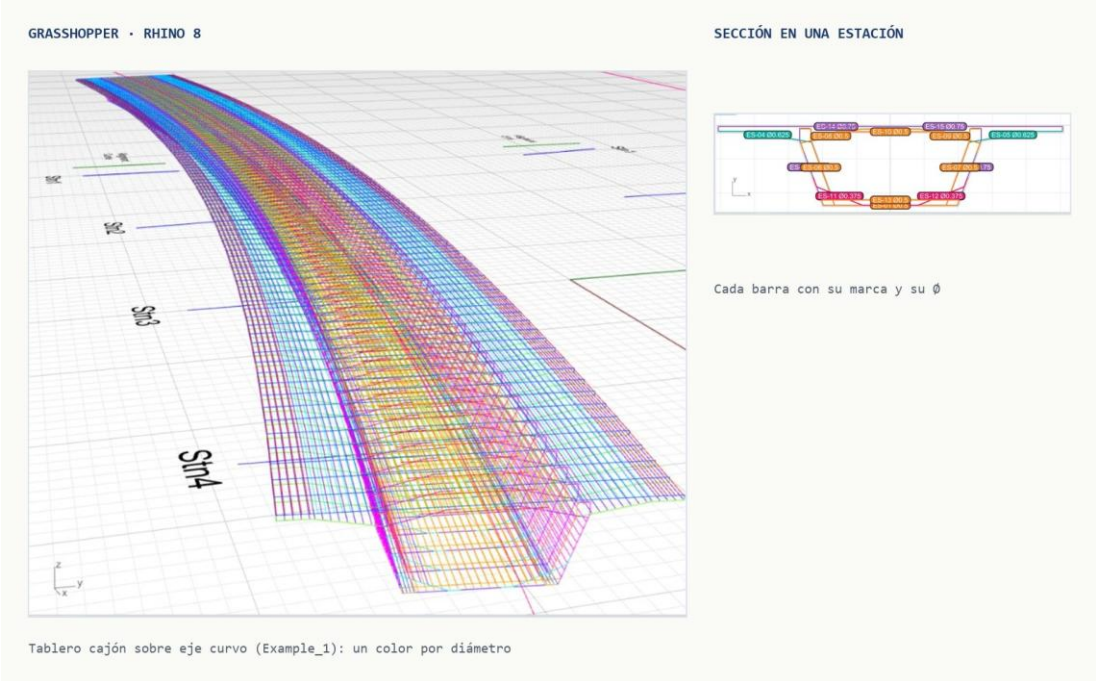
GH_BRIDGE

Guía de uso · versión 1.0.0 · Rhino 8

GH_Bridge es un sistema de modelado de puentes para Rhino y Grasshopper. Las secciones transversales paramétricas se evalúan a lo largo de un eje 3D, se barren hasta el sólido del tablero y se arman con barras transversales y longitudinales que llevan sus propios ganchos, traslapes y marcas. Después las barras se pueden enviar a Revit como Rebar nativo con Rhino.Inside.Revit.

No es un plugin de armadura genérico. El núcleo es la sección y el eje: el acero se define contra puntos con nombre de la sección, nunca en coordenadas absolutas, así que sigue al tablero cuando cambian su peralte, su ancho o sus espesores.



Cajón sobre eje curvo: acero transversal y longitudinal, un color estándar por diámetro.

La pestaña Bridge, por etapa del flujo

1. Info · Información del plugin
   └── BrInfo
2. Alignment · Eje y estaciones
   └── Axis StnCL StnD DivAx DivAxD DeStn
3. Section · Secciones transversales
   └── ParamSec RdSecJ SecCrv SecVars WrSecJ DeSec
4. Model · Geometría del tablero
   └── EvalSec SecSwp SecView PlnCut
5. Transversal · Acero transversal
   ├── RbCat RbCov RbDef RbStir RbSec RbView
   └── RbHook RbSeam
6. Longitudinal · Acero longitudinal
   ├── RbLongN RbLongP RbLong RbLongAsm RbLongView RbArc
   └── RbSplice RbLongJoin
7. Tools · Controles, metrado y salida
   ├── DeBar RbGroup RbSched RbBend RbBD RbClash

```
    └── RbNudge  Bake  RbSplBar
8. Revit  ·  Rebar nativo en Revit (solo dentro de Rhino.Inside.Revit)
    └── Crv2Rvt  Rebar2Rvt
9. Curves  ·  Herramientas de curvas para armado manual
    ├── CrvAng  DiscCrv  DivCrvN  DivCrv  ExtCrv  ExtCrv2
    ├── ExtPtCrv  LinePt  PrjCrv2  PrjCrv  OrientPl  RebuildCrv
    └── RebuildCrvH  SplCrv
```

1. INSTALACIÓN

Requisitos

- Rhino 8 con Grasshopper (Windows).
- Solo para el panel Revit: Revit 2025 + Rhino.Inside.Revit.

Opción A · Package Manager de Rhino

En Rhino ejecuta `_PackageManager`, busca `gh-bridge` e instálalo. Reinicia Rhino.

Opción B · Instalación manual (.zip)

Descarga GH_Bridge-v1.0.0.zip desde daapcloud.com/gh-bridge o Food4Rhino, y luego:

- Cierra Rhino.
- Copia la carpeta GH_Bridge (GH_Bridge.gha y sus dos .dll) en:
  `%APPDATA%\Grasshopper\Libraries\GH_Bridge\`
- Opcional, Revit: copia GH_Bridge.Revit\GH_Bridge.Revit.gha en:
  `%APPDATA%\Grasshopper\Libraries-Inside-Revit-2025\GH_Bridge.Revit\`
- Abre Rhino y Grasshopper: aparece la pestaña Bridge. Coloca Bridge Info (BrInfo) para confirmar la versión.

Notas

- Si Windows bloqueó los archivos descargados: clic derecho en cada uno > Propiedades > Desbloquear.
- No cambies el nombre del .gha ni de sus .dll.
- El .gha de Revit solo carga dentro de Revit con Rhino.Inside.Revit; en Rhino solo se ignora.

2. FLUJO DE TRABAJO

Toda definición de GH_Bridge sigue el mismo orden. Cada etapa alimenta a la siguiente:

1. Eje

```
Axis → StnCL / StnD → DivAx
```

Una curva de Rhino pasa a ser el eje. Las estaciones salen de líneas de control (estribos, pilares, esviaje) o de distancias explícitas; DivAx subdivide cada tramo para las barras.

2. Sección

```
RdSecJ / SecCrv → ParamSec → SecVars
```

La sección se lee de un JSON (o se digitaliza desde una curva cerrada). ParamSec expone sus variables como entradas; SecVars las cambia tramo a tramo.

3. Modelo

```
EvalSec → SecSwp
```

La sección se coloca en el plano de cada estación y se barre hasta el sólido del tablero (Brep).

4. Transversal

```
RbCat + RbCov + RbDef → RbSec → RbStir
```

Las barras se definen contra los puntos de la sección (anclajes, paso, ganchos) y se resuelven sobre los rieles de recubrimiento en cada estación.

