridge … escribiendo en …`, y escribe un fichero de estado cada pocos segundos; si la plataforma se ha cerrado, el panel se entera solo.
- **Sólo salen los que tienes suscritos**: corre un puente de una versión anterior, que sólo ofrecía eso. Instalado por esta aplicación, cierra JForex y vuelve a abrirlo; a mano, cópialo otra vez y relanza la estrategia.
- **En STRATEGIES hay entradas que no pusiste tú**, con rutas de otro ordenador: esa lista vive en la nube de tu cuenta, y quien usa la misma cuenta en dos ordenadores ve allí las del otro. Para cargar el puente a mano, elige el fichero con **Abrir**, no una entrada de la lista: abierta desde la lista, JForex la compila en su nube y no deja nada en el disco. El arranque automático no usa la lista.
- **Una petición parece atascada**: JForex atiende una petición cada vez, y traer años de histórico de un instrumento que nunca ha dibujado lleva un rato. La tira de descargas de arriba de la pantalla de Datos enseña lo que está en camino, y una descarga grande avanza por tramos de 20.000 velas.

Y tus robots: en la **Biblioteca**, el menú **Descargar** de una estrategia lleva, bajo **Enviarla a la carpeta de la plataforma, JForex**, que deja el `.java` en la carpeta de estrategias de tu JForex, la misma del puente. Funciona en Linux y en Windows; en Mac todavía no: descárgalo y ábrelo en JForex. Luego, en JForex, se abre con **Abrir** en la pestaña **STRATEGIES** y se arranca cuando quieras que opere: **Start → Local Run**.

Trabajar con TradingView

TradingView y MetaTrader se conectan en direcciones contrarias, y conviene saber por qué antes de ponerse a buscar un botón que no existe.

> **No hay enlace automático, y no puede haberlo.** MetaTrader es un programa de tu disco con una carpeta de añadidos, que es la que usa el conector de BotFactoryFX. TradingView es un sitio web: Pine Script corre aislado en sus servidores y no puede leer ni escribir nada en tu máquina. TradingView tampoco publica *ninguna API de datos de mercado* para particulares — sus precios están licenciados de los mercados con condiciones que no dejan redistribuirlos. Las herramientas que dicen lo contrario están haciendo ingeniería inversa de un canal privado sin documentar, y dejan de funcionar sin avisar.

Traer el histórico de precios

Usa la exportación del propio TradingView: en la barra de herramientas de arriba del gráfico, **Download chart data…** (PRO+ y Premium). Luego suelta el CSV en la pantalla Datos de aquí. Se lee tal como viene —tanto la variante con marca de tiempo ISO como la de segundos Unix— y las columnas de indicadores que TradingView añade se ignoran, en vez de confundirse con el precio.

Sacar una estrategia

Monta la estrategia aquí, pulsa **Exportar…** y elige **Pine Script** en el menú de la Biblioteca. Abre el editor de Pine en un gráfico de TradingView, pega, guarda y añádela al gráfico. La pestaña Strategy Tester la corre entonces sobre los datos de TradingView.

Comprobar que las dos coinciden — la parte que importa

Dos backtests de las mismas reglas sobre el mismo instrumento deberían dar la misma curva de capital. Cuando no la dan, uno de los dos está mal, y saber *cuál* es justo para lo que sirve un Reality Check.

Así que trae de vuelta la respuesta de TradingView: en el Strategy Tester, abre la pestaña **List of Trades** y expórtala. Luego ve a **Reality Check → Una de fuera**, elige TradingView como plataforma, adjunta ese fichero y pulsa **Ejecutar la mía sobre este histórico**. Las dos quedan una al lado de la otra, en una sola tabla.

> **Exporta la List of Trades, no el Performance Summary.** El resumen es una página de totales, y un total que nadie puede recalcular es un total que nadie puede comprobar. Todas las cifras que se enseñan aquí se recalculan desde las operaciones una a una — incluido el capital inicial, que TradingView no dice en esa exportación pero que sus columnas de beneficio y de beneficio por ciento delatan entre las dos.

Qué suele querer decir una diferencia

- **Los costes.** El guion exportado no pone ninguno —ni spread, ni comisión, ni slippage, ni carry— porque la exportación no se lleva los que cobró el backtest, y lo dice arriba del todo. Los totales de TradingView salen entonces mejores que los del backtest en lo que el backtest pagó, o peores en el carry que cobrara. Poner *Commission* y *Slippage* en las *Properties* del guion acerca los totales, no las operaciones una a una.
- **Los datos.** El feed de TradingView para un símbolo no es el de tu bróker. Otras sesiones, otros cierres, otros huecos.
- **La vela en la que entra la orden.** El backtest entra a mercado al cierre de la vela de la señal, y el guion exportado hace lo mismo — corre con `process_orders_on_close` puesto. Un guion tuyo dejado con lo que TradingView trae por defecto entra a la apertura de la vela siguiente, que después de un hueco es otro precio.

Biblioteca

Las estrategias que guardaste, cada una con un **informe exhaustivo**. Maestro-detalle: la lista a la izquierda —se busca en ella y se ordena por fecha, por nombre o por símbolo— y el detalle a la derecha.

- Los KPI de un backtest sobre el histórico cargado (P&L, acierto, factor de beneficio, drawdown máximo, Sharpe).
- La curva de capital, interactiva.
- Las reglas en cristiano.
- La gestión del riesgo y la tabla entera de operaciones.

- La definición JSON, desplegable, para mirarla por dentro.
- **Descargar** — la estrategia como fichero para cualquiera de las seis plataformas, o como JSON. Lleva la ficha del instrumento de la serie que esté abierta en ese momento. **Importar** vuelve a leer un JSON de ésos.
- **Editar** la abre en Construir › Manual; **Informe completo** abre el informe entero.
- **Duplicar** — una copia con otro nombre para cambiarla sin tocar la original: recuerda de cuál viene y se lleva su veredicto y su cuenta de variantes.
- **Notas y etiquetas** — las de cada estrategia; la lista se filtra por etiqueta, y la búsqueda también las mira.
- **Comparar con…** — dos estrategias lado a lado: en qué se diferencian sus reglas, sus cifras sobre el histórico cargado y sus dos curvas.
- **Explícamela** — tu proveedor de IA (el de Ajustes) cuenta en lenguaje llano qué hace la estrategia y qué significa su veredicto. Sólo al pulsar el botón, y lo que viaja son sus reglas en palabras y su veredicto, nada más. Queda guardada; si luego cambias las reglas, la tarjeta avisa de que caducó.

