# Demo Xamarin Divisas II

1. Cree el proyecto.
2. Agregar los Nuget: Microsoft.BCL, Microsoft.Net.Http, Newtonsoft.Json
3. Crea la HomePage.xaml:

<?xml version="1.0" encoding="UTF-8"?>  
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"   
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml" x:Class="DivisasII.HomePage">  
    <ContentPage.Content>  
        <StackLayout Padding="10,20,10,10" Spacing="10">  
            <Label Text="Divisas"  
                HorizontalOptions="Center"  
                VerticalOptions="Center"  
                Font="30"/>  
            <ActivityIndicator x:Name="waitActivityIndicator"/>  
            <Entry x:Name="amountEntry"  
                Placeholder="Entre valor a convertir"  
                Keyboard="Numeric"/>  
            <Picker x:Name="sourcePicker"   
                Title="Moneda Origen"/>   
            <Picker x:Name="targetPicker"   
                Title="Moneda Destino"/>   
            <Button x:Name="convertButton"  
                Text="Convertir"  
                BackgroundColor="Blue"  
                TextColor="White"  
                HorizontalOptions="FillAndExpand"  
                VerticalOptions="Center"/>  
            <Label x:Name="convertedLabel"  
                Font="20"  
                FontAttributes="Bold"  
                HorizontalOptions="CenterAndExpand"  
                VerticalOptions="CenterAndExpand"/>  
        </StackLayout>  
    </ContentPage.Content>  
</ContentPage>

1. Como vamos a usar el servicio: <https://openexchangerates.org/api/latest.json?app_id=f490efbcd52d48ee98fd62cf33c47b9e> para obtener las tasas de cambio. Creemos un objeto donde mapearemos la respuesta:

using System;  
using Newtonsoft.Json;  
  
namespace DivisasII  
{  
    public class ExchangeRates  
    {  
        [JsonProperty("disclaimer")]  
        public string Disclaimer { get; set; }  
  
        [JsonProperty("license")]  
        public string License { get; set; }  
  
        [JsonProperty("timestamp")]  
        public long TimeStamp { get; set; }  
  
        [JsonProperty("base")]  
        public string CurrencyBase { get; set; }  
  
        [JsonProperty("rates")]  
        public Rates Rates { get; set; }  
    }  
  
    public class Rates   
    {  
        public double BRL { get; set; } *// Real Brasilero*  
        public double CAD { get; set; } *// Dólar Canadiense*  
        public double CHF { get; set; } *// Franco Suizo*  
        public double CLP { get; set; } *// Peso Chileno*  
        public double COP { get; set; } *// Peso Colombiano*  
        public double DKK { get; set; } *// Corona Danesa*  
        public double EUR { get; set; } *// Euro*  
        public double GBP { get; set; } *// Libra Esterlina*  
        public double INR { get; set; } *// Rupia India*  
        public double JPY { get; set; } *// Yen Japonés*  
        public double MXN { get; set; } *// Peso Mexicano*  
        public double USD { get; set; } *// Peso Mexicano*  
    }  
}

1. Adicione el siguiente código a la HomePage:

using System;  
using System.Collections.Generic;  
  
using Xamarin.Forms;  
using System.Net.Http;  
using Newtonsoft.Json;  
  
namespace DivisasII  
{  
    public partial class HomePage : ContentPage  
    {  
        private ExchangeRates exchangeRates;  
  
        public HomePage ()  
        {  
            InitializeComponent ();  
  
            sourcePicker.Items.Add ("Coronas Danesas");  
            sourcePicker.Items.Add ("Dólares Canadienses");  
            sourcePicker.Items.Add ("Dólares Estadounidenses");  
            sourcePicker.Items.Add ("Euros");  
            sourcePicker.Items.Add ("Francos Suizos");  
            sourcePicker.Items.Add ("Libras Esterlinas");  
            sourcePicker.Items.Add ("Pesos Chilenos");  
            sourcePicker.Items.Add ("Pesos Colombianos");  
            sourcePicker.Items.Add ("Pesos Mexicanos");  
            sourcePicker.Items.Add ("Reales Brasileros");  
            sourcePicker.Items.Add ("Rupias Indias");  
            sourcePicker.Items.Add ("Yenes Japoneses");  
  
            targetPicker.Items.Add ("Coronas Danesas");  
            targetPicker.Items.Add ("Dólares Canadienses");  
            targetPicker.Items.Add ("Dólares Estadounidenses");  
            targetPicker.Items.Add ("Euros");  
            targetPicker.Items.Add ("Francos Suizos");  
            targetPicker.Items.Add ("Libras Esterlinas");  
            targetPicker.Items.Add ("Pesos Chilenos");  
            targetPicker.Items.Add ("Pesos Colombianos");  
            targetPicker.Items.Add ("Pesos Mexicanos");  
            targetPicker.Items.Add ("Reales Brasileros");  
            targetPicker.Items.Add ("Rupias Indias");  
            targetPicker.Items.Add ("Yenes Japoneses");  
  