5. Longitudinal

```
RbLong / RbLongN / RbLongP → RbLongAsm → RbSplice → RbLongJoin → RbArc
```

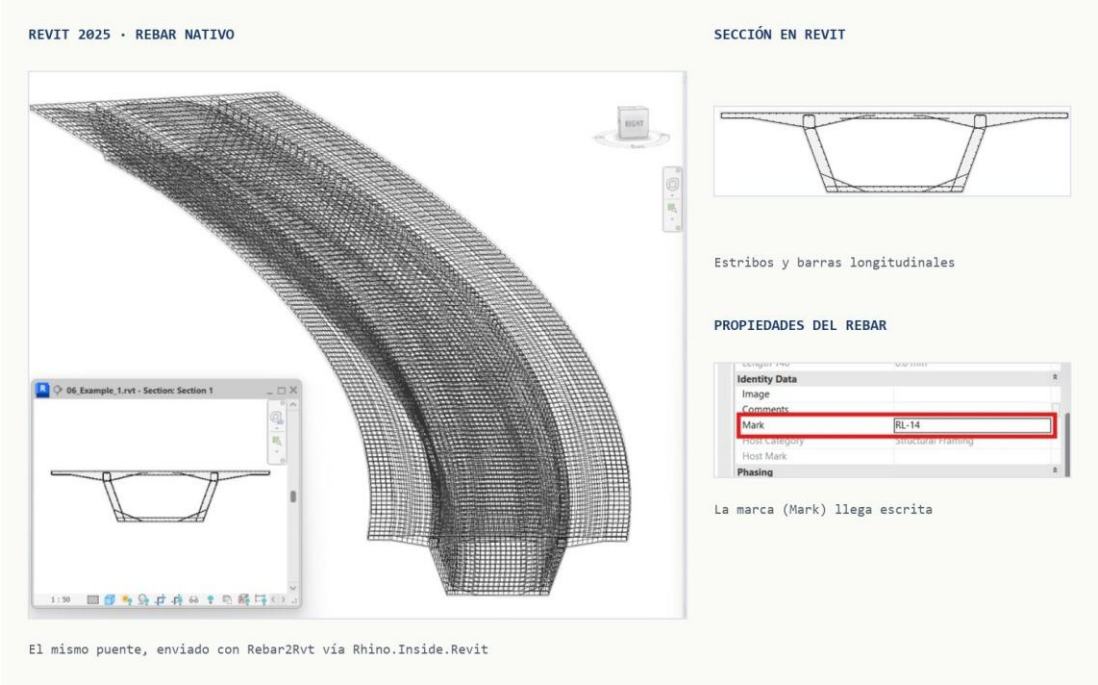
Paquetes de barras por espaciamiento, cantidad o puntos siguen al eje en 3D; al final se aplican los traslapes y el ajuste a arcos reales.

6. Salida

```
RbClash → RbNudge → RbGroup → RbSched → Bake / Rebar2Rvt
```

Control y corrección de choques, agrupación por marca con colores y capas estándar, metrado de acero, y horneado en Rhino o envío a Revit.

Pasa el mouse sobre cualquier componente o puerto en Grasshopper para ver su ayuda. Example_1 y Example_2, incluidos en el .zip, muestran la cadena completa funcionando.



Las mismas barras en Revit como Rebar nativo, con sus marcas y tipos de gancho.

3. CONCEPTOS CLAVE

3.1 La sección en JSON

Una sección es un archivo JSON con estos campos principales:

Campo	Contenido
`Id, Name, Unit`	Número, nombre y unidad de longitud (m, cm, mm, ft, in).
`Variables`	Pares nombre = valor: las dimensiones que vas a controlar (D, Wtop, Tweb...).
`Points`	Puntos de construcción; cada uno con Id y Coord [Y, Z], números o expresiones de las variables.
`Loops`	Contornos cerrados: Outer (concreto) e Inner (vacíos), cada uno con sus puntos.
`Circles`	Vacíos o pilas circulares: punto centro y radio.

- Z apunta hacia abajo: TOPC = [0, 0] es el centro superior y BOTC = [0, D] el inferior. TOPC es el punto que sigue al eje.
- Nombres: puntos del contorno exterior E1, E2...; del vacío I1, I2.... La primera letra le dice al acero en qué riel está el punto.
- Las expresiones aceptan + - * / ^ y paréntesis, por ejemplo "0.5*Wtop - Tweb".
- Un punto también se puede definir respecto de otros con Reference y Type:

Type	Reference	Significado
`Euclidean`	`0 / 1 / 2`	Absoluto; desplazamiento desde un punto; o Y del primero y Z del segundo.
`Polar`	`2`	Sistema local de ref[0] hacia ref[1]; Coord = [a lo largo, transversal].
`CY / CZ`	`3`	Sobre la recta ref[0]–ref[1], a la misma Y (o Z) que ref[2].

Tableros compuestos: pon en la primera sección puntos llamados G, G1...Gn y agrega más secciones a RdSecJ con "+"; cada una se asienta con su TOPC sobre esos puntos. Example_2 trae una biblioteca de secciones JSON para empezar.

3.2 Anclajes de las barras transversales

RbDef coloca una barra sobre el riel de recubrimiento entre un anclaje de inicio y uno de fin. Un anclaje se escribe así:

Forma	Ejemplo	Significado
Id de punto	`E3`	Justo en ese punto del riel.
From:To@dist	`E3:E4@0.25`	Sobre la arista de E3 a E4, a 0.25 (unidades del documento) de E3.
From:To@pct%	`I4:I5@30%`	Sobre la arista de I4 a I5, al 30 % de su longitud.

- Mismo inicio y fin: estribo cerrado alrededor de todo el riel. Way obliga a tomar el camino largo.

– Los ganchos son Ids del catálogo (RbCat), por ejemplo H90, H135. Un gancho direccional se escribe "Dir:Ref1:Ref2@leg1:leg2".

3.3 Tokens de traslape y distribución

Token	Ejemplo	Significado
`F@val`	`F@6.0`	Pieza de longitud fija.
`R@val`	`R@8.5`	Llena el resto con piezas de hasta val (como máximo un R).
`P@val`	`P@0.27`	Traslape centrado en val: hasta 1 es fracción de la barra; más de 1, una distancia.
`N@val`	`3@0.5, R@2.0, 3@0.5`	DivAxD: N espacios de val y el resto a R, ordenados de inicio a fin.

En RbSplice, Alt escalona los traslapes de cada segunda barra de la misma marca (en dAlt, o medio paso si está vacío); Sym refleja el patrón desde ambos extremos.

3.4 Colores y capas estándar

Las barras reciben un color estándar por diámetro (con variantes por marca) y RbGroup asigna capas estándar Bridge::Rebar::Ø<mm>::<marca>. La vista previa, el horneado y el archivo usan los mismos colores, y el concreto recibe grises estándar por tipo de elemento.

4. REFERENCIA DE COMPONENTES

Los nombres de componentes y de puertos, y la descripción de cada puerto, se dejan en inglés: es como aparecen en Grasshopper al pasar el mouse.

