



Microsoft

Desarrollo de Aplicaciones Universales con Visual Studio

Jose García
Consultor | Servicios Profesionales | sp@danysoft.com

Enero 2017

Agenda

- Introducción a las aplicaciones para la Plataforma Universal Windows
- Ejemplo en JavaScript-HTML
- Ejemplo en C#-XAML
- UWP Community Toolkit



Introducción

- ¿Qué es una aplicación para la Plataforma universal de Windows (UWP)?
 - Una aplicación para Plataforma Universal Windows (UWP) es aquella que está integrada en esta plataforma, que se presentó en Windows 8 por primera vez como Windows Runtime (WinRT).
 - Hasta ahora, se podían crear aplicaciones para Windows Phone 8.1 y Windows con un código base compartido.
 - La experiencia de este tipo de aplicación depende de cada dispositivo pero, tiene como base la del móvil.



Introducción

- ¿Qué es una aplicación para la Plataforma universal de Windows (UWP)?
 - Con Windows 10, es más fácil:
 - Una sola API.
 - Un paquete de la aplicación.
 - Una tienda para obtener acceso a todos los dispositivos de Windows 10: equipos, tabletas, teléfonos, Xbox, HoloLens, Surface Hub y mucho más.



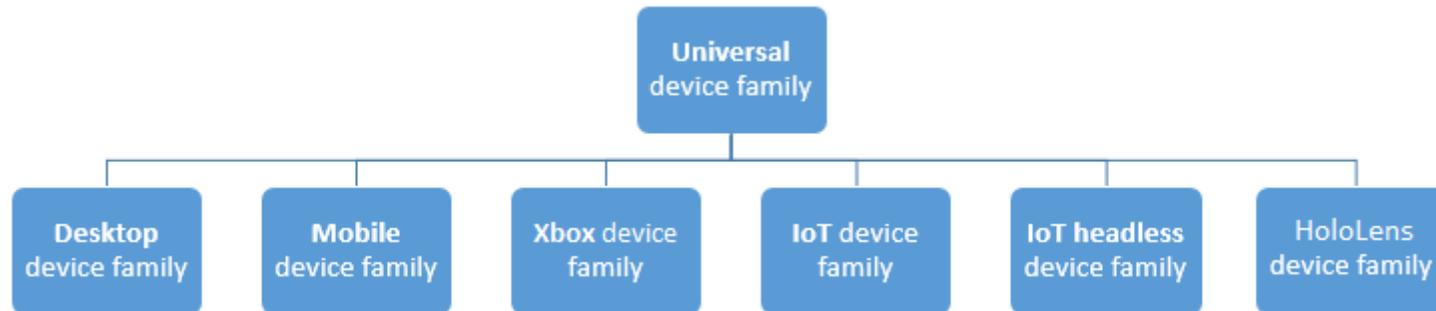
Introducción

- Características de una aplicación UWP Windows 10
 - Se abordan familias de dispositivos, no un sistema operativo.
 - Con Windows 10, ya no destinas tu aplicación a un sistema operativo, sino a una o más familias de dispositivos.



Introducción

- La familia de dispositivos identifica:
 - Las API.
 - Las características del sistema.
 - Los comportamientos que puedes esperar en todos los dispositivos de la familia.
 - Determina el conjunto de dispositivos en los que se puede instalar la aplicación desde la tienda.



Introducción

- **Características de una aplicación UWP Windows 10**
 - Todas las aplicaciones se distribuyen en formato .AppX.
 - Una tienda para todos los dispositivos:
 - Una vez registrado como desarrollador, puedes poner la aplicación disponible para los dispositivos que elijas.
 - La APIs principales de UWP son las mismas para todas las familias de dispositivos:
 - Si la aplicación solo utiliza estas APIs, se ejecutará en cualquier dispositivo Windows 10.
 - Ahora, no solo pueden llamar a las API de WinRT comunes a todos los dispositivos, sino también a las API (incluidas las de Win32 y .NET) específicas de la familia de dispositivos.



