



EEZ STUDIO E IA

**ACELERANDO O DESENVOLVIMENTO
DE INTERFACES GRÁFICAS PARA
EMBARCADOS**



27 MAI 2026
HORÁRIO: 19h30



LEANDRO QUIBAO

Embedded Technology Designer



Patrocinado por



**MOUSER
ELECTRONICS**

Agenda



1. Abertura e contexto
2. Algumas opções para UI moderno
3. A escolha do EEZ Studio
4. Demonstração do fluxo de trabalho

O desenvolvedor embarcado multifuncional



Corre: Hardware



Nada: Firmware



Voaz: Protótipo para validação

A busca pela experiência do usuário

Elementos gráficos coloridos

Animações fluidas

Feedbacks de audio

Campos de dados dinâmicos

Interfaces multi-linguagem



Os desafios no desenvolvimento

Interface com usuário costuma ser deixada para depois.

Prototipagem rápida em determinada ferramenta pode limitar arquitetura.

Interface desenvolvida manualmente consome tempo.

Desenvolvedor precisa validar fluxo, estados, mensagens, alarmes, menus e experiência do usuário o quanto antes

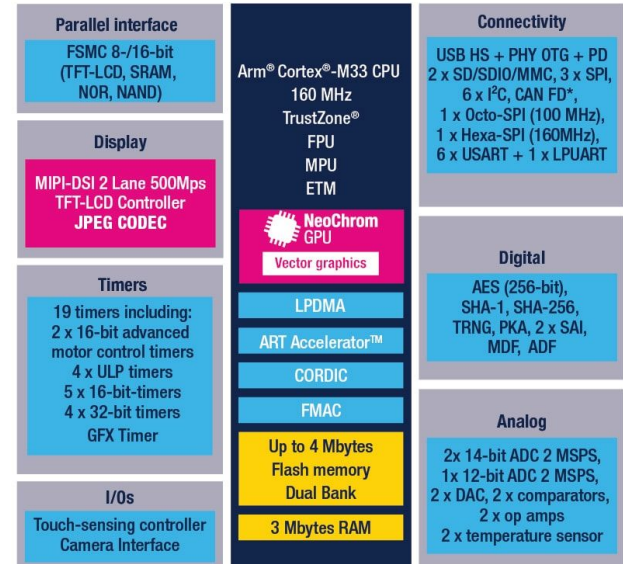
Hardware de um projeto real

Interface com usuário: Display 7", 800x480, RGB 24 bits, touch capacitivo

Microcontrolador: STM32U5G9ZJ

Audio: MAX98360, canal mono

Leds, sensores, saídas PWMs e outros periféricos...



Large embedded memory

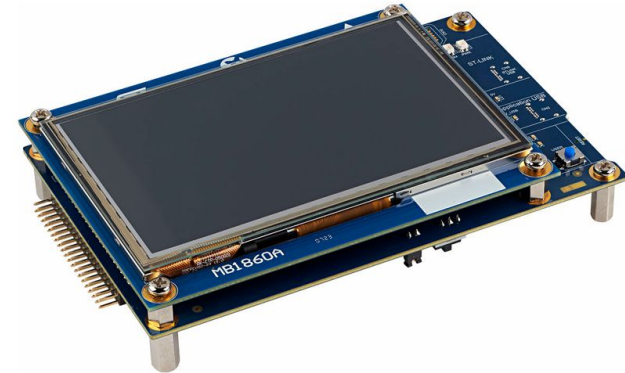
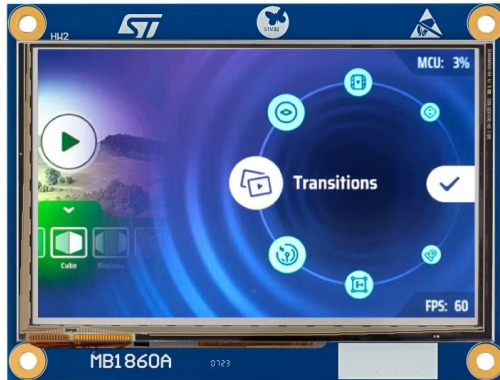
Advanced graphics

*CAN FD only available on STM32U5Gx

Inspirado em um DevKit

STM32U5G9J-DK2

Discovery kit with STM32U5G9ZJ MCU



*STM32U5G9J-DK2 global view.
Picture is not contractual.*

Qual a interface gráfica?

Específica do Fornecedor

(ex.: TouchGFX)



CPU: Variável

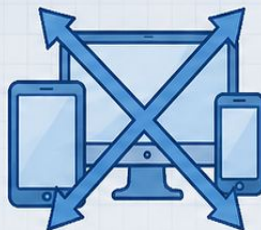
Portabilidade: Média (vinculado a determinado fabricante)

Flexibilidade: Média

Esforço: Médio

Multiplataforma

(ex.: LVGL, Qt)



CPU: Variável

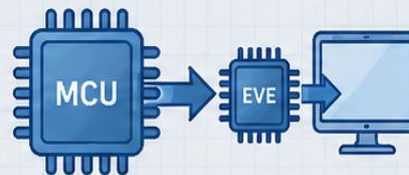
Portabilidade: Alta (independente de HW)

Flexibilidade: Alta

Esforço: Médio

Controlador Gráfico Externo

(ex.: Bridgetek EVE)



CPU: Médio (renderização externa)

Portabilidade: Baixa (vinculado a HW do fabricante)

Flexibilidade: Alta (controle total da interface)

Esforço: Alto (tanto em HW quanto em FW)

Displays Inteligentes

(ex.: DWIN, DACAI, Proculus)



CPU: Baixo (somente comandos)

Portabilidade: Baixa (depende de ferramenta e fabricante)

Flexibilidade: Baixa (elementos da interface fortemente definidos)

Esforço: Baixo

Qual a interface gráfica?

Específica do Fornecedor (ex.: TouchGFX)



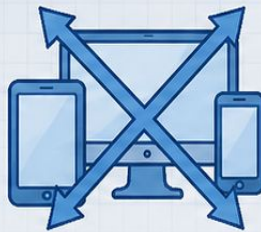
CPU: Variável

Portabilidade: Média (vinculado a determinado fabricante)

Flexibilidade: Média

Esforço: Médio

Multiplataforma (ex.: LVGL, Qt)



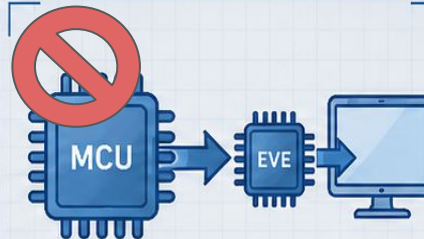
CPU: Variável

Portabilidade: Alta (independente de HW)

Flexibilidade: Alta

Esforço: Médio

Controlador Gráfico Externo (ex.: Bridgetek EVE)



CPU: Médio (renderização externa)

Portabilidade: Baixa (vinculado a HW do fabricante)

Flexibilidade: Alta (controle total da interface)

Esforço: Alto (tanto em HW quanto em FW)

Displays Inteligentes (ex.: DWIN, DACAI, Proculus)



CPU: Baixo (somente comandos)

Portabilidade: Baixa (depende de ferramenta e fabricante)

Flexibilidade: Baixa (elementos da interface fortemente definidos)

Esforço: Baixo

Qual a interface gráfica?

Específica do Fornecedor

(ex.: TouchGFX)



CPU: Variável

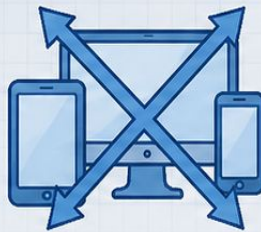
Portabilidade: Média (vinculado a determinado fabricante)

Flexibilidade: Média

Esforço: Médio

Multiplataforma

(ex.: LVGL, Qt)



CPU: Variável

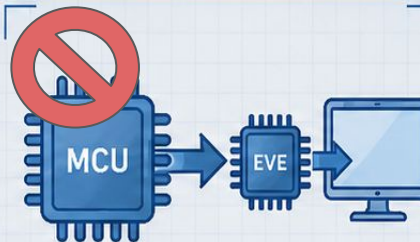
Portabilidade: Alta (independente de HW)

Flexibilidade: Alta

Esforço: Médio

Controlador Gráfico Externo

(ex.: Bridgetek EVE)



CPU: Médio (renderização externa)

Portabilidade: Baixa (vinculado a HW do fabricante)

Flexibilidade: Alta (controle total da interface)

Esforço: Alto (tanto em HW quanto em FW)

Displays Inteligentes

(ex.: DWIN, DACAI, Proculus)



CPU: Baixo (somente comandos)

Portabilidade: Baixa (depende de ferramenta e fabricante)

Flexibilidade: Baixa (elementos da interface fortemente definidos)

Esforço: Baixo

Qual a interface gráfica?