En el **informe completo**, el factor de beneficio y el resultado medio van con su margen del 90 %, sacado de volver a sortear las operaciones —con pocas, sale ancho, y es lo que tiene que verse—; y entre lo que dice debajo de los gráficos está el stop que habría conservado nueve de cada diez ganadoras, al lado del que tiene la estrategia.

> **Juzgada desde aquí, una estrategia lleva cuenta sólo si coincide.** Al Reality Check se le dice cuántas variantes se probaron para encontrar una estrategia elegida en la Biblioteca sólo cuando sus reglas, su riesgo y su gestión son los mismos que los de la estrategia abierta en Construir que lleva cuenta. Ver *La búsqueda que la encontró era grande*, en Reality Check.

Cartera

Combina varias estrategias, cada una con su **asignación** (su peso). El saldo y el tamaño del lote se escalan en proporción.

Cada estrategia se simula sobre SU histórico —su símbolo, su temporalidad, su diferencial y su comisión— y las curvas se combinan por fecha. Una cartera de varios instrumentos es el caso normal, y esto es lo que la hace medible: una pata vale lo que valía donde se juzgó. La que no tenga su histórico cargado se queda fuera y se dice cuál; nunca se corre sobre las velas de otro instrumento para que salga un número.

- **Ver sus curvas antes de combinar** — cada candidata sola, sobre su histórico, con todo el saldo inicial y sin reparto, en un mismo gráfico: para ver si caen a la vez antes de juntarlas.
- Los KPI agregados y la curva de capital, con el desglose por estrategia, y cada fila diciendo sobre qué histórico corrió.
- La matriz de correlación entre pares (sobre los rendimientos de la curva de capital).

> **La correlación.** De +0,7 a +1: van en paralelo (poca diversificación). De -0,3 a +0,3: independientes. De -1 a -0,7: opuestas (se cubren). Se mide sólo donde las dos patas tienen histórico, y si se solapan demasiado poco la pantalla lo dice en vez de dar un número.

> **Patas que no cubren las mismas fechas.** Fuera de su histórico, una estrategia cuenta como capital parado: su parte del saldo, sin operar. No se inventa nada ni se tira histórico, pero la caída conjunta cubre tramos en los que parte de la cartera no podía operar. La pantalla lista las patas que empiezan tarde o acaban pronto, y entre qué fechas; un hueco dentro de un histórico cuenta igual y no se lista, porque eso es un defecto del fichero y sale en Datos.

Una cartera se puede guardar con un nombre, volver a cargarla y actualizarla. Una cartera guardada antes del 23 de septiembre de 2026 lleva cifras medidas a la manera vieja —cada pata sobre el histórico que estuviera abierto aquel día— y lo dice al reabrirla, con un botón para volver a simularla. Al llegar de una pasada del piloto sin nada seleccionado, ya vienen marcadas las estrategias que su propia cartera combinó — las que salieron sólidas, que son las únicas que combina. Las otras que elegiste guardar al final de la pasada están en la Biblioteca, pero aquí no vienen marcadas. Si no combinó ninguna, no se marca nada, y la pantalla lo dice en vez de dejarte una página vacía — lo que encontró está en la Biblioteca, con su veredicto. Cuarentena y En vivo eran pestañas de esta pantalla y ahora son secciones propias, en el orden del trabajo: Cuarentena antes de Cartera, En vivo después. Esta pantalla hace una sola cosa — combinar estrategias y simularlas juntas.

DESPUÉS DEL BACKTEST

Cuarentena

Sección propia, entre Biblioteca y Cartera, porque ése es su sitio en el trabajo: el Reality Check decide qué merece guardarse, la Cuarentena qué merece dinero, y a una cartera sólo llega lo que pasó las dos. Un backtest mide el pasado; operar mide el presente con dinero encima. La cuarentena es lo que va en medio: la estrategia corre sobre las velas que tu bróker está publicando, sin mandar ninguna orden, y se va formando un expediente que puedes poner al lado del backtest que la aprobó.

No caduca, una estrategia tuya no pide cuenta en ninguna parte, y no hay tope de cuántas tengas dentro.

Meter una

- **Estrategia** — una de la biblioteca. Se copian su nombre y las cifras de su último backtest, así que el expediente se sigue leyendo aunque luego borres la estrategia. Si su veredicto está vigente, el expediente apunta también su nota y cómo entró —si pasó el listón o se quedó *cerca del listón*: 45 o más, o sólida sin su hermano—, para comparar después cómo le va a cada grupo en vivo. Su tarjeta enseña la nota: en verde si pasó el listón, en ámbar si no. El botón *Mandar a cuarentena* del piloto la trae aquí ya elegida. Si su stop y su objetivo van en pips, el expediente avisa también cuando el mercado se mueve ahora 1,5 veces más por vela que en el tramo que juzgó el Reality Check, o 1,5 veces menos: en proporción, sus stops ya no son los que se juzgaron, y en un mercado que se ha doblado unos stops en pips se quedan estrechos y el vaivén normal los alcanza. Si salió de una búsqueda con stops en pips, un recuadro bajo el formulario pregunta qué hacer con ellos: **adaptarlos al ATR**, lo recomendado y lo marcado de partida, o **mantenerlos fijos**. Adaptados, miden exactamente lo mismo en la última vela que juzgó el Reality Check, y desde ahí crecen y se encogen con la volatilidad; medido en ocho mercados, las estrategias que el Reality Check no rechazó batieron después al azar un 75 % de las veces en vez de un 63 %. El expediente dice entonces *stops en ATR*. A una estrategia hecha a mano no se le pregunta: sus pips son tuyos.
- **Bróker o sitio** — Hyperliquid, cualquier terminal de MetaTrader que tengas abierto, o JForex cuando su conector está corriendo. Es media pregunta, no un detalle: la misma estrategia sobre el mismo instrumento se comporta distinto en cada sitio, porque cambian la horquilla, el carry y las horas con liquidez.
- **Sin cuenta en ningún sitio** — Binance (cripto al contado) y Dukascopy (divisas, índices, metales y acciones) dicen *sin cuenta*: sus velas son públicas, así que puedes meter una estrategia en cuarentena antes de abrir cuenta en ningún bróker. Son los precios de ese sitio: si vas a operar en otro bróker, su horquilla y sus horas diferirán un poco. Dukascopy publica los días recientes de fichero en fichero, así que la primera puesta al día tarda unos minutos; después pide sólo los días nuevos. Cada expediente dice qué costes cobra —diferencial, deslizamiento y comisión— y de dónde salen: los del histórico de la estrategia, o los de fábrica si ese histórico no está cargado. Si tu bróker cobra más, tu resultado será peor que el del expediente.
- **Instrumento** y **vela** — lo que ofrezca ese sitio, de M1 a D1.