Introducción

- Características de una aplicación UWP Windows 10
 - Los SDK agregan APIs especializadas para cada familia de dispositivos.
 - Puede diseñar la aplicación para que destaque en una familia en particular, usando estas APIs, y seguir teniendo un paquete de la aplicación que se ejecute en todos los dispositivos, comprobando en qué familia de dispositivos está ejecutándose, antes de llamar a una API de extensión.



Introducción

- Características de una aplicación UWP Windows 10
 - “Píxeles eficaces”:
 - Los elementos de interfaz de usuario usan píxeles eficaces, de modo que se adapten automáticamente según el número de píxeles de pantalla disponibles en el dispositivo.
 - Se puede personalizar aún más la interfaz a un tamaño de pantalla o dispositivo específico.



Introducción

- Características de una aplicación UWP Windows 10
 - Interfaz de usuario y modo de entrada:
 - Hay aspectos que se adaptan automáticamente en todos los dispositivos.
 - Hay controles universales y paneles de diseño, para la resolución de pantalla del dispositivo.
 - Control de entrada común para táctil, lápiz, mouse, teclado o bien un controlador, como el controlador de Microsoft Xbox.
 - Algunos controles, -botones y controles deslizantes-, se adaptan automáticamente a diversas familias de dispositivos y modos de entrada.
 - Sin embargo, quizá haya que hacer adaptaciones que mejoren la experiencia de usuarios, según el tipo de dispositivo.



Introducción

- Características de una aplicación UWP Windows 10
 - Utilizamos lenguajes de programación conocidos:
 - Visual C++, C#, Visual Basic y JavaScript.
 - Con Visual C++, C# y Visual Basic, puedes usar XAML para obtener un mejor interfaz de usuario.
 - Con Visual C++, puedes dibujar con DirectX en lugar de XAML, o usar ambos.
 - Con JavaScript, la capa de presentación será HTML, que es un estándar web multiplataforma.
 - Gran parte del código y de la interfaz de usuario será universal y se ejecutará del mismo modo en cualquier dispositivo.
 - Para adaptar el código a familias de dispositivos determinadas, y la interfaz de usuario a factores de forma concretos, tendrás la opción de usar código adaptable y una interfaz de usuario adaptable.



Introducción

- Características de una aplicación UWP Windows 10
 - Llamar a una API que implementa la familia de dispositivos de destino
 - Debes saber si se implementa en la familia de dispositivos a la que está destinada la aplicación.
 - En caso de duda, puedes buscarla en la documentación de referencia de API.
 - Supongamos que tu aplicación está destinada a la versión 10.0.x.0 de la familia de dispositivos universales y quieres llamar a los miembros de la clase `Windows.UI.Core.SystemNavigationManager`. Como la familia de dispositivos es "Universal", sabes que las API tienen garantizado estar presentes en todos los dispositivos en los que puede instalarse la aplicación y que puedes llamarlas en el código como lo harías normalmente:
 - `Windows.UI.Core.SystemNavigationManager.GetForCurrentView().BackRequested += TestView_BackRequested;`