Específica do Fornecedor

(ex.: TouchGFX)



CPU: Variável

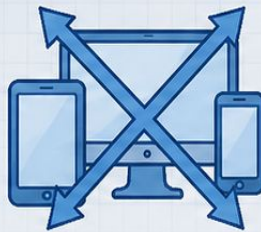
Portabilidade: Média (vinculado a determinado fabricante)

Flexibilidade: Média

Esforço: Médio

Multiplataforma

(ex.: LVGL, Qt)



CPU: Variável

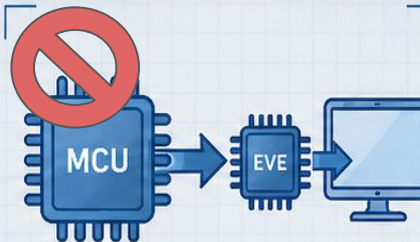
Portabilidade: Alta (independente de HW)

Flexibilidade: Alta

Esforço: Médio

Controlador Gráfico Externo

(ex.: Bridgetek EVE)



CPU: Médio (renderização externa)

Portabilidade: Baixa (vinculado a HW do fabricante)

Flexibilidade: Alta (controle total da interface)

Esforço: Alto (tanto em HW quanto em FW)

Displays Inteligentes

(ex.: DWIN, DACAI, Proculus)



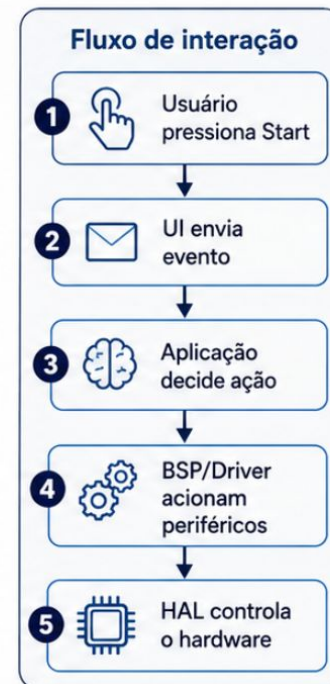
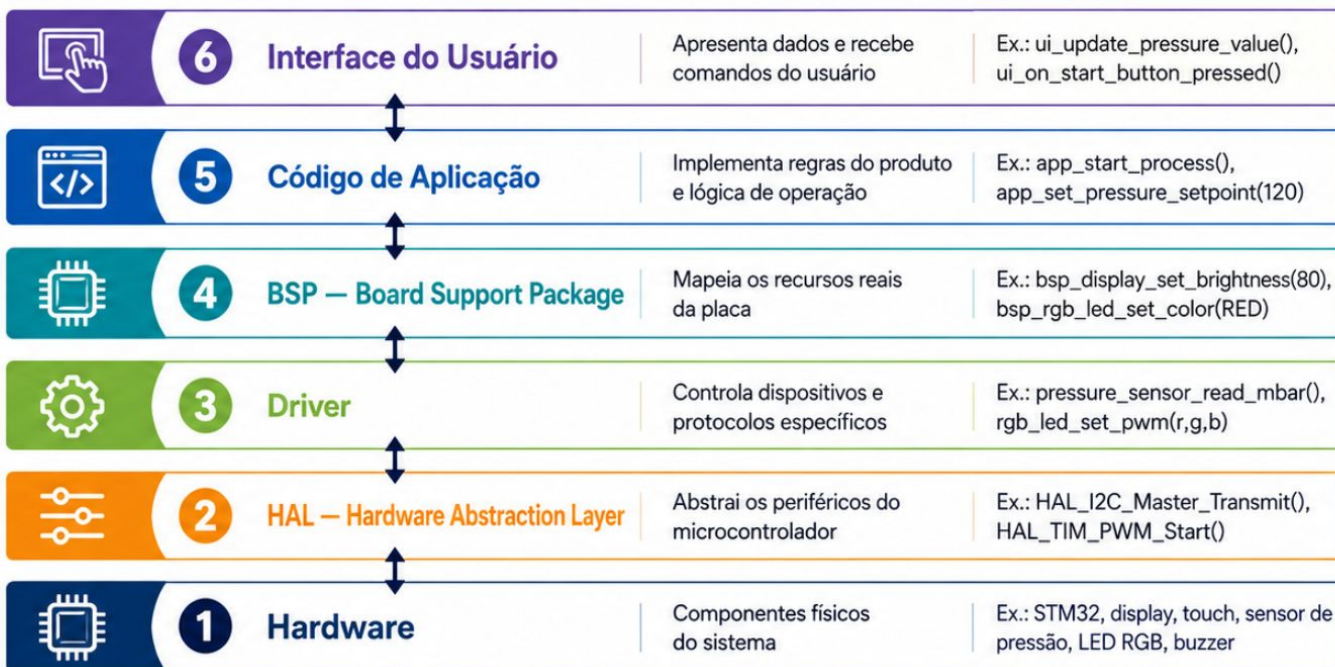
CPU: Baixo (somente comandos)

Portabilidade: Baixa (depende de ferramenta e fabricante)

Flexibilidade: Baixa (elementos da interface fortemente definidos)

Esforço: Baixo

Separando o firmware em camadas



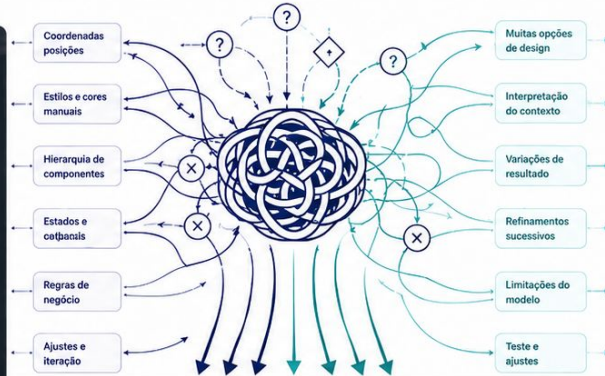
Vamos usar LVGL. E agora?

Criar interfaces visuais não é trivial



Código manual

```
tela_principal.c
1 lv_obj_t *screen = lv_scr_act();
2 lv_obj_t *panel = lv_obj_create(screen);
3 lv_obj_set_size(panel, 800, 480);
4 lv_obj_set_style(panel, bg_color, 0);
5 lv_obj_set_style_bg_color(panel, lv_color_hex(0x151867), 0);
6 lv_obj_set_style_bg_color(panel, lv_color_hex(0x151867), 0);
7
8 // Label título
9 lv_obj_t *label = lv_label_create(panel);
10 lv_label_set_text(label, "Sistema de Controle");
11 lv_obj_set_style_text_color(label, lv_color_hex(0xFF5232), 0);
12 lv_obj_set_style_text_font(label, &lv_font_montserrat_36, 0);
13 lv_obj_align(label, LV_ALIGN_TOP_MID, 0, 10);
14
15 // Botões
16 lv_obj_t *btn1 = lv_btn_create(panel);
17 lv_obj_set_size(btn1, 120, 40);
18 lv_obj_set_pos(btn1, 30, 80);
19 lv_obj_set_style_bg_color(btn1, lv_color_hex(0x23C76A), 0);
20 lv_obj_add_event_cb(btn1, on_btn_click, LV_EVENT_CLICKED, NULL);
21
22
23 // Indicadores e gráficos
24 lv_obj_t *chart = lv_chart_create(panel);
25 lv_chart_set_type(chart, LV_CHART_TYPE_LINE);
26 lv_obj_set_size(chart, 700, 100);
27 lv_obj_set_pos(chart, 50, 300);
28 lv_obj_add_event_cb(chart, on_update, LV_EVENT_ALL, NULL);
29
30 // Callbacks e regras de negócio
31 static void on_botao_click_err(void * e) {
32     // validações, estados, timers...
33 }
```