Cada vez que abres la sección se piden las velas nuevas y la estrategia se vuelve a correr sobre todo lo acumulado: las velas de cada cuarentena se guardan en este ordenador y las nuevas se les añaden. No hay nada corriendo en segundo plano: el programa no tiene que estar abierto para que el mercado siga, sólo para recoger lo que pasó. Lo único que no puede recoger es lo que el bróker ya no guarda: cada visita alcanza unos miles de velas hacia atrás —unos días en velas de un minuto, meses en velas de una hora—, y si pasa cerrada más tiempo, falta ese trozo y el expediente lo dice.

Un robot que ya opera

Arriba de la tarjeta se elige qué entra: **Una estrategia tuya** o **Un robot que ya opera**. Lo segundo es un robot que te vendieron, frente al backtest con el que te lo vendieron. Su código no se ejecuta nunca —se lee para los

números que declara— y lo que se compara es su promesa contra las operaciones que la cuenta donde corre hizo de verdad, así que aquí las fuentes son cuentas: MetaTrader 5, MetaTrader 4 o JForex con su conector, o un monedero de Ajustes. Es lo que hace BFQuarantine. En una cuenta de MetaTrader eliges además qué robot, por su número mágico, y si contar desde su primera operación allí o desde hoy.

El veredicto mira primero el dinero

Lo que importa es si gana el dinero que prometía su backtest, así que el veredicto compara el **factor de beneficio** —lo que ganaron las ganadoras dividido entre lo que perdieron las perdedoras— en vivo con el prometido. Con pocas operaciones un factor de beneficio dice poco, así que se le pone una banda remuestreando sus propias operaciones dos mil veces: sólo dice *va peor de lo que prometía su backtest* cuando lo prometido queda por encima de esa banda con un 90 % de seguridad, y nunca antes de 20 operaciones. El acierto también se enseña, como dato: un robot puede acertar menos de lo prometido y ganar su dinero, o acertar lo mismo y perderlo. Cinco respuestas: todavía sin operaciones, todavía es pronto, pierde dinero pero aún no claramente peor de lo prometido, gana dinero como prometía (o mejor), y va peor de lo prometido. Medido con 32 robots reales hechos con StrategyQuant X frente a lo que hicieron en meses que no habían visto, así se pillaron 12 de los 21 que perdían dinero sin una sola falsa alarma entre los 11 que no; mirando sólo el acierto se pillaban 10 y salían 4 falsas alarmas. Si el backtest no da factor de beneficio, se compara el acierto como antes, con un intervalo de Wilson del 95 %.

Una segunda oportunidad para lo que se quedó cerca

Una estrategia que entró *cerca del listón* no lo pasó del Reality Check, y la cuarentena es su segunda oportunidad. Cuando su veredicto dice que *gana dinero como prometía su backtest* —para lo que hacen falta al menos 20 operaciones en vivo—, el expediente la marca como **Aprobada en cuarentena**, y la Biblioteca lo enseña junto a su veredicto. Hasta entonces, la tarjeta dice cuántas operaciones lleva y, más o menos, cuántas le hacen falta. El listón no ha bajado: lo que la aprueba es un mercado que nunca había visto, no una nota. Y no es para siempre: si se va, la marca desaparece.

> **El gráfico dibuja lo que la estrategia habría hecho** — cada entrada y cada salida sobre las velas que han llegado —, y no órdenes en espera.

> **Esto no es una cuenta demo.** No manda ninguna orden a ninguna parte, así que no mide slippage, ni llenados parciales, ni rechazos, ni latencia: cosas que una demo de bróker sí toca hasta cierto punto. Lo que mide es si la estrategia se sigue pareciendo a su propio backtest sobre un mercado que es el de verdad. Para lo otro está **En vivo**. Y tampoco depende de una: no caduca a las pocas semanas como la demo de un bróker, ni necesita un servidor encendido día y noche.

Cuentas en vivo y monederos

La sección En vivo, la última del menú, lee cuentas de verdad y dibuja sus operaciones sobre el gráfico. Un robot suelto contra un mercado opera a ciegas: los llenados salen por una consola y el gráfico está en otra ventana, de otro programa y a otra escala.

- **Hyperliquid** — desde la dirección del monedero. Capital, flotante, posiciones abiertas, operaciones cerradas y la curva de la cuenta.
- **MetaTrader 5** — por el mismo conector que trae el histórico, sin clave y sin puerto abierto. Arriba, la cuenta: balance, capital, flotante, nivel de margen y lo que hay abierto, con el robot de cada posición (hace falta el

conector 1.4.0). Lee *operaciones* y no apuntes: empareja cada entrada con su salida, descuenta comisión y swap, y separa los robots por número mágico. Pide sólo las operaciones nuevas, no el historial entero cada vez.