            this.LoadRates ();  
  
            convertButton.Clicked += convertButton\_Clicked;  
        }  
  
        private async void convertButton\_Clicked(object sender, EventArgs e)  
        {  
            if (string.IsNullOrEmpty(amountEntry.Text)) {  
                await DisplayAlert("Error",   
                    "Debe ingresar una cantidad mayor a cero (0)", "Aceptar");  
                amountEntry.Focus();  
                return;  
            }  
  
            var amount = decimal.Parse(amountEntry.Text);  
            if (amount <= 0) {  
                await DisplayAlert("Error",   
                    "Debe ingresar una cantidad mayor a cero (0)", "Aceptar");  
                amountEntry.Focus();  
                return;  
            }  
  
            if (sourcePicker.SelectedIndex == -1) {  
                await DisplayAlert("Error",   
                    "Debe seleccionar una moneda origen", "Aceptar");  
                sourcePicker.Focus();  
                return;  
            }  
  
            if (targetPicker.SelectedIndex == -1) {  
                await DisplayAlert("Error",   
                    "Debe seleccionar una moneda destino", "Aceptar");  
                targetPicker.Focus();  
                return;  
            }  
  
            var amountConverted = this.Convert (amount,   
                sourcePicker.SelectedIndex, targetPicker.SelectedIndex);  
            convertedLabel.Text = string.Format (  
                "{0:N2} {1} = {2:N2} {3}",   
                amount,   
                sourcePicker.Items[sourcePicker.SelectedIndex],   
                amountConverted,   
                targetPicker.Items[targetPicker.SelectedIndex]);  
        }  
  
        private decimal Convert(decimal amount, int source, int target)  
        {  
            double rateSource = this.GetRate (source);  
            double rateTarget = this.GetRate (target);  
            return amount / (decimal)rateSource \* (decimal)rateTarget;  
        }  
  
        private double GetRate(int index)  
        {  
            switch (index) {  
            case 0:    return exchangeRates.Rates.DKK;  
            case 1:    return exchangeRates.Rates.CAD;  
            case 2:    return exchangeRates.Rates.USD;  
            case 3:    return exchangeRates.Rates.EUR;  
            case 4:    return exchangeRates.Rates.CHF;  
            case 5:    return exchangeRates.Rates.GBP;  
            case 6:    return exchangeRates.Rates.CLP;  
            case 7:    return exchangeRates.Rates.COP;  
            case 8:    return exchangeRates.Rates.MXN;  
            case 9:    return exchangeRates.Rates.BRL;  
            case 10: return exchangeRates.Rates.INR;  
            case 11: return exchangeRates.Rates.JPY;  
            default: return 1;  
            }  
        }  
  
        private async void LoadRates()  
        {  
            waitActivityIndicator.IsRunning = true;  
            HttpClient client = new HttpClient();  
            client.BaseAddress = new Uri("https://openexchangerates.org");              
            string url = string.Format ("/api/latest.json?app\_id=f490efbcd52d48ee98fd62cf33c47b9e");  
            var response = await client.GetAsync(url);  
            var result = response.Content.ReadAsStringAsync().Result;  
            waitActivityIndicator.IsRunning = false;  
  
            if (string.IsNullOrEmpty (result)) {  
                await DisplayAlert("Error",   
                    "No hubo respuesta del servidor, intente más tarde", "Aceptar");  
                return;  
            }  
  
            this.exchangeRates = JsonConvert.Deserial

izeObject<ExchangeRates> (result);  
        }  
    }  
}

//new:

**<?xml version="1.0" encoding="utf-8" ?>**

**<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"**

**xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"**

**x:Class="XDivisasApi.MainPage" Title="Main Page.....">**

**<ScrollView>**

**<StackLayout Spacing="10">**

**<Label Text="Divisas" FontSize="30" FontAttributes="Bold" HorizontalOptions="Center" VerticalOptions="Center"></Label>**

**<Label Text="Ingrese Valor, Seleccione moneda origen, seleccione moneda destino y presione el botón de convertir."**

**HorizontalOptions="Start"**

**FontSize="18"**

**VerticalOptions="Center"></Label>**

**<Entry x:Name="amountEntry" Placeholder="Ingrese Valor......"**

**Keyboard="Numeric"**

**HorizontalOptions="FillAndExpand"**

**VerticalOptions="Center"></Entry>**

**<Label Text="- Selecciones modeda Origen." FontSize="15" HorizontalOptions="FillAndExpand" VerticalOptions="Center"></Label>**

