

MercadoPagoLocal – Delphi (Android)

Componente Delphi para integração com máquinas de cartão Mercado Pago usando integração intent-based no Android.

Instalação:

Abra o arquivo “**MercadoPagoDelphi.groupproj**”, e no Delphi, no painel de projetos clique com o botão direito sobre “**MercadoPago4Delphi.bpl**” e clique em “**Install**”, feito isso o componente “**TMercadoPagoLocal**” aparecerá na aba “**Mercado Pago**” na paleta de componentes do Delphi.

Copie a unit “**MercadoPagoDelphi.pas**” para o diretório “C:\Program Files (x86)\Embarcadero\Studio\20.0\lib\android\release” **OU** inclua a pasta com a unit no “**Library Path**” do seu Delphi.

Ou se preferir copie a unit “**MercadoPagoDelphi.pas**” para dentro da pasta do seu projeto.

Funcionamento:

O componente utiliza comunicação através de intent com o App Oficial do Mercado Pago, sendo necessário que o app esteja instalado e logado com as credenciais da conta que receberá os pagamentos.

Link da documentação baseada em intent-based:

https://github.com/mercadopago/point-android_integration

OBS: Caso o app não esteja instalado no dispositivo, o componente consegue detectar e solicitar a instalação do app automaticamente.

Sobre a necessidade da conta estar logada no dispositivo:

O componente utiliza a conta Mercado Pago já configurada no app para realizar as vendas e em alguns casos, por motivos de segurança, essa conta não pode ser distribuída em mãos de terceiros. Nesse caso pode ser usado o recurso de “**Colaboradores**” do Mercado Pago, onde são criadas sub-contas com permissões configuráveis de acordo com a necessidade e que não possuem acesso a transações e saldo da conta principal, mas que podem efetuar cobranças em nome da conta principal.

Saiba mais em: https://www.mercadopago.com.br/ajuda/preciso-autorizar-pessoas-a-usar-o-leitor-point_2659 e também https://www.mercadopago.com.br/ajuda/colaboradores_4026

Propriedades:

MensagemInstalarMercadoPago : Defina aqui a mensagem a ser mostrada quando o app MercadoPago não for encontrado, o componente ira mostrar essa mensagem solicitando se deseja instalar o app através da Play Store.

Por padrão a mensagem é: “Para efetuar a venda usando sua conta do Mercado Pago, é necessário que o aplicativo oficial do Mercado Pago esteja instalado e logado com as suas credenciais.

Deseja instalar o App do Mercado Pago agora?”

Uso do componente:

O componente possui a procedure **“TMercadoPagoLocal.ProcessarPagamento”** para iniciar o processo de cobrança de acordo com as preferencias (parâmetros opcionais podem ser ignorados):

```
MercadoPagoLocal1.ProcessarPagamento(  
Valor: double;  
CartaoCredito: boolean;  
Parcelas: integer;  
Descricao: string;  
ReferenciaExterna: string = '';  
EmailPagador: string = '';  
);
```

Valor: Valor da compra/transação, exemplo 29.90;

CartaoCredito: indica se a venda será efetuada no cartão de crédito(True) ou débito(False);

Parcelas: número de parcelas da compra (Apenas cartão de crédito);

Descrição: Descrição da venda, sem espaços e máximo de 20 caracteres;

ReferenciaExterna: (Opcional) Uso interno, pode ser um código ou outro valor que pode ser usado como referência em uma futura consulta na API do MercadoPago;

EmailPagador: (Opcional) Preencha este campo para que o MercadoPago envie um e-mail ao destinatário com as informações da cobrança;

Exemplos de uso:

Cobrar R\$ 299,90 com cartão de crédito em 2 parcelas, sem os parâmetros opcionais (ReferenciaExterna e EmailPagador):

```
MercadoPagoLocal1.ProcessarPagamento(299.90, true, 2, 'VendaExemplo1');
```

Cobrar R\$ 30,00 no débito, com referência externa e enviando e-mail ao pagador:

```
MercadoPagoLocal1.ProcessarPagamento(30, false, 1, 'VendaExemplo2', 'Cod12345', 'cliente@hotmail.com');
```

Evento retorno da venda:

O resultado da cobrança é obtido através do evento **TMercadoPagoLocal.OnResultadoVenda** contendo todos os dados no objeto Resultado: **TResultadoMercadoPagoDelphi**:

```
TMercadoPagoLocal.ResultadoVenda(  
Sender: TObject;  
Resultado: TResultadoMercadoPagoDelphi  
);
```

Resultado: Objeto do tipo **TResultadoMercadoPagoDelphi** com os dados da cobrança que podem ser acessados nas propriedades:

Resultado.VendaEfetuada (Boolean): indica se a venda foi efetuada com sucesso(True) ou se houve erro(False);

Resultado.IdPagamento (String): Id gerado no MercadoPago para a transação, que poderá ser usado futuramente para consulta na API do MercadoPago;

