

# Painel de assinatura (web)

## Função

O componente **Painel de assinatura** permite a realização de uma assinatura digital feita através do toque (touchscreen, mesa digitalizadora ou mouse) e obter o conteúdo assinado através de uma imagem.

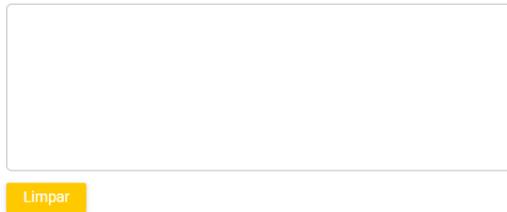


Figura 1 - Componente Painel de assinatura rodando na web.

## Principais propriedades

Na tabela abaixo estão descritas as principais propriedades do componente.

Nome	Propriedade	Função
Valor	ng-model	Propriedade Angular que pode ser usada para obter e alimentar o conteúdo pelo bloco de programação.
Identificador	id	Atributo que especifica um ID exclusivo para o componente. Por padrão, um ID é gerado automaticamente e atribuído ao componente.
Estilo	style	Altera o estilo geral do componente da forma <i>inline</i> .
Exibir	ng-show	Propriedade Angular usada para mostrar ou oculta o componente.
Repetir	crn-repeat	Propriedade que seleciona uma fonte de dados contida na tela e repete o componente baseado no número de itens dessa fonte de dados.
Segurança	cronapp-security	Propriedade do Cronapp que trata as permissões de segurança da aplicação. Usado para permitir ou não edição e visualização do componente com base nos perfis de selecionados.
Skin	crn-skin	Permite selecionar um skin que afetará apenas o componente selecionado.
Classes CSS	class	Adiciona classes CSS que já foram criadas.

## Exemplo de uso

O componente Painel de assinatura só recebe a assinatura do usuário e limpa o campo, assim, é preciso associa-lo a uma fonte de dados ou obter a imagem através de um bloco de programação. No exemplo a seguir, iremos capturar a assinatura do usuário através de um bloco de programação, chamado por um **botão**, e a assinatura será renderizada no **componente imagem**. Como a imagem é convertida em base64, é necessário concatenar o Data URI scheme com a base64, como mostra a figura 2.

### Dica

Como o componente transforma a assinatura em imagem e depois converte para base64, é possível obter seu valor para tratar de diversas formas: salvar em banco, vincular a um campo da fonte de dados, fazer uma lógica via bloco de programação e outros.

## Criando o bloco de programação

### Nome em inglês

Signature Pad

### Nesta página

- [Função](#)
- [Principais propriedades](#)
- [Exemplo de uso](#)
  - [Criando o bloco de programação](#)
  - [Exemplo de uso](#)

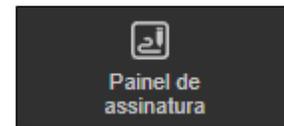
### Compatibilidade

- Formulário web

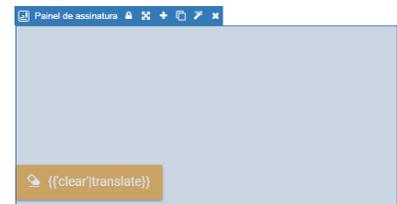
### Equivalente mobile

[Painel de assinatura \(mobile\)](#)

### Botão do Componente



### Imagem no Editor Visual



Para realizar a captura da assinatura, inicialmente precisamos criar o [Bloco de programação](#) que irá realizar tal função, confira na imagem abaixo. Utilizaremos o bloco de programação [Definir valor do atributo](#) para obter o valor do campo assinatura. No parâmetro **Valor do atributo** estamos obtendo o valor do input (assinatura) e concatenando com o Data URI Scheme, desse jeito podemos renderizar a imagem no componente imagem. **OBS:** é necessário vincular o bloco de programação a um formulário de referência, confira no tópico [Propriedades do Blockly](#) da documentação [Bloco de programação](#).

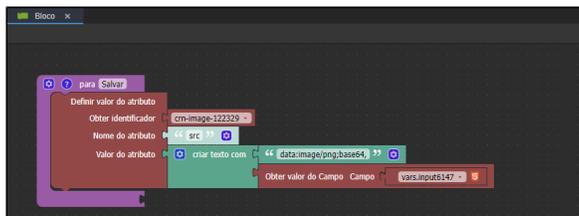


Figura 2 - Bloco de programação responsável para salvar a imagem da assinatura.

#### Data URI scheme usado no exemplo

```
data:image/png;base64,
```

Agora precisamos incluir o [componente imagem](#) e [botão](#) na view junto com componente [Painel de assinatura](#), confira na figura 2.1.

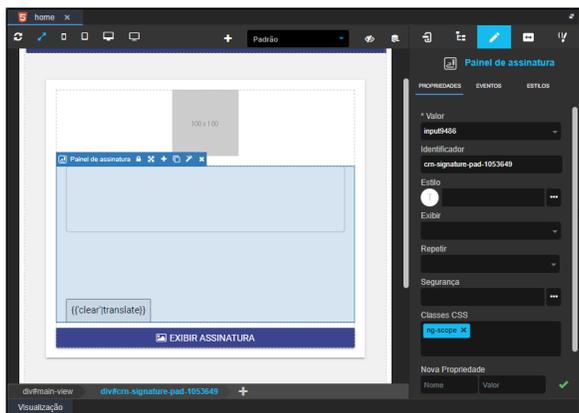


Figura 2.1 - Adicionando o componente botão e imagem.

Por fim, basta associar o evento **Ao Clicar** do botão ao bloco de programação criado na figura 2. Em seguida execute a aplicação, o resultado obtido pode ser visto na figura 2.2.



**Figura 2.2 - Componente Painel de assinatura rodando na web**

1. Componente imagem exibindo a assinatura capturada pelo botão;
2. Área destinada à assinatura;
3. Botão para limpar o campo da assinatura;
4. Botão responsável por capturar a assinatura.