- **JForex** — por su conector, como MetaTrader: una estrategia dentro de la plataforma que sólo lee. Debajo de la cuenta de MetaTrader, su cuenta: el balance al cierre de ayer, el capital, el flotante —capital menos capital base, porque el balance de JForex es el del día anterior—, el nivel de margen y lo que hay abierto, con el robot de cada posición (hace falta el conector 1.3.0).
- Mientras MetaTrader 5, MetaTrader 4 o JForex no están conectados, esta sección ofrece instalar su conector, como Datos.

> **Leer una cuenta de Hyperliquid no necesita ninguna clave.** Su API contesta a la dirección, que es pública y está en la cadena; la «API wallet» que ofrece el mercado es para *firmar órdenes*, y aquí no se firma nada. Guardar un secreto que no se usa nunca es riesgo sin beneficio, así que no se pide.

Cada robot, con su estrategia y su deriva

Cada robot exportado a MetaTrader o a JForex lleva un número mágico sacado de su estrategia, que no cambia al renombrarla —en JForex viaja dentro de la etiqueta de cada orden—. En *Tus robots* eso basta para ponerle al lado su estrategia de la Biblioteca y medir si sigue haciendo lo que prometía: se corre la estrategia sobre las velas de la misma plataforma, se toman las operaciones que cerró *antes* de ponerla en marcha, y se mira dónde caen las que el robot ha cerrado desde entonces. Por debajo de 20 operaciones cerradas dice que todavía es pronto, y cuántas faltan. JForex sale en el desplegable junto a los terminales. Cuenta los tamaños en millones, así que fuera de los pares de divisas no se dice cuántos lotes lleva una operación, y la deriva compara dinero sin normalizar por lote, como avisa. Los robots exportados con versiones anteriores llevan todos el mismo número, 20250810, y no se pueden atribuir solos: elige en el desplegable de qué estrategia es cada uno.

Si al exportarla desde la Biblioteca elegiste **adaptar sus stops al ATR**, el robot no lleva los pips de la Biblioteca, y su backtest tampoco: el gráfico, la deriva y el diario decisión a decisión corren la estrategia con los stops en ATR con que salió hacia esa plataforma —la última vez que se exportó a ella— y lo dicen debajo. Si se dejaron fijos, o se exportó antes de que existiera la elección, con los pips de siempre.

Lo que costó entrar

Debajo del gráfico, al lado de lo que operó el backtest, sale lo que costaron de verdad las entradas a mercado: el precio al que se llenaron contra el cierre de la **vela de la señal**, la última cerrada cuando se llenaron, que es la que un robot que decide al cierre acaba de ver. Sólo cuenta lo que se llenó en los primeros segundos de una vela —el 10 % de la vela, entre 5 segundos y 2 minutos—: una orden en espera o una entrada a mano se alejan de ese cierre a propósito, y se dice cuántas quedan fuera. Con el pip de cada operación —los decimales del bróker en MetaTrader, lo que dice JForex, la regla de los perpetuos en Hyperliquid— y compras y ventas por separado: las velas de MetaTrader y de JForex son precios de oferta (bid), así que una compra se llena al de demanda (ask) y su cifra lleva dentro el diferencial; una venta lo paga al salir. Compáralas con lo que el backtest cobra al entrar: el diferencial, y el deslizamiento en las órdenes stop. Y el backtest de al lado corre con la ficha de ese instrumento y, si no se le dan, con los costes de su familia: nunca a coste cero.

Decisión a decisión

Los robots de MetaTrader (MQL5 y MQL4) exportados desde esta versión llevan un diario: en la carpeta común de los terminales, junto al conector, apuntan cada vela que deciden —su cierre como lo vieron, lo que dijeron sus tres reglas (la de salida, la de largos y la de cortos) y, si no entraron, por qué: una operación abierta, una orden esperando o los topes del día— y cada orden que mandan, con el precio pedido, lo que contestó el bróker y el precio al que se llenó. Un fichero por mes; no se manda nada a ninguna parte, y en el Probador no se escribe.

Los robots de JForex exportados desde esta versión llevan el mismo diario, en la carpeta de ficheros de JForex —la que una estrategia usa sin pedir acceso total—, dentro de `BotFactoryFX`. Cada orden se apunta cuando JForex contesta, con lo que contestó; nada en el robot espera esa respuesta, así que opera igual que sin diario. Al lado dejan su latido, y BotFactoryFX avisa del robot de JForex que deja de latir mientras llegan precios, que deja de decidir o cuyo corte salta. Sólo JForex sabe dónde está esa carpeta: se lo dice a BotFactoryFX su conector (1.3.0). En el Probador histórico de JForex no se escribe nada, y una estrategia que corre en los servidores de Dukascopy lo escribe allí, donde BotFactoryFX no lo puede leer.

En *Tus robots*, debajo de la deriva, cada robot con diario se pone al lado de su backtest sobre las mismas velas del mismo terminal, o de JForex, en el reloj con que cuenta el robot y empezando plano donde empezó él: señal a señal y decisión a decisión —entradas, órdenes de entrada, sus llenados y caducidades, y salidas por la regla o por tiempo; el stop y el objetivo los ejecuta el bróker, no el robot, y se comparan en la deriva—. Cada diferencia sale con su causa: el robot estaba apagado, el bróker la rechazó (y con qué código), las reglas leyeron distinto, o uno de los dos tenía la puerta cerrada. Lo que viene detrás de la primera se marca como consecuencia, porque veinte diferencias que salen de una sola no son veinte problemas. Una señal distinta sobre la **misma** vela es un fallo de la exportación, y conviene avisarnos; sobre **otra** vela, el histórico no es lo que el robot vio. Y de lo que se llenó a mercado, lo obtenido contra lo pedido: el deslizamiento de verdad, orden a orden.

> **Los robots exportados antes no escriben diario.** Expórtalo otra vez desde la Biblioteca y vuelve a ponerlo en su gráfico. Los de Python y Pine todavía no llevan diario.