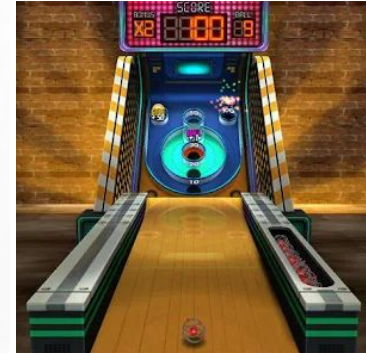
Prompt

Nova conversa

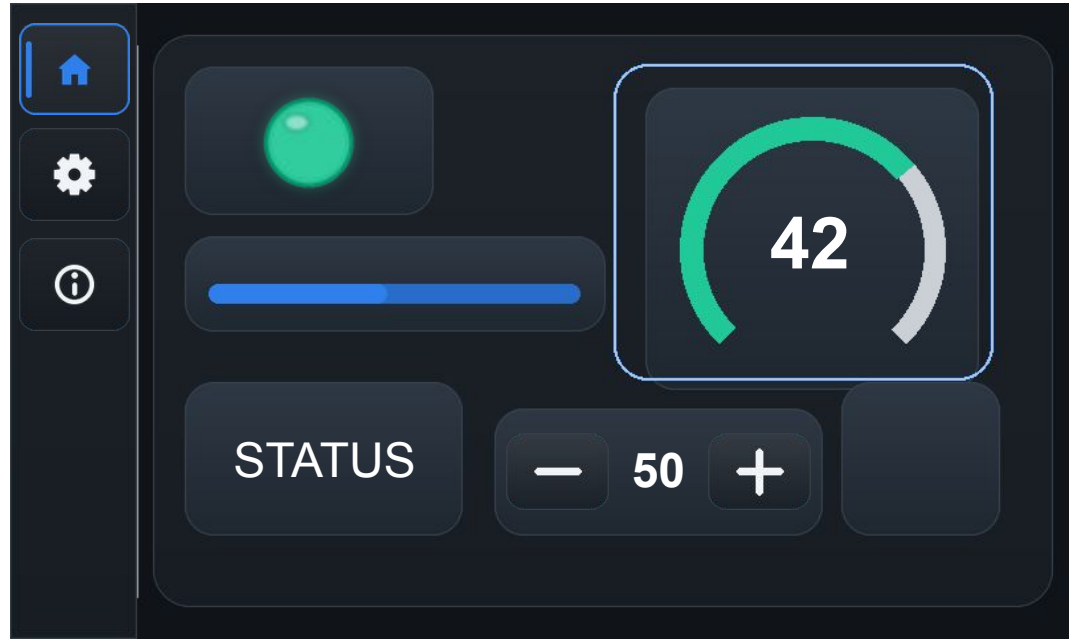
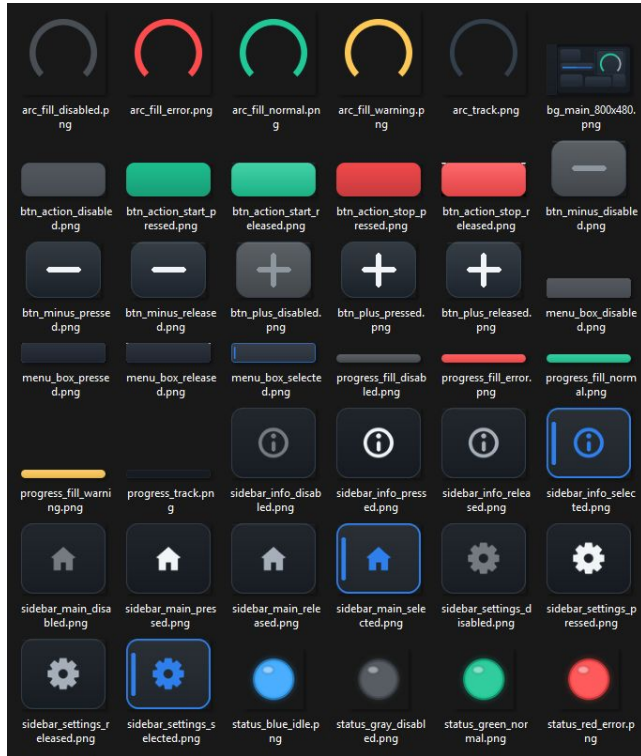
Crie uma interface para um painel de controle industrial:

- Tela inicial com resumo do sistema
- Mostrar pressão, temperatura e vazão
- Botões para Iniciar, Parar e Alarmes
- Gráfico de tendência dos últimos 60 minutos
- Alarmes ativos em destaque
- Usar cores modernas e contraste
- Dever funcionar bem em tela 4:3" touch
- Interface clara e intuitiva

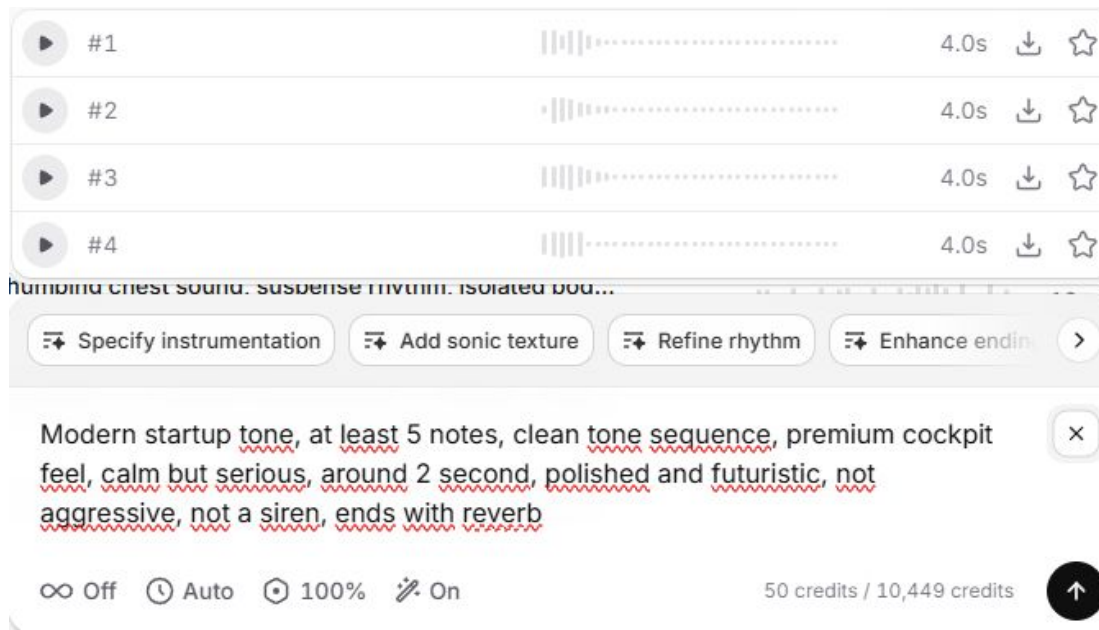
Interface gerada (exemplo)



A “Visão” da IA



E audição também!



#1 4.0s

#2 4.0s

#3 4.0s

#4 4.0s

humbina chest sound. suspense rhythm. isolated bod...

Specify instrumentation Add sonic texture Refine rhythm Enhance ending

Modern startup tone, at least 5 notes, clean tone sequence, premium cockpit feel, calm but serious, around 2 second, polished and futuristic, not aggressive, not a siren, ends with reverb

Off Auto 100% On 50 credits / 10,449 credits







elevenlabs.io/app/sound-effects

What you see, is what you get...

1. EEZ Studio



Gratuito / Código aberto

-  Editor de interface drag-and-drop
-  Lógica visual opcional com EEZ Flow
-  Suporta LVGL 8.x e 9.x
-  Simulador e depurador desktop
-  Bom para GUIs embarcadas e interfaces baseadas em menu
-  Gera código de projeto em C/C++











Ideal para: projetos de GUI embarcada open-source

2. LVGL Pro



Ferramenta oficial do LVGL

-  Fluxo de trabalho orientado a XML
-  Pré-visualização ao vivo e inspetor
-  Integração com Figma
-  Suporte a CLI / CI
-  Suporte a testes de interface
-  Traduções e data bindings
-  Fluxo de trabalho amigável ao Git
-  Voltado para LVGL 9.x