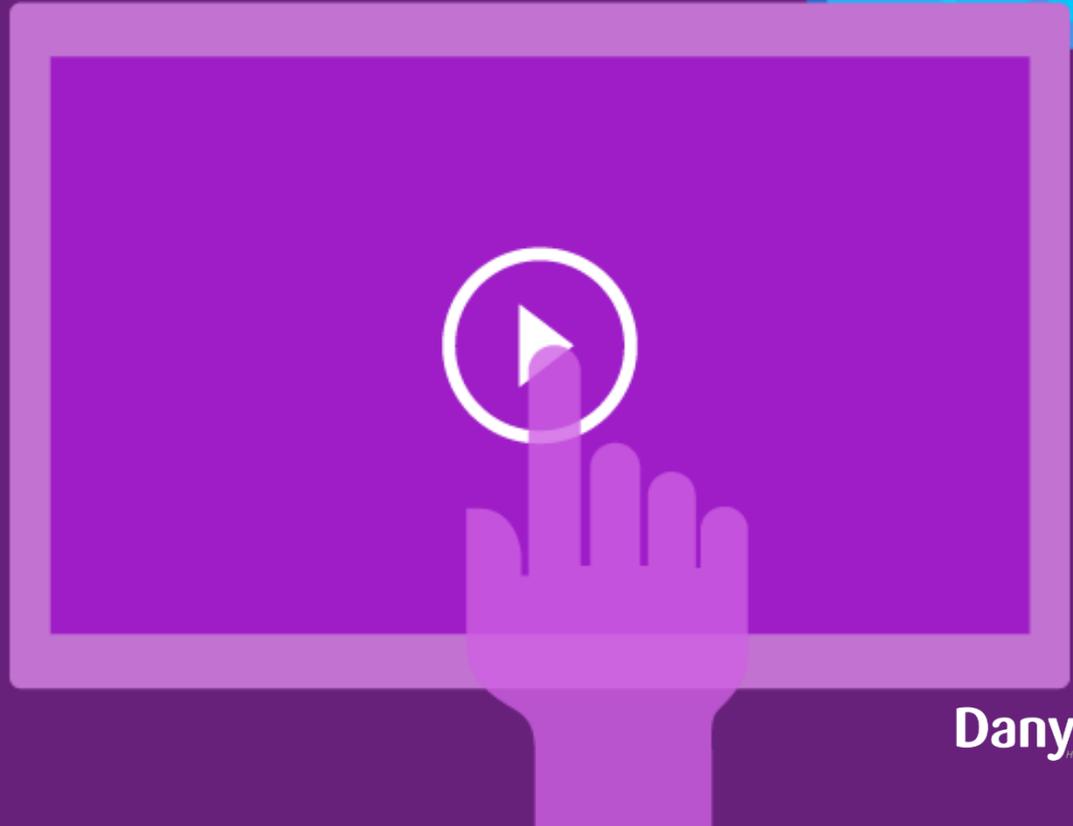
Introducción

- Una de las características de las aplicaciones multi-dispositivo es la importancia que toma la experiencia de usuario.
- Los elementos del interfaz a tener en cuenta:
 - Diseño
 - Controles
 - Entradas



Demo

Crear una aplicación "Hello, world" en JavaScript



Objetivos

- Crear un nuevo proyecto de Visual Studio 2015 diseñado para Windows 10 y UWP.
- Controlar la entrada táctil, de pluma y mouse.
- Ejecutar el proyecto en el escritorio local y en el emulador de teléfono de Visual Studio.
- Usar un control de la Biblioteca de Windows para JavaScript.



Requisitos

- Visual Studio 2015 / 2017
- Instalación de Visual Studio Tools para aplicaciones Windows universales
- En la primera ejecución va a solicitar cambiar la configuración del equipo a Modo programador
- Instalar los Emuladores, (no es obligatorio).



Proyecto WinJS App (Windows universal).

- Crear un nuevo proyecto de Visual Studio 2015/2017 JavaScript / Windows Universal / Aplicación WinJS.
- Le indicamos el rango de versiones de compilador.
- La primera vez que lo ejecutamos nos aparece la ventana de configuración para que usemos, también en nuestro ordenador, el modo de programador.



Proyecto WinJS App (Windows universal).

- El proyecto se crea con unos ficheros, de los que son principales:
 - Un archivo de manifiesto (package.appxmanifest) que describe la aplicación (nombre, descripción, icono, página de inicio, etc.) y enumera los archivos que contiene la aplicación.
 - Un conjunto de imágenes para el menú de inicio, el logo de la aplicación en la Tienda Windows, y para la pantalla de presentación cuando se inicia la aplicación.
 - Una página de inicio (index.html) donde se crea el interfaz de usuario.
 - Una hoja CSS vacía para implementar estilos, (vacía).
 - El archivo (main.js) que se ejecuta cuando se inicia la aplicación.



