

STRIPE

GENERADOR DE PAGOS EN MVC



1.- REGISTRARSE EN STRIPE

- TENEMOS QUE REGISTRARNOS EN STRIPE.



2.- OBTENER CLAVES

- OBTENER NUESTRAS CLAVES
- ESTAS SE ENCUENTRAN EN EL DASHBOARD

Te damos la bienvenida. Para empezar, sigue estos pasos.

> Busca la integración adecuada para tu empresa

✓ Consigue tus claves de API de prueba

Clave publicable

pk_test_51HkOQBAHmBGqUw1CBvfyaCFtJnYUGcVQpQYtik932bf0oBUwD
izwUECj1FZeGWA2PCbgbyAuwQ7kXa01AZo39T00zv10DZ6w

Clave secreta

Revelar la clave de prueba

3.- AÑADIR A WEBCONFIG

- AÑADIR LAS LLAVES PUBLICABLE Y SECRETA A NUESTRO WEBCONFIG

```
<appSettings>
  <add key="webpages:Version" value="3.0.0.0"/>
  <add key="webpages:Enabled" value="false"/>
  <add key="ClientValidationEnabled" value="true"/>
  <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
  <add key="stripeSecretKey" value="sk_test_51HkOQBAHmBGqUW1CBvfyaCFtJnjYUGcVQpQYtik932bf0oBUwDizwUECj1fZeGWZA2PCbgbyAuWQ7kXa01AZo39T00zv1ODZ6w"/>
  <add key="stripePublishKey" value="pk_test_51HkOQBAHmBGqUW1CBvfyaCFtJnjYUGcVQpQYtik932bf0oBUwDizwUECj1fZeGWZA2PCbgbyAuWQ7kXa01AZo39T00zv1ODZ6w"/>
</appSettings>
<system.web>
  <compilation debug="true" targetFramework="4.7.2"/>
  <httpRuntime targetFramework="4.7.2"/>
</system.web>
```


4.- INSTALAR STRIPE

- AÑADIR POR CONSOLA EL SIGUIENTE PAQUETE
 - INSTALL-PACKAGE STRIPE.NET
 - AL INSTALARLO HAY QUE REVISAR QUE SU VERSIÓN SEA COMPATIBLE CON LA VERSIÓN DE LA API WEB DONDE NOS REGISTRAMOS.

```
PM> Install-Package Stripe.net
```

```
Intentando recopilar información sobre la dependencia para el paquete 'Stripe.net.39.22.0' respecto
La recopilación de información de dependencia tardó 24,84 sec
Intentando resolver las dependencias para el paquete 'Stripe.net.39.22.0' con DependencyBehavior 'Lo
La resolución de información de dependencia tardó 0 ms
Resolviendo acciones para instalar el paquete 'Stripe.net.39.22.0'
Se resolvieron las acciones para instalar el paquete 'Stripe.net.39.22.0'
  GET https://api.nuget.org/v3-flatcontainer/system.threading.tasks.extensions/4.5.2/system.threadin
  GET https://api.nuget.org/v3-flatcontainer/system.runtime.interopservices.runtimeinformation/4.3.0
  GET https://api.nuget.org/v3-flatcontainer/microsoft.bcl.asyncinterfaces/1.1.0/microsoft.bcl.async
  OK https://api.nuget.org/v3-flatcontainer/system.runtime.interopservices.runtimeinformation/4.3.0/
Recuperando paquete 'Microsoft.Bcl.AsyncInterfaces 1.1.0' de 'nuget.org'.
Recuperando paquete 'Stripe.net 39.22.0' de 'nuget.org'.
Recuperando paquete 'System.Runtime.CompilerServices.Unsafe 4.5.2' de 'nuget.org'.
```

Versión de la API

VERSIÓN
2020-08-27 🔗 Valor predeterminado
Última
2020-03-02 🔗

Es probable que estés usando varias versiones de API a causa de las librerías o los plugin

Instalado: 37.35.0 Desinstalar

Versión: 37.35.0 Instalar

Opciones

Descripción

Stripe.net is a sync/async .NET 4.5+ client, and a portable class library for the Stripe API. (Official Library)

Versión: 37.35.0

Autores: Stripe, Jayme Davis

Licencia: [Ver licencia](#)

Fecha de publicación: jueves, 27 de agosto de 2020 (27/08/2020)