Los monederos se añaden en Ajustes

Una dirección es de quien tiene la llave, no de un sitio, así que la lista no es «la cuenta de Hyperliquid» sino los monederos que quieras, con el nombre que tú les pongas. Al pegar una dirección se prueba en los sitios que se pueden leer y se rellena lo demás. Hyperliquid enseña dos direcciones distintas: la de tu cuenta, en el menú de tu cuenta arriba a la derecha, y la de su ventana «Depositar», que es un buzón que reenvía el dinero a la cuenta. Es la primera la que hay que pegar; con la segunda, el programa avisa de que ahí no hay ninguna cuenta. El campo quita además los caracteres invisibles que arrastra un copia y pega desde una página, que antes hacían rebotar una dirección buena sin que se viera nada raro.

Inicio deja elegir qué curva estás mirando

Simulada, cuarentena o en vivo — y debajo del título dice cuál de las tres: lo que *habría* pasado sobre el histórico que importaste, lo que *habría* pasado sobre las velas que han llegado desde entonces, o lo que *pasó* con dinero. No se suman nunca; cada una arranca de su propia línea.

Dr. Quant

Dr. Quant vive en dos sitios. En la conversación del **piloto automático**: lo que escribas ahí que sea un encargo de construir se lee en código y se construye. Desde el 08-10-2026 hay dos órdenes más que también lee el programa, sin gastar tu clave: **juzgar una estrategia guardada** —«juzga Ruptura 7», o «júzgala» con una abierta— y **llevarla a la selección de Cartera o sacarla** —«lleva Ruptura 7 a la cartera»—. Las dos enseñan antes una ficha con lo que va a pasar exactamente —sobre qué histórico corre el juicio, con qué cuenta de variantes— y nada corre sin tu clic en esa ficha; el juicio es el mismo Reality Check de siempre, sobre el histórico de la propia estrategia, y su veredicto se guarda con ella. Una pregunta —una duda sobre una métrica, qué hacer ahora— va a tu proveedor de IA como conversación; y una orden que no es ninguna de ésas —«optimiza el periodo del RSI»— no se ejecuta desde ahí: Dr. Quant te dice en qué pantalla se hace. **Qué viaja con una pregunta**: la conversación, el diario de veredictos resumido —lo que ya probaste, para que no te lo proponga como idea nueva— y sólo lo que la pregunta

nombre de lo tuyo, resumido por el programa: si preguntas por tu Biblioteca, los nombres con su veredicto; si nombras una estrategia, su ficha del juez; si preguntas por tus datos o tus carteras guardadas, lo suyo. Tus reglas, tus velas y tus operaciones no viajan nunca desde aquí. Y en la pantalla de **Datos**, donde **Ask Dr. Quant** busca las condiciones de tu bróker para rellenar lo que te cuesta operar (ver *Dr. Quant en la web*, más abajo).

Contesta como a quien empieza: dice qué significa cada término la primera vez que lo usa, y acaba en lo siguiente que mirar. Cuando la respuesta no está en lo que sabe de este programa, lo dice y te manda al capítulo de esta Ayuda que lo cuenta, en vez de rellenar el hueco con cómo funcionan otros programas. Cuando preguntas por el Reality Check —el veredicto, una comprobación, qué recorta la nota, el sobreajuste— se le cuenta además cómo juzga el juez de este programa, sacado del propio programa: las comprobaciones con su peso, dónde empieza cada palabra del veredicto y qué recorta la nota. Nada de tu estrategia va con ello. Y cuando la pregunta va del programa, le llegan los trozos de esta Ayuda más cercanos a ella, copiados tal cual y con el capítulo del que salen, para que pueda mandarte allí. Todo eso tiene un techo —lo que se le cuenta antes de cada pregunta no pasa de unos 2.500 tokens, y tu clave lo paga en cada mensaje— y la parte que no cambia va delante, para que los proveedores que cobran menos el texto que se repite de un mensaje a otro te lo puedan cobrar más barato.

Ejemplos: «¿Es razonable un Sharpe de 1,8 con 12 operaciones?», «¿Cómo valido esta estrategia fuera de muestra?», «Propón 3 condiciones de salida para un cruce de medias».

> **Ya no hay botón de Dr. Quant en el menú, ni modo agente.** Si vienes de una versión anterior: el panel lateral, con sus dos modos, conversación y agente, se retiró. Dos puertas con el mismo nombre y poderes distintos eran peor que una, y el modo agente era un modelo improvisando sobre tu estrategia con herramientas y sin forma de repetir lo que había hecho — justo lo que este programa te enseña a no creerte. Lo que se le pedía, «búscame una estrategia para EURUSD en H1», lo hace ahora el piloto: lee el encargo en código y enseña lo que ha supuesto antes de empezar.

Dr. Quant funciona con tu propia cuenta de IA. Aquí dentro no viene ningún modelo y no se pasa nada por los servidores de nadie. Añade un proveedor en **Ajustes** con tu clave de la API. La lista empieza **vacía**, y la clave le dice al programa de quién es: `sk-ant-…` es Anthropic, `gsk_…` es Groq. Hay cinco clases: **OpenAI o compatible**, **Anthropic**, **Google Gemini**, **OpenRouter** —el que puede buscar en la web con cualquier modelo— y **En tu máquina**, que no pide clave y cuyo botón del modelo lista los que te hayas descargado. La primera y la última llevan una **Dirección**: vacía es OpenAI u Ollama, y cualquier servicio que hable el dialecto de OpenAI —Groq, DeepSeek, Mistral, LM Studio, un servidor tuyo— es su dirección pegada ahí. Una clave con la firma de uno de ésos, `gsk_…` de Groq, entra como *OpenAI o compatible* con la dirección ya puesta. Las preguntas te las cobran ellos a ti, a sus precios, y la clave se queda en esta máquina.

Debajo de cada respuesta, en pequeño, va lo que tu proveedor dice que costó: los tokens que leyó —y cuántos salieron de su caché, que cobra más baratos— y los que escribió. Cada pregunta lleva además la conversación de antes, y eso también se paga: **Conversación nueva**, arriba de la conversación, empieza otra vez desde cero, y **Copiar la conversación** la saca entera como texto.