Ejecución / Depuración

- La aplicación ya se puede ejecutar:
 - En el dispositivo de escritorio
 - En Simulador
 - Equipo remoto
 - Emuladores



Revisión del código de main.js

- `(function () { /*Crea una función anónima, que se ejecuta automáticamente, donde se va a encapsular el resto del código. Es un procedimiento recomendado de programación*/`
- `"use strict"; /*Activa el modo strict. Proporciona una comprobación adicional de errores para el código. Por ejemplo, impide que uses variables declaradas implícitamente o que asignes un valor a una propiedad de solo lectura.*/`
- El resto del código controla los eventos `activated` y `checkpoint`. El evento `activated` se desencadena cuando se inicia la aplicación.



Controlador para de eventos para el botón

- El controlador de eventos obtiene el nombre de usuario del control input y lo pasa para mostrar un saludo al elemento "saludo".

```
/*se crea una función de controlador de eventos click denominada manejadorBoton que tome un único parámetro llamado eventInfo.*/
```

```
function manejadorBoton(eventInfo) {  
    var nombreUsuario = document.getElementById("nombre").value; /*recupera el valor del nombre de usuario del control input*/  
    var saludoString = "Hola, " + nombreUsuario; /*lo usa para crear el saludo*/  
    document.getElementById("saludo").innerText = saludoString; /*lo muestra en el elemento div "saludo"*/  
}
```



Registrar el controlador de eventos (HTML)

- Es necesario registrar el controlador de eventos para el control HTML. La forma recomendada es llamar a `addEventListener`. Un buen lugar es cuando se activa nuestra aplicación
- `//Registramos el controlador de eventos para el boton, para el evento click`
- `var nombreBoton = document.getElementById("botonHola");`
- `nombreBoton.addEventListener("click", manejadorBoton, false);`
- Ejecutamos la aplicación y probamos.



Biblioteca de Windows para JavaScript

- Vamos a utilizar un control de la Biblioteca de Windows para JavaScript: WinJS.UI.Rating.
- Hay otros como WinJS.UI.DatePicker, WinJS.UI.FlipView, WinJS.UI.ListView



Agregar el control Rating

- Para agregar un control de WinJS, crea un elemento div y usa el atributo data-win-control para especificar el tipo de control. Para Rating, se establece el atributo en "WinJS.UI.Rating". En index.html, añadimos:

```
<!--Control Rating-->  
  <label for="controlValoracion">  
    Valora este saludo:  
  </label>  
  <div id="controlValoracion" data-win-control="WinJS.UI.Rating">  
</div>
```



Registrar el controlador de eventos (JS)

- Es necesario registrar el controlador de eventos para el control WinJS.
- Primero creamos el div en index.html, donde mostrar el valor.

```
<!--Div para mostrar el valor de Rating-->  
  <div id="salidaValoracion"></div>
```



Registrar el controlador de eventos (JS)

- En main.js, se crea un controlador de eventos para el evento change del control Rating. El parámetro eventInfo contiene una propiedad detail.tentativeRating que proporciona una nueva clasificación de usuario. Recupera este valor y muéstralo en la salida div.

```
/*funcion para crear el controlador para el evento change
del control Rating.
detail.tentativeRating proporciona la valoración del usuario*/
function cambioValoracion(eventInfo) {
    var salidaValoracion =
        document.getElementById("salidaValoracion");
    salidaValoracion.innerHTML = eventInfo.detail.tentativeRating;
```



Registrar el controlador de eventos (JS)

- Actualiza el controlador de eventos onactivated que llama a WinJS.UI.processAll. Agregamos una llamada al método then y pasa una función completed, en la que recupera el elemento controlValidacion que hospeda el control Rating. Después, usa la propiedad winControl para recuperar el control Rating real.



