

# **ANÁLISE COMPARATIVA DE FERRAMENTAS DE DESENVOLVIMENTO DE APLICATIVOS MÓVEIS MULTIPLATAFORMA**

Aluno: Nikson Barth

Orientador: Marcel Hugo

# Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Questionário de Avaliação
- Especificação dos Protótipos
- Implementação
- Resultados e Discussões
- Conclusões e Sugestões

# Introdução

- Porque aplicativos multiplataforma?
- Utilizar uma ferramenta de desenvolvimento multiplataforma móvel?
- Qual ferramenta utilizar?

# Objetivos

- Realizar uma análise comparativa entre ferramentas de desenvolvimento móvel multiplataforma.
  - Definir os critérios de avaliação
  - Especificar os aplicativos
  - Desenvolver os aplicativos especificados
  - Realizar a análise observando os critérios de avaliação.

# Fundamentação Teórica

- **Desenvolvimento móvel multiplataforma.**
- **Xamarin**
- **PhoneGap**
- **Delphi XE5**

# Fundamentação Teórica

- **NBR ISO/IEC 25000 SQuaRE**
  - medidas de qualidade do processo;
  - medidas de qualidade internas;
  - medidas de qualidade externas;
  - medidas de qualidade do software em uso.
- ISO/IEC 2500n – Divisão gestão de qualidade;
- ISO/IEC 2501n – Divisão modelo de qualidade;
- ISO/IEC 2502n – Divisão medição da qualidade;
- ISO/IEC 2503n – Divisão requisitos de qualidade;
- ISO/IEC 2504n – Divisão avaliação da qualidade.

# Trabalhos Correlatos

- **Análise comparativa entre Groovy e Java (REZENDE, 2011)**
- **Desenvolvimento de aplicativo móvel multiplataforma integrado ao sistema de alerta de cheias da bacia do Itajaí (SOUZA, 2012)**

