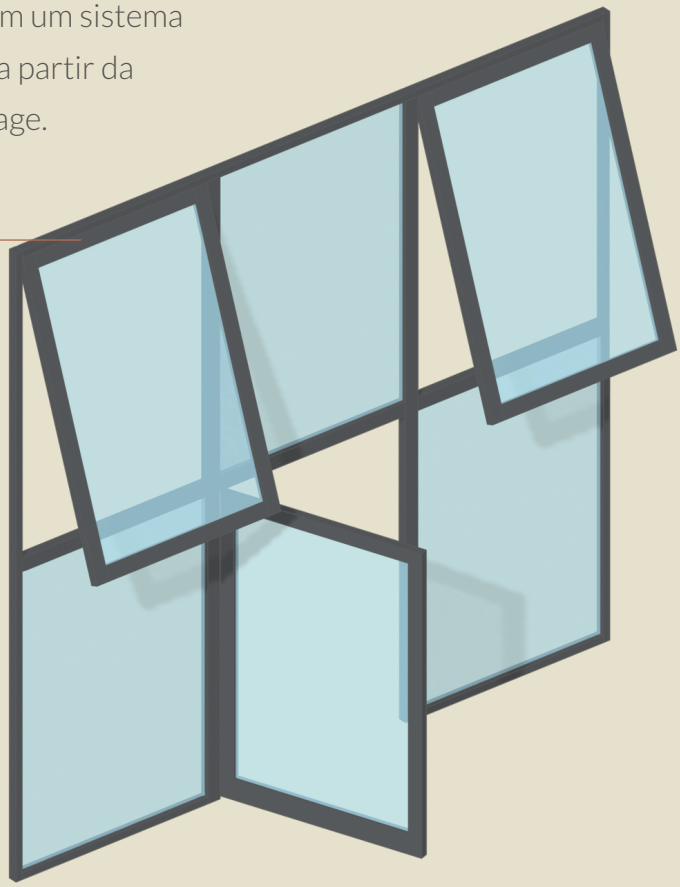
OBJETOS GDL COMPLEXOS COM INTELIGÊNCIA ARTIFICIAL

Método para construir objetos GDL com um sistema de código com IA, isolado do Archicad, a partir da biblioteca de esquadrias e portas da Stage.

Se interessou? Baixe a biblioteca de graça em stageaec.com.br/biblioteca



stage Architecture Engineering Construction

ANTES DE COMEÇAR

A biblioteca STG de esquadrias e portas tem quatro famílias (correr, modular, porta de madeira e porta camarão), 14 arquivos de objeto, cerca de 8.200 linhas de GDL e 399 parâmetros. Quem escreveu o código foi a IA. Quem decidiu o que o objeto devia fazer, conferiu cada revisão e recusou o que não servia fomos nós.

O método deste texto é isolado do Archicad. A IA não conecta, não automatiza e não opera o programa. Você trabalha só no sistema de código com IA da sua preferência (nós usamos o Claude Code): ele escreve o GDL, compila o objeto e mostra uma prévia no navegador. O Archicad entra só quando você abre o objeto pronto para conferir, e o que você vê volta para a IA como instrução da próxima revisão.

A divisão de trabalho é esta: a IA escreve, compila, verifica e documenta; o arquiteto define o objeto, fornece as fontes e julga o resultado. Os erros mais caros que tivemos vieram de deixar a IA decidir uma medida sem fonte.

Um aviso antes de tudo: um bom objeto vem da tentativa e erro. A esquadria de correr passou por 34 revisões; nos dois primeiros dias, foram cerca de 30 versões seguidas. Nenhuma revisão sai certa de primeira. O que faz o objeto ficar bom é o ciclo curto: pedir, abrir, apontar o que está errado, pedir de novo.

DOWNLOAD GRATUITO

Se interessou? Baixe a biblioteca de graça. As quatro famílias deste ebook (correr, modular, porta de madeira e porta camarão) estão prontas para usar no Archicad 28 em diante. Entre em stageaec.com.br/biblioteca, faça o cadastro e baixe.

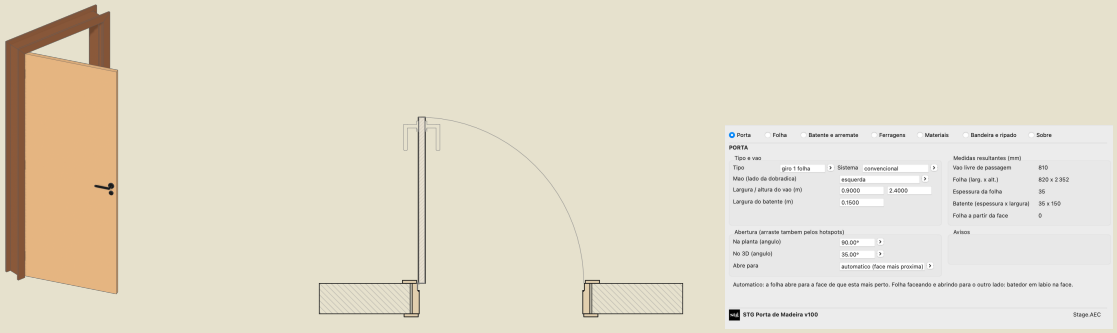
01 O QUE É O ARCHICAD E O QUE É GDL

O Archicad é o software BIM da Graphisoft. Nele, o projeto é um modelo: paredes, lajes e esquadrias são elementos com dados, e as plantas, cortes, elevações e quantitativos saem desse modelo.