Ideal para: fluxos de trabalho profissionais de interface

3. SquareLine Studio



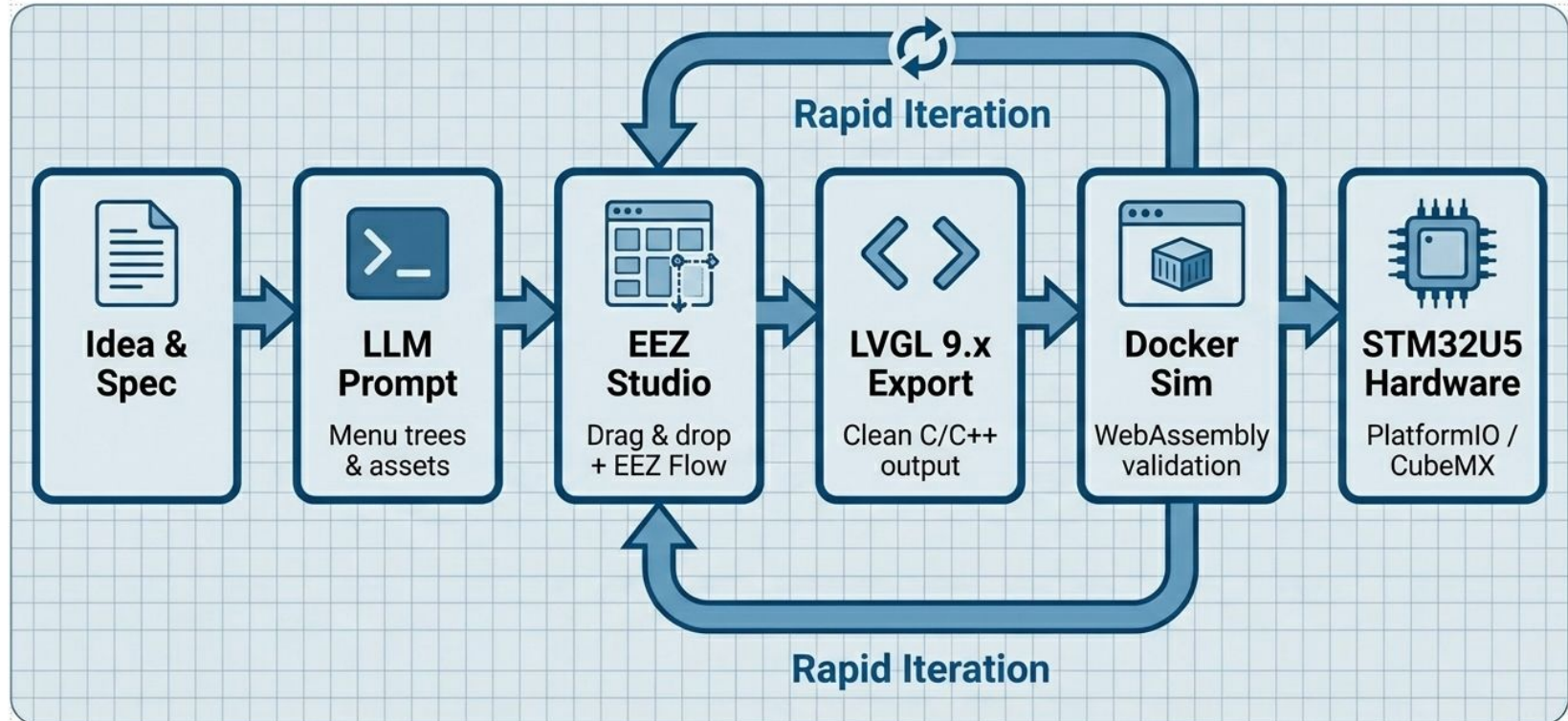
Designer visual maduro

-  Editor drag-and-drop
-  Biblioteca rica de componentes
-  Estilos e temas
-  Eventos e animações
-  Gerenciamento de ativos
-  Suporte a múltiplos idiomas
-  Exporta para C e Python / MicroPython
-  Templates de placas

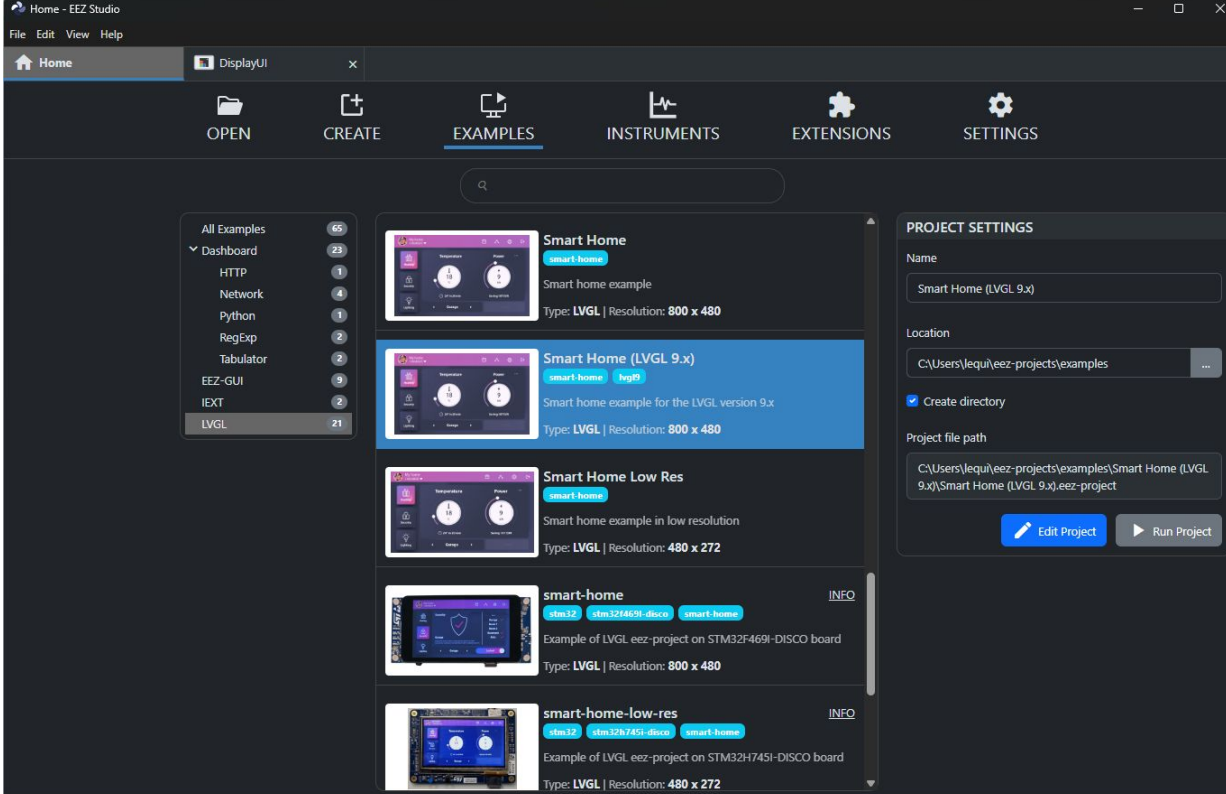


Ideal para: criação rápida de GUI visual

Comparação focada em destaques práticos de recursos



Tela inicial do EEZ Studio



The screenshot shows the EEZ Studio Home interface. At the top, there is a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it is a toolbar with icons for 'OPEN', 'CREATE', 'EXAMPLES', 'INSTRUMENTS', 'EXTENSIONS', and 'SETTINGS'. The 'EXAMPLES' tab is selected, displaying a list of project examples. On the left, a sidebar shows a tree view of examples, with 'LVGL' selected. The main area lists five examples, each with a preview image and details. On the right, a 'PROJECT SETTINGS' panel is visible, showing fields for Name, Location, and Project file path, along with 'Edit Project' and 'Run Project' buttons.

Home - EEZ Studio
File Edit View Help

Home DisplayUI

OPEN CREATE EXAMPLES INSTRUMENTS EXTENSIONS SETTINGS

All Examples 65
Dashboard 23
HTTP 1
Network 4
Python 1
RegExp 2
Tabulator 2
EEZ-GUI 9
IEXT 2
LVGL 21

Smart Home
Smart home example
Type: LVGL | Resolution: 800 x 480

Smart Home (LVGL 9.x)
Smart home example for the LVGL version 9.x
Type: LVGL | Resolution: 800 x 480

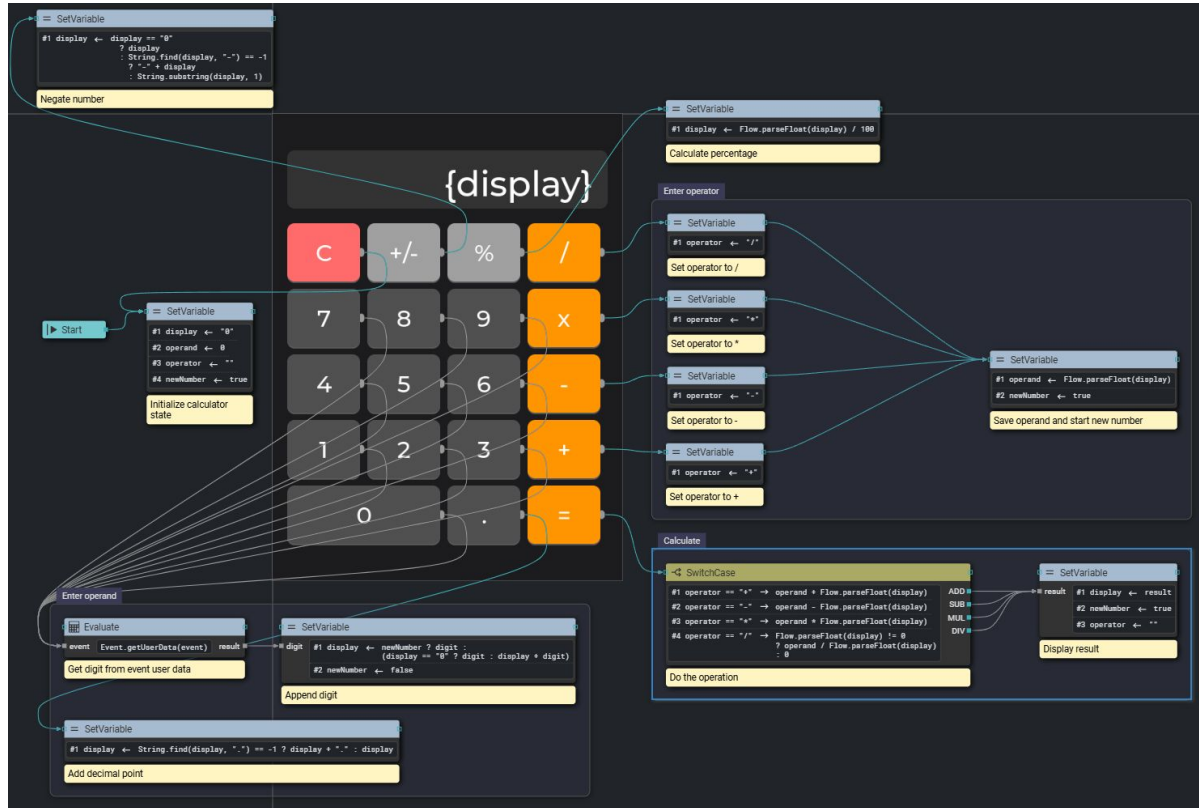
Smart Home Low Res
Smart home example in low resolution
Type: LVGL | Resolution: 480 x 272

smart-home INFO
stm32 stm32f469i-discv smart-home
Example of LVGL eez-project on STM32F469I-DISCO board
Type: LVGL | Resolution: 800 x 480