# Questionário de avaliação

Quadro 2 – Questionário de avaliação

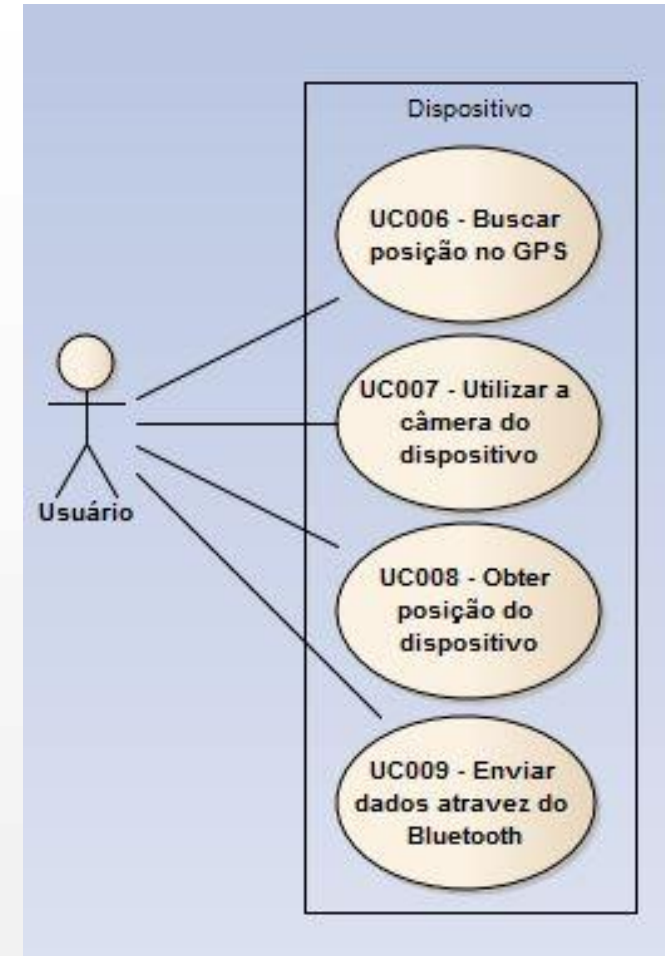
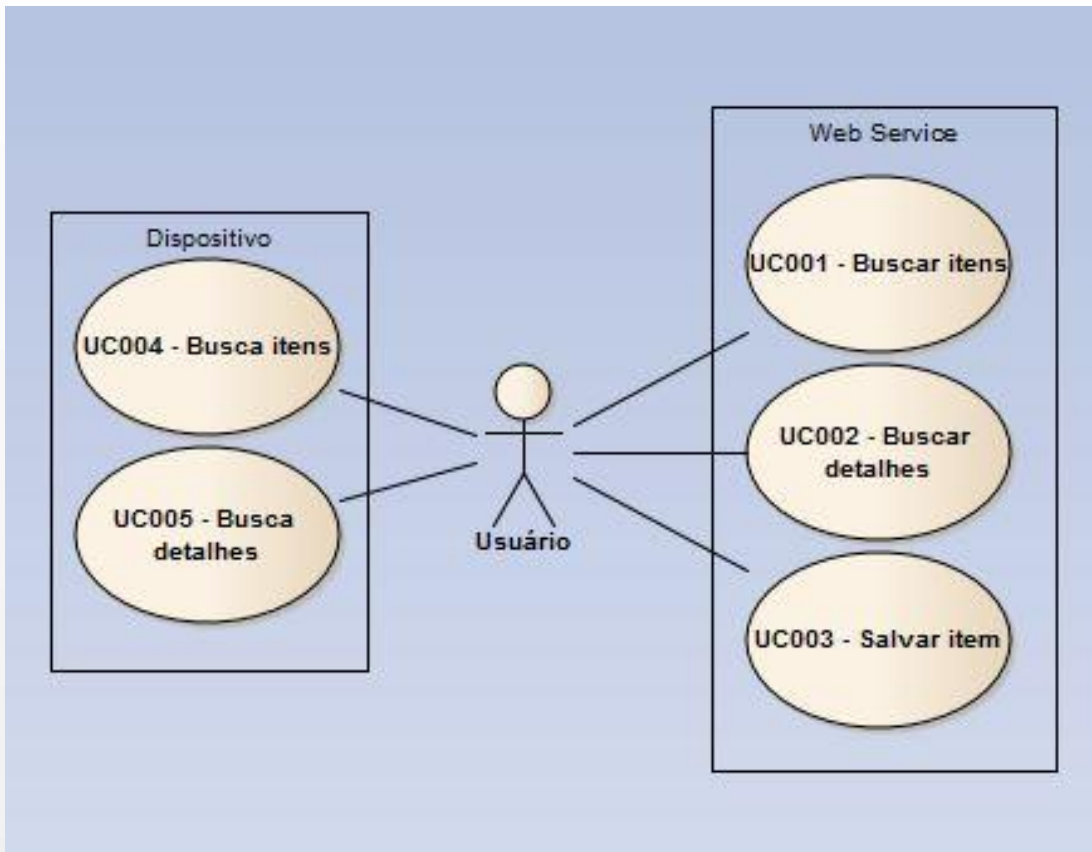
Adequação funcional	Completude funcional	Compila o mesmo código para as plataformas <u>Android</u> e <u>iOS</u> ?
		Compila no emulador?
		Compila no dispositivo?
		Executa a construção rapidamente?
		Possui suporte e documentação?
	<u>Corretude funcional</u>	A interface gráfica corresponde ao apresentado no dispositivo? O código funciona corretamente?
	Funcionalidade apropriada	O software possui <u>IntelliSense</u> ? O software possui uma interface de modelagem?
Confiabilidade	Maturidade	Funciona sem problemas durante a codificação?
	Disponibilidade	Quando necessário, o software está operacional?
	Tolerância a falhas	Se algo inesperado ocorrer durante a instalação no dispositivo, o software continua funcionando?
		Se algo inesperado ocorrer durante a instalação no emulador, o software continua funcionando?
	<u>Recuperabilidade</u>	Quando ocorre um defeito, o software consegue recuperar os dados que estavam sendo trabalhados?
Usabilidade	Apropriação reconhecível	O software é apropriado para o desenvolvimento dos aplicativos especificados?
	Inteligibilidade	Os conceitos-chave do software são de fácil aprendizagem?
	Operabilidade	É fácil instalar o aplicativo no emulador?
		É fácil instalar o aplicativo no dispositivo?
		É intuitiva a criação de projetos?
	Proteção contra erro de usuário	O software auxilia o usuário a não cometer erros?
	Estética de interface com usuário	A interface do software proporciona prazer e interação satisfatórios?
Acessibilidade	O software foi projetado para atender usuários com necessidades especiais?	