**<Picker x:Name="sourcePicker"  HorizontalOptions="FillAndExpand" VerticalOptions="Center"></Picker>**

**<Label Text="- Selecciones modeda Destino." FontSize="15" HorizontalOptions="FillAndExpand" VerticalOptions="Center"></Label>**

**<Picker x:Name="targetPicker"  HorizontalOptions="FillAndExpand" VerticalOptions="Center"></Picker>**

**<Button x:Name="convertButton"**

**Text="CovertT.!"**

**BackgroundColor="Maroon"**

**IsEnabled="False"**

**TextColor="White"**

**HorizontalOptions="FillAndExpand"**

**VerticalOptions="Start"></Button>**

**<ActivityIndicator x:Name="waitActivityIndicator" IsRunning="False" HorizontalOptions="FillAndExpand"**

**VerticalOptions="Center"></ActivityIndicator>**

**<Label x:Name="covertedLabel"**

**Text="No se han realizado conversiones."**

**IsEnabled="False"**

**FontSize="22"**

**FontAttributes="Bold"**

**HorizontalTextAlignment="Center"**

**VerticalTextAlignment="Center"**

**BackgroundColor="Teal"**

**TextColor="Red"**

**HorizontalOptions="FillAndExpand"**

**VerticalOptions="FillAndExpand"></Label>**

**</StackLayout>**

**</ScrollView>**

**</ContentPage>**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**namespace XDivisasApi**

**{**

**public class ExchangeRates**

**{**

**public string disclaimer { get; set; }**

**public string license { get; set; }**

**public double  timestamp { get; set; }**

**public string @base { get; set; }**

**public Rates rates { get; set; }**

**}**

**public class Rates**

**{**

**public double AED { get; set; }**

**public double AFN { get; set; }**

**public double ALL { get; set; }**

**public double AMD { get; set; }**

**public double ANG { get; set; }**

**public double AOA { get; set; }**

**public double ARS { get; set; }**

**public double AUD { get; set; }**

**public double AWG { get; set; }**

**public double AZN { get; set; }**

**public double BAM { get; set; }**

**public double  BBD { get; set; }**

**public double BDT { get; set; }**

**public double BGN { get; set; }**

**public double BHD { get; set; }**

**public double BIF { get; set; }**

**public double  BMD { get; set; }**

**public double BND { get; set; }**

**public double BOB { get; set; }**

**public double BRL { get; set; }**

**public double  BSD { get; set; }**

**public double BTC { get; set; }**

**public double BTN { get; set; }**

**public double BWP { get; set; }**

**public double BYN { get; set; }**

**public double BYR { get; set; }**

**public double BZD { get; set; }**

**public double CAD { get; set; }**

**public double CDF { get; set; }**

**public double CHF { get; set; }**

**public double CLF { get; set; }**

**public double CLP { get; set; }**

**public double CNY { get; set; }**

**public double COP { get; set; }**

**public double CRC { get; set; }**

**public double  CUC { get; set; }**

**public double CUP { get; set; }**

**public double CVE { get; set; }**

**public double CZK { get; set; }**

**public double DJF { get; set; }**

**public double DKK { get; set; }**

**public double DOP { get; set; }**

**public double DZD { get; set; }**

**public double EEK { get; set; }**

**public double EGP { get; set; }**

**public double ERN { get; set; }**

**public double ETB { get; set; }**

**public double EUR { get; set; }**

**public double FJD { get; set; }**

**public double FKP { get; set; }**

**public double GBP { get; set; }**

**public double GEL { get; set; }**

**public double GGP { get; set; }**

**public double GHS { get; set; }**

**public double GIP { get; set; }**

**public double GMD { get; set; }**

**public double GNF { get; set; }**

**public double GTQ { get; set; }**

**public double GYD { get; set; }**

**public double HKD { get; set; }**

**public double HNL { get; set; }**

**public double HRK { get; set; }**

**public double HTG { get; set; }**

**public double HUF { get; set; }**

**public double IDR { get; set; }**

**public double ILS { get; set; }**

**public double IMP { get; set; }**

**public double INR { get; set; }**