1. Info · Información del plugin

a. Bridge Info (BrInfo)

Versión, autor y licencia de GH_Bridge

ENTRADAS:

- **(ninguna)**

SALIDAS:

- **Plugin (P)**: GH_Bridge version and author
- **License (L)**: License terms
- **Links (U)**: Source code, documentation and contact

2. Alignment · Eje y estaciones

a. Axis From Curve (Axis)

Construye el eje del puente a partir de una curva de Rhino, parametrizado por estación

ENTRADAS:

- **Curve (Crv)**: Alignment curve
- **Name (N)**: Axis name

SALIDAS:

- **Axis (Ax)**: bridge axis

b. Stations From Control Lines (StnCL)

Una estación por cada línea de control que cruza el eje (estribos, pilares, líneas de esviaje), ordenadas y medidas a lo largo del eje

ENTRADAS:

- **Axis (Ax)**: bridge axis
- **Control Lines (CL)**: Lines/curves crossing the axis, one per station (e.g. abutment, pier or skew lines); named Stn1, Stn2, ... in the order given here
- **Tolerance (Tol)**: Intersection tolerance

SALIDAS:

- **Stations (Stns)**: Stations, one per control line, ordered along the axis; Station = running length from the first one
- **Planes (Pl)**: Station planes (same order as Stations) - feed into EvaluateSectionComponent's Plane input

- **Gaps (Gaps)**: Running-length gap from each station to the next (Count - 1 values) - not the same thing as Divide Axis's own Sp input, which sets the rebar gaps WITHIN a tramo
- **Length (L)**: Total bridge length, from the first station to the last

c. Stations From Distances (StnD)

Estaciona el eje en distancias explícitas medidas a lo largo de él, perpendiculares al eje (sin líneas de control)

ENTRADAS:

- **Axis (Ax)**: bridge axis
- **Distances (D)**: Running-length distances from the start of the Axis curve, e.g. 0, 25, 50, 75, 100; named Stn1, Stn2, ... in the order given here

SALIDAS:

- **Stations (Stns)**: Stations at the requested distances, perpendicular to the Axis
- **Planes (Pl)**: Station planes (same order as Stations)
- **Gaps (Gaps)**: Running-length gap from each station to the next, in output order (Count - 1 values) - not the same thing as Divide Axis's own Sp input, which sets the rebar spacing WITHIN a tramo
- **Length (L)**: Total length covered, from the first distance given to the last

d. Divide Axis (DivAx)

Divide el tramo entre dos estaciones con nombre a un espaciamiento fijo, para colocar el acero a lo largo de un tramo

ENTRADAS:

- **Axis (Ax)**: bridge axis (needed to evaluate points/tangents between the two named stations)
- **Stations (Stns)**: already-resolved stations to pick the domain from (e.g. StnCL/StnD output) - also used as direction-interpolation guides inside the domain
- **From (From)**: name of the station marking one end of the domain (e.g. "Stn1")
- **To (To)**: name of the station marking the other end of the domain (e.g. "Stn2") - order doesn't matter, the domain always runs low-to-high running length
- **Spacing (Sp)**: running-length spacing between infill stations inside the domain

SALIDAS:

- **Stations (Stns)**: From, To (with their original names) and the fixed-spacing infill stations between them (unnamed)
- **Planes (Pl)**: Station planes (same order as Stations)

e. Divide Axis By Distribution (DivAxD)

Divide el tramo entre dos estaciones con nombre según una distribución simétrica de tokens N@val/R@val

ENTRADAS:

- **Axis (Ax)**: bridge axis (needed to evaluate points/tangents between the two named stations)
- **Stations (Stns)**: already-resolved stations to pick the domain from (e.g. StnCL/StnD output) - also used as direction-interpolation guides inside the domain
- **From (From)**: name of the station marking one end of the domain (e.g. "Stn1")
- **To (To)**: name of the station marking the other end of the domain (e.g. "Stn2") - order doesn't matter, the domain always runs low-to-high running length
- **Distribution (Dist)**: Tokens ordered start-to-end, e.g. "3@0.5", "R@2.0", "3@0.5" (mirrored) - see component description

SALIDAS:

- **Stations (Stns)**: From, To (with their original names) and the distribution's infill stations between them (unnamed)
- **Planes (Pl)**: Station planes (same order as Stations)

f. Deconstruct Station (DeStn)

Descompone una lista de estaciones en nombre, progresiva y plano

ENTRADAS:

- **Stations (Stns)**: Axis stations

SALIDAS:

- **Name (Nm)**: Station name
- **Station (Stn)**: Running length along the axis
- **Plane (Pl)**: Station plane

3. Section · Secciones transversales

a. Parametric Section (ParamSec)

Expone una entrada por cada variable de la sección para controlar sus dimensiones en vivo. Acepta la lista ensamblada de RdSecJ: un bloque de entradas y una salida por sección (la receptora arriba; las variables compartidas, solo en su bloque)

ENTRADAS:

- **Section (Sec)**: Parametric section template (Variables + expressions), or RdSecJ's receiver + adhered sections

SALIDAS:

- **Section (Sec)**: Modified section (same geometry/topology, Variables overridden)

b. Read Section JSON (RdSecJ)

Lee una sección desde un texto JSON o una ruta de archivo (formato propio o exportación del esquema de referencia). Con el zoom y el botón "+" se adhieren más secciones: cada una se asienta (su TOPC) sobre cada punto G/G1..Gn de la primera

ENTRADAS:

- **File Path (Path)**: Receiver section: file path to a .json file, or a JSON description pasted in directly

SALIDAS:

- **Section (Sec)**: Receiver section, followed by one copy of each adhered section per G point

c. Section From Curve (SecCrv)

Convierte una curva plana cerrada (dibujada en Rhino o traída de un .dwg) en una sección

ENTRADAS:

- **Outer Curve (Out)**: Closed planar curve describing the outer boundary (straight segments only - no arcs)
- **Inner Curves (In)**: Closed planar curves describing voids/holes (e.g. a box girder cell)
- **Plane (Pl)**: Section plane; defaults to the outer curve's own planar frame
- **Name (Nm)**: Section name
- **Unit (U)**: reference-schema-style unit code (m/cm/mm/ft/in); defaults to the active Rhino document's model unit

SALIDAS:

- **Section (Sec)**: cross section

d. Section Set Variables (SecVars)

Genera una sección por estación a partir de una plantilla, arrastrando los cambios de variables tramo a tramo

ENTRADAS:

- **Section (Sec)**: parametric template (Variables + expressions)
- **Stations (Stns)**: stations to build one Section per - order can be arbitrary, matched by Name
- **Overrides (Ov)**: "StationName: Var=Value, Var2=Value2" entries - one per station where a change starts; later stations carry it forward until the next override

SALIDAS:

- **Sections (Sec)**: one Section per station, same order as Stations

e. Write Section JSON (WrSecJ)

Escribe una sección como texto JSON compatible con el formato de secciones del esquema de referencia

ENTRADAS:

- **Section (Sec)**: cross section
- **Indented (I)**: Pretty-print the JSON output

SALIDAS:

- **Json Text (Txt)**: JSON description

f. Deconstruct Section (DeSec)

Descompone una sección en sus puntos, contornos y círculos para revisarla

ENTRADAS:

- **Section (Sec)**: cross section

SALIDAS:

- **Name (Nm)**: Section name
- **Unit (U)**: Section unit
- **Points (P)**: Top-level construction points
- **Loops (L)**: Section loops
- **Circles (C)**: Section circles

4. Model · Geometría del tablero

a. Evaluate Section (EvalSec)

Evalúa la geometría de una sección (curvas de contorno y puntos) para vista previa o para usarla aguas abajo

ENTRADAS:

- **Section (Sec)**: cross section
- **Plane (Pl)**: Where to place the section in 3D; defaults to the world XY plane

SALIDAS:

- **Curves (C)**: Outline curves: one per loop, then one per circle (circular piers, round duct voids)
- **Points (P)**: All resolved section points (top-level and loop points)

b. Section Sweep (SecSwp)

Barre los contornos de una sección a lo largo del eje, pasando por las curvas de cada estación (de EvalSec con la lista de planos), hasta un sólido Brep

ENTRADAS:

- **Rail (R)**: The Axis alignment curve the sweep follows between stations
- **Curves (C)**: Per-station loop curves - one branch per station, one curve per loop in the same order at every station (EvalSec's Curves output, fed a list of station planes). Branches {piece; station} (grafted Sections) are swept piece by piece - the station index is the last path index that varies
- **Tolerance (Tol)**: Sweep/join/cap tolerance; defaults to the active document's absolute tolerance

SALIDAS:

- **Solid (S)**: One solid per piece (one for a single Section; deck + each girder for a grafted assembly) - a capped tube, or (for a hollow section) the outer tube minus every inner tube
- **Walls (W)**: The per-loop swept skin surfaces (open, before capping), for inspection

c. Section Viewer (SecView)

Muestra una sección con sus propios colores (Outer verde, Inner rojo, TOPC/BOTC morado, demás puntos azul) y el Id de cada punto

ENTRADAS:

- **Section (Sec)**: cross section
- **Plane (Pl)**: Where to place the section in 3D; defaults to the world XY plane
- **Labels (Lbl)**: Show every point as a labeled dot (off by default - many dots slow the viewport down)

SALIDAS:

- **Curves (C)**: Outline curves, one per loop (same as EvalSec)
- **Points (P)**: All resolved section points

d. Plane Cutter (PlnCut)

Crea un cortador plano rectangular (Brep) centrado en cada plano

ENTRADAS:

- **Planes (Pl)**: Planes to build cutters on
- **Size (Sz)**: Half-width of each square cutter
- **Tolerance (Tol)**: Tolerance for building the planar Brep

SALIDAS:

- **Cutters (Cut)**: Planar cutter Breps, one per plane

5. Transversal · Acero transversal

a. Hook Catalog (RbCat)

Lee el catálogo de formas de gancho (Id, ángulo y extensión en diámetros de barra) desde un texto JSON o una ruta, o usa el catálogo incorporado si no se da ninguno

ENTRADAS:

- **File Path (Path)**: File path to a .json file, or a JSON description pasted in directly - leave empty to use the built-in default catalog (H90/H135/H180)

SALIDAS:

- **Hooks (Hks)**: hook shapes in the catalog
- **Ids (Ids)**: hook Ids only, same order as Hooks - handy to feed a Value List

b. Rebar Cover (RbCov)

Agrupa el recubrimiento exterior e interior en un solo valor para RbSec/RbLongAsm y deja pasar la sección

ENTRADAS:

- **Section (Sec)**: cross section, passed through unchanged (optional)

- **Cover Exterior (CovE)**: Concrete cover for Outer loop(s) - the rail moves inward by this much
- **Cover Interior (CovI)**: Concrete cover for Inner/void loop(s) - the rail moves outward by this much

SALIDAS:

- **Section (Sec)**: the input Section, passed through
- **Cover (Cov)**: CoverExterior/CoverInterior bundled - wire into RbSec/RbLongAsm's Cover input

c. Transversal Bar (RbDef)

Define una barra transversal: Id y diámetro, anclajes de inicio, paso y fin sobre el riel de la sección, y gancho

ENTRADAS:

- **Id (Id)**: bar identifier, e.g. "ES-01"
- **Diameter (Ø)**: bar diameter
- **Start (Sta)**: start anchor: a point Id, "From:To@dist", or "From:To@pct%"
- **Waypoint (Way)**: optional - an intermediate point Id forcing the LONGER of the two paths around the closed rail; leave empty to just take the shorter path (the usual case)
- **End (End)**: end anchor - same forms as Start; the same anchor as Start gives a closed stirrup around the whole rail
- **Hook (Hk)**: hook Id (or "Dir:Ref1:Ref2@leg1:leg2" for a directional hook aimed at a reference edge instead of the rail's own corner), matched against a Rebar Hook Catalog (Hooks/Ids output) - default for both ends unless overridden by HkS/HkE
- **Hook Start (HkS)**: override hook just for the Start end (same forms as Hook); leave empty to use Hook
- **Hook End (HkE)**: override hook just for the End end (same forms as Hook); leave empty to use Hook

SALIDAS:

- **Bar (Bar)**: the resulting bar definition

d. Closed Stirrup (RbStir)

Abre los estribos cerrados en una esquina, con traslape cruzado y ganchos hacia adentro: la forma que Revit reconoce como estribo cerrado

ENTRADAS:

- **Bars (Bars)**: Closed Rebar Bars (e.g. RbSec stirrups); open bars pass through unchanged
- **Corner (Cn)**: Index of the loop vertex where the stirrup closes
- **Overrun (Ov)**: How far each leg runs past the corner, in mm
- **Hook (Hk)**: Hook length at each end, in mm
- **Angle (A)**: Angle between each hook and its leg, in degrees (the hook turns back into the stirrup)

SALIDAS:

- **Bars (Bars)**: Open stirrups ready for Rebar2Rvt
- **Geometry (G)**: The open stirrup polyline

e. Transversal Assemble (RbSec)

Construye los rieles desplazados por el recubrimiento a partir de la sección y resuelve sobre ellos una lista de barras, ganchos incluidos

ENTRADAS:

- **Section (Sec)**: cross section (one station)
- **Plane (Pl)**: Where the section is placed in 3D; defaults to the world XY plane
- **Cover (Cov)**: CoverExterior/CoverInterior bundle (RbCov output)
- **Bars (Bars)**: bar definitions (RbDef output)
- **Hooks (Hks)**: hook catalog (RbCat output)

SALIDAS:

- **Section (Sec)**: the input Section, passed through - wire straight into the viewer's Section input (RbView / RbLongView) instead of cabling it separately
- **Bars (Bars)**: resolved bar geometry, hooks included

f. Transversal Viewer (RbView)

Muestra la sección (atenuada) con las barras resueltas encima, con un color estándar por diámetro

ENTRADAS:

- **Section (Sec)**: cross section, drawn muted for context (optional)
- **Plane (Pl)**: must match the plane Bars were resolved on; defaults to WorldXY
- **Bars (Bars)**: resolved bars (RbSec output)
- **Labels (Lbl)**: Label each bar with its Mark and diameter (off by default - many dots slow the viewport down)

SALIDAS:

- **(ninguna)**

g. Hook Preview (RbHook)

Dibuja la forma de un gancho por separado (tramo de entrada y doblez) para un diámetro de barra dado

ENTRADAS:

- **Hook (Hk)**: one hook from a Rebar Hook Catalog (Hooks output)
- **Diameter (Ø)**: bar diameter, scales the hook's extension
- **Plane (Pl)**: where to draw it; the leg arrives along the plane's XAxis, the hook turns toward YAxis
- **Leg Length (Leg)**: length of the straight lead-in segment, for visualization only; 0 = auto (8x diameter)
- **Flip (Flip)**: turn toward -YAxis instead of +YAxis

SALIDAS:

- **Curve (C)**: the hook's shape: leg + bend, as a polyline

h. Seam Match (RbSeam)

Uniformiza el sentido de recorrido y el vértice de inicio de un conjunto de curvas cerradas, para que se correspondan de estación en estación

ENTRADAS:

- **Curves (C)**: Closed curves to normalize (one per station)
- **Planes (Pl)**: Reference plane per curve (single plane reused for all if only one given)
- **Flip (Flip)**: Flip the winding direction
- **Right (Right)**: Seam on the right side (false: left side)
- **Segments (Nseg)**: Advance the seam by this many extra vertices (can be negative)

SALIDAS:

- **Curves (C)**: Seam-matched curves, same order as input
- **Bad (Bad)**: True where the curve needed a fallback repair

6. Longitudinal · Acero longitudinal

a. Longitudinal By Count (RbLongN)

Define un paquete de barras longitudinales (una marca y un diámetro) con una cantidad fija, repartidas por igual entre dos puntos del riel

ENTRADAS:

- **Mark (Mk)**: shared mark for the whole package, e.g. "RL-01"
- **Diameter (Ø)**: bar diameter, shared by the whole package
- **Section (Sec)**: cross section used to resolve the real From-To distance and this package's offset rail
- **Offset Diameter (ØOfs)**: how far this package's rail sits off the raw section face (e.g. to clear the transversal bars) - not the structural cover; direction follows From/To's own E-/I- point Id
- **From (Fr)**: starting rail point Id (e.g. "E10") - the first bar lands here
- **To (To)**: ending rail point Id - the edge From->To the package is placed on
- **Count (Cnt)**: fixed number of bars, evenly spaced from From to To
- **Start Offset On (OffS)**: enable a start offset
- **Start Offset (dS)**: start offset distance
- **End Offset On (OffE)**: enable an end offset
- **End Offset (dE)**: end offset distance
- **Extend Start (ExtS)**: when Start Offset is on, dS pushes the first bar OUTWARD past From (same edge line) instead of trimming inward
- **Extend End (ExtE)**: when End Offset is on, dE pushes the last bar OUTWARD past To (same edge line) instead of trimming inward

SALIDAS:

- **Bars (Bars)**: one longitudinal bar definition per point in the package
- **Count (N)**: how many bars this package generated

b. Longitudinal By Points (RbLongP)

Define un paquete de barras longitudinales (una marca y un diámetro) en coordenadas locales de la sección escritas a mano

ENTRADAS:

- **Mark (Mk)**: shared mark for the whole package, e.g. "RL-01"
- **Diameter (Ø)**: bar diameter, shared by the whole package
- **Points (Pts)**: one point per bar, already in the Rhino document's own units (no Section input to convert from) - X is the section's local horizontal axis, Y is vertical as seen on screen (positive up), origin = the section's TOPC; Z is ignored

SALIDAS:

- **Bars (Bars)**: one longitudinal bar definition per point in the package
- **Count (N)**: how many bars this package generated

c. Longitudinal By Spacing (RbLong)

Define un paquete de barras longitudinales (una marca y un diámetro) con un espaciamiento fijo entre dos puntos del riel

ENTRADAS:

- **Mark (Mk)**: shared mark for the whole package, e.g. "RL-01"
- **Diameter (Ø)**: bar diameter, shared by the whole package
- **Section (Sec)**: cross section used to resolve the real From-To distance and this package's offset rail
- **Offset Diameter (ØOfs)**: how far this package's rail sits off the raw section face (e.g. to clear the transversal bars) - not the structural cover; direction follows From/To's own E-/I- point Id
- **From (Fr)**: starting rail point Id (e.g. "E10") - distances are measured from here
- **To (To)**: ending rail point Id - the edge From->To the package is placed on
- **Spacing (Spc)**: fixed real spacing between bars
- **Mode (Mode)**: 0 = Beginning, 1 = Middle, 2 = End, 3 = Extreme (leftover distance distributed accordingly)
- **Start Offset On (OffS)**: enable a start offset
- **Start Offset (dS)**: start offset distance
- **End Offset On (OffE)**: enable an end offset
- **End Offset (dE)**: end offset distance
- **Extend Start (ExtS)**: when Start Offset is on, dS pushes the first bar OUTWARD past From (same edge line) instead of trimming inward
- **Extend End (ExtE)**: when End Offset is on, dE pushes the last bar OUTWARD past To (same edge line) instead of trimming inward

SALIDAS:

- **Bars (Bars)**: one longitudinal bar definition per point in the package
- **Count (N)**: how many bars this package generated

d. Longitudinal Assemble (RbLongAsm)

Resuelve el anclaje de cada barra longitudinal sobre el riel de cada estación y une los puntos en una polilínea

ENTRADAS:

- **Section (Sec)**: cross section(s) - one Section broadcasts across every station, or a list matching Plane one-to-one for a tapering run
- **Plane (Pl)**: one Plane per station, in order along the run; with nothing wired, defaults to a single WorldXY station (same as RbSec)
- **Cover (Cov)**: CoverExterior/CoverInterior bundle (RbCov output)
- **Bars (Bars)**: longitudinal bar definitions (RbLong output)

SALIDAS:

- **Section (Sec)**: the first station's Section, passed through - wire straight into the viewer's Section input (RbView / RbLongView) instead of cabling it separately
- **Cover (Cov)**: the cover-offset rail(s) actually used by the resolved bars, one branch per station - wire into Rebar Longitudinal Viewer's Cover input
- **Bars (Bars)**: resolved bar geometry, one per longitudinal bar

e. Longitudinal Viewer (RbLongView)

Muestra la sección (atenuada) con las barras de un paquete longitudinal como puntos en cada estación, con un color estándar por diámetro

ENTRADAS:

- **Section (Sec)**: cross section, drawn muted for context (optional)
- **Plane (Pl)**: plane the Section context is drawn on; defaults to WorldXY
- **Cover (Cov)**: cover-offset rail(s) (RbLongAsm's Cover output), drawn in magenta (optional)
- **Bars (Bars)**: resolved longitudinal bars (RbLongAsm output)
- **Labels (Lbl)**: Label each bar with its Mark and diameter (off by default - many dots slow the viewport down)

SALIDAS:

- **(ninguna)**

f. Fit Arcs (RbArc)

Reconstruye las polilíneas densas de las barras como pocas líneas y arcos reales dentro de una tolerancia (1 mm por defecto), para un envío limpio a Revit

ENTRADAS:

- **Curves (C)**: Bar curves to rebuild (polylines through every station, e.g. RbGroup's Geometry)
- **Tolerance (Tol)**: Maximum deviation from the original points, in millimetres

SALIDAS:

- **Curves (C)**: Rebuilt curves: one line, one arc, or a chain of lines and arcs
- **Segments (N)**: Number of lines/arcs in each rebuilt curve

g. Lap Splice (RbSplice)

Define una regla de traslape (patrón F@/R@, longitud de traslape, escalonado) para las marcas longitudinales indicadas; se conecta a la entrada Splice de RbLongJoin

ENTRADAS:

- **Mark (Mk)**: Mark(s) this rule applies to (same Mk as RbLong/RbLongN/RbLongP)
- **Tokens (Tok)**: Ordered tokens: "F@val" fixed piece length, "R@val" fills the rest (at most one) - e.g. "R@8.5" or "F@6.0", "R@8.5". Or fixed lap positions only: "P@val" centers a lap at val (<= 1: fraction of the bar length, > 1: distance) - e.g. "P@0.27", "P@0.73"
- **Lap (Lap)**: Lap length between consecutive pieces, fixed, in document units
- **Symmetric (Sym)**: False: pattern applied start-to-end / True: mirror all but the last token at both ends
- **Alternate (Alt)**: Stagger splices: every 2nd bar of the same Mark gets its splice pattern shifted by dAlt
- **Alternate Shift (dAlt)**: Shift applied to alternate bars; leave empty for half the pattern pitch (first piece length - Lap)

SALIDAS:

- **Splice (Spl)**: splice rule - wire into RbLongJoin's Splice input

h. Longitudinal Join (RbLongJoin)