URL del proyecto: <https://github.com/stripe/stripe-dotnet>

Notificar abuso: <https://www.nuget.org/packages/Stripe.net/37.35.0/ReportAbuse>

Etiquetas: stripe, payment, credit, cards, money, gateway, paypal, braintree

5.- AJUSTAR API KEY

- AÑADIR EN GLOBAL.ASAX:
 - AÑADIR LA LLAVE SECRETA EN LA CONFIGURACIÓN DE STRIPE.

```
using Stripe;
using System.Configuration;

namespace StripeFundamentals
{
    0 referencias
    public class MvcApplication : System.Web.HttpApplication
    {
        0 referencias
        protected void Application_Start()
        {
            Configuración General

            StripeConfiguration.ApiKey = ConfigurationManager.AppSettings["stripeSecretKey"];
        }
    }
}
```


5.- AJUSTAR API KEY

- AÑADIR EN GLOBAL.ASAX:
 - AÑADIR LA LLAVE SECRETA EN LA CONFIGURACIÓN DE STRIPE.
 - LOS TIPOS RESALTADOS TIENEN QUE TENER ESTE MISMO NOMBRE.

```
4 referencias
public class StripeViewModel
{
    1 referencia
    public string PublishableKey    { get; set; }
    1 referencia
    public string stripeToken      { get; set; }
    0 referencias