Portas, janelas e mobiliário são objetos de biblioteca. Cada objeto é escrito em GDL, a linguagem de programação do próprio Archicad. Um objeto GDL reúne:

- **parâmetros:** os campos que o usuário ajusta, como largura, número de folhas e material;
- **script 3D:** a geometria do objeto no modelo;
- **script 2D:** o símbolo da planta, que não é um corte do 3D e precisa ser desenhado à parte;
- **script de interface:** a janela de configurações, com abas, listas e travas;
- **hotspots:** os pontos que o usuário arrasta na planta e no 3D.

Um objeto simples tem poucas dezenas de linhas. Uma esquadria de correr com 12 folhas, trilhos, bolsão e persiana passa de 2.500.



SCRIPT 3D SCRIPT 2D SCRIPT DE INTERFACE

Um objeto, três scripts: a mesma porta de madeira no 3D, na planta e na janela de configurações.

02 CHAT OU CÓDIGO: DUAS FORMAS DE USAR IA

A maior parte das pessoas usa IA pela interface típica, o chat: você escreve, a IA responde, e o que ela produz fica na conversa. Para um objeto GDL complexo, isso não basta. A IA não vê os seus arquivos, não compila o objeto e não sabe se o código funcionou; quem copia, cola, testa e relata o erro é você.

O método deste ebook usa a interface de codificação: um agente de IA que trabalha dentro da pasta do projeto, no seu computador.

	CHAT (INTERFACE TÍPICA)	INTERFACE DE CODIFICAÇÃO
Onde a IA trabalha	Numa janela de conversa	Na pasta do projeto, no seu computador
Arquivos	Você copia e cola trechos	Ela lê catálogos, PDFs e DWG e grava os arquivos do objeto
Comandos	Não roda nada	Compila, roda o verificador e gera a prévia
Erros	Você descobre e relata	Ela lê a mensagem do compilador e corrige
Memória do projeto	A conversa	Arquivos, revisões e anotações na pasta

O Claude Code, da Anthropic, é a ferramenta de codificação que usamos. Ele roda no terminal, no aplicativo de desktop ou dentro de um editor de código. Você conversa em português, como num chat, mas cada pedido vira arquivo escrito, comando executado e resultado conferido.

03 QUAL IA USAR

O fluxo não depende de uma marca. O critério é um só: a ferramenta precisa trabalhar na pasta do projeto, ler e gravar arquivos e rodar comandos no terminal. Hoje, entre outras, cumprem esse papel:

FERRAMENTA	EMPRESA	ONDE RODA
Claude Code	Anthropic	Terminal, aplicativo de desktop e editores
Codex	OpenAI	Terminal, aplicativo e editores
Cursor	Anysphere	Editor de código próprio, com agente
Gemini CLI	Google	Terminal
GitHub Copilot (modo agente)	GitHub / Microsoft	Editores de código
Kimi Code	Moonshot AI	Terminal e editores

Dois avisos. Nós construímos a biblioteca STG com o Claude Code; nas outras ferramentas, o método vale, mas não refizemos o percurso completo. E nomes, versões e preços nesse mercado mudam a cada poucos meses: esta lista é de outubro de 2026.

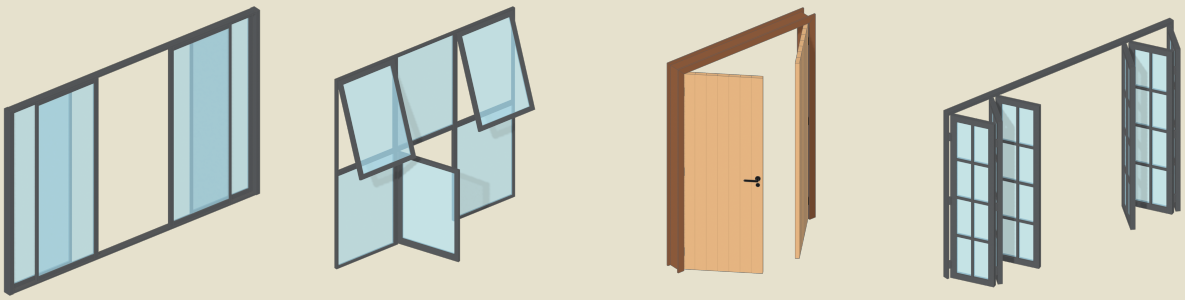
04 DIGA O QUE É O OBJETO E PARA QUE ELE SERVE

O primeiro texto que você entrega à IA decide metade do resultado. Não basta pedir "uma esquadria de correr". É preciso dizer:

- **o que o objeto é:** esquadria de correr multifolhas, em linha de catálogo de alumínio ou madeira;
- **onde ele vive no Archicad:** porta, janela ou objeto livre, e se corta a parede;
- **o que precisa variar:** número de folhas, sentido, trilhos, preenchimento de cada folha, bandeiras;
- **o que ele precisa superar:** no nosso caso, a porta de correr multifolhas da biblioteca nativa do Archicad 28 foi a referência mínima;
- **como você vai aceitar o resultado:** uma lista de casos que o objeto tem que desenhar certo.

O documento que mais rendeu foi um caderno de requisitos escrito a partir de uma pergunta simples: se eu fosse contratar este objeto de um fornecedor, o que eu exigiria? Dele saiu uma matriz de casos de aceite (2 folhas centrais, 4 folhas para a esquerda com 3 trilhos, folha fixa no meio, bolsão, giro no canto) que virou a lista de conferência de cada revisão.

> PROMPT
>
> Quero um objeto GDL para Archicad 28: [o que é]. Ele vai ser usado como [porta / janela / objeto livre]. Precisa variar [lista]. A referência mínima é [objeto nativo ou de mercado]. Antes de escrever código, me devolva um caderno de requisitos com a matriz de casos de aceite e as perguntas que você precisa que eu responda.



CORRER MODULAR PORTA DE MADEIRA PORTA CAMARÃO

As quatro famílias da biblioteca STG. Cada uma faz porta, janela ou objeto livre com o mesmo código.

05 ENTREGUE AS FONTES CERTAS

A IA não sabe a medida de um perfil de catálogo. Se você não der a fonte, ela inventa, e inventa com confiança. Tivemos um detalhe de perfil desenhado com uma peça que não existia no catálogo. Daí saiu a regra que vale até hoje: **detalhe de perfil só com base em corte de catálogo.**

O que entregar, em ordem de qualidade:

1. **Catálogo do fabricante em PDF.** Usamos seis catálogos de alumínio. A IA leu o texto, rasterizou as páginas de corte em alta resolução e extraiu as cotas.
2. **DWG.** Um detalhe de porta de madeira chegou em DWG. A IA converteu o arquivo para dados legíveis e redesenhou a geometria por script.
3. **PDF de projeto executivo.** Pranchas de detalhamento de projetos reais deram as medidas de batente, guarnição e folga.
4. **Foto, em último caso.** Uma foto de obra definiu a bandeira superior inclinada. Serve para a forma, não para a medida.

O passo que fez diferença foi exigir que cada número extraído viesse marcado com a origem. Nas nossas extrações, toda cota tem página e etiqueta: `(texto)` quando está escrita no catálogo, `(desenho)` quando foi lida no corte, `(medido)` quando foi medida com régua sobre o desenho calibrado por uma cota conhecida, `(inferência)` quando foi deduzida, e "não consta" quando não existe. Assim você sabe em qual número confiar e qual conferir em corte.

> PROMPT
>
> Leia os catálogos em [pasta]. Para cada linha, extraia vistas, espessuras, trilhos e passo. Marque cada número com a página e a origem: texto, desenho, medido ou inferência. O que não estiver no catálogo fica "não consta". Não complete lacunas por conta própria.

06 MONTE O AMBIENTE

O sistema de código com IA trabalha no seu computador, com acesso às pastas e ao terminal. Ele instala o que precisa conforme a tarefa pede. Nada disso conversa com o Archicad aberto:

FERRAMENTA	PARA QUE SERVE
Sistema de código com IA (nós usamos o Claude Code)	Escreve o GDL, os scripts de build e a documentação; roda os comandos.
LP_XMLConverter	Programa de linha de comando que vem na instalação do Archicad. Compila o GDL em `.gsm`, verifica a biblioteca e monta o LCF, sem abrir o Archicad.
Python	Scripts de build, extração de catálogo, prévia e manual.
Three.js	Prévia 3D no navegador, para acertar a geometria antes do GDL.
Blender (sem interface)	Renderiza imagens do objeto para manual e divulgação.

