



mercado
pago

| Developers

<dev>program

Instrucciones de certificación
Checkout PRO



Para formar parte de nuestra comunidad y acceder a todos los beneficios extras, es necesario que apruebes este examen.

Te recomendamos leer todo antes de empezar y revisar el paso a paso del examen donde encontrarás los requerimientos técnicos y la documentación necesaria

Para el ejercicio necesitas tener conocimientos en:

- ★ **Desarrollo backend.**
- ★ **Desarrollo front-end para experiencia “modal”**
- ★ **API Rest.**



CRITERIOS

EJERCICIO

EVALUACIÓN

INFORMACIÓN ADICIONAL

Conoce los criterios de evaluación...

Deberás subir tu ejercicio resuelto a un servidor (Heroku o similar) que permita disponibilizar los archivos de forma online y pública. Si programas en PHP o NodeJS, debés utilizar la tienda dummy creada por el equipo de Mercado Pago para la realización de este examen. De lo contrario, podés generar tu propia tienda. Es importante que la tienda no tenga ningún tipo de autenticación para que los usuarios puedan acceder.

Los principales puntos evaluados son:

1

El desarrollo de un **receptor de notificaciones** y su especificación en **notification_url**.

Revisaremos el correcto manejo de estas notificaciones de pago.

2

El correcto manejo de las **back_url** para los diferentes estados de pagos y la visualización de los campos requeridos indicados en el ejercicio en la página de éxito de las back_urls

3

La **completitud de la preferencia** de acuerdo a la información ingresada al momento de simular un pago.

4

La correcta inclusión del **script de seguridad** para el comportamiento del usuario en las páginas de tu ejercicio.

La simulación de un pago aprobado para obtener un payment ID.

CRITERIOS

EJERCICIO

EVALUACIÓN

INFORMACIÓN ADICIONAL

Documentación

Para ayudarte con la ejecución del examen, confía las documentaciones abajo.

[Checkout PRO configuración](#)

[API Reference | Crear preferencia](#)

[API Reference | Obtener pago](#)

[Código de seguridad para mejorar la aprobación](#)

[Repositorio GitHub PHP](#)

[Repositorio GitHub Nodejs](#)

Desafío

La empresa “Tienda azul” que vende celulares está terminando su sitio de eCommerce. Te contrataron para que implementes el Checkout de Mercado Pago como solución de pago sobre el proyecto que, actualmente, tiene la empresa.



Requerimientos

Integrator ID

Se debe incluir el **integrator id*** que es un código único para identificar a los desarrolladores o agencias que realizaron la integración. Para este ejercicio debes utilizar el Integrator ID:

dev_24c65fb163bf11ea96500242ac130004

Medios de Pago

El cliente desea que el número de cuotas (**mensualidades**) en el que sus compradores pueden diferir el pago sea **máximo 6**. A su vez, **no** quiere permitir pagos con tarjetas de crédito **Visa**.

Información del pagador

- ★ **Nombre y apellido:** Lalo Landa
- ★ **Email:** email del test-user pagador (vea en [información adicional](#) los test-users y sus informaciones)
- ★ **Teléfono:** ingresa un número telefónico válido en tu país
- ★ **Dirección de la calle del pagador:** calle falsa 123
- ★ **Código postal de la dirección del pagador:** ingresa un código postal de tu país

External reference

Deberás indicar tu **Correo electrónico*** ligado a la cuenta de Mercado Pago

*Cuando apruebes la certificación tu propio integrator ID será generado

Requerimientos

⚙️ Producto

- ★ **ID:** ingresa **4 dígitos numéricos**.
- ★ **Nombre del Producto:** Nombre del producto seleccionado del carrito del ejercicio.
- ★ **Descripción del Producto:** “Dispositivo móvil de Tienda e-commerce”
- ★ **URL Imagen:** Foto del producto seleccionado. (**url válida**)
- ★ **Cantidad:** 1
- ★ **Precio:** Mayor a **1 USD**

⚙️ Páginas de retorno (back_url)

Una vez que el pagador termine el flujo de Mercado Pago, se deberá retornar al sitio web del cliente y manejar una pantalla diferente para cada estado de pago:

- ★ El pago es “rechazado” o no ha finalizado (**failure**)
- ★ El usuario a decidido pagar con un medio de pago offline (ticket) y el pago quede en un estado “pending” o “in_process” (**pending**)
- ★ El pago es exitoso. (**success**)

En la pantalla se deberá mostrar la información proveniente de los parámetros que enviamos en el Query String como: payment_method_id que se usó para pagar, el valor del campo external_reference y el ID de pago (payment_id o collection_id) de Mercado Pago. Se deberá configurar para que desde el checkout de Mercado Pago el retorno en caso de pago aprobado sea automático (auto_return).