**public double IQD { get; set; }**

**public double IRR { get; set; }**

**public double ISK { get; set; }**

**public double JEP { get; set; }**

**public double JMD { get; set; }**

**public double JOD { get; set; }**

**public double JPY { get; set; }**

**public double KES { get; set; }**

**public double KGS { get; set; }**

**public double KHR { get; set; }**

**public double KMF { get; set; }**

**public double KPW { get; set; }**

**public double KRW { get; set; }**

**public double KWD { get; set; }**

**public double KYD { get; set; }**

**public double KZT { get; set; }**

**public double LAK { get; set; }**

**public double LBP { get; set; }**

**public double LKR { get; set; }**

**public double LRD { get; set; }**

**public double LSL { get; set; }**

**public double LTL { get; set; }**

**public double LVL { get; set; }**

**public double LYD { get; set; }**

**public double MAD { get; set; }**

**public double MDL { get; set; }**

**public double MGA { get; set; }**

**public double MKD { get; set; }**

**public double MMK { get; set; }**

**public double MNT { get; set; }**

**public double MOP { get; set; }**

**public double MRO { get; set; }**

**public double MTL { get; set; }**

**public double MUR { get; set; }**

**public double MVR { get; set; }**

**public double MWK { get; set; }**

**public double MXN { get; set; }**

**public double MYR { get; set; }**

**public double MZN { get; set; }**

**public double NAD { get; set; }**

**public double NGN { get; set; }**

**public double NIO { get; set; }**

**public double NOK { get; set; }**

**public double NPR { get; set; }**

**public double NZD { get; set; }**

**public double OMR { get; set; }**

**public double  PAB { get; set; }**

**public double PEN { get; set; }**

**public double PGK { get; set; }**

**public double PHP { get; set; }**

**public double PKR { get; set; }**

**public double PLN { get; set; }**

**public double PYG { get; set; }**

**public double QAR { get; set; }**

**public double RON { get; set; }**

**public double RSD { get; set; }**

**public double RUB { get; set; }**

**public double RWF { get; set; }**

**public double SAR { get; set; }**

**public double SBD { get; set; }**

**public double SCR { get; set; }**

**public double SDG { get; set; }**

**public double SEK { get; set; }**

**public double SGD { get; set; }**

**public double SHP { get; set; }**

**public double  SLL { get; set; }**

**public double SOS { get; set; }**

**public double SRD { get; set; }**

**public double STD { get; set; }**

**public double SVC { get; set; }**

**public double SYP { get; set; }**

**public double SZL { get; set; }**

**public double THB { get; set; }**

**public double TJS { get; set; }**

**public double TMT { get; set; }**

**public double TND { get; set; }**

**public double TOP { get; set; }**

**public double TRY { get; set; }**

**public double TTD { get; set; }**

**public double TWD { get; set; }**

**public double TZS { get; set; }**

**public double UAH { get; set; }**

**public double UGX { get; set; }**

**public double  USD { get; set; }**

**public double UYU { get; set; }**

**public double UZS { get; set; }**

**public double VEF { get; set; }**

**public double VND { get; set; }**

**public double VUV { get; set; }**

**public double WST { get; set; }**

**public double XAF { get; set; }**

**public double XAG { get; set; }**

**public double XAU { get; set; }**

**public double XCD { get; set; }**

**public double XDR { get; set; }**

**public double XOF { get; set; }**

**public double XPD { get; set; }**

**public double XPF { get; set; }**

**public double XPT { get; set; }**

**public double YER { get; set; }**

**public double ZAR { get; set; }**

**public double ZMK { get; set; }**

**public double ZMW { get; set; }**

**public double ZWL { get; set; }**

**}**

**}**

**using Newtonsoft.Json;**

**using System;**

**using System.Collections.Generic;**

**using System.Linq;**

**using System.Net.Http;**

**using System.Text;**