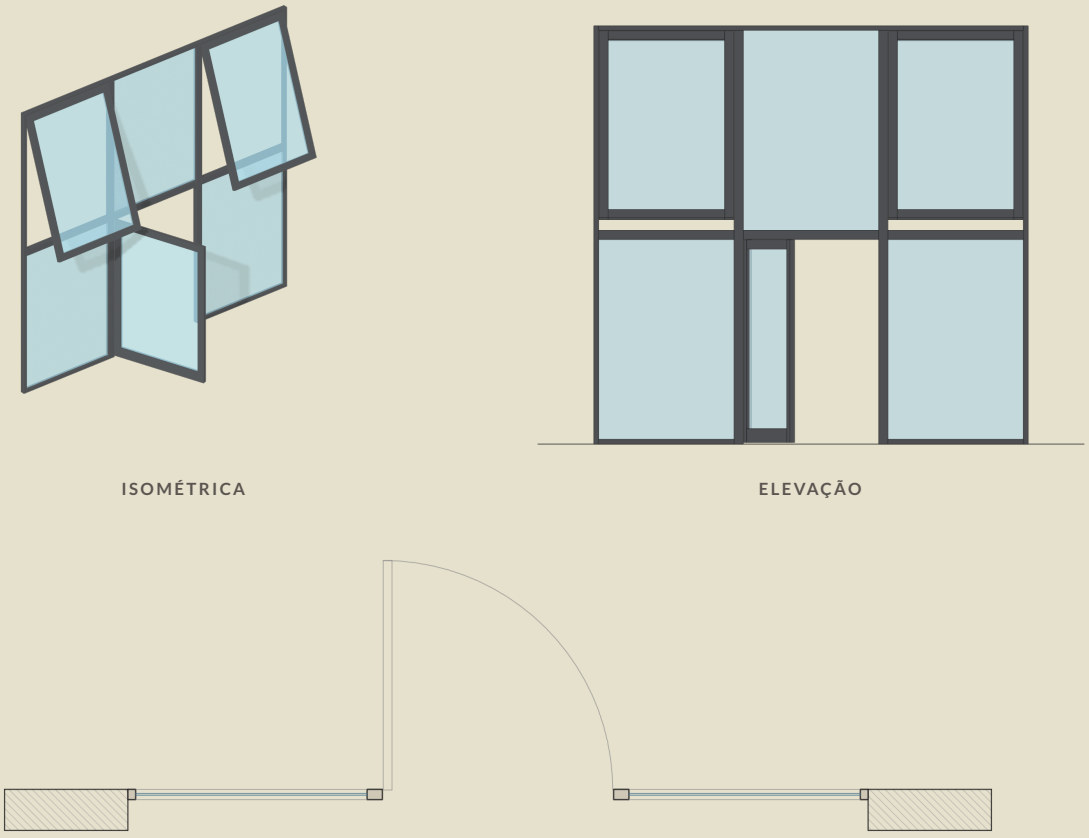
O seu lado também precisa de ordem: um arquivo `.pln` só para teste, onde você carrega cada revisão nova. Objeto em teste não entra em projeto de cliente.

07 VEJA ANTES DE ABRIR O ARCHICAD

Para geometria nova, uma prévia em Three.js no navegador economiza voltas. A rampa paramétrica da Stage foi assim: a forma, as inclinações e os patamares foram acertados num modelo 3D interativo, e a especificação fechada virou o GDL.

Na biblioteca de esquadrias fomos além: a IA construiu um visualizador que lê o próprio código GDL e desenha, no navegador, a interface do objeto, a planta, a elevação e a vista lateral. Assim, boa parte dos erros aparece antes de o objeto chegar ao Archicad.

A prévia não substitui a conferência no programa. Fizemos um protótipo da interface em HTML e o descartamos: o que interessa é como o diálogo se comporta dentro do Archicad, com as travas e as listas reais. A regra que ficou: **a interface se confere no Archicad, a cada revisão do `.gsm`.**



Prévia no navegador: planta e elevação desenhadas pelo visualizador a partir do código GDL, sem abrir o Archicad.

08 PROGRAME O GDL

O objeto não é escrito direto no editor do Archicad. Ele vive como texto: um arquivo XML com os parâmetros e um arquivo `.gdl` para cada script (2D, 3D, parâmetros, interface). Um script de build em Python junta tudo e chama o LP_XMLConverter para gerar o `.gsm`.

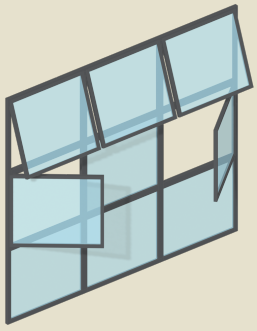
Essa organização permite três coisas que o editor não permite:

- **revisão numerada:** cada build gera uma revisão nova (R01, R02...), com nome e identificador próprios, e nunca grava por cima da anterior. A esquadria de correr passou por 34 revisões; a modular, por 9; as duas portas, por 18 cada;
- **código compartilhado:** cada família tem uma macro núcleo, que desenha tudo, e cascas finas para porta, janela e objeto livre. A modular chama o núcleo da correr para os seus módulos de correr;
- **código gerado:** a grade da modular tem 18 módulos. O script que escreve os 18 blocos de GDL é Python, não digitação.