# Questionário de avaliação

Eficiência de desempenho	Comportamento em relação ao tempo	Tempo necessário para compilar o projeto.
		Tempo necessário para instalar no dispositivo.
		Tempo necessário para instalar no emulador.
	Utilização de recursos	Recursos utilizados no computador.
		Recursos utilizados com emulador.
		Recursos utilizados no dispositivo.
Capacidade	O software atendeu todos os requisitos especificados nos aplicativos?	
Compatibilidade	Coexistência	O software é capaz de coexistir com outros softwares do computador?
		O software é capaz de coexistir com outros softwares do dispositivo?
Portabilidade	Interoperabilidade	O software é capaz de interagir com o Mac OS? (Necessário para compilar para iOS)
		O software funciona na plataforma Windows?
	Adaptabilidade	O software funciona na plataforma Mac OS?
		O software é facilmente instalado?
Efetividade	Efetividade	O software pode substituir outro?
		O software permite o desenvolvimento dos aplicativos especificados por completo?
Eficiência	Eficiência	A compilação é rápida? Ou faz o usuário esperar?
Satisfação	Utilidade	O usuário possui satisfação ao utilizar o software?
	Prazer	O usuário sente prazer em usar o software?
	Conforto	O uso de funções e componentes simples é de fácil acesso?
Cobertura de contexto	Completude de contexto	O software pode ser usado com efetividade, eficiência, sem riscos e com satisfação em todos os contextos especificados?
	Flexibilidade	O software pode ser usado com efetividade, eficiência, sem riscos e com satisfação em todos os contextos especificados?

# Especificação dos protótipos



# Especificação dos protótipos

- RF001 – Buscar registros em um Web Service
  - RF002 – Inserir registro no banco de dados do dispositivo móvel
  - RF003 – Buscar registros no banco de dados do dispositivo móvel
  - RF004 – Buscar geolocalização do dispositivo através do GPS
  - RF005 – Realizar o envio de dados através do recurso Bluetooth do dispositivo móvel
  - RF006 – Buscar a posição do dispositivo móvel utilizando os recursos acelerômetro e giroscópio.
  - RF007 – Efetuar a captura de uma imagem utilizando a câmera do dispositivo.
- 
- RNF001 – Utilizar ferramentas de desenvolvimento móvel multiplataforma para desenvolver os aplicativos que serão utilizados nas análises comparativas
  - RNF002 – Utilizar a norma NBR ISO/IEC 25000 SQuaRE para elencar os critérios que serão avaliados na análise comparativa
  - RNF003 – Efetuar a análise comparativa entre as ferramentas de desenvolvimento móvel multiplataforma.

# Implementação

- Aplicativo 1
  - Banco de dados
    - Delphi XE 5:



```
Form1.SQLQuery.ParamByName('nome').AsString := txt;  
Form1.SQLQuery.ExecSQL();
```

# Implementação

- Aplicativo 1
  - Banco de dados
    - Xamarin:

```
SQLiteConnection  
    conn = new SQLiteConnection(System.IO.Path.Combine(folder, "TCC_Receitas.db"));  
conn.Insert(receita);  
conn.Insert(receita.Ingredientes);  
conn.Insert(receita.Preparo);
```