Con más de un proveedor activo contesta el primero por prioridad, y si está saturado con un error de límite de peticiones (429) la pregunta pasa sola al siguiente activo.

El piloto no necesita nada de esto para construir: lee lo que le pides en código, no con un modelo. Al modelo sólo van sus preguntas.

Dr. Quant en la web, y lo que cuesta

Dr. Quant puede buscar cosas en internet: las especificaciones de contrato de un bróker, horquillas típicas, tablas de comisiones. Bajo esa misma palabra pasan dos cosas distintas, y se cobran distinto. Importa lo bastante como para decirlo claro.

Leer una página es gratis

Cuando le das a Dr. Quant una **URL**, el servidor del propio BotFactoryFX descarga esa página y le pasa el texto al modelo. Eso no te lo cobra nadie: ni tu proveedor de IA ni nosotros. Si pegas el enlace de la página de especificaciones de contrato de tu bróker en **Datos → Lo que te cuesta operar esto**, la consulta entera cuesta **cero**, y además lee la página oficial y no lo que a un buscador le haya parecido relevante.

Buscar en la web no es gratis, ni siquiera con un modelo gratuito

Una *búsqueda* es un servicio aparte que tu proveedor compra por ti, y se cobra aparte del modelo. **Que el modelo sea gratis no hace gratis la búsqueda.** OpenRouter lo dice con todas las letras: usar búsqueda web tiene coste añadido, también con modelos gratuitos.

A los precios de hoy de OpenRouter, una búsqueda cuesta unos **0,007 $** con el buscador por defecto (Exa, modo automático) o **0,005 $** con el modo básico de Parallel. En ese precio entran diez resultados; BotFactoryFX pide cinco, así que una consulta nunca se pasa en silencio al cargo por resultado de más.

En la práctica: buscar las condiciones de tu bróker cien veces cuesta unos **70 céntimos**. A ti no te cambia nada, pero es tu dinero en tu propia clave, así que se te dice en vez de sorprenderte.

> **La costumbre barata.** Pega la URL cuando la tengas, escribe el nombre del bróker cuando no. Lo primero es gratis y es mejor prueba; lo segundo cuesta menos de un céntimo.

> **Todo lo que se trae de la web se trata como información, nunca como instrucciones.** El texto descargado se acota antes de que el modelo lo vea, las direcciones privadas y locales se rechazan, y cada redirección se vuelve a comprobar — así una página no puede convencer a Dr. Quant de que toque tu biblioteca.

Preguntas frecuentes

¿Hay que instalar algo?

No. La aplicación de escritorio es el programa entero: la búsqueda genética, el Reality Check y la exportación corren dentro de ella, y tu trabajo se queda en esta máquina. Conectar una plataforma —MetaTrader 5, MetaTrader 4 o JForex— sí pide instalar su conector una vez, desde la pantalla de Datos o desde En vivo.

¿Qué formato CSV se admite?

Cualquier CSV con columnas de fecha y hora, OHLC y, si lo hay, volumen. El importador reconoce él solo las cabeceras más corrientes.

¿Cómo se calcula el P&L?

En pips, por lotes, por lo que vale un pip de un lote — y las tres cosas salen de la ficha del instrumento de la serie (ver *Los datos y las temporalidades*): el tamaño del pip, las unidades que tiene un lote, y de qué lado del par está el dólar. En EURUSD un pip es 0,0001 y un lote 100.000 unidades, así que un pip de un lote son 10 $. En USDJPY un pip es 0,01 y su valor en dólares se mueve con el precio. Un lote de oro son 100 onzas; un lote de un perpetuo de cripto es una moneda. En un cruce sin dólar, EURGBP por ejemplo, el valor del pip se estima y el informe avisa de que sus cifras son orientativas. El stop y el objetivo se llenan dentro de la vela que los alcanza, en su propio nivel.

¿Por qué mi estrategia da 0 operaciones?

Las condiciones son demasiado estrictas o los periodos demasiado largos. Prueba con periodos más cortos (SMA(10) en vez de SMA(50)) o simplifica las reglas.

¿Sobreajusta la búsqueda genética?

Sí, si la dejas. El panel de preparación ya pide un mínimo de operaciones sacado de tu histórico —entre 20 y 120— y aparta el último tramo sin enseñárselo; en *Ajustes avanzados* puedes cambiar *Operaciones mín.* y poner tope al *DD máx. %*. Después juzga lo que encontró en el Reality Check, al que se le dice cuántos juegos de reglas probó la búsqueda y que le exige más al mejor de una búsqueda grande.

¿Puedo combinar estrategias largas y cortas?

Sí, y es lo recomendable. En Cartera, combina LARGA con CORTA para bajar el drawdown. La matriz de correlación te dirá si de verdad están diversificadas.

¿Dónde se guardan mis estrategias?

En la Biblioteca, una base de datos SQLite de esta máquina: las reglas, el riesgo, la gestión monetaria, el tamaño de posición, el lote y el capital inicial. Guarda también el último veredicto del juez: la nota, la palabra, contra qué histórico, con qué cuenta de variantes y cuándo, y las cifras del tramo que la estrategia no vio —operaciones, PF, Sharpe y drawdown—, por las que se puede filtrar la lista; lo enseña en la ficha, y si cambias las reglas después, lo marca como de una versión anterior y dice qué cambió: cada condición añadida o quitada, y cada ajuste que el motor usa con su antes y su después. El informe en PDF describe entonces la versión juzgada. Y guarda lo que costó llegar a ella, la cuenta de variantes, que sólo sube. Cada una puede llevar notas y etiquetas, y la lista se filtra y se busca por etiqueta. **Duplicar** hace una copia con otro nombre para cambiarla sin tocar la original: recuerda de cuál viene y se lleva su veredicto y su cuenta de variantes, porque una copia no empieza de cero. **Comparar con…** pone dos de tus estrategias lado a lado: en qué se diferencian sus reglas, sus cifras sobre el histórico cargado con los mismos costes, y sus dos curvas en un mismo gráfico. Esas cifras cubren también el tramo que el juez guarda aparte, así que para decidir entre las dos lee sus veredictos. Filtrar por cifras y elegir con el filtro puesto se cobra: la elegida suma las cuentas de todas las comparadas, y la lista lo dice antes de que elijas. Las operaciones no las guarda — sus cifras se calculan cuando pulsas **Backtest**, sobre el histórico que esté cargado entonces.