Nota: Estos son los parámetros que enviamos en la QueryString cuando redirigimos a las back_url establecidas:

```
https://www.tusitio.com/success.php?collection_id=[  
PAYMENT_ID]&collection_status=app  
roved&external_refe  
rence=[EXTERNAL_REFERENCE]&payment_type=cre  
dit_card&preference_id=[PREFER  
ENCE_ID]&site_id=[  
SITE_ID]&processing_mode=aggregator&merchant_  
account_id=null
```

Checkout PRO

El cliente solicitó que el Checkout de Mercado Pago se abra en la misma ventana en la que se muestra la tienda (**modal**). El botón que inicia el flujo del Checkout debe decir: **“Pagar la compra”**

Al generar la **preferencia de pago** se debe:

- Enviar la información del ítem que se está comprando.
- Enviar los datos del pagador.
- Enviar la URL donde se van a recibir las notificaciones de pago.
- Enviar external_reference.
- Limitar la cantidad de cuotas según lo solicitado.
- No ofrecer los medios de pago según lo solicitado.



Notificaciones Webhook

Deberás generar un **endpoint** que sea capaz de recibir nuestras notificaciones Webhook según la [documentación](#) y, en el campo notification_url de la preferencia, deberá apuntar a éste.



Código de seguridad

Es necesario que incluyas, en cada uno de tus archivos, este [javascript](#) de seguridad que sirve para detectar comportamientos inusuales de los compradores en el sitio. Éste deberá aparecer en el flujo completo de la compra y se deberá enviar correctamente el parámetro **“view”** asignando el valor correspondiente a la página del sitio que se está visitando (**“home”, “search”, “item” o vacío**).

CRITERIOS

EJERCICIO

EVALUACIÓN

INFORMACIÓN ADICIONAL

<dev>program

Para la simulación del pago, deberás redireccionar al usuario al “**init_point**” productivo que obtendrás de nuestra API de preferencias (no utilizar “sandbox_init_point”).

Una vez completada la integración, será necesario que realices una simulación de pago.

Al completar la simulación de pago recibirás en el **URL de notificaciones** el **payment_id** es importante que conserves este identificador ya que será necesario que lo envíes en el formulario de respuestas:

ENVÍO DE ID DE PAGO

Certificación de Mercado Pago Checkout Pro

Ante todo muchas gracias por tomarte el tiempo para realizar la prueba para convertirte en Desarrollador Certificado de Mercado Pago. Rellena el siguiente formulario después de haber hecho la integración y las pruebas. Muchas gracias y éxitos!



¿Recibiste la notificación de pago? indicanos el payment_id *

El payment_id es el identificador del pago. Por ejemplo, si recibiste "data": {"id": "9999999999"}, el payment_id será 9999999999.

Ingresa el id del pago

[Volver](#)[Avanzar](#)

Para la configuración del plugin deberán utilizar las credenciales productivas del test-user de Mercado Pago que indicamos en la siguiente página más abajo.

CRITERIOS
EJERCICIO
EVALUACIÓN

INFORMACIÓN ADICIONAL

Argentina

Integrator ID:

★ `dev_24c65fb163bf11ea96500242ac130004`

Test User (Vendedor) | Producción

★ **Access Token:**

`APP_USR-8709825494258279-092911-227a84b3ec8d8b30fff364888abeb67a-1160706432`

★ **Public Key:**

`APP_USR-ff96fe80-6866-4888-847e-c69250754d38`

Test User (Comprador o pagador)

★ **Email:**

`test_user_36961754@testuser.com`

Para probar los flujos, deberán usar un test-user que será el pagador o comprador y podrán utilizar las tarjetas de prueba.

Utiliza [estas tarjetas](#) para hacer simulaciones de pago en el testeo.

Brasil

Integrator ID:

★ dev_24c65fb163bf11ea96500242ac130004

Test User (Vendedor) | Producción

★ Access Token:

APP_USR-2453313229452572-092911-2df2d24eb035a4c0852f3455a89d1459-1160953381

★ Public Key:

APP_USR-5c4239f9-0e8a-42ba-8eb1-920b1183d7f7

Test User (Comprador o pagador)

★ Email:

test_user_33467020@testuser.com

Para probar los flujos, deberán usar un test-user que será el pagador o comprador y podrán utilizar las tarjetas de prueba.