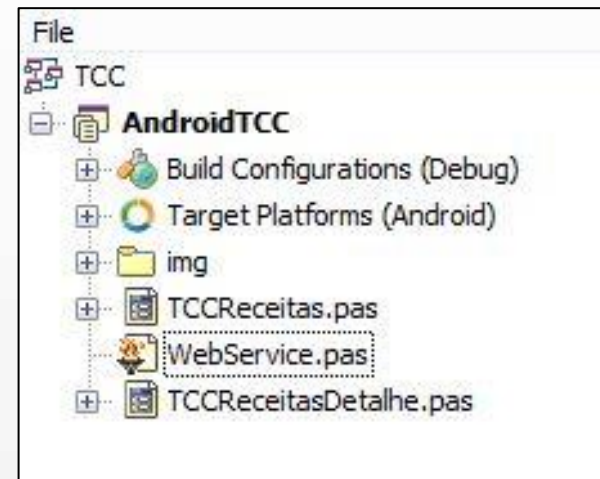
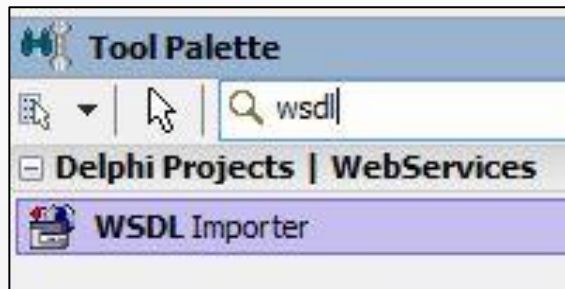
# Implementação

- Aplicativo 1
  - Banco de dados
    - PhoneGap:

```
function criarTabelas(tx) {  
  tx.executeSql('CREATE TABLE IF NOT EXISTS RECEITAS (id unique, nome, imagem)');  
  tx.executeSql('CREATE TABLE IF NOT EXISTS INGREDIENTES (id unique, id_receita, descricao)');  
  tx.executeSql('CREATE TABLE IF NOT EXISTS PREPARO (id unique, id_receita, descricao)');  
}  
  
function buscarReceitas(tx) {  
  tx.executeSql('SELECT * FROM RECEITAS WHERE NOME LIKE \'%\' + $('#TxtBuscar').val() + \'%\'' ,  
    [], buscaSucesso, error);  
}
```

# Implementação

- Aplicativo 1
  - Web Service
    - DelphiXE5:



```
resultado := (HTTPRIOReceitas as WebServiceSoap).GetReceitasResumo(TxtEdit.Text);
```

# Implementação

- Aplicativo 1
  - Web Service
    - Xamarin:

```
public static WebService.WebService GetWebService
{
    get
    {
        WebService.WebService web = new WebService.WebService();
        web.Url = "http://192.168.0.102/WebService.asmx";
        return web;
    }
}
```

```
Util.GetWebService.GetReceitasResumo(txt.Text)
```



# Implementação

- Aplicativo 1
  - Web Service
    - PhoneGap:

```
function BuscaReceitasWeb(ids) {
    var response = "";
    $.ajax({
        type: "GET",
        url: "http://192.168.0.100:80/WebService.asmx/GetReceitasResumoJQuery",
        data: { "param": $('#TxtBuscar').val() },
        contentType: "text/plain; charset=utf-8",
        dataType: "text",
        success: function (response) {
            var json_obj = $.parseJSON(response);
            var output = "";
            for (var i in json_obj) {
                var x = ids.indexOf(json_obj[i].Id);
                if (x == -1) {
                    output += "<li><a href=\javascript:goDetail(" + json_obj[i].Id + ",0)>"
                        + json_obj[i].Nome + "</a></li>";
                }
            }
            $('#details').append(output);
        },
        error: function (response) {
            alert(response);
        }
    });
    return true;
}
```

# Implementação

- Aplicativo 2
  - Acelerômetro
    - DelphiXE5:



```
for LProp in MotionSensor1.Sensor.AvailableProperties do
begin
  case LProp of
    TCustomMotionSensor.TProperty.AccelerationX:
      begin
        lbAccelerationX.Text := Format('Acceleration X: %6.2f', [MotionSensor1.Sensor.AccelerationX]);
      end;
    TCustomMotionSensor.TProperty.AccelerationY:
      begin
        lbAccelerationY.Text := Format('Acceleration Y: %6.2f', [MotionSensor1.Sensor.AccelerationY]);
      end;
    TCustomMotionSensor.TProperty.AccelerationZ:
      begin
        lbAccelerationZ.Text := Format('Acceleration Z: %6.2f', [MotionSensor1.Sensor.AccelerationZ]);
      end;
  end;
end;
```