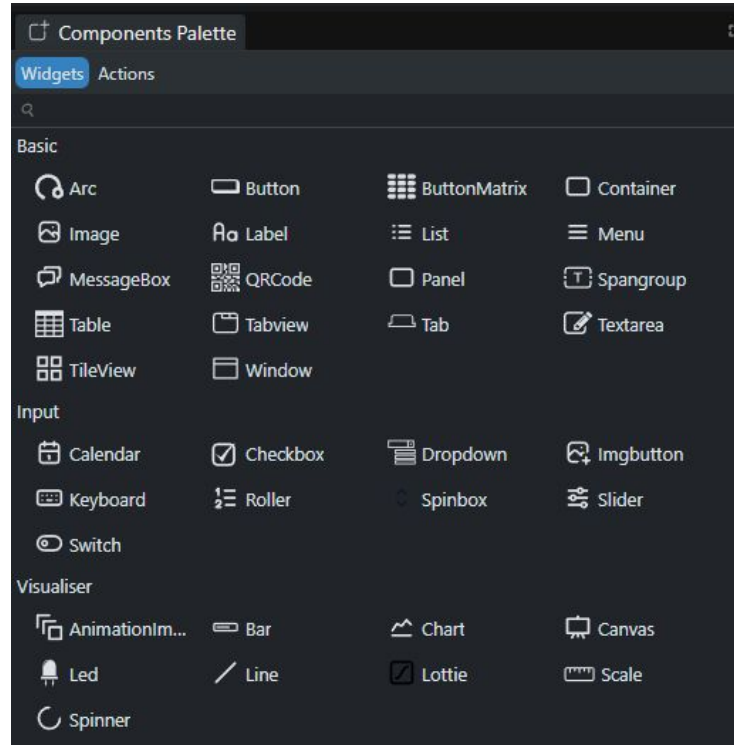
smart-home-low-res INFO
stm32 stm32f469i-discv smart-home
Example of LVGL eez-project on STM32F469I-DISCO board
Type: LVGL | Resolution: 480 x 272

PROJECT SETTINGS
Name
Smart Home (LVGL 9.x)
Location
C:\Users\lequ\eez-projects\examples
Create directory
Project file path
C:\Users\lequ\eez-projects\examples\Smart Home (LVGL 9.x)\Smart Home (LVGL 9.x).eez-project
Edit Project Run Project