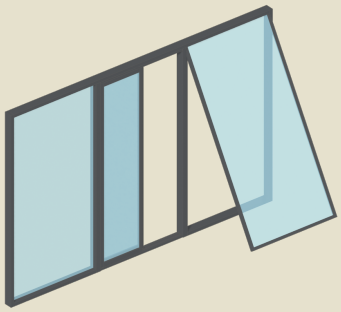
Duas decisões de método pesaram mais que qualquer linha de código:

Reorganizar, não reescrever. Numa revisão crítica, a IA concluiu que o núcleo da primeira versão "não era de objeto profissional" e propôs reescrever do zero. Recusamos. A regra passou a ser: a versão nova é a anterior reorganizada, nunca reescrita. O que já tinha sido conferido não volta à estaca zero.

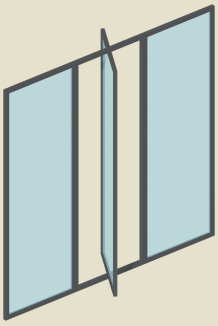
Uma família por vez. Começamos pelo núcleo e pela correr. Só depois vieram modular, madeira e camarão, cada uma reaproveitando o que a anterior provou.



GRADE 3 X 3 MISTA



PORTA DE CORRER ENTRE FIXO E MAXIM-AR



PIVOTANTE E FIXOS

Modular: os 18 módulos da grade são escritos por script. A porta de correr chama o núcleo da família correr.

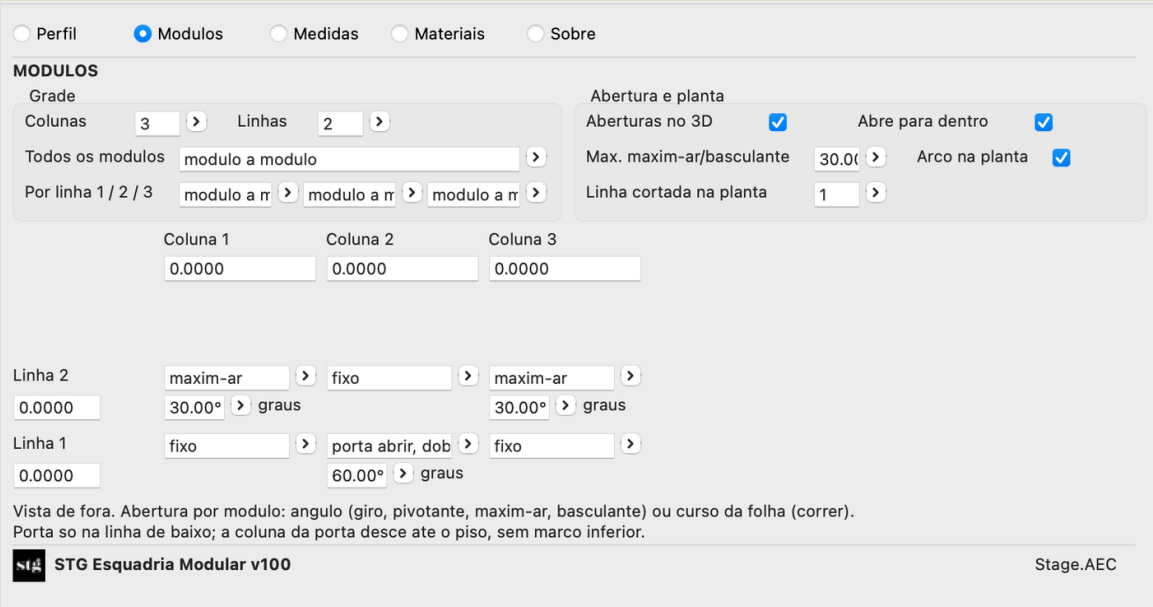
09 PEÇA A INTERFACE

A interface é o que o usuário vê. Vale a pena descrever como se fosse para um designer:

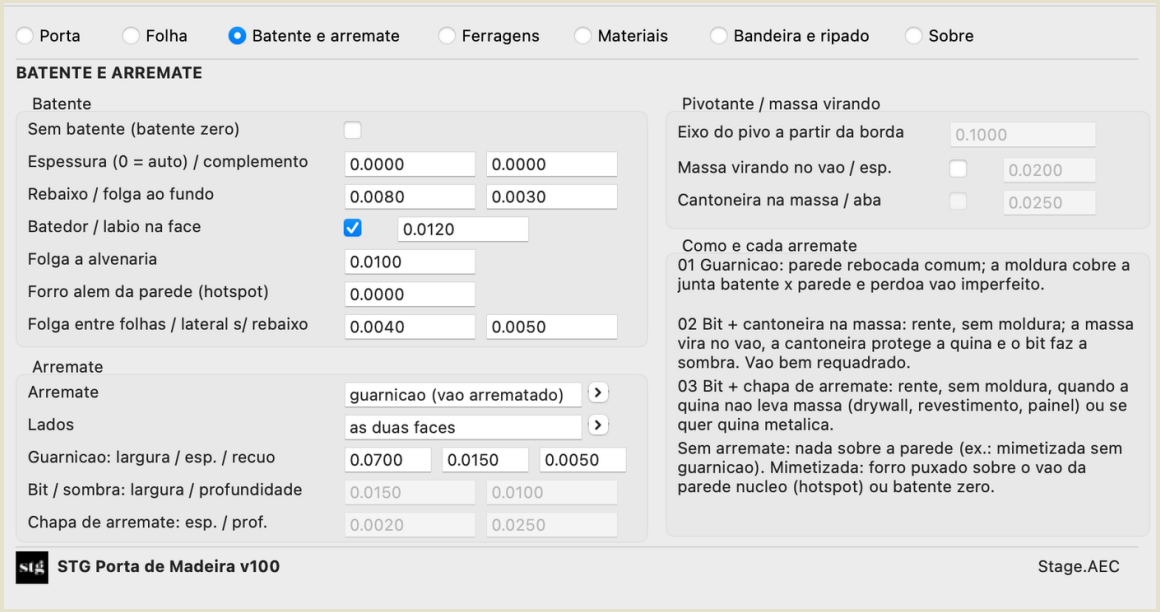
- **abas:** quais são e o que vai em cada uma. A correr tem 10 abas; a madeira, 7; a modular e a camarão, 5;
- **listas fechadas:** linha de catálogo, sentido de abertura, tipo de módulo. Lista fechada evita valor inválido;
- **travas:** campo que não se aplica fica cinza. Na porta de madeira, a mão da dobradiça trava quando a porta tem 2 folhas;
- **padrões em cascata:** trocar a linha de catálogo preenche vistas, trilhos e espessuras de uma vez;
- **avisos:** o objeto avisa quando a configuração não fecha ("vão livre muito pequeno para porta");
- **o que não entra:** nome de projeto, de cliente ou de fornecedor de referência não aparece na interface.

> PROMPT
>
> Interface em [n] abas: [nome da aba: campos]. Use listas fechadas para [campos]. Trave [campo] quando [condição]. Ao trocar [campo], preencha [campos] com os valores da linha. Mostre um aviso quando [condição]. Rode o verificador a cada mudança.

Um alerta: a IA errou a sintaxe da interface mais de uma vez, com segurança total. A navegação por abas nativa não funcionou como a documentação sugeria, e um comando de trava que ela usou não existe em GDL. O que resolveu foi tratar o verificador do LP_XMLConverter como juiz: nenhuma revisão sai sem passar por ele com zero erros.



ABA MÓDULOS · ESQUADRIA MODULAR



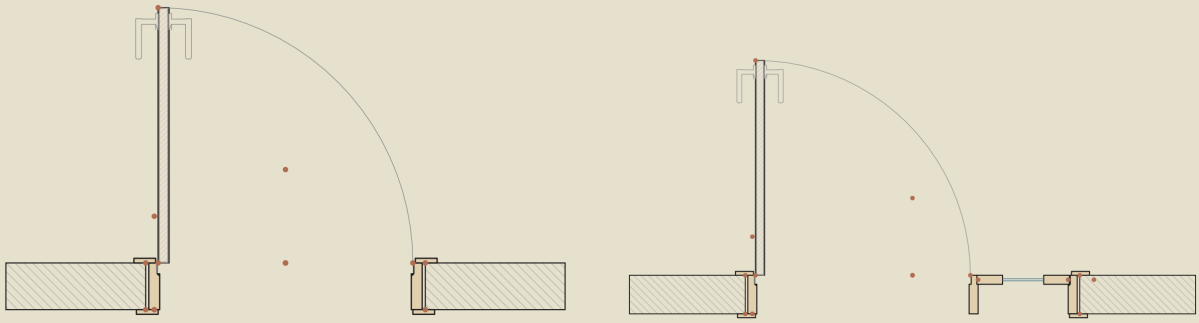
ABA BATENTE E ARREMATE · PORTA DE MADEIRA

10 FALE DOS HOTSPOTS

Hotspot é o ponto que o usuário arrasta na planta ou no 3D. Se você não pedir, a IA faz o mínimo. Nossas duas regras:

- **hotspot para tudo que mexe:** abertura de cada folha, ângulo de giro, curso de correr, largura de bandeira, posição do puxador;
- **hotspot de referência fixo** nos centros e eixos, para encaixar o objeto com precisão.

Peça o mapa por escrito, separado por vista, antes do código: "na planta, arrasta-se X, Y, Z; no 3D, arrasta-se A, B". A prévia no navegador mostra onde cada hotspot está. Se ele se comporta bem na mão, só você confirma, arrastando no Archicad.



GIRO 1 FOLHA BANDEIRA LATERAL

Pontos terracota: hotspots da planta. Ângulo da folha, posição da folha no batente e largura da bandeira se ajustam arrastando.

11 O CICLO: TENTATIVA E ERRO

Cada revisão segue o mesmo caminho:

1. **A IA** escreve a mudança, compila e passa no verificador com zero erros.
2. **A IA** mostra a prévia no navegador: interface, planta e elevação dos casos de aceite.
3. **Você** carrega o `.gsm` no arquivo de teste, coloca o objeto e confere planta, 3D, interface e hotspots.
4. **Você** devolve o que viu: print da tela e o que está errado, em frases curtas.
5. **A IA** sobe a revisão e corrige, sem mexer no que já estava certo.