# Implementação

- Aplicativo 2
  - Acelerômetro
    - Xamarin (iOS):

```
CMMotionManager _motionManager = new CMMotionManager ();  
_motionManager.StartAccelerometerUpdates (NSOperationQueue.CurrentQueue, (data, error) =>  
{  
    this.lblX.Text = data.Acceleration.X.ToString ("0.00000000");  
    this.lblY.Text = data.Acceleration.Y.ToString ("0.00000000");  
    this.lblZ.Text = data.Acceleration.Z.ToString ("0.00000000");  
});
```

# Implementação

- Aplicativo 2
  - Acelerômetro
    - Xamarin (Android):

```
protected override void OnCreate(Bundle bundle)
{
    base.OnCreate(bundle);

    SetContentView(Resource.Layout.Main);
    _sensorManager = (SensorManager)GetSystemService(Context.SensorService);
    _sensorTextView = FindViewById<TextView>(Resource.Id.accelerometer_text);
}

public void OnSensorChanged(SensorEvent e)
{
    lock (_syncLock)
    {
        var text = new StringBuilder("x = ")
            .Append(e.Values[0])
            .Append(", y=")
            .Append(e.Values[1])
            .Append(", z=")
            .Append(e.Values[2]);
        _sensorTextView.Text = text.ToString();
    }
}

protected override void OnResume()
{
    base.OnResume();
    _sensorManager.RegisterListener(this, _sensorManager.GetDefaultSensor(SensorType.Accelerometer), SensorDelay.Ui);
}
```

# Implementação

- Aplicativo 2
  - Acelerômetro
    - PhoneGap (Android):

```
document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {
    navigator.accelerometer.getCurrentAcceleration(onSuccess, onError);
}

function onSuccess(acceleration) {
    alert('Acceleration X: ' + acceleration.x + '\n' +
        'Acceleration Y: ' + acceleration.y + '\n' +
        'Acceleration Z: ' + acceleration.z + '\n' +
        'Timestamp: ' + acceleration.timestamp + '\n');
}

function onError() {
    alert('onError!');
}
```

# Implementação

- Aplicativo 2
  - Acelerômetro
    - PhoneGap (iOS e Android):

```
var watchID = null;

document.addEventListener("deviceready", onDeviceReady, false);

function onDeviceReady() {
    startWatch();
}

function startWatch() {
    var options = { frequency: 3000 };
    watchID = navigator.accelerometer.watchAcceleration(onSuccess, onError, options);
}

function stopWatch() {
    if (watchID) {
        navigator.accelerometer.clearWatch(watchID);
        watchID = null;
    }
}

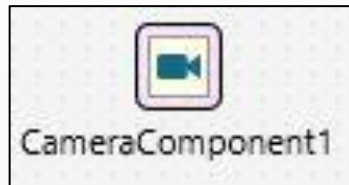
function onSuccess(acceleration) {
    var element = document.getElementById('accelerometer');
    element.innerHTML = 'Acceleration X: ' + acceleration.x + '<br />' +
        'Acceleration Y: ' + acceleration.y + '<br />' +
        'Acceleration Z: ' + acceleration.z + '<br />' +
        'Timestamp: ' + acceleration.timestamp + '<br />';
}

function onError() {
    alert('onError!');
}
```



# Implementação

- Aplicativo 2
  - Câmera
    - DelphiXE5:



```
//Liga a câmera
CameraComponent1.Active := True;

//Desliga a câmera
CameraComponent1.Active := False;

//Captura a imagem e atribui a um componente TImage
procedure TCameraComponentForm.CameraComponent1SampleBufferReady(
Sender: TObject; const ATime: Integer);
begin
    CameraComponent1.SampleBufferToBitmap(imgCameraView.Bitmap, True);
    imgCameraView.Width := imgCameraView.Bitmap.Width;
    imgCameraView.Height := imgCameraView.Bitmap.Height;
end;
```