Registrar el controlador de eventos (JS)

```
//Añidimos código para que Promise nos diga cuando se ha completado:
document.addEventListener("visibilitychange", onVisibilityChanged);
args.setPromise(WinJS.UI.processAll().then(function completed() {
    // Recupera el div donde está el control Rating.
    var controlValoracion = document.getElementById("controlValoracion");
    // Recupera el valor del control Rating.
    var salidaValoracion = controlValoracion.winControl;
    // Recupera el manejador del evento.
    salidaValoracion.addEventListener("change", cambioValoracion, false);
}));
//Registramos el controlador de eventos para el boton, para el evento click
var nombreBoton = document.getElementById("botonHola");
nombreBoton.addEventListener("click", manejadorBoton, false);
}

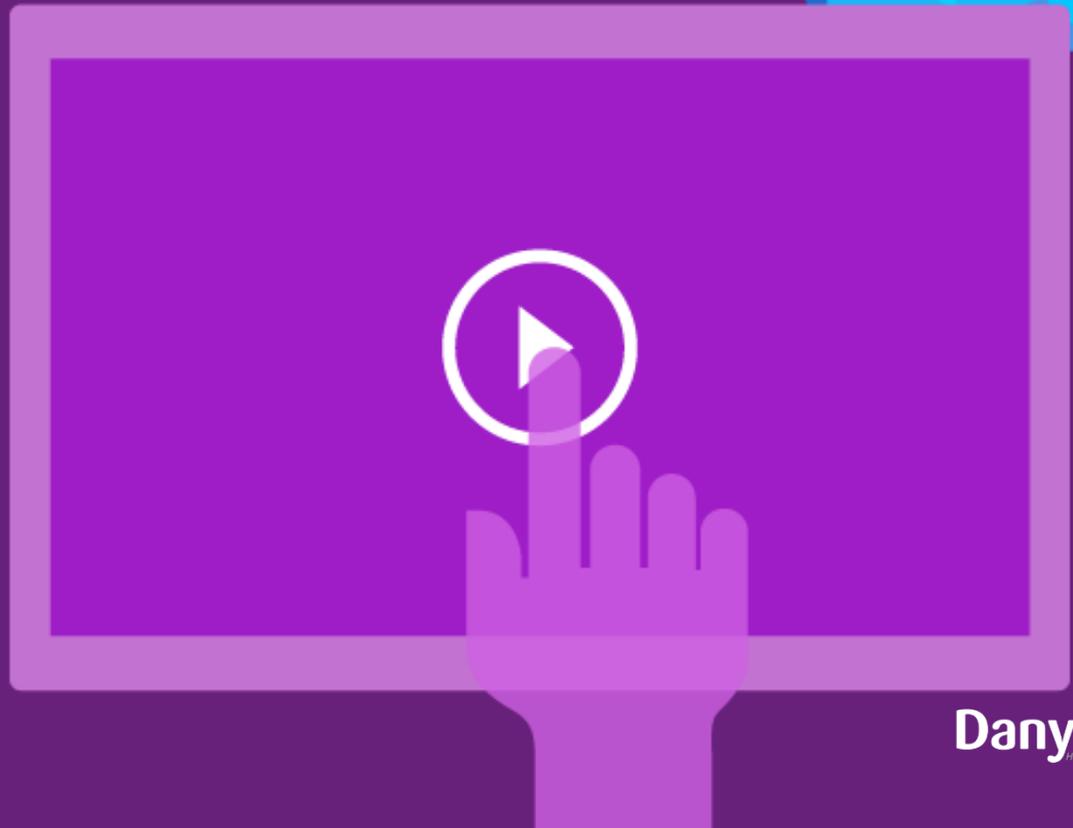
isFirstActivation = false;
```

```
};
```



Demo

Crear una aplicación tipo "Hello, world" en C#



Objetivos

- XAML y el panel de diseño, para cambiar la interfaz de usuario en la página de inicio.
- Ejecuta el proyecto en el escritorio local y en el emulador de teléfono de Visual Studio.
- Usa un SpeechSynthesizer para que la aplicación hable al presionar un botón.