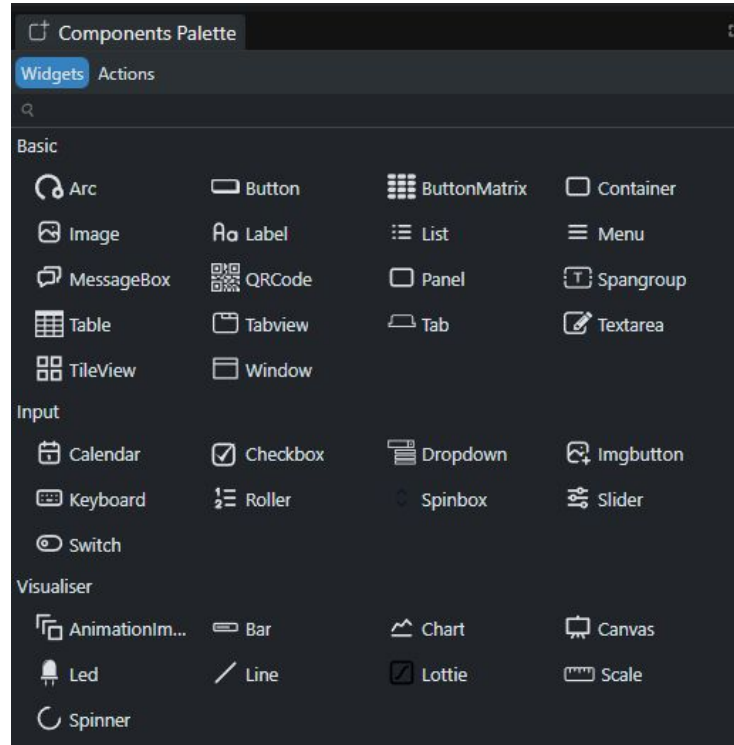
Um fluxo com EEZ Flow



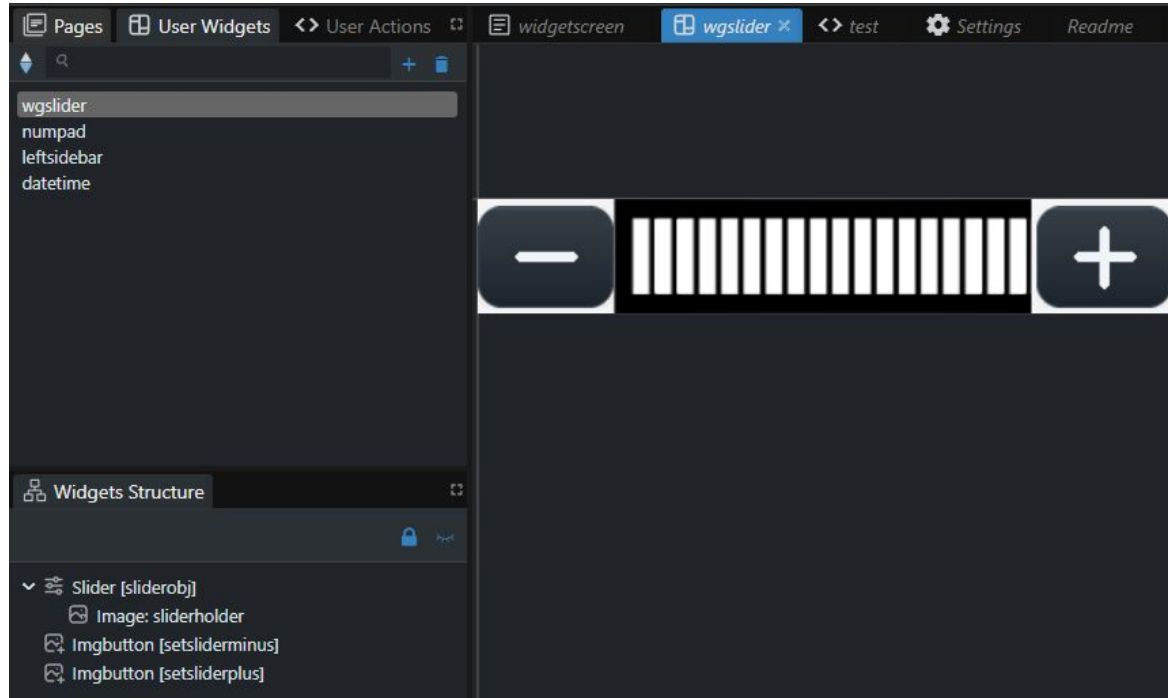
A Paleta de componentes padrão LVGL



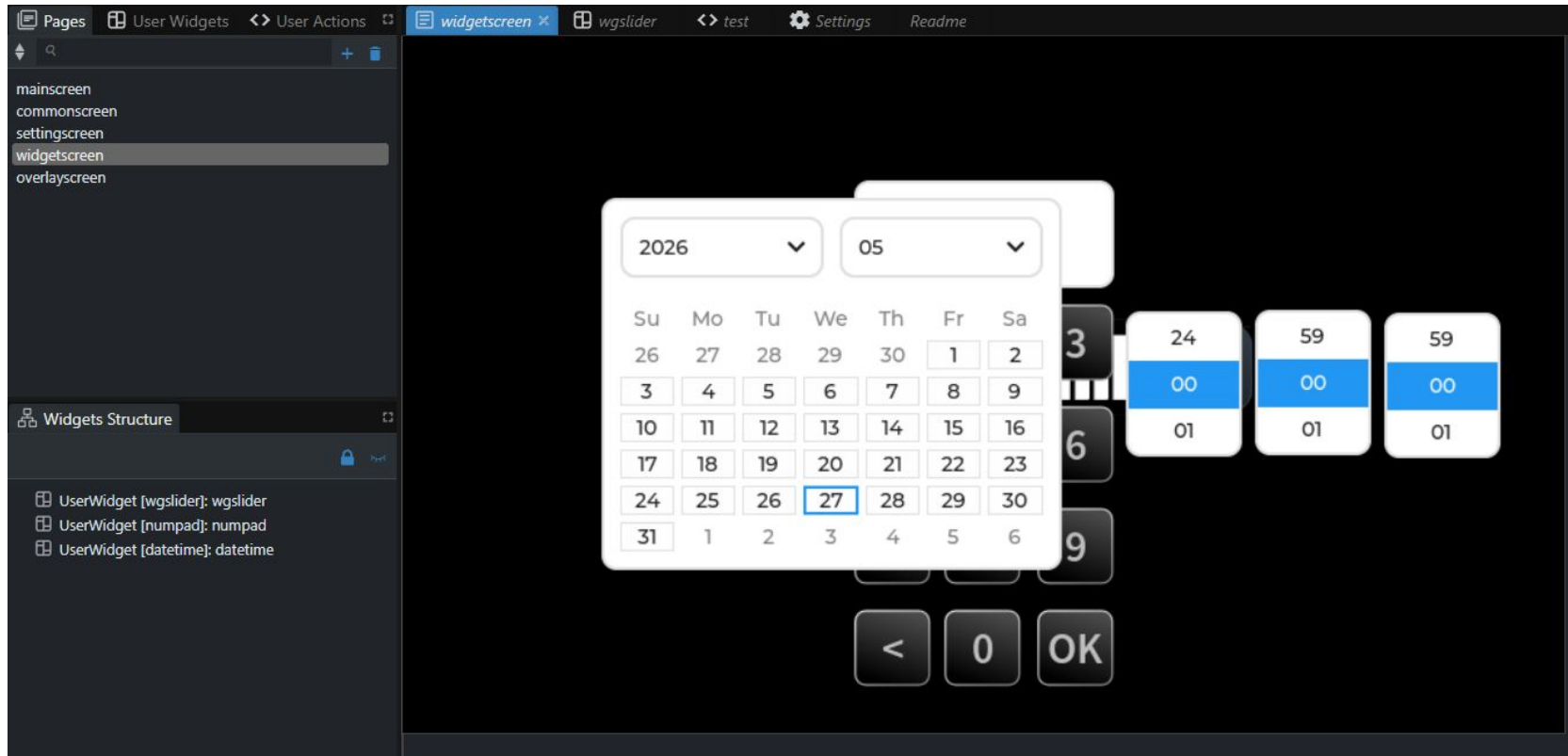
A Paleta de componentes padrão LVGL



Combinando elementos em user widgets

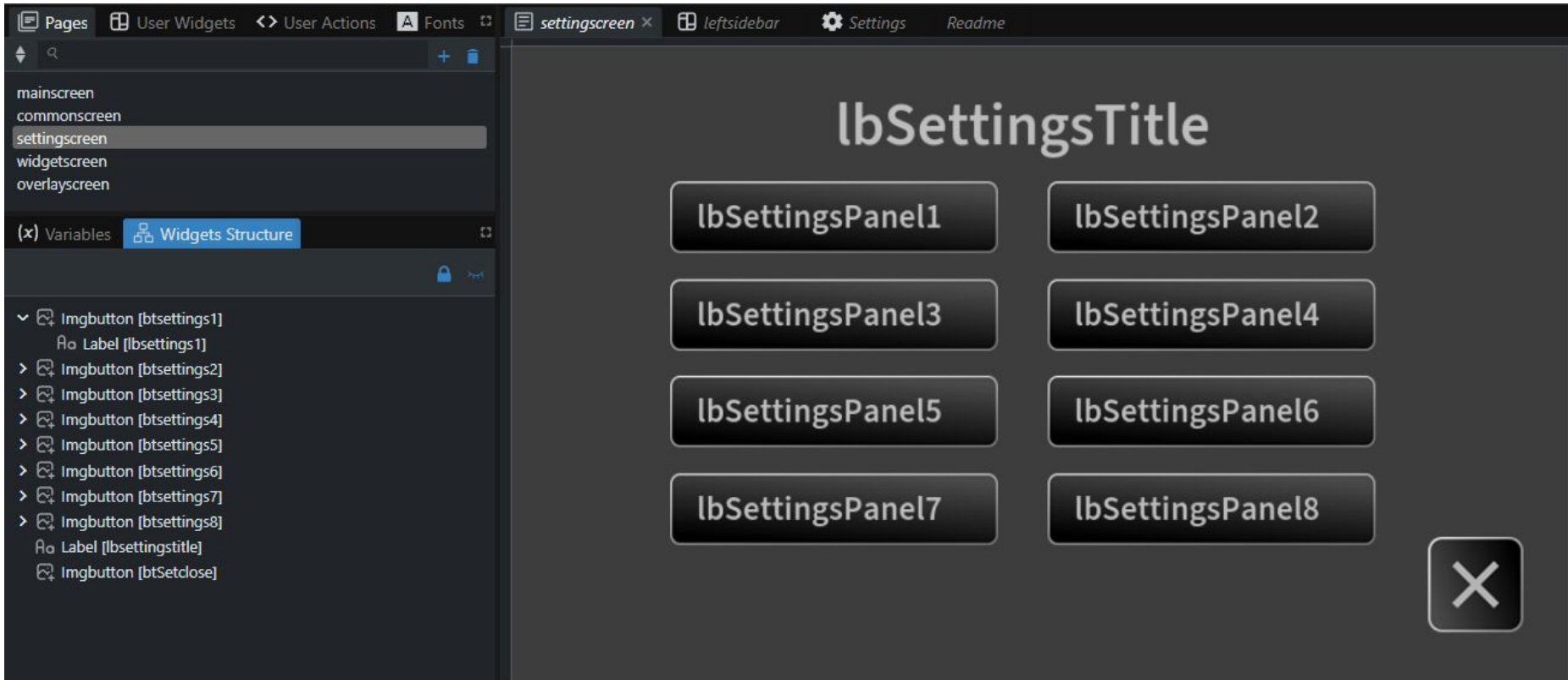


Para conjuntos facilmente gerenciáveis e reaproveitáveis



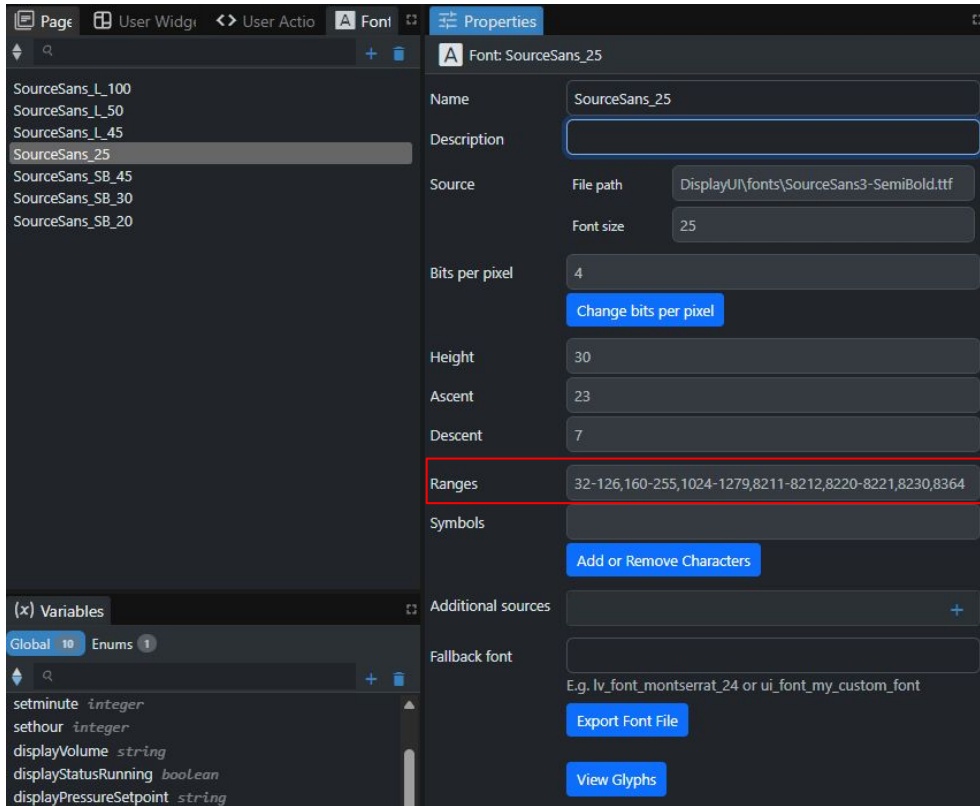
The screenshot displays a development environment with a dark theme. On the left, a sidebar shows a 'Pages' list with 'widgetscreen' selected, and a 'Widgets Structure' panel listing three widgets: 'UserWidget [wgslider]: wgslider', 'UserWidget [numpad]: numpad', and 'UserWidget [datetime]: datetime'. The main area shows a 'widgetscreen' with a date picker widget. The date picker has two dropdown menus for year (2026) and month (05), a calendar grid with the 27th highlighted, and a numeric keypad with three columns of digits (24, 59, 59) and two rows of '00' and '01' options. Below the keypad are navigation buttons for back, 0, and OK.

