



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“DESARROLLO DE APLICACIONES METRO SOBRE PLATAFORMA
WINDOWS 8 UTILIZANDO IDE VISUAL STUDIO 2012”**

AUTOR: BYRON PATRICIO PINEDA PILLAJO

DIRECTOR: ING. JOSÉ LÚIS RODRÍGUEZ

IBARRA – ECUADOR

2015



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1 IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	100275025-3		
APELLIDOS Y NOMBRES:	PINEDA PILLAJO BYRON PATRICIO		
DIRECCIÓN:	OTAVALO PIEDRAHITA Y ATAHUALPA		
EMAIL:	pineda.sistemas@gmail.com		
TELÉFONO FIJO:	06 2927760	TELÉFONO MOVIL:	0988438965
DATOS DE LA OBRA			
TÍTULO:	"DESARROLLO DE APLICACIONES METRO SOBRE PLATAFORMA WINDOWS 8 UTILIZANDO IDE VISUAL STUDIO 2012"		
AUTOR:	PINEDA PILLAJO BYRON PATRICIO		
FECHA:	ABRIL DEL 2015		
PROGRAMA:	<input type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO		
TÍTULO POR EL QUE OPTA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES		
DIRECTOR:	ING. JOSÉ LÚIS RODRÍGUEZ		

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Byron Patricio Pineda Pillajo, con cedula de identidad Nro. 100275025-3, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.



Firma

Nombre: Byron Patricio Pineda Pillajo

Cédula: 100275025-3

Ibarra, Abril del 2015



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Firma

Nombre: Byron Patricio Pineda Pillajo

Cédula: 100275025-3

Ibarra, Abril del 2015



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA
UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, Byron Patricio Pineda Pillajo, con cédula de identidad Nro. 100275025-3, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: "Desarrollo De Aplicaciones Metro Sobre Plataforma Windows 8 Utilizando IDE Visual Studio 2012" que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte

Firma

Nombre: Byron Patricio Pineda Pillajo

Cédula: 100275025-3

Ibarra, Abril del 2015



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN

Certifico que el proyecto de Trabajo de Grado "Desarrollo De Aplicaciones Metro Sobre Plataforma Windows 8 Utilizando IDE Visual Studio 2012" ha sido realizada en su totalidad por el señor: Byron Patricio Pineda Pillajo portador de la cédula de identidad número: 100275025-3



ING. JOSÉ LUIS RODRÍGUEZ
DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

A Dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mi madre Martha por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me permitió ser una persona de bien, pero más que nada, por su amor.

A mi hermana Albita por ser ejemplo de hermana y de la cual aprendí aciertos y de momentos difíciles; a mis hermanos Pablito, Edwin, Fernanda, a mis sobrinos Mateo, Ashley, Piere y a todos aquellos que participaron directamente en la elaboración de esta tesis.

A mi hermano Jonathan gracias por darme un gran ejemplo de vida, que a pesar de todo se pone en contra siempre hay una salida y sobre todo por enseñarme a vivir en familia.

Byron Patricio Pineda Pillajo



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

A Dios por haberme guiado por el camino de la felicidad hasta ahora; a mi madre Martha por toda la paciencia que ha tenido hasta la culminación de mi trabajo de grado, y a cada uno de los que son parte de mi familia.

A mis hermanos y a todos mis tíos; por siempre haberme dado su fuerza de apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora. Por ultimo a mis compañeros de trabajo porque en esta armonía grupal he logrado terminar mi trabajo y a mi director de tesis quien en todo momento con sus conocimientos, y sabiduría que sin ellos hubiera declinado en algún momento, Ing. José Luis Rodríguez.

Byron Patricio Pineda Pillajo

RESUMEN

El objetivo fundamental del presente trabajo de tesis, es investigar el nuevo Sistema Operativo Windows 8, y el desarrollo de aplicaciones metro sobre Windows 8. Esta es la reciente y última versión del ya conocido sistema operativo de Microsoft (Windows) que llega con la idea de revolucionar nuestra experiencia como usuarios, con la consigna de dar respuesta a los principales problemas de sus antecesores y acoplarse con vista futurista a las nuevas tendencias tecnológicas (pantallas táctiles, Aero glass, etc). Con su vocación multi-dispositivos, Windows 8 y las aplicaciones metro centran el protagonismo en el usuario que será quien decida en cada momento el hardware e incluso el software a utilizar.

Capítulo I, Define el análisis y el planteamiento del problema al cual se dará solución.

Capítulo II, Define en forma detallada y profunda los conceptos y definiciones de la nueva plataforma Windows 8, los principios de la nueva interfaz que las aplicaciones metro permiten al usuario, las herramientas para el desarrollo de las aplicaciones estilo metro.