Rehace cada barra longitudinal (salida de RbLongAsm) como un trazo interpolado limpio, aplica los traslapes de RbSplice por marca y vuelve a discretizar cada pieza a una longitud mínima de segmento

ENTRADAS:

- **Bars (Bars)**: longitudinal bars (RbLongAsm output)
- **Distance (Dist)**: minimum segment length of the final polylines, in document units; empty = 1 m, lowest allowed 0.30 m
- **Splice (Spl)**: optional splice rules (RbSplice output), matched by Mark; empty = bars come out whole
- **Max Length (MaxL)**: maximum commercial bar length, in document units; empty = 9 m. Bars with any piece longer than this are reported in Rej, and removed once any Splice rule is wired

SALIDAS:

- **Bars (Bars)**: final bars (one per spliced piece), same Mark as the source bar - once Splice rules are wired, only bars within MaxL
- **Rejected (Rej)**: Marks removed because at least one of their pieces exceeds MaxL (with bar count and longest piece)

7. Tools · Controles, metrado y salida

a. Deconstruct Rebar (DeBar)

Descompone una lista de barras en marca, diámetro, grado del acero y geometría

ENTRADAS:

- **Bars (Bars)**: Reinforcement bars

SALIDAS:

- **Mark (Mk)**: Bar mark
- **Diameter (Ø)**: Bar diameter
- **Steel Grade (Grd)**: Reinforcement steel grade
- **Geometry (G)**: Bar centerline geometry

b. Group By Mark (RbGroup)

Agrupa las barras resueltas (salida de RbSec, con cualquier forma de árbol) por marca, una rama por marca, para hornear o exportar por categoría

ENTRADAS:

- **Bars (Bars)**: resolved bars (RbSec output) - any tree shape, e.g. one branch per station
- **Filter (Flt)**: optional - only keep these Mark(s) (e.g. "E-01"); leave empty to keep every mark. When given, output branch order follows THIS list's order instead of first-seen order

SALIDAS:

- **Marks (Mk)**: distinct bar marks kept, one per branch of Geometry (same order) - use to build a per-branch Layer path downstream
- **Diameter (Ø)**: that mark's bar diameter, one per Mark
- **Geometry (G)**: bar geometry, grouped one branch per Mark
- **Bars (Bars)**: the kept bars themselves (Mark/Diameter/Geometry), grouped one branch per Mark - wire into Rebar2Rvt
- **Colors (Col)**: one color per Mark from the plugin's rebar standard (a fixed color per diameter, lighter/darker variants per mark), on the same branch as that Mark's Geometry - wire into Bake's Object Color
- **Layers (Ly)**: one standard layer path per Mark, "Bridge::Rebar::Ø<mm>::<Mark>", on the same branch as that Mark's Geometry - wire into Bake's Layer

c. Rebar Schedule (RbSched)

Metrado de acero por marca: diámetro, piezas, longitud total y peso (kg), con CSV opcional

ENTRADAS:

- **Bars (Bars)**: Rebar Bars (any tree shape), e.g. RbSec and RbLongJoin outputs
- **CSV Path (Path)**: Optional .csv file to write the schedule to

SALIDAS:

- **Marks (Mk)**: One row per Mark, in natural order
- **Diameter (Ø)**: Bar diameter, mm
- **Pieces (N)**: Number of pieces
- **Length (L)**: Total length, m
- **Weight (W)**: Total weight, kg
- **Table (T)**: The schedule as text, with totals

d. Bend Check (RbBend)

Marca las barras en las que un tramo recto no alcanza para su diámetro de doblez en un vértice (demasiado corto o giro demasiado cerrado; es el mismo control)

ENTRADAS:

- **Bars (Bars)**: Reinforcement bars to check
- **Bend Diameter (Bd)**: Optional mandrel diameter in model units, for every bar. Leave empty to use each bar's own default: 4 x its diameter for closed stirrups/ties, 6 x for the rest

SALIDAS:

- **Valid (V)**: True if that bar's bends all fit, one per input bar
- **Messages (M)**: Empty if valid, otherwise which segment(s) came up short and by how much
- **Points (P)**: Midpoints of every offending segment, across all bars, for viewport highlighting

e. Bend Diameter (RbBD)

Diámetro mínimo de doblez estándar (al estilo ACI 318) para un diámetro de barra en pulgadas, por rangos de tamaño

ENTRADAS:

- **Diameter (Ø)**: Bar diameter in inches (e.g. from the Rebar Size value list)
- **Stirrup Or Tie (St)**: True for a stirrup/tie (closed loop) bar, false for a straight/longitudinal bar

SALIDAS:

- **Bend Diameter (Bd)**: Standard minimum bend (mandrel) diameter, in inches

f. Rebar Clash (RbClash)

Encuentra barras que se cortan entre sí con sus diámetros reales: contacto (solape de pocos milímetros) o cruce

ENTRADAS:

- **Bars A (A)**: Rebar Bars to check (any tree shape)
- **Bars B (B)**: Optional second set; empty = check A against itself
- **Contact (Ct)**: Overlap still counted as contact, in mm

SALIDAS:

- **Points (P)**: Midpoint between the two bars at each clash
- **Clashes (Cl)**: "MarkA x MarkB: crossing|contact, overlap N mm" per clash
- **Crossing (X)**: True for a real crossing, false for contact - one per clash
- **Summary (S)**: Clash count per pair of Marks

g. Rebar Nudge (RbNudge)

Desplaza las barras que chocan (por ejemplo, estribos) a lo largo del tablero, unos milímetros cada vez, hasta librar los obstáculos o hasta la posición de menor interferencia

ENTRADAS:

- **Bars (Bars)**: Bars allowed to move (planar, e.g. stirrups) - any tree shape, kept in the output
- **Obstacles (Obs)**: Bars that stay where they are (e.g. longitudinal bars)
- **Max Shift (Max)**: Largest shift either way, in mm
- **Step (St)**: Shift increment, in mm
- **Contact (Ct)**: Overlap still accepted as contact, in mm

SALIDAS:

- **Bars (Bars)**: Bars at their final position, same tree as the input
- **Shift (Sh)**: Shift applied to each bar, in mm (+ along its plane normal)
- **Overlap (Ov)**: Remaining overlap of each bar, in mm (0 = clear)
- **Summary (S)**: How many bars stayed, moved, or stay with an overlap



h. Bake Geometry (Bake)

Hornea geometría en el documento de Rhino bajo una capa dada (anidada), con colores y agrupación opcionales

ENTRADAS:

- **Geometry (G)**: Geometry to bake
- **Layer (Layer)**: Target layer path, e.g. "Bridge::Rebar::Stirrups" (created if missing)
- **Layer Color (LCol)**: Color to (re)apply to the target layer
- **Object Color (GCol)**: Per-object color override; leave unset to use the layer's color
- **Group (Group)**: Group the baked objects together
- **Bake (Bake)**: Set True to actually bake

SALIDAS:

- **IDs (IDs)**: Object IDs of the baked geometry
- **Geometry (G)**: Passthrough of the input geometry, for wiring straight into Custom Preview
- **Object Color (GCol)**: Passthrough of the input object color, for wiring straight into Custom Preview



i. Splice Bars (RbSplBar)

Corta en piezas traslapadas las barras más largas que el largo comercial (patrón F@/R@ o posiciones fijas P@), conservando marca y diámetro

ENTRADAS:

- **Bars (Bars)**: Rebar Bars, any tree shape
- **Tokens (Tok)**: Piece pattern: e.g. "R@9.0" (fill with 9 m pieces), "F@6","R@9", or fixed lap centers "P@0.5"
- **Lap (Lap)**: Lap length, in model units
- **Symmetric (Sym)**: Mirror all but the last token at both ends (F@/R@ only)
- **Max Length (MaxL)**: Only bars longer than this are spliced, in model units
- **Marks (Mk)**: Optional: only these Marks are spliced

SALIDAS:

- **Bars (Bars)**: Bars after splicing (one item per piece), same tree as the input
- **Spliced (N)**: How many bars were spliced

8. Revit · Rebar nativo en Revit (solo dentro de Rhino.Inside.Revit)

a. Send Curves To Revit (Crv2Rvt)

Crea Rebar nativo de Revit a partir de curvas simples (polilíneas, líneas y arcos); las barras no planas pasan como Rebar de forma libre (acero manual, o Geometry/Marks de RbGroup), con marca opcional por rama; fijadas y rastreadas como los componentes de elementos de Rhino.Inside

ENTRADAS:

- **Curves (C)**: Curves to turn into Revit Rebar: polylines, lines and arcs (arcs go to Revit as true arcs; a non-planar bar, e.g. an arc with a leg out of its plane, becomes free-form Rebar) - any tree shape; each branch is one group sharing a Mark/Bar Type/Material. Disconnecting it removes the bars this component sent
- **Marks (Mk)**: optional - Revit Mark, one per branch of Curves (last one repeats; a single value marks everything)
- **Host (H)**: Revit element the rebar will be hosted on (a DirectShape works fine)
- **Bar Type (BT)**: Revit RebarBarType (e.g. #4 for a 1/2" bar), one per branch of Curves (last one repeats; a single one applies to every curve)
- **Material (Mat)**: optional - Revit material, one per branch (last one repeats). Revit keeps material per BAR TYPE, so this recolours every bar of that type in the project. Leave empty to keep the bar type's current material
- **Run (Run)**: optional - False leaves Revit untouched (no delete, no create); empty or True sends
- **Elements (E)**: Created Revit Rebar elements, same branches as Curves (null where a curve was skipped)

SALIDAS:

- **(ninguna)**

b. Send Rebar To Revit (Rebar2Rvt)

Crea Rebar nativo de Revit a partir de barras de GH_Bridge (por ejemplo, la salida Bars de RbGroup) y escribe la marca de cada una; fijadas y rastreadas como los componentes de elementos de Rhino.Inside

ENTRADAS:

- **Rebar Bars (Bars)**: GH_Bridge reinforcement bars to create in Revit - any tree shape (e.g. RbGroup's Bars, one branch per Mark); Elements output keeps the same branches. Catalog hooks (H90, H135...) and RbStir's closing hooks go to Revit as Rebar Hook Types (library shapes such as 02, 08, T1; a matching hook type is created when the project has none); other hooks go as drawn. Disconnecting it removes the bars this component sent
- **Host (H)**: Revit element the rebar will be hosted on - Rebar.CreateFromCurves only needs a generic Element (a DirectShape works fine), not a true HostObject

- **Bar Type (BT)**: Revit RebarBarType (e.g. #4 for a 1/2" bar), one per branch of Bars (last one repeats; a single one applies to every bar)
- **Material (Mat)**: optional - Revit material, one per branch (last one repeats). Revit keeps material per BAR TYPE, so this recolours every bar of that type in the project. Leave empty to keep the bar type's current material
- **Run (Run)**: optional - False leaves Revit untouched (no delete, no create); empty or True sends
- **Elements (E)**: Created Revit Rebar elements, same branches as Bars (null where a bar was skipped)

SALIDAS:

- **(ninguna)**

9. Curves · Herramientas de curvas para armado manual

a. Curve Angle At End (CrvAng)

Ángulo de la tangente (grados, antihorario desde el eje X del plano) en el extremo inferior de una curva en el plano

ENTRADAS:

- **Curve (C)**: Curve to measure
- **Plane (Pl)**: Reference plane

SALIDAS:

- **Angle (Ang)**: Tangent angle in degrees, 0-360
- **Plane (Pl)**: Reference plane used

b. Discretize Curve (DiscCrv)

Discretiza una curva según el tipo de tramo: los rectos conservan solo sus extremos y los arcos se subdividen por distancia

ENTRADAS:

- **Curves (C)**: Curves to discretize
- **Distance (D)**: Subdivision distance for arc segments
- **Vertex Mode (V)**: Add points just before/after every straight-segment vertex instead of only at it

SALIDAS:

- **Points (P)**: Discretization points, one branch per curve
- **Polylines (Pl)**: Polyline through the discretization points, one branch per curve

c. Divide Curve By Count (DivCrvN)

Divide curvas en N tramos iguales; entrega una rama por índice de punto a través de todas las curvas

ENTRADAS:

- **Curves (C)**: Curves to divide

- **Divisions (N)**: Number of divisions
- **Start Offset On (OffS)**: Enable a start offset
- **Start Offset (dS)**: Start offset distance
- **End Offset On (OffE)**: Enable an end offset
- **End Offset (dE)**: End offset distance

SALIDAS:

- **Points (P)**: Division points, one branch per point index across all curves

d. Divide Curve By Length (DivCrv)

Divide curvas con un paso de longitud fijo; entrega una rama por índice de punto a través de todas las curvas

ENTRADAS:

- **Curves (C)**: Curves to divide
- **Distance (D)**: Division step (arc length)
- **Mode (Mode)**: 0 = Beginning, 1 = Middle, 2 = End, 3 = Extreme (leftover distributed accordingly)
- **Start Offset On (OffS)**: Enable a start offset
- **Start Offset (dS)**: Start offset distance
- **End Offset On (OffE)**: Enable an end offset
- **End Offset (dE)**: End offset distance

SALIDAS:

- **Points (P)**: Division points, one branch per point index across all curves

e. Extend Curve (ExtCrv)

Extiende una curva en su inicio o en su fin una distancia fija

ENTRADAS:

- **Curves (C)**: Curves to extend
- **Distance (D)**: Distance to extend
- **End (E)**: False: extend the Start / True: extend the End
- **Plane (Pl)**: Reference plane for straight curves

SALIDAS:

- **Curves (C)**: Extended curves

f. Extend Curve Both Ends (ExtCrv2)

Extiende una curva en sus dos extremos la misma distancia fija

ENTRADAS:

- **Curves (C)**: Curves to extend
- **Distance (D)**: Distance to extend at each end
- **Plane (Pl)**: Reference plane for straight curves

SALIDAS:

- **Curves (C)**: Extended curves

g. Extend Point To Curve (ExtPtCrv)

Lanza un rayo desde cada punto con un ángulo fijo y lo recorta donde toca una curva objetivo

ENTRADAS:

- **Points (P)**: Ray origins
- **Target (T)**: Target curve
- **Angle (A)**: Ray angle in degrees, 0 = plane's X axis, CCW
- **Plane (Pl)**: Reference plane

SALIDAS:

- **Curves (C)**: Line from each point to its hit on the target (null if no hit)
- **Distances (D)**: Distance from each point to its hit (null if no hit)

h. Line From Point (LinePt)

Dibuja una línea desde un punto con un ángulo en el plano y una distancia dados

ENTRADAS:

- **Points (P)**: Start points
- **Angle (A)**: Angle in degrees, 0 = plane's X axis, CCW
- **Distance (D)**: Line length; negative flips the direction 180 degrees
- **Plane (Pl)**: Reference plane