Utiliza [estas tarjetas](#) para hacer simulaciones de pago en el testeo.

México

Integrator ID:

★ `dev_24c65fb163bf11ea96500242ac130004`

Test User (Vendedor) | Producción

★ **Access Token:**
`APP_USR-2926550097213535-092911-5eded40868803c83f12e9eef1afa99fa-1160956296`

★ **Public Key:**
`APP_USR-a68157fb-5513-4dc7-adbf-709ba3e46766`

Test User (Vendedor) | Sandbox

★ **Access Token:**
`TEST-2926550097213535-092911-9f95049df86420b3055add7ae6e0e56e-1160956296`

★ **Public Key:**
`TEST-179e1cbc-8f9d-4b85-944b-112d3c67838e`

Test User (Comprador o pagador)

★ **Email:**
`test_user_94708656@testuser.com`

Para probar los flujos, deberán usar un test-user que será el pagador o comprador y podrán utilizar las tarjetas de prueba.

Utiliza [estas tarjetas](#) para hacer simulaciones de pago en el testeo.

Colombia

Integrator ID:

★ `dev_24c65fb163bf11ea96500242ac130004`

Test User (Vendedor) | Producción

★ **Access Token:**

`APP_USR-8902774665784533-092911-fab78ca802b6475923ebb446b02fee62-1160743707`

★ **Public Key:**

`APP_USR-9a464dc4-1ad8-4646-af03-1f18d41cd90c`

Test User (Vendedor) | Sandbox

★ **Access Token:**

`TEST-8902774665784533-092911-61bed7c951173114ca299d6da3df87bd-1160743707`

★ **Public Key:**

`TEST-913814df-94e9-4e3e-958b-89caac1a3a9c`

Test User (Comprador o pagador)

★ **Email:**

`test_user_51300629@testuser.com`

Para probar los flujos, deberán usar un test-user que será el pagador o comprador y podrán utilizar las tarjetas de prueba.

Utiliza [estas tarjetas](#) para hacer simulaciones de pago en el testeo.

Uruguay

Integrator ID:

★ `dev_24c65fb163bf11ea96500242ac130004`

Test User (Vendedor) | Producción

★ **Access Token:**
`APP_USR-2815099995655791-092911-c238fdac299eadc66456257445c5457d-1160950667`

★ **Public Key:**
`APP_USR-3fa91575-503e-476a-b21d-4f49cae39fa6`

Test User (Vendedor) | Sandbox

★ **Access Token:**
`TEST-2815099995655791-092911-147d97da9f81833ec976b282edb7f40e-1160950667`

★ **Public Key:**
`TEST-aa4575ea-bf83-4ba7-b8d0-ae78899dc9b9`

Test User (Comprador o pagador)

★ **Email:**
`test_user_17805074@testuser.com`

Para probar los flujos, deberán usar un test-user que será el pagador o comprador y podrán utilizar las tarjetas de prueba.

Utiliza [estas tarjetas](#) para hacer simulaciones de pago en el testeo.

Perú

Test User (Vendedor) | Producción

- ★ **Access Token:**
APP_USR-5803173797352206-050918-e3faf048a99e583771de8f2d1fb47065-1120797873
- ★ **Public Key:**
APP_USR-4ee10498-d923-440e-90fa-f21866fba100

Test User (Vendedor) | Sandbox

- ★ **Access Token:**
TEST-5803173797352206-050918-eaf289baab83a68af05297e96f458a9a-1120797873
- ★ **Public Key:**
TEST-1636a0d8-2f90-4ab6-85e2-320908e8dc1b

Test User (Comprador o pagador)

- ★ **Email:**
test_user_81131286@testuser.com

Para probar los flujos, deberán usar un test-user que será el pagador o comprador y podrán utilizar las tarjetas de prueba.

Utiliza [estas tarjetas](#) para hacer simulaciones de pago en el testeo.

Tarjetas de PRUEBA

Tarjeta	Número	Código de seguridad	Fecha de caducidad
Mastercard	5031 7557 3453 0604	123	11/25
Visa	4509 9535 6623 3704	123	11/25
American Express	3711 803032 57522	1234	11/25

Es necesario que pongas **APRO** en el nombre del titular de la tarjeta para que este sea **APROBADO**.

THANK YOU!