Capítulo III, Se profundiza la investigación de la nueva plataforma su interfaz amigable con el usuario, su diseño exclusivo para las aplicaciones metro, así como sus herramienta principal Visual Studio 2012 y su relación con HTML5 y CSS3, sus principales características y directrices.

Capitulo IV, Se expondrá la metodología de la investigación y se construirá un lector de noticias RSS con la funcionalidad de recolectar desde varias páginas web los RSS y listarlos en el aplicativo con ellos poder observar los resultado de la investigación, cabe señalar que esto pretende ser una guía inicial para que los estudiantes de la Universidad Técnica del Norte, empiecen a desarrollar sus propias aplicaciones estilo metro

SUMARY

The main objective of this thesis is to investigate the new operating system Windows 8, Metro and developing Metro applications on Windows 8. This is the latest and final versión of the well-known Microsoft operating system (Windows) that comes with the idea revolutionizing our user experience, with the slogan of responding to the major problems of its predecessors and coupled with futuristic view of the new technological trends (touch screens, Aero glass). With his vocation multi - devices, Windows 8 Metro applications and focus the spotlight on the user will decide at each time hardware and even software to use.

Chapter I. Define the analysis and approach to the problem which will solve.

Chapter II. Define in detail and profound concepts and definitions of the new Windows 8 platform, the principles of the new Metro interface that allows the user applications, tools for the development of Metro style apps.

Chapter III, Research new SSOO deepens friendly interface with the user, its unique design for underground applications as well as its main tool Visual Studio 2012 and its relationship with HTML5 and CSS3, its main characteristics and guidelines.

Chapter IV, The research methodology will be exposed and RSS news reader be built with functionality to collect from various websites RSS and list them in the application with them to observe the outcome of the investigation, it should be noted that this is to be an initial guide for students of the Universidad Técnica del Norte begin to develop their own applications Metro style

INDICE DE CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICACIÓN	II
CONSTANCIA	IV
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	V
CERTIFICACIÓN	VI
DEDICATORIA	VII
AGRADECIMIENTO	VIII
RESUMEN.....	IX
SUMARY	X
INDICE DE CONTENIDO.....	XI
INDICE DE ILUSTRACIONES	XVIII
INDICE DE TABLAS	XXII
CAPITULO I	1
1 INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA	4
1.2 OBJETIVO GENERAL	4
1.3 OBJETIVOS ESPECÍFICOS	4
1.4 ALCANCE DEL PROYECTO	5
1.5 ANTECEDENTES	6
1.6 JUSTIFICACIÓN DEL PROBLEMA.....	7
CAPÍTULO II	8
2 LA PLATAFORMA DE WINDOWS 8 PARA LAS APLICACIONES DEL NUEVO ESTILO DE INTERFAZ DE USUARIO.....	8
2.1 INTRODUCCIÓN A LA EXPERIENCIA WINDOWS	8
2.2 WINDOWS RE IMAGINADO.....	9
2.3 INTRODUCCIÓN AL RUN – TIME DE WINDOWS 8	14
2.3.1 ADQUISICIÓN DE APLICACIONES	17
	XI

2.3.2 LA ESTRUCTURA DE UN PAQUETE DE INSTALACIÓN DE APLICACIONES .	17
2.3.2.1 MANIFIESTO DE UN PAQUETE DE INSTALACIÓN DE APLICACIONES	18
2.3.2.2 BLOCKMAP	18
2.3.3 FIRMA.....	18
2.3.3.1 AISLAMIENTO DE PROCESOS	18
2.3.4 TRANSICIÓN DE ESTADO DE LOS PROCESOS.....	18
2.3.5 DATOS DE LA APLICACIÓN VS DATOS DEL USUARIO	20
2.4 LA TIENDA DE WINDOWS 8.....	20
2.4.1 CATEGORÍA O RESULTADO DE LA BÚSQUEDA.....	21
2.4.2 LAS PÁGINAS DE APLICACIONES	22
2.4.3 INSTALAR APLICACIONES	22
2.4.4 GUARDAR Y ACTUALIZAR APLICACIONES.....	23
2.4.5 SEGURIDAD Y CONTROL PARENTAL.....	23
2.5 DISEÑO DE APLICACIONES CON LOS PRINCIPIOS Y PERSONALIDAD DEL NUEVO ESTILO DE INTERFAZ DE USUARIO	24
2.5.1 DISEÑO DEL NUEVO ESTILO DE INTERFAZ DE USUARIO	24
2.5.2 PILARES DEL DISEÑO METRO.....	25
2.5.3 PRINCIPIOS Y PERSONALIDAD DEL ESTILO METRO	25
2.5.3.1 SACAR PROVECHO AL ESTILO DE DISEÑO MICROSOFT	25
2.5.3.2 APLICACIONES WINDOWS 8 ESTILO METRO.....	27
2.5.3.3 CARACTERÍSTICAS.....	28
2.6 DESARROLLO DE APLICACIONES CON EL NUEVO ESTILO DE INTERFAZ DE USUARIO CON XAML	31
2.6.1 WINRT PARA EL DESARROLLO .NET	31
2.6.1.1 PLATAFORMA WINRT	31
2.6.1.2 LENGUAJES DE DESARROLLO SOPORTADO	34
2.6.1.3 WINRT Y METRO PARA DESARROLLO.....	34
2.6.1.4 WINRT Y .NET	34