¿Qué hace Dr. Quant?

Contesta preguntas y tiene dos manos con ficha. En la conversación del **piloto automático**, «juzga *nombre*» y «lleva *nombre* a la cartera» las lee el programa —una ficha dice qué va a pasar y nada corre sin tu clic, sin gastar tu clave—. Una pregunta va a tu propio proveedor de IA con sólo lo que nombre de lo tuyo, resumido: la Biblioteca con sus veredictos, una estrategia que nombres, tus datos o tus carteras guardadas, y el diario de veredictos; tus reglas y tus velas no viajan. Una orden que no es ninguna de ésas —«optimiza el periodo del RSI»— no se ejecuta desde ahí: Dr. Quant te dice en qué pantalla se hace. Y en la pantalla de **Datos** busca las condiciones de tu bróker para rellenar los costes. No construye nada ni ejecuta nada: construir es cosa del piloto, que lee tu encargo en código. No es asesoramiento financiero.

¿Puedo elegir qué proveedor de IA usa Dr. Quant?

Por su orden en **Ajustes**: contesta el proveedor activo con el número de prioridad más bajo. Si está saturado con un error de límite de peticiones (429), la pregunta pasa sola al siguiente activo.

¿Los modelos de IA gratuitos me cobran las búsquedas en internet?

El modelo es gratis; la **búsqueda no**. Una búsqueda web es un servicio aparte que tu proveedor compra por ti y te cobra aparte — OpenRouter lo dice con todas las letras: la búsqueda web tiene coste añadido también con modelos gratuitos. Unos **0,007 $** por búsqueda con el buscador por defecto, unos 70 céntimos por cien consultas. Sale de tu propia clave, no de la nuestra. **Leer una página de la que das la URL es gratis** — la descarga el servidor del propio BotFactoryFX —, así que pegar el enlace de las especificaciones de contrato de tu bróker no cuesta nada.

¿Por qué no puedo elegir una temporalidad menor que la de mi fichero?

Porque no se puede construir. Cuatro velas de M15 se juntan en una de H1, pero no hay aritmética que parta una vela de H1 en cuatro de M15 — ese detalle nunca estuvo en el fichero. En vez de inventarlo, el selector sólo ofrece la resolución de tu fichero y las más gruesas. Importa un CSV más fino si necesitas velas más finas.

He cambiado la temporalidad: ¿cambia también el backtest?

Sí. Cambiar la temporalidad reconstruye las velas de verdad, y el gráfico, el backtest, la búsqueda genética y el código exportado corren todos sobre la serie reconstruida. No pueden discrepar nunca sobre qué velas se están probando. Las versiones anteriores sólo cambiaban la etiqueta, que es justo el fallo que esto vino a sustituir.

¿Cómo muevo el gráfico hacia atrás en el tiempo?

Arrástralo, usa **mayús** con la rueda, usa los botones de flecha de encima del gráfico, o haz clic en el gráfico y usa ← → (Re Pág y Av Pág para una pantalla entera, Inicio y Fin para los dos extremos). La rueda sola hace zoom en vez de desplazar — es la convención de todos los gráficos, y por eso parece que sólo hace zoom hasta que uno se entera de lo de mayús.

¿Puedo exportar estrategias a MetaTrader?

Sí. Abre el menú de descarga de una estrategia en la **Biblioteca** —o pulsa **Exportar…** en **Construir › Manual**, que la guarda y te abre ese menú— y elige **MQL4** (MetaTrader 4) o **MQL5** (MetaTrader 5). El fichero es un Expert Advisor completo: las reglas de entrada, la orden de entrada, el stop y el objetivo, el punto de equilibrio, el trailing stop, las salidas por tiempo, los topes del día y el tamaño de posición, tal como los corrió el backtest. Cópialo en la carpeta `MQL4/Experts/` o `MQL5/Experts/` de la carpeta de datos de MetaTrader, compílalo y ponlo en un gráfico — paso a paso, con los ajustes que añade el fichero, en *Instalar el robot en cada plataforma*.

¿El robot exportado da lo mismo que el backtest de BotFactoryFX?

Se parece todo lo que cada plataforma permite, y cada fichero dice arriba dónde no puede. Las reglas de entrada, leídas sobre velas cerradas; las entradas en espera STOP y LIMIT, que caducan al mismo número de velas; el stop y el objetivo; el punto de equilibrio y el trailing stop; las salidas por tiempo; los dos topes del día, contados sobre el día UTC; y los tres modos de tamaño —lote fijo, porcentaje de riesgo y Kelly— con el pip, el lote y el apalancamiento del instrumento que mediste (exporta con su histórico abierto): todo eso lo simula el backtest y lo lleva el fichero exportado, que mantiene una posición a la vez, como el backtest. **Excepciones de hoy:** JForex sólo dimensiona por riesgo o por Kelly en pares de divisas, y sólo opera pares de divisas, oro, plata, bitcoin y éter — en cualquier otra cosa se niega a exportar; y el guion de TradingView no pone ningún coste, así que sus totales no llevan el spread, la comisión, el slippage y el carry que el backtest cobró. **Lo que ningún fichero puede llevar** es lo que la simulación tuvo que suponer: el spread y el slippage de verdad de tu bróker, los llenados parciales, y una orden en espera que se salta su nivel de un hueco, que un bróker llena a mercado mientras el backtest la llenaba en el nivel.

¿Modo Guiado o Personalizado en Construir › Manual?