Controlador de eventos

- Hacemos doble clic en botón para crear el controlador de eventos para el botón, y nos lleva a MainPage.xaml.cs.
- El controlador de eventos predeterminado es:

```
private void button_Click(object sender, RoutedEventArgs e)
{
}
}
```



Controlador de eventos

- Sustituimos el código, para usar algunas API de Windows para crear un objeto de síntesis de voz, y luego le proporciona algo de texto para que lo lea:

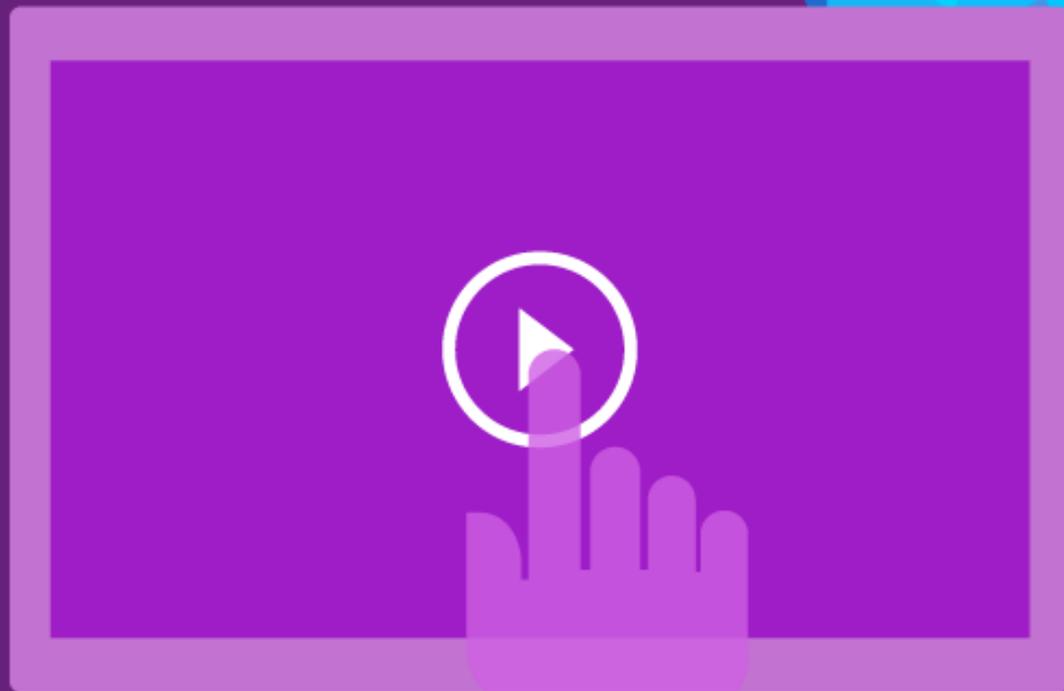
```
private async void button_Click(object sender, RoutedEventArgs e)
{
    MediaElement mediaElement = new MediaElement();
    var voz = new Windows.Media.SpeechSynthesis.SpeechSynthesizer();
    Windows.Media.SpeechSynthesis.SpeechSynthesisStream
        cadena = await voz.SynthesizeTextToStreamAsync("Hola, cómo estás");
    mediaElement.SetSource(cadena, cadena.ContentType);
    mediaElement.Play();
}
```



Demo

Utilizar Microsoft Toolkit UWP

sp@danysoft.com



Microsoft Toolkit UWP

- Es una herramienta gratuita con diferentes ejemplos de código útil para diferentes familias de dispositivos.
- Se descarga de:
 - <https://developer.microsoft.com/en-us/windows/uwp-community-toolkit/getting-started>
- Crear un nuevo proyecto C# Windows Universal en blanco.
- En el Solution Explorer, haz clic derecho sobre el proyecto y clic en Manage NuGet Packages. Busca Microsoft.Toolkit.UWP e instálalo.
- En la página App.xaml añadir la referencia:
 - `xmlns:controls="using:Microsoft.Toolkit.Uwp.UI.Controls"`
- En la página C# (MainPage.xaml.cs) añadir el espacio de nombres:
 - `using Microsoft.Toolkit.Uwp;`



Muchas gracias



sp@danysoft.com
www.danysoft.com

- Formación
- Consultoría
- Desarrollo