**using System.Threading.Tasks;**

**using Xamarin.Forms;**

**namespace XDivisasApi**

**{**

**public partial class MainPage : ContentPage**

**{**

**private ExchangeRates exchangeRates;**

**public MainPage()**

**{**

**InitializeComponent();**

**this.Padding = Device.OnPlatform(new Thickness(10, 20, 10, 10), new Thickness(10), new Thickness(10));**

**LoadPickers();**

**LoadRates();**

**convertButton.Clicked += ConvertButton\_Clicked;**

**}**

**private async void ConvertButton\_Clicked(object sender, EventArgs e)**

**{**

**if (string.IsNullOrEmpty(amountEntry.Text))**

**{**

**await DisplayAlert("Error", "Debes ingresar la cantidad a Convertir.", "Aceptar");**

**amountEntry.Focus();**

**return;**

**}**

**if (sourcePicker.SelectedIndex == -1)**

**{**

**await DisplayAlert("Error", "Debes seleccionar la Moneda Origen.", "Aceptar");**

**sourcePicker.Focus();**

**return;**

**}**

**if (targetPicker.SelectedIndex == -1)**

**{**

**await DisplayAlert("Error", "Debes seleccionar la Moneda Destino.", "Aceptar");**

**targetPicker.Focus();**

**return;**

**}**

**var amount = decimal.Parse(amountEntry.Text);**

**var sourceRate = GetRate(sourcePicker.SelectedIndex);**

**var targetRate = GetRate(targetPicker.SelectedIndex);**

**var amountConverted =  amount / Convert.ToDecimal(sourceRate) \* Convert.ToDecimal(targetRate);**

**covertedLabel.Text = $"{amount:N2} {sourcePicker.Items[sourcePicker.SelectedIndex]} = {amountConverted:N2} " +**

**$"{targetPicker.Items[targetPicker.SelectedIndex]}";**

**}**

**private object GetRate(int index)**

**{**

**switch (index)**

**{**

**case 0: return exchangeRates.rates.AED;**

**case 1: return exchangeRates.rates.AFN;**

**case 2: return exchangeRates.rates.ALL;**

**case 3: return exchangeRates.rates.AMD;**

**case 4: return exchangeRates.rates.ANG;**

**case 5: return exchangeRates.rates.AOA;**

**case 6: return exchangeRates.rates.ARS;**

**case 7: return exchangeRates.rates.AUD;**

**case 8: return exchangeRates.rates.AWG;**

**case 9: return exchangeRates.rates.AZN;**

**case 10: return exchangeRates.rates.BAM;**

**case 11: return exchangeRates.rates.BBD;**

**case 12: return exchangeRates.rates.BDT;**

**case 13: return exchangeRates.rates.BGN;**

**case 14: return exchangeRates.rates.BHD;**

**case 15: return exchangeRates.rates.BIF;**

**case 16: return exchangeRates.rates.BMD;**

**case 17: return exchangeRates.rates.BND;**

**case 18: return exchangeRates.rates.BOB;**

**case 19: return exchangeRates.rates.BRL;**

**case 20: return exchangeRates.rates.BSD;**

**case 21: return exchangeRates.rates.BTC;**

**case 22: return exchangeRates.rates.BTN;**

**case 23: return exchangeRates.rates.BWP;**

**case 24: return exchangeRates.rates.BYN;**

**case 25: return exchangeRates.rates.BYR;**

**case 26: return exchangeRates.rates.BZD;**

**case 27: return exchangeRates.rates.CAD;**

**case 28: return exchangeRates.rates.CDF;**

**case 29: return exchangeRates.rates.CHF;**

**case 30: return exchangeRates.rates.CLF;**

**case 31: return exchangeRates.rates.CLP;**

**case 32: return exchangeRates.rates.CNY;**

**case 33: return exchangeRates.rates.COP;**

**case 34: return exchangeRates.rates.CRC;**

**case 35: return exchangeRates.rates.CUC;**

**case 36: return exchangeRates.rates.CUP;**

**case 37: return exchangeRates.rates.CVE;**

**case 38: return exchangeRates.rates.CZK;**

**case 39: return exchangeRates.rates.DJF;**

**case 40: return exchangeRates.rates.DKK;**

**case 41: return exchangeRates.rates.DOP;**

**case 42: return exchangeRates.rates.DZD;**

**case 43: return exchangeRates.rates.EEK;**

**case 44: return exchangeRates.rates.EGP;**

**case 45: return exchangeRates.rates.ERN;**

**case 46: return exchangeRates.rates.ETB;**

**case 47: return exchangeRates.rates.EUR;**

**case 48: return exchangeRates.rates.FJD;**

**case 49: return exchangeRates.rates.FKP;**

**case 50: return exchangeRates.rates.GBP;**

**case 51: return exchangeRates.rates.GEL;**

**case 52: return exchangeRates.rates.GGP;**

**case 53: return exchangeRates.rates.GHS;**

**case 54: return exchangeRates.rates.GIP;**

**case 55: return exchangeRates.rates.GMD;**

**case 56: return exchangeRates.rates.GNF;**

**case 57: return exchangeRates.rates.GTQ;**

**case 58: return exchangeRates.rates.GYD;**

**case 59: return exchangeRates.rates.HKD;**

**case 60: return exchangeRates.rates.HNL;**

**case 61: return exchangeRates.rates.HRK;**

**case 62: return exchangeRates.rates.HTG;**

**case 63: return exchangeRates.rates.HUF;**

**case 64: return exchangeRates.rates.IDR;**

**case 65: return exchangeRates.rates.ILS;**

**case 66: return exchangeRates.rates.IMP;**

**case 67: return exchangeRates.rates.INR;**

**case 68: return exchangeRates.rates.IQD;**

**case 69: return exchangeRates.rates.IRR;**

**case 70: return exchangeRates.rates.ISK;**

**case 71: return exchangeRates.rates.JEP;**

**case 72: return exchangeRates.rates.JMD;**

**case 73: return exchangeRates.rates.JOD;**

**case 74: return exchangeRates.rates.JPY;**

**case 75: return exchangeRates.rates.KES;**

**case 76: return exchangeRates.rates.KGS;**

**case 77: return exchangeRates.rates.KHR;**

**case 78: return exchangeRates.rates.KMF;**

**case 79: return exchangeRates.rates.KPW;**

**case 80: return exchangeRates.rates.KRW;**

**case 81: return exchangeRates.rates.KWD;**

**case 82: return exchangeRates.rates.KYD;**

**case 83: return exchangeRates.rates.KZT;**

**case 84: return exchangeRates.rates.LAK;**

**case 85: return exchangeRates.rates.LBP;**

**case 86: return exchangeRates.rates.LKR;**

**case 87: return exchangeRates.rates.LRD;**

**case 88: return exchangeRates.rates.LSL;**

**case 89: return exchangeRates.rates.LTL;**

**case 90: return exchangeRates.rates.LVL;**

**case 91: return exchangeRates.rates.LYD;**

**case 92: return exchangeRates.rates.MAD;**

**case 93: return exchangeRates.rates.MDL;**

**case 94: return exchangeRates.rates.MGA;**

**case 95: return exchangeRates.rates.MKD;**

**case 96: return exchangeRates.rates.MMK;**

**case 97: return exchangeRates.rates.MNT;**

**case 98: return exchangeRates.rates.MOP;**

**case 99: return exchangeRates.rates.MRO;**

**case 100: return exchangeRates.rates.MTL;**

**case 101: return exchangeRates.rates.MUR;**

**case 102: return exchangeRates.rates.MVR;**

**case 103: return exchangeRates.rates.MWK;**

**case 104: return exchangeRates.rates.MXN;**

**case 105: return exchangeRates.rates.MYR;**

**case 106: return exchangeRates.rates.MZN;**

**case 107: return exchangeRates.rates.NAD;**

**case 108: return exchangeRates.rates.NGN;**

**case 109: return exchangeRates.rates.NIO;**

**case 110: return exchangeRates.rates.NOK;**

**case 111: return exchangeRates.rates.NPR;**

**case 112: return exchangeRates.rates.NZD;**

**case 113: return exchangeRates.rates.OMR;**

**case 114: return exchangeRates.rates.PAB;**

**case 115: return exchangeRates.rates.PEN;**

**case 116: return exchangeRates.rates.PGK;**

**case 117: return exchangeRates.rates.PHP;**

**case 118: return exchangeRates.rates.PKR;**

**case 119: return exchangeRates.rates.PLN;**

**case 120: return exchangeRates.rates.PYG;**

**case 121: return exchangeRates.rates.QAR;**

**case 122: return exchangeRates.rates.RON;**

**case 123: return exchangeRates.rates.RSD;**

**case 124: return exchangeRates.rates.RUB;**

**case 125: return exchangeRates.rates.RWF;**

**case 126: return exchangeRates.rates.SAR;**

**case 127: return exchangeRates.rates.SBD;**

**case 128: return exchangeRates.rates.SCR;**

**case 129: return exchangeRates.rates.SDG;**

**case 130: return exchangeRates.rates.SEK;**

**case 131: return exchangeRates.rates.SGD;**

**case 132: return exchangeRates.rates.SHP;**

**case 133: return exchangeRates.rates.SLL;**

**case 134: return exchangeRates.rates.SOS;**

**case 135: return exchangeRates.rates.SRD;**

**case 136: return exchangeRates.rates.STD;**

**case 137: return exchangeRates.rates.SVC;**

**case 138: return exchangeRates.rates.SYP;**