2.6.2 PROGRAMACIÓN ASINCRÓNICA.....	35
2.6.2.1 ¿QUÉ ES MÉTODO ASINCRÓNICO?.....	35
2.6.2.2 PROGRAMACIÓN ASINCRÓNICA EN WINDOWS EN TIEMPO DE EJECUCIÓN.....	36
2.6.2.3 MODELOS ASINCRÓNICOS EN APLICACIONES DE LA TIENDA WINDOWS CON JAVASCRIPT.....	37
2.6.2.4 MODELOS ASINCRÓNICOS EN APLICACIONES DE LA TIENDA WINDOWS CON C# Y MICROSOFT VB.....	37
2.6.2.5 MODELOS ASINCRÓNICOS EN APLICACIONES DE LA TIENDA WINDOWS CON C++.....	38
2.6.2.6 CONTROL DE ERRORES.....	38
2.6.2.7 RESUMEN Y NEXT STEPS.....	38
2.7 DESARROLLO DE APLICACIONES DEL NUEVO ESTILO DE INTERFAZ DE USUARIO CON HTML Y JAVASCRIPT.....	39
2.7.1 HTML 5.0.....	39
2.7.2 CONCEPTOS BÁSICOS.....	40
2.7.3 ESTRUCTURA DEL CUERPO.....	41
2.7.4 NUEVOS Y VIEJOS ELEMENTOS.....	42
2.8 HTML 5.0 E INTERNET EXPLORER Y MICROSOFT DEVELOPER TOOLS.....	45
2.9 CSS3.....	47
2.9.1 HISTORIA.....	47
2.9.2 ¿PARA QUÉ SIRVE CSS3?.....	48
2.9.3 CARACTERÍSTICAS PRINCIPALES.....	48
2.9.4 VENTAJAS CSS3.....	49
2.9.5 VENTAJAS CSS3.....	50
CAPÍTULO III.....	51
3 CONSTRUYENDO UNA INTERFAZ DE USUARIO CON EL NUEVO ESTILO DE WINDOWS 8.....	51

3.1 HERRAMIENTAS DE DESARROLLO PARA APLICACIONES METRO SOBRE WINDOWS 8.....	51
3.1.1 VISUAL STUDIO 2012	51
3.1.2 INTERFAZ A ESTILO METRO.....	52
3.1.3 INTELLISENSE DE JAVASCRIPT	52
3.1.4 PLANTILLAS DE PROYECTO	53
3.1.5 ARCHIVOS IMPORTANTES DE UN PROYECTO METRO EN VISUAL STUDIO 2012.....	53
3.1.6 BLEND.....	54
3.1.7 AREA DE TRABAJO	55
3.1.8 VISTA DE DISEÑO	55
3.1.9 MESA DE TRABAJO.....	56
3.1.10 PANEL DE ACTIVOS.....	57
3.1.11 PANEL DE PROYECTOS	58
3.1.12 PANEL DE PROPIEDADES.....	59
3.1.14 LICENCIA DE DESARROLLADOR PARA WINDOWS 8.....	62
3.1.14 OBTENCIÓN DE LA LICENCIA PARA DESARROLLO DE APLICACIONES METRO SOBRE WINDOWS 8.....	62
3.1.16 OBTENCIÓN DE LA LICENCIA PARA PROGRAMADOR MEDIANTE SÍMBOLO DEL SISTEMA	63
3.2 DIRECTRICES PARA EL DESARROLLO DE APLICACIONES METRO	63
3.2.1 MUESTRA BUEN HACER	63
3.2.2 ESFUÉRZATE EN LOS DETALLES	64
3.2.3 HAZ MÁS CON MENOS	64
3.2.4 PON EL CONTENIDO ANTES QUE EL CONTENEDOR VISUAL	64
3.2.5 INSPIRA CONFIANZA AL USUARIO	64
3.3 DIRECTRICES PARA INTERACCIÓN DEL USUARIO CON LA APLICACIÓN.....	65
3.4 DIRECTRICES DE INTERACCIÓN TÁCTIL	65