Resultado.ValorPago (Double): Valor final pago pelo cliente;

Resultado.TipoCartao (String): Tipo do cartão utilizado na venda;

Resultado.Parcelas (Integer): Número de parcelas da compra;

Resultado.TitularCartao: Nome do titular do cartão. (OBS: o formato do nome (Abreviado, Truncado...) pode variar de acordo com as políticas de segurança de cada cartão);

Resultado.Erro (String): Código do erro em caso de falha;

Resultado.DescricaoErro (String): Mensagem com descrição amigável do erro que pode ser mostrada ao usuário;

Exemplos de uso:

Confere se a venda foi efetuada e exibe o resultado:

```
if Resultado.VendaEfetuada = true then
```

```
begin
```

```
Label1.Text:= 'Id da Venda: ' + Resultado.IdPagamento + sLineBreak +
```

```
'Valor pago: R$' + FormatFloat('#####0.00', Resultado.ValorPago).Replace('.', ',') + sLineBreak +
```

```
'Tipo de cartão: ' + Resultado.TipoCartao + sLineBreak +
```

```
'Parcelas: ' + IntToStr(Resultado.Parcelas) + sLineBreak +
```

```
'Titular Cartão: ' + Resultado.TitularCartao;
```

```
end
```

```
else
```

```
begin
```

```
Label1.Text:= 'Erro: ' + Resultado.DescricaoErro;
```

```
end;
```

Erros:

O componente faz o tratamento dos erros de forma automática e gera uma mensagem amigável para o usuário de acordo com o erro. Mas caso seja necessário o erro pode ser tratado manualmente pegando o código do erro retornado (Resultado.Erro).

Abaixo segue tabela de retornos de erro, e suas respectivas constantes pré definidas no componente para uso:

Código do erro	Descrição oficial	Constante pré definida no componente
USER_ABORTED	When the user goes back at the first payment screen	mplUSER_ABORTED
NO_BT	Device has not bluetooth (only android)	mplNO_BT
EXPIRED_CARD	Expired card entered/swiped.	mplEXPIRED_CARD
TRY_AGAIN	In some payment rejected cases	mplTRY_AGAIN
WRONG_MCC	Invalid Merchant Category Code selected	mplWRONG_MCC
OWNERSHIP_ERROR	Point device belongs to another account	mplOWNERSHIP_ERROR

Código do erro	Descrição oficial	Constante pré definida no componente
LOW_BATTERY	Point device with critically low battery	mplLOW_BATTERY
UNRECOGNIZED_DEVICE	Can't recognize the point (bluetooth) device	mplUNRECOGNIZED_DEVICE
VISA_BLACKLISTED	Visa blacklisted entered/swiped card	mplVISA_BLACKLISTED
USE_OTHER_CARD	Payment rejected, use other card requested	mplUSE_OTHER_CARD
INVALID_DATA_PROVIDED	Invalid card type selected (credit/debit)	mplINVALID_DATA_PROVIDED
USER_CANCELLED_ERROR	User cancelled payment (on the point device)	mplUSER_CANCELLED_ERROR
READER_DISABLED	Point device blocked	mplREADER_DISABLED
CANCELLED_PAYMENT_ATTEMPT	Intent integration: cancelled attempt (only android)	mplCANCELLED_PAYMENT_ATTEMPT
PAYMENT_REJECTED	Payment rejected	mplPAYMENT_REJECTED
INVALID_FINANCING_COST	Invalid installments selected	mplINVALID_FINANCING_COST

Abaixo tabela das descrições geradas para cada tipo de erro:

Código do erro	Descrição
USER_ABORTED	Cancelado pelo usuário
NO_BT	Dispositivo não possui bluetooth
EXPIRED_CARD	Cartão expirado
TRY_AGAIN	Tente novamente
WRONG_MCC	Merchant Category Code inválido
OWNERSHIP_ERROR	Dispositivo Point pertence a outra conta

Código do erro	Descrição
LOW_BATTERY	Dispositivo Point com pouca bateria
UNRECOGNIZED_DEVICE	Não foi possível reconhecer o dispositivo Point
VISA_BLACKLISTED	Cartão na Blacklist da Visa
USE_OTHER_CARD	Pagamento rejeitado, utilize outro cartão
INVALID_DATA_PROVIDED	Tipo de cartão selecionado inválido
USER_CANCELLED_ERROR	Pagamento cancelado no dispositivo Point
READER_DISABLED	Dispositivo Point bloqueado
CANCELLED_PAYMENT_ATTEMPT	Tentativa cancelada
PAYMENT_REJECTED	Pagamento rejeitado
INVALID_FINANCING_COST	Número de parcelas inválido

Demo:

Confira a demo “**DemoDelphiMercadoPago**” dentro de “**MercadoPagoDelphi.groupproj**” e veja um exemplo de uso do componente pronto.