    public string stripeTokenType  { get; set; }
    1 referencia
    public string stripeEmail      { get; set; }
}
```


5.- AÑADIR FORM

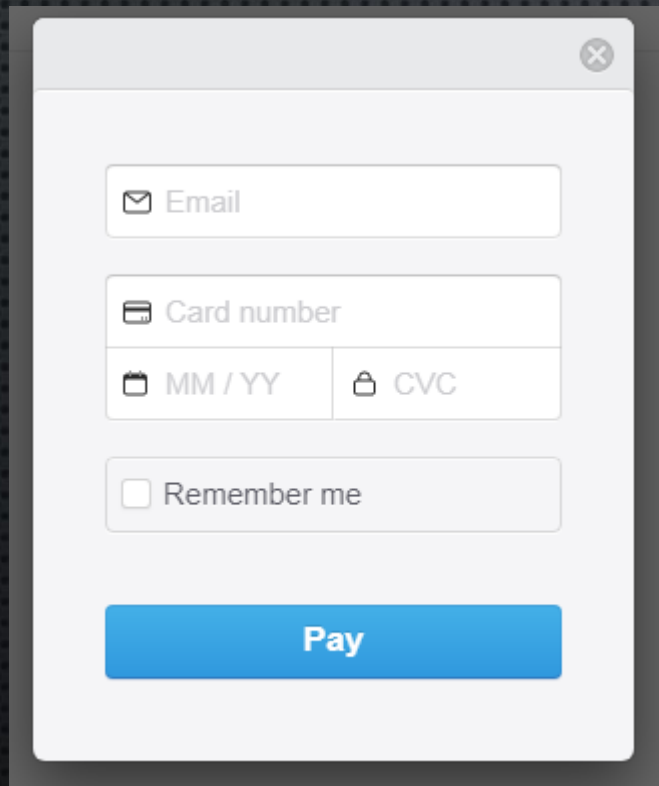
- 1) HACER REFERENCIA A NUESTRO MODELO CREADO ANTERIORMENTE.
- 2) HACER REFERENCIA AL JS CHECKOUT.JS.
- 3) PONER EN EL DATA-KEY NUESTRA LLAVE PUBLICABLE.
- 4) ES BUENA PRACTICA PONER TANTO UNA VALIDACIÓN DE TOKEN COMO UN MENSAJE DE ERROR.

```
@model StripeFundamentals.ViewModels.StripeViewModel 1

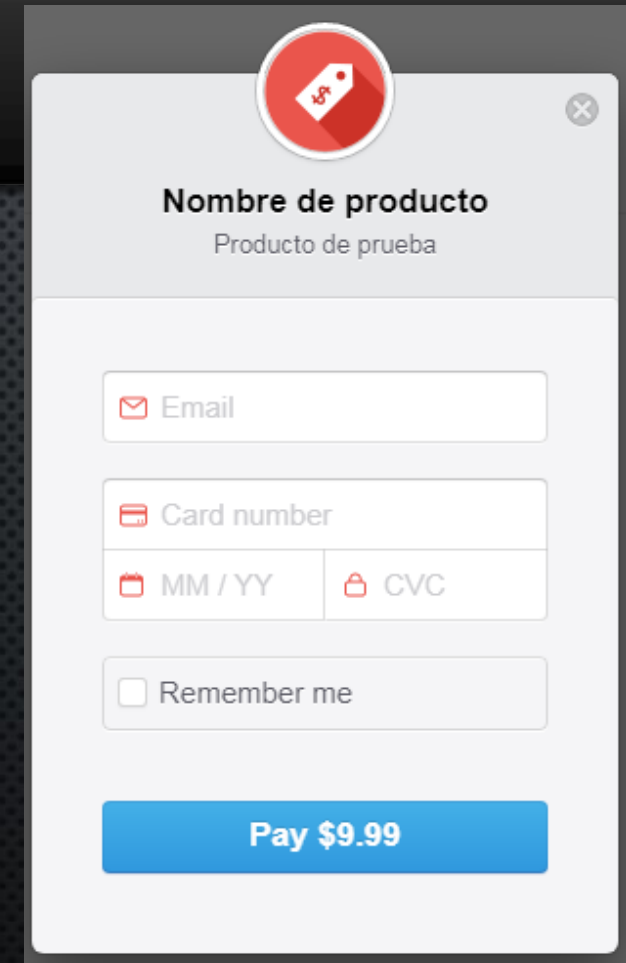
@using (Html.BeginForm("PagoIncrustado", "Home", FormMethod.Post))
{
    <script src="https://checkout.stripe.com/checkout.js" 2
        class="stripe-button"
        data-key="@Model.PublishableKey" 3
        data-amount="999"
        data-name="Nombre de producto"
        data-description="Producto de prueba"
        data-image="https://www.newneuromarketing.com/media/zoo/images/NNM-2015-019-Cost-consciousnes
    </script>
    @Html.ValidationSummary(true, "There was an error", new { @class = "alert alert-danger" }) 4
    @Html.AntiForgeryToken()
}
```