Reaproveitamento em estruturas de menu comum



The screenshot displays a software development environment. On the left, a sidebar shows a project structure with pages like 'mainscreen', 'commonscreen', 'settingscreen', 'widgetscreen', and 'overlayscreen'. Below this, a 'Widgets Structure' tree lists components such as 'Imgbutton [btsettings1]', 'Label [lbsettings1]', and 'Label [lbsettingstitle]'. The main area shows a visual design of a settings screen with a title 'lbSettingsTitle' and eight buttons labeled 'lbSettingsPanel1' through 'lbSettingsPanel8'. A close button (X) is located in the bottom right corner of the design area.

Adicione fontes customizadas no projeto



1) ASCII 32–126

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|----|
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| ! | " | # | \$ | % | & | ' | (|) | * | + | , | - | / | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | | | |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 92 | 93 | 94 | 95 | | |
| @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | | |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 117 | 118 | 119 | 120 | 122 | 123 | 124 | 125 | 126 | | | | |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | { | } | | | | |

2) Latin-1 Supplement 160–255

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| À | Á | Â | Ã | Ä | Å | Æ | Ç | È | É | Ê | Ë | Ì | Í | Î | Ï | Ð | Ñ | Ò | Ó | Ô | Õ | × | Ø | Ù | Ú | Û | Ü | Ý | ÿ | | |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 206 | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | | |
| À | Á | Â | Ã | Ä | Å | Æ | Ç | È | É | Ê | Ë | Ì | Í | Î | Ï | Ð | Ñ | Ò | Ó | Ô | Õ | × | Ø | Ù | Ú | Û | Ü | Ý | ÿ | | |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 234 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 249 | 248 | 249 | 251 | 252 | 253 | 254 | 255 | | |
| à | á | â | ã | ä | å | æ | ç | è | é | ê | ë | ì | í | î | ï | ð | ñ | ò | ó | ô | õ | ÷ | ø | ù | ú | û | ü | ý | ÿ | | |

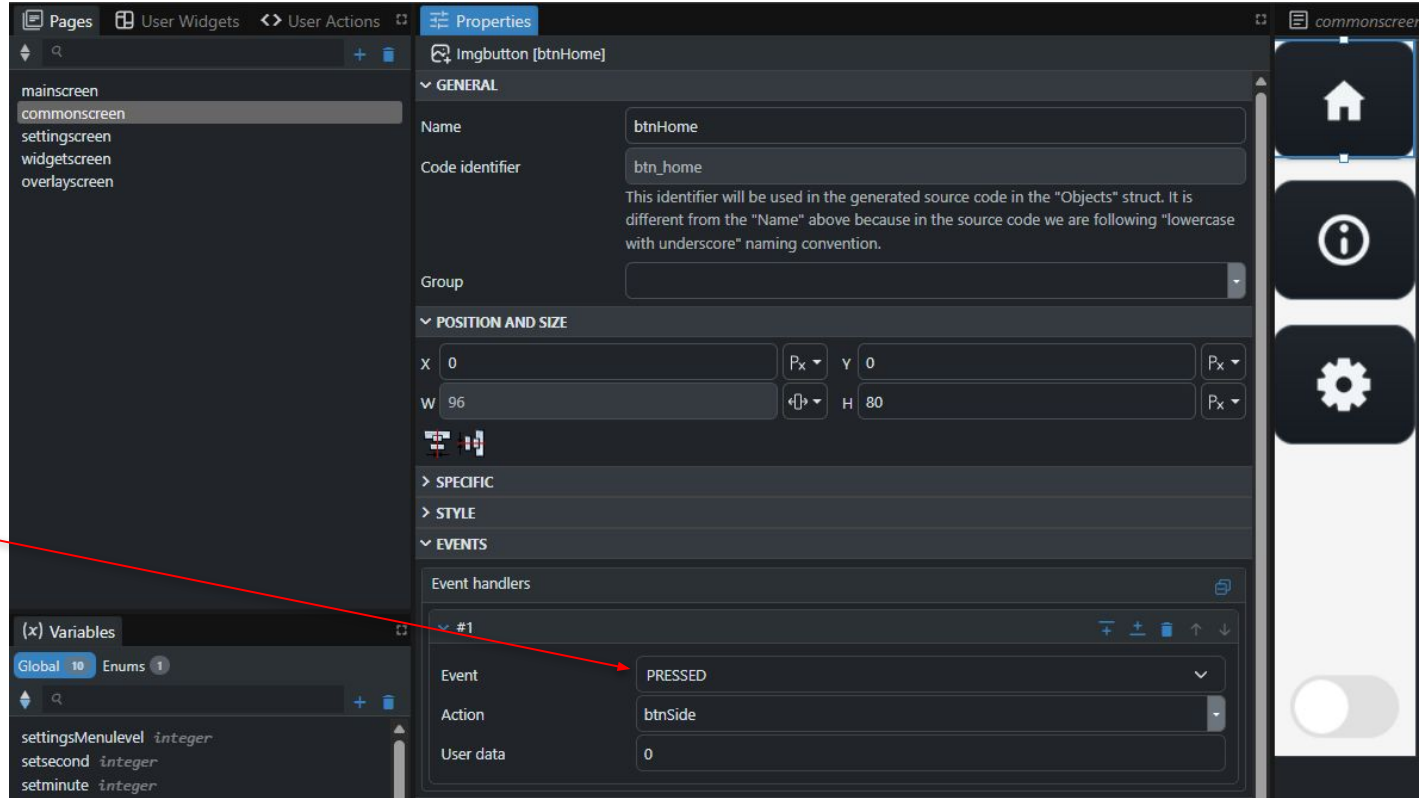
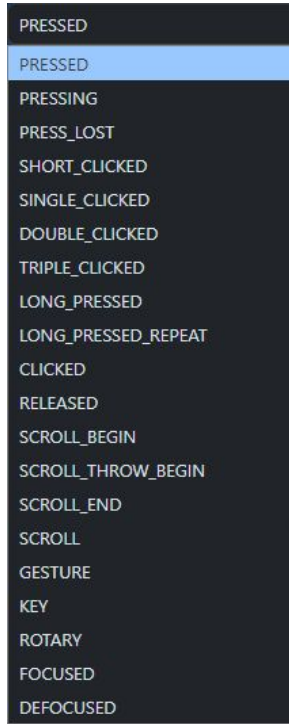
3) Cyrillic 1024–1279