# Implementação

- Aplicativo 2
  - Câmera
    - Xamarin (iOS):

```
TweetStation.Camera.TakePicture (this, (obj) =>{
    var photo = obj.ValueForKey(new NSString("UIImagePickerControllerOriginalImage")) as UIImage;
    var meta = obj.ValueForKey(new NSString("UIImagePickerControllerMediaMetadata")) as NSDictionary;
    ALAssetsLibrary library = new ALAssetsLibrary();
    library.WriteImageToSavedPhotosAlbum (photo.CGImage, meta, (assetUrl, error) =>{
        Console.WriteLine ("assetUrl:"+assetUrl);
    });
});
```



# Implementação

- Aplicativo 2
  - Câmera
    - Xamarin (Android):

```
private void TakeAPicture(object sender, EventArgs EventArgs)
{
    Intent intent = new Intent(MediaStore.ActionImageCapture);
    _file = new File(_dir, String.Format("myPhoto_{0}.jpg", Guid.NewGuid()));
    intent.PutExtra(MediaStore.ExtraOutput, Android.Net.Uri.FromFile(_file));
    StartActivityForResult(intent, 0);
}
protected override void OnActivityResult(int requestCode, Result resultCode, Intent data)
{
    base.OnActivityResult(requestCode, resultCode, data);
    Intent mediaScanIntent = new Intent(Intent.ActionMediaScannerScanFile);
    Android.Net.Uri contentUri = Android.Net.Uri.FromFile(_file);
    mediaScanIntent.SetData(contentUri);
    SendBroadcast(mediaScanIntent);
    int height = _imageView.Height;
    int width = Resources.DisplayMetrics.WidthPixels;
    using (Bitmap bitmap = _file.Path.LoadAndResizeBitmap(width, height))
    {
        _imageView.SetImageBitmap(bitmap);
    }
}
```

# Implementação

- Aplicativo 2
  - Câmera
    - PhoneGap:

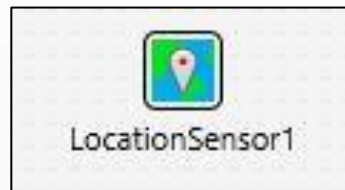
```
function capturePhoto() {  
    navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 50,  
        destinationType: destinationType.DATA_URL });  
}
```

```
function onPhotoDataSuccess(imageData) {  
    var smallImage = document.getElementById('smallImage');  
    smallImage.style.display = 'block';  
  
    smallImage.src = "data:image/jpeg;base64," + imageData;  
}
```

```
<button onclick="getPhoto(pictureSource.PHOTOLIBRARY);">From Photo Library</button>
```

# Implementação

- Aplicativo 2
  - GPS
    - DelphiXE5:



```
procedure TLocationForm.LocationSensor1LocationChanged(Sender: TObject;
const OldLocation, NewLocation: TLocationCoord2D);
const
  LGoogleMapsURL: String = 'https://maps.google.com/maps?q=%s,%s&output=embed';
begin
  lblLatitude.Text := 'Latitude: ' + NewLocation.Latitude.ToString;
  lblLongitude.Text := 'Longitude: ' + NewLocation.Longitude.ToString;
  WebBrowser1.Navigate(Format(LGoogleMapsURL, [NewLocation.Latitude.ToString, NewLocation.Longitude.ToString]));
end;
```

# Implementação

- Aplicativo 2
  - GPS
    - Xamarin (iOS):

```
var LocMgr = new CLLocationManager();

//Iniciar a localização
if (CLLocationManager.LocationServicesEnabled) {
    LocMgr.StartMonitoringSignificantLocationChanges ();
} else {
    Console.WriteLine ("GPS não está ativo");
}

//Dispara quando feita a localização
LocMgr.LocationsUpdated += (o, e) => Console.WriteLine ("Localização atualizada");

//Para a localização
LocMgr.StopMonitoringSignificantLocationChanges ();
```