(OPCIONAL) CONFIGURAR FORM

```
<script src      = "https://checkout.stripe.com/checkout.js"
      class      = "stripe-button"
      data-key    = "@Model.PublishableKey"
      data-amount = "999"
      data-name   = "Nombre de producto"
      data-description = "Producto de prueba"
      data-image  = "/img/documentation/checkout/marketplace.png"
      data-locale = "auto"
>
</script>
```



A mockup of a Stripe checkout form. It features a light gray header with a close button (X). Below the header, there is a white form area with a light gray border. The form contains an email input field with an envelope icon, a card number input field with a card icon, and two input fields for the expiration date (MM / YY) and CVC (CVC) with their respective icons. A checkbox labeled "Remember me" is located below the card fields. At the bottom of the form is a blue button with the text "Pay".



A mockup of a Stripe checkout form, similar to the one on the left but with additional branding. It features a light gray header with a close button (X) and a red circular icon with a white tag and dollar sign. Below the header, the text "Nombre de producto" is displayed in bold, followed by "Producto de prueba" in a smaller font. The form area is white with a light gray border and contains the same input fields and "Remember me" checkbox as the left mockup. At the bottom is a blue button with the text "Pay \$9.99".



6.- MANDAR LLAVE A FORM

- AL LLAMAR A NUESTRA VISTA PRINCIPAL DEBEMOS TRAER NUESTRA LLAVE DE APP SETTINGS

```
StripeViewModel Stripe = new StripeViewModel()
{
    PublishableKey = ConfigurationManager.AppSettings["stripePublishKey"]
};
GM = new GrandModelViewHome()
{
    Stripe = Stripe
};

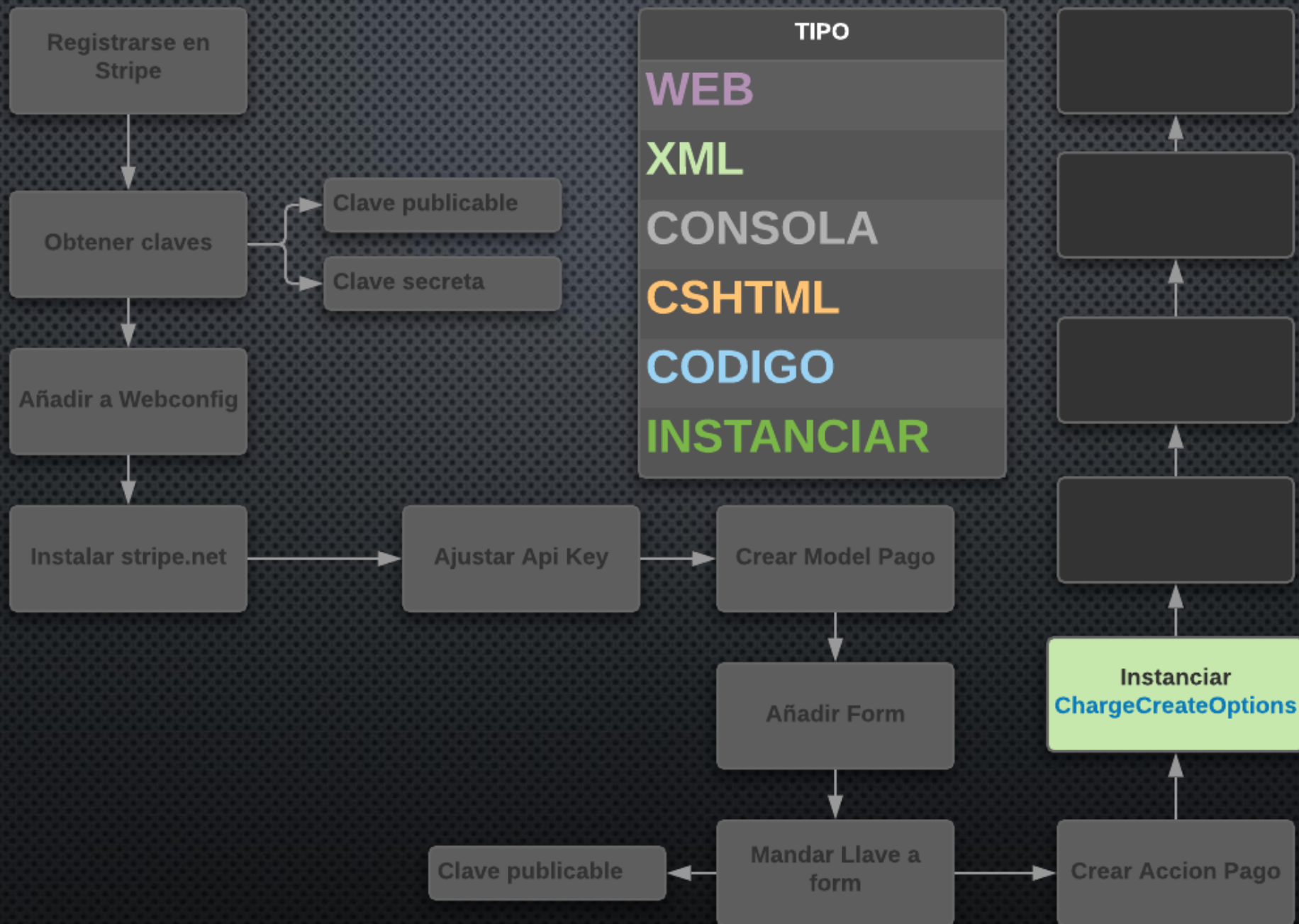
return View(GM);
```




7.- CREAR ACCIÓN PAGO

AL HACER EL PAGO NUESTRO FORM LLAMARA A NUESTRO SERVIDOR, CREAMOS UNA ACTION QUE RECIBA LA INFORMACIÓN RECIBIDA.

```
[HttpPost]
[ValidateAntiForgeryToken]
0 referencias
public ActionResult PagoIncrustado(StripeViewModel InformacionPago)
{
    Capsula
    return RedirectToAction("Incrustado");
}
```

8.- INSTANCIAR CHARGE OPTIONS

- CHARGE OPTIONS ES EL OBJETO QUE CONTIENE LAS OPCIONES DE NUESTRA COMPRA, HAY DETALLES OPCIONALES Y OTROS REQUERIDOS.