É aqui que o objeto fica bom. As primeiras revisões quebram, e isso é esperado. O que tira o objeto do lugar é a quantidade de voltas curtas, não uma instrução perfeita no começo.

O que aprendemos conferindo:

- **erro na planta não aparece no 3D.** Um puxador mal escrito apagou a planta inteira da porta enquanto o 3D seguia perfeito. Confira as duas vistas, sempre;
- **índice de pena, hachura e material é do projeto,** não do objeto. Um valor que existe no seu template pode não existir no do vizinho;
- **mande o print junto com a frase.** "A folha da direita está passando do marco" com a imagem não deixa dúvida sobre qual folha nem sobre o quanto.

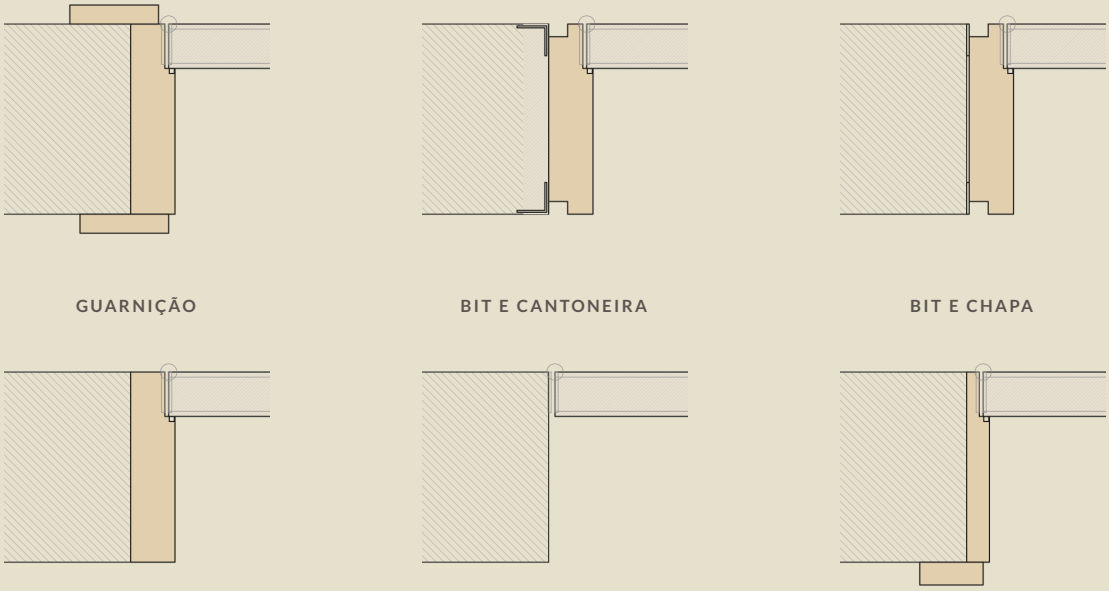
> PROMPT
>
> Carreguei a R[nn]. Na planta: [o que está errado]. No 3D: [o que está errado]. Na interface: [o que está errado]. Segue o print. Suba a revisão e corrija só isso; não mexa no que já está certo.

12 EXPORTE E ENTREGUE

A entrega tem quatro cuidados:

1. **Código fechado.** A versão final é compilada com senha. O objeto funciona normalmente, mas o editor do Archicad não mostra os scripts.
2. **Identificador estável.** Cada objeto tem um GUID fixo. Quando sai uma versão nova com o mesmo GUID, ela substitui a anterior nos projetos. Renomear um parâmetro obriga a trocar o GUID.
3. **Um arquivo só.** A biblioteca vai num `.lcf`. Carregar o LCF e a pasta solta juntos gera peça duplicada.
4. **Nunca sobrescrever.** Cada versão final vai para uma pasta nova (R01, R02...). O build se recusa a gravar numa pasta que já existe.

E a documentação: um manual de uso, fichas técnicas por família e um manual visual com isométrica, elevação e planta de cada recurso, todos gerados a partir do próprio código.



Documentação gerada do código: detalhe do batente em planta nos seis arremates da porta de madeira.

13 COMO CHEGAMOS AQUI

- **29/06/2026:** estudo inicial a partir de um objeto de madeira existente e de dois catálogos de alumínio.
- **29 e 30/06:** cerca de 30 versões em sequência, em ciclo curto de correção. A v90 foi entregue cifrada.
- **19/10:** revisão crítica do código, caderno de requisitos e plano por famílias. Decisão de reorganizar, não reescrever.
- **19 e 20/10:** as quatro famílias, revisão após revisão. Versão final R05 no fim do dia 20.
- **21/10:** manuais e organização da pasta.
- **27/10:** visualizador, manual com plantas e elevações, material de divulgação.