# Implementação

- Aplicativo 2
  - GPS
    - Xamarin (Android):

```
void InitializeLocationManager()
{
    _locationManager = (LocationManager)SystemService(LocationService);
    Criteria criteriaForLocationService = new Criteria
    {
        Accuracy = Accuracy.Fine
    };
    IList<string> acceptableLocationProviders = _locationManager.GetProviders(criteriaForLocationService, true);

    if (acceptableLocationProviders.Any())
    {
        _locationProvider = acceptableLocationProviders.First();
    }
    else
    {
        _locationProvider = String.Empty;
    }
}

public void OnLocationChanged(Location location)
{
    _currentLocation = location;
    if (_currentLocation == null)
    {
        _locationText.Text = "Unable to determine your location.";
    }
    else
    {
        _locationText.Text = String.Format("{0},{1}", _currentLocation.Latitude, _currentLocation.Longitude);
    }
}
```

# Implementação

- Aplicativo 2
  - GPS
    - PhoneGap:

```
function onDeviceReady() {
    navigator.geolocation.getCurrentPosition(onSuccess, onError);
}

function onSuccess(position) {
    var element = document.getElementById('geolocation');
    element.innerHTML = 'Latitude: ' + position.coords.latitude + '<br />' +
        'Longitude: ' + position.coords.longitude + '<br />' +
        'Altitude: ' + position.coords.altitude + '<br />' +
        'Accuracy: ' + position.coords.accuracy + '<br />' +
        'Altitude Accuracy: ' + position.coords.altitudeAccuracy + '<br />' +
        'Heading: ' + position.coords.heading + '<br />' +
        'Speed: ' + position.coords.speed + '<br />' +
        'Timestamp: ' + position.timestamp + '<br />';
}
```



# Resultados e Discussões

Tabela 1 – Tempo em min de instalação no dispositivo Android

	<u>Delphi XE5</u>	<u>Xamarin</u>	<u>PhoneGap</u>
Aplicativo 1	01:40:508	00:13:990	00:05:844
Aplicativo 2	01:56:259	00:14:281	00:06:327

Tabela 2 – Tempo em min de instalação no emulador Android

	<u>Delphi XE5</u>	<u>Xamarin</u>	<u>PhoneGap</u>
Aplicativo 1	02:45:740	02:30:771	00:13:716
Aplicativo 2	03:20:165	03:11:005	00:16:846

Tabela 3 – Tempo em min de instalação no emulador iOS

	<u>Delphi XE5</u>	<u>Xamarin</u>	<u>PhoneGap</u>
Aplicativo 1	01:41:249	00:28:108	00:04:293
Aplicativo 2	02:02:752	00:35:468	00:05:346

Tabela 4 – Tempo em min de compilação dos projetos

	<u>Delphi XE5</u>	<u>Xamarin</u>	<u>PhoneGap</u>
Aplicativo 1	01:02:473	00:05:532	00:02:574
Aplicativo 2	01:12:165	00:03:372	00:03:413

Tabela 5 – Recurso utilizando no dispositivo Android

	<u>Delphi XE5</u>	<u>Xamarin</u>	<u>PhoneGap</u>
Aplicativo 1	RAM: 38,52mb, CPU 1,08%	RAM: 14,97mb, CPU 0,01%	RAM: 27,52mb, CPU 1,00%
Aplicativo 2	RAM: 45,30mb, CPU 1,51%	RAM: 18,01mb, CPU 0,01%	RAM: 39,41mb, CPU 1,33%

Tabela 6 – Recurso utilizando pelas ferramentas no computador

<u>Delphi XE5</u>	<u>Xamarin - Visual Studio 2012</u>	<u>PhoneGap - Eclipse</u>
RAM: 352,3mb, CPU 3,5%	RAM: 191,6mb, CPU 0,05%	RAM: 345,7mb, CPU 0,01%

# Resultados e Discussões

Quadro 27 – Questionário de avaliação

		Delphi XE5			Xamarin			PhoneGap		
		0	5	10	0	5	10	0	5	10
Adequação funcional	Compila o mesmo código para as plataformas <u>Android</u> e <u>iOS</u> ?			X	X					X
	Compila no emulador?			X			X			X
	Compila no dispositivo?			X			X			X
	Executa a construção rapidamente?		X				X			X
	Possui suporte e documentação?			X	X				X	
	A interface gráfica corresponde ao apresentado no dispositivo?			X			X		X	
	O código funciona corretamente?			X			X			X
	O software possui <u>IntelliSense</u> ?			X			X	X		
	O software possui uma interface de modelagem?			X			X	X		
				85			75			60
Confiabilidade	Funciona sem problemas durante a codificação?		X				X			X
	Quando necessário, o software está operacional?			X			X			X
	Se algo inesperado ocorrer durante a instalação no dispositivo, o software continua funcionando?		X				X			X
	Se algo inesperado ocorrer durante a instalação no emulador, o software continua funcionando?		X				X			X
	Quando ocorre um defeito, o software consegue recuperar os dados que estavam sendo trabalhados?			X			X			X
				35			40			50