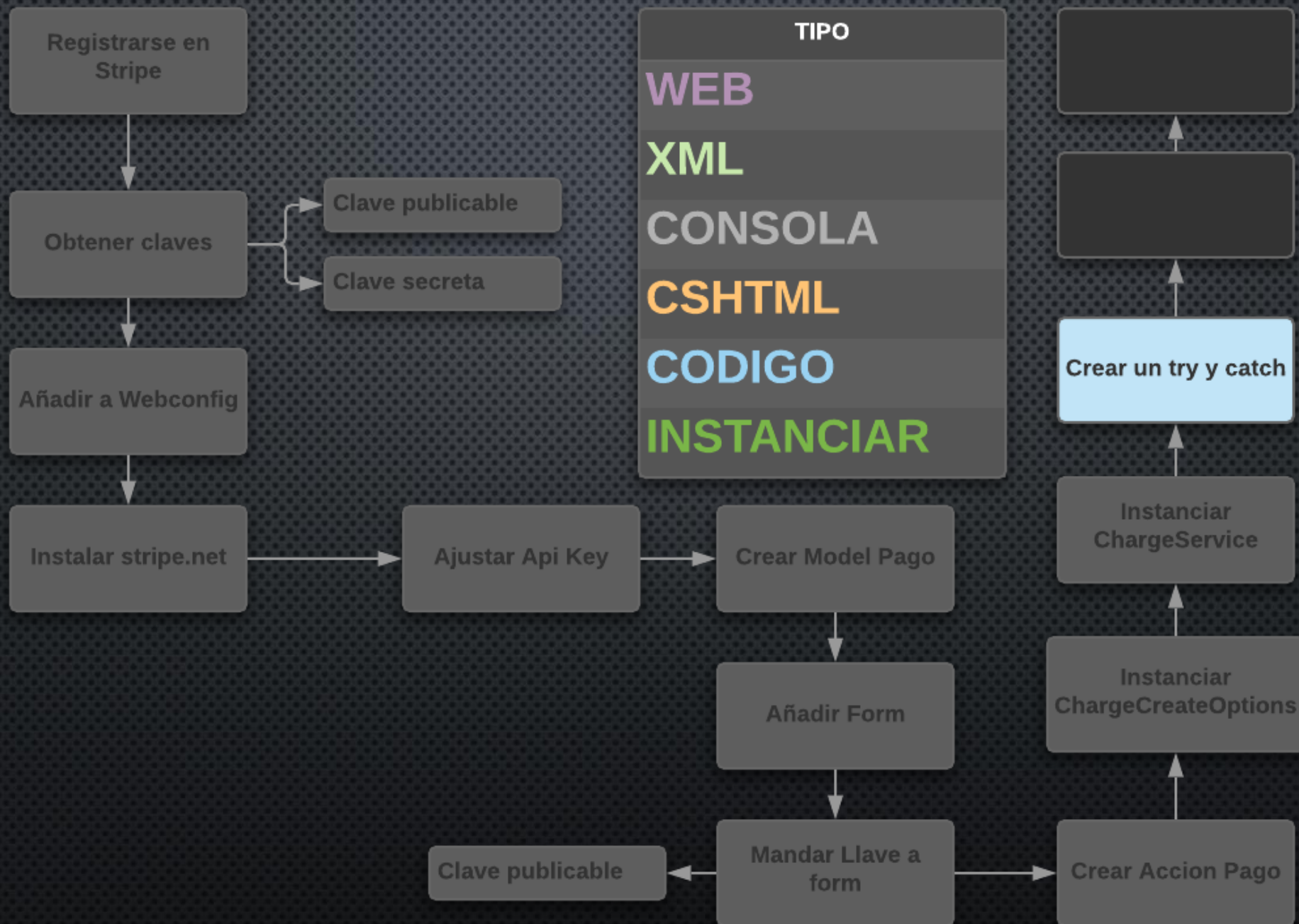
```
ChargeCreateOptions ChargeOptions = new ChargeCreateOptions()
{
    // Requerido
    Amount = 3900,
    Currency = "USD",
    Source = InformacionPago.stripeToken,
    // Opcional
    Description = "Pago de Servicio por 1 mes",
    ReceiptEmail = InformacionPago.stripeEmail
};
```