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1024 | 1025 | 1026 | 1027 | 1028 | 1029 | 1030 | 1031 | 1032 | 1033 | 1034 | 1036 | 1037 | 1038 | 1039 | 1040 | 1041 | 1042 | 1043 | 1044 | 1045 | 1046 | 1047 | 1048 | 1049 | 1050 | 1051 | 1052 | 1053 | 1054 | 1055 |
| А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ш | Щ | Ъ | Ы | Ь | Ю | Я | |
| 1056 | 1057 | 1058 | 1059 | 1060 | 1061 | 1062 | 1063 | 1064 | 1066 | 1067 | 1068 | 1069 | 1070 | 1071 | 1072 | 1073 | 1074 | 1075 | 1076 | 1077 | 1078 | 1079 | 1080 | 1081 | 1082 | 1083 | 1084 | 1085 | 1086 | 1087 |
| а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ш | щ | ъ | ы | ь | ю | я | |
| 1088 | 1089 | 1090 | 1091 | 1092 | 1093 | 1094 | 1095 | 1096 | 1097 | 1098 | 1099 | 1100 | 1102 | 1103 | 1104 | 1106 | 1108 | 1109 | 1110 | 1111 | 1112 | 1114 | 1115 | 1116 | 1117 | 1118 | 1119 | | | |
| Ё | ё | Є | є | ѕ | і | ї | ї | ј | ј | љ | њ | њ | ћ | ќ | ќ | љ | љ | џ | џ | Ѡ | ѡ | Ѣ | ѣ | Ѥ | ѥ | Ѧ | ѧ | Ѩ | ѩ | |
| 1120 | 1121 | 1122 | 1123 | 1124 | 1125 | 1126 | 1127 | 1128 | 1129 | 1130 | 1131 | 1132 | 1133 | 1134 | 1135 | 1136 | 1139 | 1140 | 1141 | 1142 | 1143 | 1144 | 1145 | 1146 | 1147 | 1148 | 1149 | 1150 | 1151 | |
| Ѱ | ѱ | Ѳ | ѳ | Ѵ | ѵ | Ѷ | ѷ | Ѹ | ѹ | Ѻ | ѻ | Ѽ | ѽ | Ѿ | ѿ | Ѡ | ѡ | Ѣ | ѣ | Ѥ | ѥ | Ѧ | ѧ | Ѩ | ѩ | Ѫ | ѫ | Ѭ | ѭ | |
| 1152 | 1153 | 1154 | 1155 | 1156 | 1157 | 1158 | 1159 | 1160 | 1161 | 1162 | 1163 | 1164 | 1166 | 1167 | 1168 | 1169 | 1170 | 1171 | 1172 | 1173 | 1174 | 1175 | 1176 | 1177 | 1178 | 1179 | 1180 | 1181 | 1182 | |
| Ѱ | ѱ | Ѳ | ѳ | Ѵ | ѵ | Ѷ | ѷ | Ѹ | ѹ | Ѻ | ѻ | Ѽ | ѽ | Ѿ | ѿ | Ѡ | ѡ | Ѣ | ѣ | Ѥ | ѥ | Ѧ | ѧ | Ѩ | ѩ | Ѫ | ѫ | Ѭ | ѭ | |
| 1184 | 1185 | 1186 | 1187 | 1188 | 1189 | 1190 | 1191 | 1192 | 1193 | 1194 | 1196 | 1197 | 1198 | 1199 | 1200 | 1201 | 1203 | 1204 | 1205 | 1206 | 1207 | 1208 | 1209 | 1210 | 1211 | 1212 | 1213 | 1214 | 1215 | |
| Ѱ | ѱ | Ѳ | ѳ | Ѵ | ѵ | Ѷ | ѷ | Ѹ | ѹ | Ѻ | ѻ | Ѽ | ѽ | Ѿ | ѿ | Ѡ | ѡ | Ѣ | ѣ | Ѥ | ѥ | Ѧ | ѧ | Ѩ | ѩ | Ѫ | ѫ | Ѭ | ѭ | |
| 1216 | 1217 | 1218 | 1219 | 1220 | 1221 | 1222 | 1223 | 1224 | 1226 | 1227 | 1228 | 1229 | 1230 | 1231 | 1232 | 1233 | 1234 | 1235 | 1237 | 1238 | 1239 | 1240 | 1241 | 1242 | 1243 | 1244 | 1245 | 1246 | 1247 | |
| Ѱ | ѱ | Ѳ | ѳ | Ѵ | ѵ | Ѷ | ѷ | Ѹ | ѹ | Ѻ | ѻ | Ѽ | ѽ | Ѿ | ѿ | Ѡ | ѡ | Ѣ | ѣ | Ѥ | ѥ | Ѧ | ѧ | Ѩ | ѩ | Ѫ | ѫ | Ѭ | ѭ | |
| 1248 | 1249 | 1250 | 1251 | 1252 | 1253 | 1254 | 1255 | 1256 | 1257 | 1258 | 1259 | 1261 | 1262 | 1263 | 1264 | 1265 | 1266 | 1267 | 1268 | 1269 | 1270 | 1271 | 1273 | 1274 | 1275 | 1276 | 1277 | 1278 | 1279 | |
| Ѱ | ѱ | Ѳ | ѳ | Ѵ | ѵ | Ѷ | ѷ | Ѹ | ѹ | Ѻ | ѻ | Ѽ | ѽ | Ѿ | ѿ | Ѡ | ѡ | Ѣ | ѣ | Ѥ | ѥ | Ѧ | ѧ | Ѩ | ѩ | Ѫ | ѫ | Ѭ | ѭ | |

4) Extra punctuation and symbols: 8211, 8212, 8220, 8221, 8230, 8364

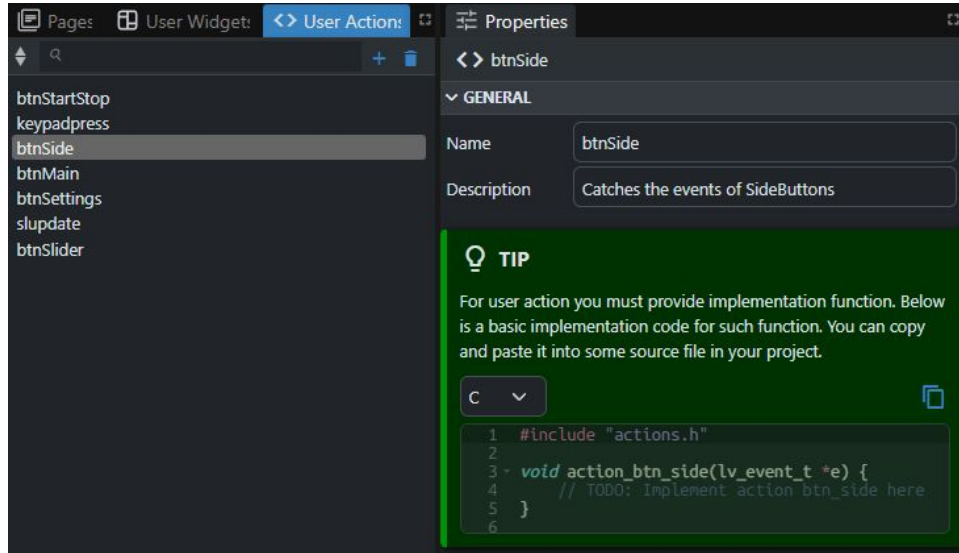
| 8211 | 8212 | 8220 | 8221 | 8230 | 8364 |
|---------|---------|----------------------------|-----------------------------|---------------------|-----------|
| — | — | “ | ” | … | € |
| EN DASH | EM DASH | LEFT DOUBLE QUOTATION MARK | RIGHT DOUBLE QUOTATION MARK | HORIZONTAL ELLIPSIS | EURO SIGN |

Eventos em um objeto, chamam determinada ação



E mais...

Toda ação, é implementada no seu código



The screenshot shows a software development interface with a 'User Action' configuration window for 'btnSide'. The 'GENERAL' tab is active, showing the name 'btnSide' and a description 'Catches the events of SideButtons'. A 'TIP' box provides instructions on implementing a user action function. Below the tip, a code editor shows a basic implementation for the 'action_btn_side' function.

```
1 #include "actions.h"
2
3 void action_btn_side(lv_event_t *e) {
4     // TODO: Implement action btn_side here
5 }
6
```

```
void action_btn_side(lv_event_t* e) {
    /* Sidebar pass their target via user_data:
    0 = btnHome, 1 = btnParameter, 2 = btnSettings. */
    uintptr_t which = (uintptr_t)lv_event_get_user_data(e);
    switch (which) {
        case 0: loadScreen(SCREEN_ID_MAINSCREEN);
            break;
        case 1: loadScreen(SCREEN_ID_SETTINGSSCREEN);
            break;
        case 2: loadScreen(SCREEN_ID_SETTINGSSCREEN);
            break;
        default:
            break;
    }
}
```

Simulando no EEZ studio

Um Hardware real

FECHAMENTO

1. FERRAMENTAS QUE ACELERAM

| EEZ Studio | LVGL | IA |
|--|--|--|
| | | |
| <ul style="list-style-type: none"> ✓ Design de UI ✓ Navegação ✓ Geração de código | <ul style="list-style-type: none"> ✓ Widgets prontos ✓ Temas e estilos ✓ Leve e otimizado | <ul style="list-style-type: none"> ✓ Gera interfaces ✓ Sugere código ✓ Automatiza tarefas |

2. SIMULAÇÃO E PROTOTIPAÇÃO

Compilação sem erros

🕒 **Teste, visualize e valide antes do hardware. Menos retrabalho, mais aprendizado.**

3. HARDWARE E VALIDAÇÃO

Hardware real

- ✓ Testes funcionais e de integração

Validação de produto

- ✓ Produto pronto, validado e confiável

Código gerado precisa ser revisado

```
if (temp > 30) {
    alert_high_temp();
}
// Revisar lógica,
// limites e tratamento
// de erros
```

✓

Arquitetura precisa ser controlada

```

graph TD
    UI[UI LVGL] --- Aplicação
    UI --- Serviços
    UI --- Comunicação
    Aplicação --- Drivers[Drivers / HAL / BSP]
    
```

✓

Integração com o firmware exige critérios técnicos

- Interface bem definida
- Sincronização de dados
- Gerenciamento de recursos
- Testes e cobertura
- Performance e segurança

✓

OBRIGADO!



Patrocinado por



www.embarcados.com.br



linkedin.com/embarcados



[@portalembarcados](https://instagram.com/portalembarcados)



[youtube/Embarcados TV](https://youtube.com/EmbarcadosTV)