# Resultados e Discussões

Usabilidade	O software é apropriado para o desenvolvimento dos aplicativos especificados?		X	X			X
	Os conceitos-chave do software são de fácil aprendizagem?		X		X		X
	É fácil instalar o aplicativo no emulador?		X		X		X
	É fácil instalar o aplicativo no dispositivo?		X		X		X
	É intuitiva a criação de projetos?		X		X	X	
	O software auxilia o usuário a não cometer erros?		X		X	X	
	A interface do software proporciona prazer e interação satisfatórios?		X		X	X	
	O software foi projetado para atender usuários com necessidades especiais?	X		X		X	
			63		58		42.5
Eficiência de desempenho	Tempo necessário para compilar o projeto	X		X			X
	Tempo necessário para instalar no dispositivo	X		X			X
	Tempo necessário pra instalar no emulador	X		X			X
	Recursos utilizados no computador	X			X	X	
	Recursos utilizados no dispositivo	X			X	X	
	O software atendeu todos os requisitos especificados nos aplicativos?		X	X			X
			10		40		50

# Resultados e Discussões

Compatibilidade	O software é capaz de coexistir com outros softwares do computador?		X		X		X
	O software é capaz de coexistir com outros softwares do dispositivo?		X		X		X
	O software é capaz de interagir com o Mac OS? (Necessário para compilar para <u>iOS</u> )		X		X	X	
		30		30		20	
Portabilidade	O software funciona na plataforma Windows?		X		X		X
	O software funciona na plataforma Mac OS?	X			X		X
	O software é facilmente instalado?		X		X	X	
	O software pode substituir outro?		X		X		X
	30		40		30		
Efetividade	O software permite o desenvolvimento dos aplicativos especificados por completo?		X	X			X
		10		5		10	
Eficiência	A compilação é rápida? Ou faz o usuário esperar?	X			X		X
		5		10		10	

# Resultados e Discussões

Satisfação	O usuário possui satisfação ao utilizar o software?	X			X			X	
	O usuário sente prazer em usar o software?	X			X			X	
	O uso de funções e componentes simples é de fácil acesso?	X				X	X		
		7.5		10		5			
Cobertura de contexto	O software pode ser usado com efetividade, eficiência, sem riscos e com satisfação em todos os contextos especificados?	X				X			X
	O software pode ser usado com efetividade, eficiência, sem riscos e com satisfação em todos os contextos especificados?		X			X			X
		15		20		20			
Total	290.5		353		297.5				

# Resultados e Discussões

Quadro 28 – Comparativo de características

	<b>Delphi XE5</b>	<b>Xamarin</b>	<b>PhoneGap</b>
Adequação funcional	X		
Confiabilidade			X
Usabilidade	X		
Eficiência de desempenho			X
Compatibilidade	X	X	
Portabilidade		X	
Efetividade	X		X
Eficiência		X	X
Satisfação		X	
Cobertura de contexto	X	X	X

# Conclusões

- Delphi XE5:
  - adequado para o desenvolvimento;
  - grande usabilidade;
  - efetivo;
  - baixo desempenho.
- Xamarin:
  - satisfação em relação ao uso e falsidades de acesso a funções;
  - compatível com Windows e OS X;
  - muitas vezes foi preciso desenvolver para cada plataforma.
- PhoneGap:
  - dificuldade na instalação, criação de projetos e uso de funções;
  - confiável e eficiente.

# Sugestões para futuros trabalhos

- análise comparativa sobre o recurso Bluetooth;
- análise comparativa sobre os componentes de interface com usuário e sua exibição e funcionalidade nas diferentes plataformas;
- nova versão Xamarin (lançada há 20 dias).