9.- INSTANCIAR CHARGE SERVICE

- EL OBJETO CHARGE SERVICE SERÁ EL QUE HAGA LA LLAMADA A STRIPE

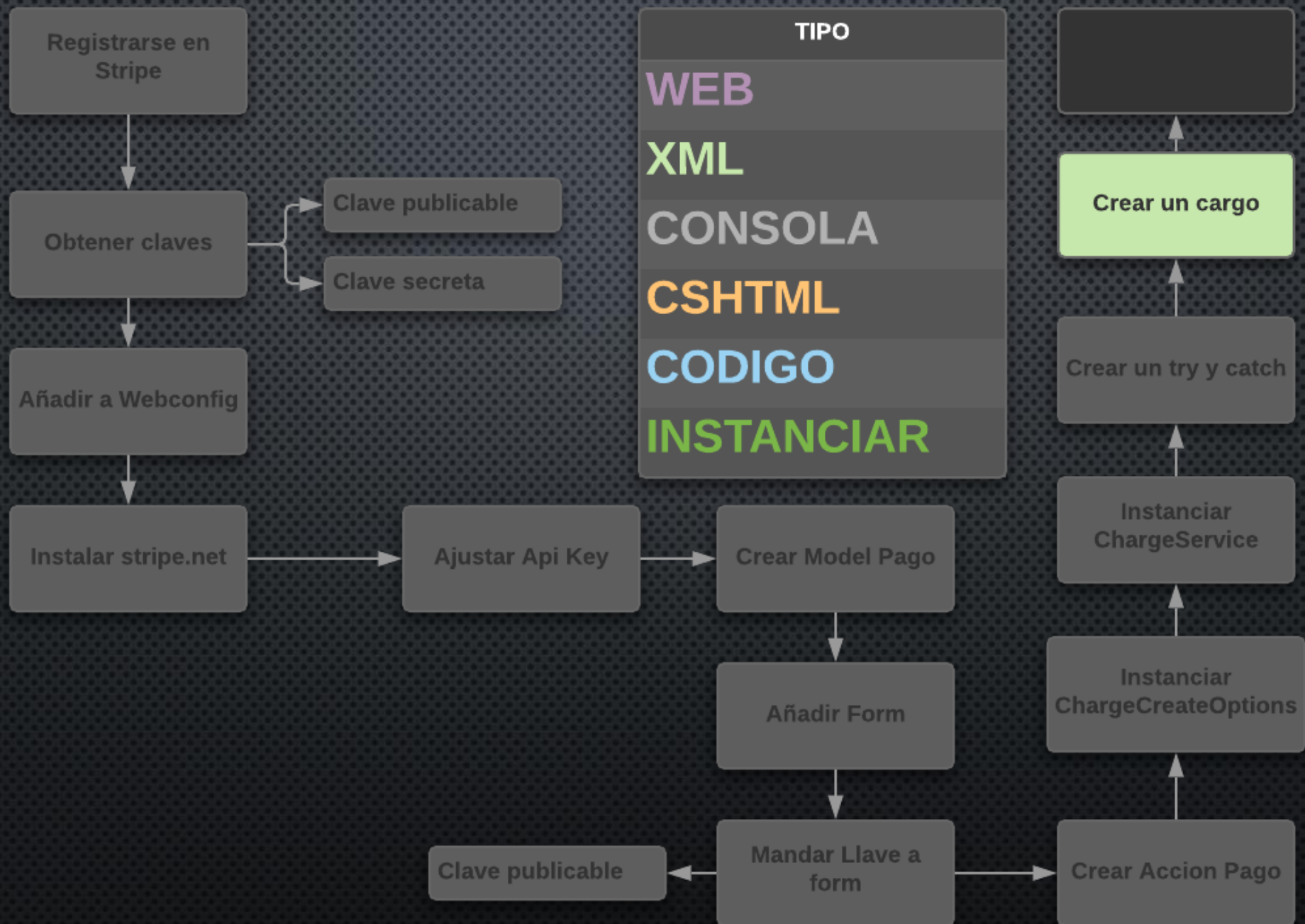
```
ChargeCreateOptions ChargeOptions = new ChargeCreateOptions();  
ChargeService ChargeService = new ChargeService();
```

10.- CREAR TRY Y CATCH

- DEBIDO A QUE PUEDEN OCURRIR EXCEPCIONES UTILIZAREMOS UN TRY Y UN CATCH

```
try
{
    Capsula
}
catch (StripeException e)
{
    Capsula
}
```

11.- CREAR UN CARGO

- MANDAREMOS A LLAMAR EL OBJETO CRÉATE DE CHARGE SERVICE..

```
try
{
    Charge ChargeResponse = ChargeService.Create(ChargeOptions);
}
catch (StripeException e)
{
    Capsula
}
```




12.- MANEJAR EXCEPCIONES

- EN CASO DE TENER UN ERROR, AGREGAREMOS ESTE ERROR Y LO REGRESAREMOS A NUESTRA VISTA PRINCIPAL.

-

```
}  
catch (StripeException e)  
{  
  
    ModelState.AddModelError(string.Empty, e.Message);  
    GM = new GrandModelViewHome()  
    {  
        Stripe = InformacionPago  
    };  
    return View("Incrustado", GM);  
}
```