**case 139: return exchangeRates.rates.SZL;**

**case 140: return exchangeRates.rates.THB;**

**case 141: return exchangeRates.rates.TJS;**

**case 142: return exchangeRates.rates.TMT;**

**case 143: return exchangeRates.rates.TND;**

**case 144: return exchangeRates.rates.TOP;**

**case 145: return exchangeRates.rates.TRY;**

**case 146: return exchangeRates.rates.TTD;**

**case 147: return exchangeRates.rates.TWD;**

**case 148: return exchangeRates.rates.TZS;**

**case 149: return exchangeRates.rates.UAH;**

**case 150: return exchangeRates.rates.UGX;**

**case 151: return exchangeRates.rates.USD;**

**case 152: return exchangeRates.rates.UYU;**

**case 153: return exchangeRates.rates.UZS;**

**case 154: return exchangeRates.rates.VEF;**

**case 155: return exchangeRates.rates.VND;**

**case 156: return exchangeRates.rates.VUV;**

**case 157: return exchangeRates.rates.WST;**

**case 158: return exchangeRates.rates.XAF;**

**case 159: return exchangeRates.rates.XAG;**

**case 160: return exchangeRates.rates.XAU;**

**case 161: return exchangeRates.rates.XCD;**

**case 162: return exchangeRates.rates.XDR;**

**case 163: return exchangeRates.rates.XOF;**

**case 164: return exchangeRates.rates.XPD;**

**case 165: return exchangeRates.rates.XPF;**

**case 166: return exchangeRates.rates.XPT;**

**case 167: return exchangeRates.rates.YER;**

**case 168: return exchangeRates.rates.ZAR;**

**case 169: return exchangeRates.rates.ZMK;**

**case 170: return exchangeRates.rates.ZMW;**

**case 171: return exchangeRates.rates.ZWL;**

**default:**

**return 1;**

**}**

**}**

**private async void LoadRates()**

**{**

**waitActivityIndicator.IsRunning = true;**

**//Consu,ir el servicio(api)**

**try**

**{**

**var client = new HttpClient();**

**client.BaseAddress = new Uri("https://openexchangerates.org");**

**var url = "/api/latest.json?app\_id=7ce0119051774ef5a478eefc8f8070d1";**

**var response = await client.GetAsync(url);**

**if (!response.IsSuccessStatusCode)**

**{**

**await DisplayAlert("Error", response.StatusCode.ToString(), "Aceptar");**

**waitActivityIndicator.IsRunning = false;**

**convertButton.IsEnabled = false;**

**return;**

**}**

**//Aqui recibo lo que llegue la respuesta de y las convieto en string:**

**var result = await response.Content.ReadAsStringAsync();**

**exchangeRates = JsonConvert.DeserializeObject<ExchangeRates>(result);**

**}**

**catch (Exception ex)**

**{**

**await DisplayAlert("Error", ex.Message, "Aceptar");**

**waitActivityIndicator.IsRunning = false;**

**convertButton.IsEnabled = false;**

**return;**

**}**

**convertButton.IsEnabled = true;**

**waitActivityIndicator.IsRunning = false;**

**}**

**private void LoadPickers()**

**{**

**LoadPicker(sourcePicker);**

**LoadPicker(targetPicker);**

**}**

**private void LoadPicker(Picker picker)**

**{**

**picker.Items.Add("AED");**

**picker.Items.Add("AFN");**

**picker.Items.Add("ALL");**

**picker.Items.Add("AMD");**

**picker.Items.Add("ANG");**

**picker.Items.Add("AOA");**

**picker.Items.Add("ARS");**

**picker.Items.Add("AUD");**

**picker.Items.Add("AWG");**

**picker.Items.Add("AZN");**

**picker.Items.Add("BAM");**

**picker.Items.Add("BBD");**

**picker.Items.Add("BDT");**

**picker.Items.Add("BGN");**

**picker.Items.Add("BHD");**

**picker.Items.Add("BIF");**

**picker.Items.Add("BMD");**

**picker.Items.Add("BND");**

**picker.Items.Add("BOB");**

**picker.Items.Add("BRL");**

**picker.Items.Add("BSD");**

**picker.Items.Add("BTC");**

**picker.Items.Add("BTN");**

**picker.Items.Add("BWP");**

**picker.Items.Add("BYN");**

**picker.Items.Add("BYR");**

**picker.Items.Add("BZD");**

**picker.Items.Add("CAD");**

**picker.Items.Add("CDF");**

**picker.Items.Add("CHF");**

**picker.Items.Add("CLF");**

**picker.Items.Add("CLP");**

**picker.Items.Add("CNY");**

**picker.Items.Add("COP");**

**picker.Items.Add("CRC");**

**picker.Items.Add("CUC");**

**picker.Items.Add("CUP");**

**picker.Items.Add("CVE");**

**picker.Items.Add("CZK");**