O que ainda está aberto: a tabela folha a folha da correr não aparece na interface, a modular em madeira depende de decisões nossas, e o arraste dos hotspots da versão final não foi conferido na tela.



14 QUEM SOMOS: A STAGE

A Stage.AEC foi fundada em São Paulo, em 2023, por José Rocha, César Sallum e Helena Obino. Trabalhamos de forma integrada, da concepção à obra, cuidando do projeto e do investimento: projetamos casas e restaurantes de alto padrão e ficamos até o último dia da obra.

Os sócios participam diretamente da condução do trabalho e da conversa com o cliente:

- **José Rocha:** arquitetura, processos e tecnologia. Mais de 20 anos. Estrutura fluxos, integra BIM e inteligência artificial.
- **César Sallum:** obras, planejamento e custos. Mais de 18 anos. Aproxima contratação, orçamento, coordenação e execução.
- **Helena Obino:** coordenação técnica e BIM. Mais de 13 anos. Liga concepção, documentação, compatibilização e qualidade.

Os três passaram por escritórios como Bernardes, Gui Mattos, Capella Garcia e LZA. Trabalhos da Stage estão entre as 100 Melhores Casas da América Latina do ArchDaily 2025 e na CasaCor 2025 e 2026.

A biblioteca de esquadrias faz parte da nossa frente de pesquisa aplicada em IA e Archicad.

15 SOBRE O AUTOR

José Rocha é arquiteto e cofundador da Stage.AEC, com mais de 20 anos em projetos de arquitetura e construção e longa trajetória em BIM e Archicad:

- **2004:** estágio em Santiago, Chile, no escritório Gonzalo Mardones Viviani & Asociados.
- **2007 e 2008:** Capella Garcia Arquitectura, em Barcelona e Granada, Espanha.
- **2008 a 2013:** coordenador de arquitetura e BIM Manager na Arquitetura Gui Mattos, responsável por templates, padrões gráficos e documentação.
- **2014 e 2015:** consultor BIM e instrutor oficial de Archicad, com implantação em escritórios como Aflalo & Gasperini, Bernardes, Dal Pian, DMDV e MW.
- **2015 a 2017:** implantação do Archicad e da metodologia BIM na Bernardes Arquitetura, com treinamento de equipes em São Paulo e no Rio de Janeiro.
- **2017 a 2022:** coordenador de arquitetura na Arquitetura Gui Mattos.
- **Desde 2023:** cofundador da Stage.AEC, onde aplica inteligência artificial a arquitetura, orçamento, gestão de projetos e tomada de decisão.

Formado em Arquitetura e Urbanismo pela Belas Artes de São Paulo, com master pela Universitat Politècnica de Catalunya, em Barcelona.

16 QUER LEVAR ISSO PARA O SEU ESCRITÓRIO?

Se ficou alguma dúvida sobre o método, ou se você quer entender como implementar esse fluxo no seu escritório, fale com a gente. Conversamos sobre o seu caso: a equipe, os objetos que vocês mais usam e por onde faz sentido começar.

- **E-mail:** stage@stageaec.com.br
- **Site:** stageaec.com.br
- **Instagram:** @stage.aec

DOWNLOAD GRATUITO

Se interessou? Baixe a biblioteca de graça. As quatro famílias deste ebook (correr, modular, porta de madeira e porta camarão) estão prontas para usar no Archicad 28 em diante. Entre em stageaec.com.br/biblioteca, faça o cadastro e baixe.

ANEXO: OS PROMPTS QUE MAIS USAMOS

1. **Requisitos:** "Antes de escrever código, me devolva o caderno de requisitos, a matriz de casos de aceite e as perguntas que você precisa que eu responda."
2. **Fontes:** "Extraia de [catálogo] e marque cada número com página e origem. O que não estiver lá fica 'não consta'."
3. **Revisão:** "Suba a revisão, compile, rode o verificador e me mostre a prévia dos casos de aceite no navegador. Não grave por cima de nenhuma revisão."
4. **Retorno da conferência:** "Carreguei a R[nn]. Na planta: [...]. No 3D: [...]. Segue o print. Corrija só isso."
5. **Revisão crítica:** "Leia o código como se fosse contratar este objeto. O que não é de objeto profissional? Proponha correções sem reescrever o que já foi conferido."
6. **Interface:** "Interface em [n] abas: [campos por aba]. Listas fechadas para [campos]. Trave [campo] quando [condição]."
7. **Hotspots:** "Me devolva o mapa de hotspots por vista antes do código: o que se arrasta na planta e o que se arrasta no 3D."
8. **Entrega:** "Gere a versão final cifrada numa pasta nova, com GUIDs estáveis, monte o LCF e prove que ele é igual à pasta."

stage Architecture Engineering Construction

Biblioteca STG de esquadrias e portas, para Archicad 28 em diante.

Se interessou? Baixe a biblioteca STG de graça, após o cadastro:
stageaec.com.br/biblioteca

STAGEAEC.COM.BR