Empieza en **Guiado**: elige una familia (Tendencia, Ruptura, Reversión, Acción del precio, Impulso, Volatilidad, Divergencia) y una plantilla — las reglas vienen puestas y tú sólo tocas parámetros. Pásate a **Personalizado (Alchemist)** cuando quieras mezclar indicadores de familias distintas sin ninguna barandilla. El comprobador de redundancias corre en los dos modos.

¿Guiado o Aleatorio en la búsqueda genética?

Guiado —el de por defecto, y el único modo del panel de preparación— ciñe la búsqueda genética a los indicadores y los periodos de las familias que marques, y siembra cerca de la mitad de la primera generación con sus plantillas, así que explora alrededor de patrones de reglas que significan algo en vez de probar indicadores al azar. **Aleatorio**, en *Ajustes avanzados*, no se ciñe a ninguna familia y no siembra con plantillas. No es el más ancho de los dos: sortea del repertorio que se ve en *Conjunto de indicadores activo*, un núcleo corto — cierre, SMA, EMA, RSI, ATR, ADX y MACD —, que sólo se amplía con el esfuerzo *Profunda*. Con varias familias marcadas, Guiado sortea de muchos más indicadores que ésos.

¿Puedo mezclar familias de estrategias?

Sí: marca las que quieras, en **Construir › Búsqueda genética** y en la rueda, o nombra varios estilos en la frase del piloto. Hay familias que se complementan: la acción del precio no trae osciladores que estorben a nadie, y una ruptura con filtro de tendencia es una estrategia que ninguna de las dos familias sabe escribir sola, porque los máximos de la sesión no están en el repertorio de tendencia y el ADX no está en el de ruptura. Mezclar junta el repertorio de indicadores, el de periodos y las plantillas con las que se siembra la primera generación, y la siembra sortea primero una familia y luego una de sus plantillas, así que cada familia marcada se lleva la misma parte y no una parte proporcional a cuántas plantillas le hayan escrito. Hace posibles las híbridas sin forzarlas, y una divergencia se queda entera en una mezcla igual que sola. **Lo que cuesta:** un repertorio más ancho quiere decir que se prueban más variantes distintas, y el Reality Check exige más pruebas fuera de muestra para la misma nota cuantas más candidatas se probaron — la línea de debajo de las fichas dice cuánto ha crecido el repertorio antes de que empieces. Con las siete marcadas, el repertorio tiene todos los indicadores de serie que trae el motor; la siembra desde plantillas la sigue separando del modo Aleatorio, cuyo repertorio es mucho más corto.

¿Por qué hay indicadores marcados como redundantes?

El comprobador en vivo caza: (1) condiciones idénticas en el mismo bloque, (2) el mismo indicador comparado consigo mismo (por ejemplo `SMA(20) > SMA(20)`, falso siempre), (3) periodos casi iguales, como SMA(20) y SMA(22), (4) mezclar RSI + ROC (los dos son osciladores de impulso), (5) tres o más medias simples o exponenciales en un bloque, (6) entradas LARGA y CORTA idénticas, que saltan en la misma vela. Son *avisos*, no errores: puedes ignorarlos si sabes lo que haces.

Glosario de métricas

Factor de beneficio

`gross profit / gross loss`

> 1 = rentable, > 1,5 = bueno, > 2 = excelente. ∞ si no hay pérdidas.

Acierto

`winning trades / total × 100`

% de operaciones positivas. Sólo sirve con contexto.

Sharpe

`(mean / stddev) × √(trades per year)`

Rentabilidad ajustada por volatilidad, operación a operación, anualizada según lo a menudo que opera la estrategia — no por un √252 que daría por supuestas velas diarias. > 1 aceptable, > 2 muy bueno. **No medible** cuando todas las operaciones devuelven lo mismo — sin pérdidas, o todas en el mismo stop: sin variación no hay riesgo que medir, y un número ahí sería dividir por casi cero. La búsqueda lo puntúa como sin ventaja.

Drawdown máximo

`(peak - valley) / peak × 100`

Mayor caída desde un máximo. < 20% razonable, > 40% peligroso.

Rentabilidad/DD (Calmar)

`Total P&L / Max DD`

Rentabilidad por unidad de riesgo asumido.

Correlación

`cov(r_a, r_b) / (σ_a × σ_b)`

Cuánto se mueven juntas dos estrategias. -1 opuestas, 0 independientes, +1 paralelas.

Atajos y consejos

- En Construir › Búsqueda genética, cada ficha de familia se enciende por su cuenta: marca varias para mezclar esas familias en una sola búsqueda.
- En la Biblioteca, el menú de descarga guarda una estrategia para cualquiera de las seis plataformas o como JSON, e Importar vuelve a leer un JSON.
- Las preguntas van en la caja del piloto automático: lo que no es un encargo de construir lo contesta Dr. Quant, con tu propio proveedor de IA; una orden sobre una estrategia que ya tienes te dice en qué pantalla se hace.
- En Cartera, ponle un peso distinto a cada estrategia para repartir a tu gusto.
- Si la búsqueda genética deja de mejorar, abre Ajustes avanzados y prueba otra Mutación — pero deja Operaciones mín. donde lo puso el panel: ya pide entre 20 y 120 operaciones de tu histórico, y bajarlo deja

pasar ruido.

- Los cruces («cruza hacia arriba» / «cruza hacia abajo») se evalúan contra la vela anterior.
- ¿Primera vez en Construir › Manual? Empieza en modo Guiado, elige una familia y luego una plantilla — después puedes pasarte a Personalizado (Alchemist) sin perder las reglas.
- Ctrl+K busca en todo el programa —secciones, estrategias de la Biblioteca, históricos y esta Ayuda— y lleva a lo que eliges; no importan las tildes ni las mayúsculas.
- En Construir › Manual, Deshacer y Rehacer —o Ctrl+Z y Ctrl+Mayús+Z— recorren tus cambios de las reglas, «Limpiar» incluido; dentro de un campo de texto, Ctrl+Z deshace lo que escribes en él.

El contenido de esta aplicación es educativo. No constituye asesoramiento financiero. Operar conlleva riesgo de pérdida.

Si algo no va, contesta al correo de tu compra o escribe a support@botfactoryfx.com.