**picker.Items.Add("DJF");**

**picker.Items.Add("DKK");**

**picker.Items.Add("DOP");**

**picker.Items.Add("DZD");**

**picker.Items.Add("EEK");**

**picker.Items.Add("EGP");**

**picker.Items.Add("ERN");**

**picker.Items.Add("ETB");**

**picker.Items.Add("EUR");**

**picker.Items.Add("FJD");**

**picker.Items.Add("FKP");**

**picker.Items.Add("GBP");**

**picker.Items.Add("GEL");**

**picker.Items.Add("GGP");**

**picker.Items.Add("GHS");**

**picker.Items.Add("GIP");**

**picker.Items.Add("GMD");**

**picker.Items.Add("GNF");**

**picker.Items.Add("GTQ");**

**picker.Items.Add("GYD");**

**picker.Items.Add("HKD");**

**picker.Items.Add("HNL");**

**picker.Items.Add("HRK");**

**picker.Items.Add("HTG");**

**picker.Items.Add("HUF");**

**picker.Items.Add("IDR");**

**picker.Items.Add("ILS");**

**picker.Items.Add("IMP");**

**picker.Items.Add("INR");**

**picker.Items.Add("IQD");**

**picker.Items.Add("IRR");**

**picker.Items.Add("ISK");**

**picker.Items.Add("JEP");**

**picker.Items.Add("JMD");**

**picker.Items.Add("JOD");**

**picker.Items.Add("JPY");**

**picker.Items.Add("KES");**

**picker.Items.Add("KGS");**

**picker.Items.Add("KHR");**

**picker.Items.Add("KMF");**

**picker.Items.Add("KPW");**

**picker.Items.Add("KRW");**

**picker.Items.Add("KWD");**

**picker.Items.Add("KYD");**

**picker.Items.Add("KZT");**

**picker.Items.Add("LAK");**

**picker.Items.Add("LBP");**

**picker.Items.Add("LKR");**

**picker.Items.Add("LRD");**

**picker.Items.Add("LSL");**

**picker.Items.Add("LTL");**

**picker.Items.Add("LVL");**

**picker.Items.Add("LYD");**

**picker.Items.Add("MAD");**

**picker.Items.Add("MDL");**

**picker.Items.Add("MGA");**

**picker.Items.Add("MKD");**

**picker.Items.Add("MMK");**

**picker.Items.Add("MNT");**

**picker.Items.Add("MOP");**

**picker.Items.Add("MRO");**

**picker.Items.Add("MTL");**

**picker.Items.Add("MUR");**

**picker.Items.Add("MVR");**

**picker.Items.Add("MWK");**

**picker.Items.Add("MXN");**

**picker.Items.Add("MYR");**

**picker.Items.Add("MZN");**

**picker.Items.Add("NAD");**

**picker.Items.Add("NGN");**

**picker.Items.Add("NIO");**

**picker.Items.Add("NOK");**

**picker.Items.Add("NPR");**

**picker.Items.Add("NZD");**

**picker.Items.Add("OMR");**

**picker.Items.Add("PAB");**

**picker.Items.Add("PEN");**

**picker.Items.Add("PGK");**

**picker.Items.Add("PHP");**

**picker.Items.Add("PKR");**

**picker.Items.Add("PLN");**

**picker.Items.Add("PYG");**

**picker.Items.Add("QAR");**

**picker.Items.Add("RON");**

**picker.Items.Add("RSD");**

**picker.Items.Add("RUB");**

**picker.Items.Add("RWF");**

**picker.Items.Add("SAR");**

**picker.Items.Add("SBD");**

**picker.Items.Add("SCR");**

**picker.Items.Add("SDG");**

**picker.Items.Add("SEK");**

**picker.Items.Add("SGD");**

**picker.Items.Add("SHP");**

**picker.Items.Add("SLL");**

**picker.Items.Add("SOS");**

**picker.Items.Add("SRD");**

**picker.Items.Add("STD");**

**picker.Items.Add("SVC");**

**picker.Items.Add("SYP");**

**picker.Items.Add("SZL");**

**picker.Items.Add("THB");**

**picker.Items.Add("TJS");**

**picker.Items.Add("TMT");**

**picker.Items.Add("TND");**

**picker.Items.Add("TOP");**

**picker.Items.Add("TRY");**

**picker.Items.Add("TTD");**

**picker.Items.Add("TWD");**

**picker.Items.Add("TZS");**

**picker.Items.Add("UAH");**

**picker.Items.Add("UGX");**

**picker.Items.Add("USD");**

**picker.Items.Add("UYU");**

**picker.Items.Add("UZS");**

**picker.Items.Add("VEF");**

**picker.Items.Add("VND");**

**picker.Items.Add("VUV");**

**picker.Items.Add("WST");**

**picker.Items.Add("XAF");**

**picker.Items.Add("XAG");**

**picker.Items.Add("XAU");**

**picker.Items.Add("XCD");**

**picker.Items.Add("XDR");**

**picker.Items.Add("XOF");**

**picker.Items.Add("XPD");**

**picker.Items.Add("XPF");**

**picker.Items.Add("XPT");**

**picker.Items.Add("YER");**

**picker.Items.Add("ZAR");**

**picker.Items.Add("ZMK");**

**picker.Items.Add("ZMW");**

**picker.Items.Add("ZWL");**

**}**

**}**

**}**