SALIDAS:

- **Curves (C)**: Lines, one per point

i. Project Curve Between (PrjCrv2)

Extiende los dos extremos de una curva, cada uno hasta su propia curva objetivo

ENTRADAS:

- **Curve (C)**: Curve to extend
- **Start Target (TS)**: Target(s) for the Start end
- **End Target (TE)**: Target(s) for the End end
- **Plane (Pl)**: Reference plane

SALIDAS:

- **Curves (C)**: Curves extended toward their Start/End targets

j. Project Curve To Curve (PrjCrv)

Extiende una curva siguiendo su propia tangente hasta que toca una curva objetivo

ENTRADAS:

- **Curve (C)**: Curve to extend
- **Target (T)**: Target curve(s) to extend toward - one, or one per Crv1
- **From Start (S)**: False: extend from the End (default) / True: extend from the Start
- **Plane (Pl)**: Reference plane

SALIDAS:

- **Curves (C)**: Extended curves (null where no hit was found)
- **Plane (Pl)**: Reference plane used

k. Orient Curves To Planes (OrientPl)

Copia curvas plantilla desde un punto de referencia a cada plano objetivo (filtrado)

ENTRADAS:

- **Curves (C)**: Template curves, defined around Point in world XY
- **Reference (R)**: Reference origin the template curves are built around
- **Planes (Pl)**: Target planes to copy the template onto
- **Side (S)**: 0 = All, 1 = Right half, 2 = Left half
- **Alternate (Alt)**: 0 = All, 1 = Even indices only, 2 = Odd indices only

SALIDAS:

- **Curves (C)**: Copied curves, one branch per target plane index
- **Points (P)**: Reference point copied onto each target plane

l. Rebuild Curve (RebuildCrv)

Reparte de nuevo los puntos de una curva densamente discretizada a espaciamiento uniforme, conservando los dos extremos

ENTRADAS:

- **Curves (C)**: Densely-discretized curves to rebuild
- **Spacing (S)**: Target spacing between points
- **Interpolated (I)**: False: rebuild as a Polyline / True: rebuild as an interpolated curve

SALIDAS:

- **Curves (C)**: Rebuilt curves
- **Points (P)**: New points, one branch per curve
- **Count (N)**: Number of points generated, per curve
- **Actual Spacing (AS)**: Actual resulting spacing, per curve

m. Rebuild Curve With Hook (RebuildCrvH)

Reparte de nuevo una curva densamente discretizada a espaciamiento uniforme, conservando el gancho que detecta en un extremo

ENTRADAS:

- **Curves (C)**: Densely-discretized curves to rebuild

- **Spacing (S)**: Target spacing for the longitudinal (non-hook) span
- **Hook Length (Hk)**: Known length of the hook's single end segment, used to detect it
- **Tolerance (Tol)**: Length-match tolerance for hook detection
- **Prefer End (E)**: If both ends match the hook length: False prefers Start, True prefers End
- **Interpolated (I)**: False: rebuild the longitudinal span as a Polyline / True: as an interpolated curve (the hook always keeps its original points)

SALIDAS:

- **Curves (C)**: Rebuilt curves (null where no hook was detected)
- **Points (P)**: New points, one branch per curve
- **Side (Side)**: "S" (hook at Start), "E" (hook at End), or null if not detected
- **Hook Point (HP)**: Vertex where the hook begins
- **Count (N)**: Number of points in the rebuilt curve
- **Actual Spacing (AS)**: Actual resulting spacing of the longitudinal span

n. Splice Curves (SplCrv)

Divide curvas en piezas traslapadas según un patrón de tokens F@val/R@val, secuencial o simétrico

ENTRADAS:

- **Curves (C)**: Curves to splice
- **Tokens (Tok)**: Ordered tokens: "F@val" fixed length, "R@val" fills the rest (at most one, plain numbers are read as F@val). Or fixed lap positions only: "P@val" centers a lap at val (<= 1: fraction of the length, > 1: distance)
- **Lap (Lap)**: Overlap between consecutive pieces
- **Symmetric (Sym)**: False: sequential start-to-end / True: mirror all but the last token at both ends

SALIDAS:

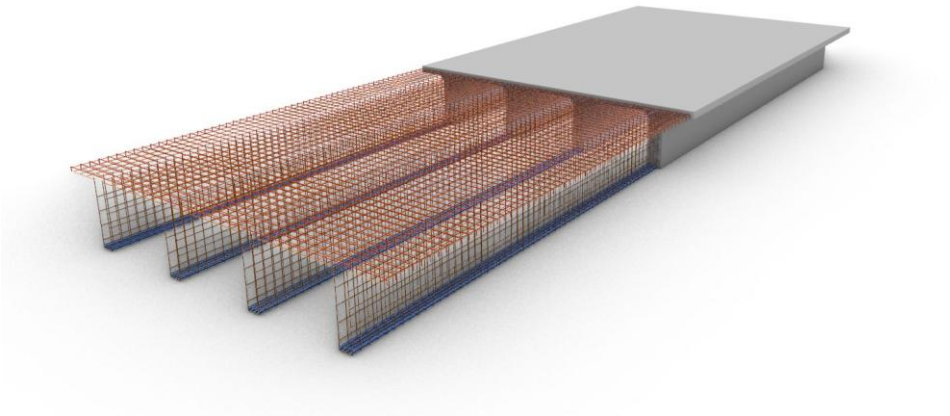
- **Curves (C)**: Resulting spliced pieces, in order
- **Token (Tok)**: "F" fixed, "R" fill piece, "R!" the fill piece adjusted to absorb rounding - same order as Curves
- **Length (L)**: Nominal length of each piece - same order as Curves
- **Check (Chk)**: Per input curve: covered length minus actual curve length (0 = exact fit)
- **Warning (W)**: Per input curve: True if something didn't fit exactly (see Check)

5. EJEMPLOS

- **Example_1**: cajón unicelular con acero transversal y longitudinal, y envío a Revit.
- **Example_2**: siete tableros con vigas cajón, de acero y prefabricadas, ensamblados por asiento de secciones, con la biblioteca de secciones JSON.

Los dos vienen en la carpeta Examples del .zip. Abre Example.3dm antes que el .gh: la definición toma curvas de ese archivo. Las rutas a los JSON son relativas a la carpeta del ejemplo.

El kit de ejemplos completos, desde Example_3 (un puente cajón curvo de 3 vanos y un puente viga-losa de 25 m, de punta a punta, con metrado), está en daapcloud.com/gh-bridge.



Example_5 (kit): tablero viga-losa de 25 m con su acero.

Limitaciones conocidas (v1.0)

- Un gancho de 180° en una barra recta no define un plano: se envía a Revit dibujado (forma libre) en lugar de como tipo de gancho de Revit.
- Con Rhino.Inside.Revit, Revit procesa los envíos cuando su ventana tiene actividad: si un envío no avanza, haz un clic en la vista de Revit.

Licencia y responsabilidad

Freeware: gratuito para cualquier uso, incluido el trabajo profesional. No se permite venderlo, modificarlo ni descompilarlo. Texto completo en LICENSE.txt. Copyright (c) 2026 Andrei Damian Pacheco.

GH_Bridge es una herramienta de modelado. La geometría, el acero y el metrado que produce deben ser revisados por un ingeniero calificado, que es el responsable del diseño.

Soporte y actualizaciones: daapcloud.com/gh-bridge