3.5 DIRECTRICES PARA EL TAMAÑO DE OBJETOS	68
3.6 DIRECTRICES DE NAVEGACIÓN	69
3.6.1 SISTEMA JERÁRQUICO	69
3.6.2 SISTEMA PLANO	70
3.6.3 ANATOMÍA DE NAVEGACIÓN	71
3.7 DIRECTRICES DE DISEÑO	73
3.7.1 DISEÑO FLUIDO Y ADAPTABLE	73
3.7.2 DISEÑO ESTÁTICO O FIJO	74
3.7.3 RESOLUCIÓN DE PANTALLA.....	74
3.7.4 ESTADO DE VISUALIZACIÓN.....	75
3.7.5 COMPATIBILIDAD CON LAS ORIENTACIONES DE LA PANTALLA	75
3.7.6 NAVEGAR CON LAS ETIQUETAS DE LA SECCIÓN Y LOS MENÚS DE ENCABEZADOS	76
3.7.7 NAVEGAR CON FILTROS, TABLAS DINÁMICAS, ORDENACIONES Y VISTAS	77
3.7.8 BARRA DE APLICACIÓN SUPERIOR	78
3.7.9 CAMBIO DE VISTA DE LA BARRA DE APLICACIÓN	78
3.8 DIRECTRICES DE TEXTO	78
3.9 DIRECTRICES PARA BOTONES	80
3.10 DIRECTRICES PARA CASILLAS	81
3.11 DIRECTRICES PARA BOTONES DE RADIO	81
3.12 DIRECTRICES PARA VÍNCULOS	82
3.13 DIRECTRICES PARA CUADROS DE DIALOGO.....	82
3.14 DIRECTRICES PARA ICONOS	83
CAPÍTULO IV	84
4 METODOLOGÍA	84
4.1 PROCESO DE INVESTIGACIÓN.....	84
4.1.1 ANÁLISIS.....	84

4.1.2 TIPO DE INVESTIGACIÓN	84
4.1.3 MÉTODO	84
4.1.4 TÉCNICA	85
4.1.5 INSTRUMENTO.....	85
4.2 METODOLOGÍA INFORMÁTICA	85
4.2.1 PROCESO UNIFICADO DE DESARROLLO.....	85
4.2.2 CARACTERÍSTICAS.....	85
4.2.3 FASES DEL PROCESO UNIFICADO DE DESARROLLO	86
4.2.3.1 FASE INICIO.....	86
4.2.3.2 DECLARACIÓN DEL PROBLEMA.....	86
4.2.3.3 STAKEHOLDERS	87
4.2.3.4 CASO DE USO DE NEGOCIO.....	87
4.2.3.5 LISTA DE RIESGO	88
4.2.4 FASE DE ELABORACIÓN	88
4.2.4.1 CASO DE USO DE SISTEMA.....	88
4.2.4.2 ESCENARIO CASOS DE USO	89
4.2.4.3 DIAGRAMA DE ACTIVIDADES.....	91
4.2.4.4 DIAGRAMA DE SECUENCIA.....	92
4.2.4.5 DIAGRAMA DE PAQUETES.....	93
4.2.4.6 ARQUITECTURA DEL APLICATIVO	93
4.2.5 FASE DE CONSTRUCCIÓN.....	94
4.2.5.1 DESARROLLO DEL APLICATIVO “LECTOR DE RSS”	94
4.2.5.2 ESTRUCTURA.....	94
4.2.5.3 INTERFAZ.....	102
4.2.6 FASE DE TRANSICIÓN.....	105
4.2.6.1 REQUERIMIENTOS TÉCNICOS	106
4.2.6.2 INTERFAZ LECTOR RSS.....	106

CAPÍTULO V	108
5 CONCLUSIONES Y RECOMENDACIONES.....	108
5.1 CONCLUSIONES	108
5.2 RECOMENDACIONES	108
5.3 BIBLIOGRAFÍA	109

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 1: Logo Windows 8”	1
ILUSTRACIÓN 2: “Windows Runtime”	1
ILUSTRACIÓN 3: “todo en la nube”	2
ILUSTRACIÓN 4: ” Experiencia de usuario”	2
ILUSTRACIÓN 5: “Transición de programas”	3
ILUSTRACIÓN 6: “Imagen tienda Windows 8”	4
ILUSTRACIÓN 7: “Aplicativo Lector de Feeds”	6
ILUSTRACIÓN 8: “Inicio Windows 8”	8
ILUSTRACIÓN 9: “Pantalla principal Windows 8”	9
ILUSTRACIÓN 10: ”Presentación Windows 8 para móviles”	9
ILUSTRACIÓN 11: “Pantalla Principal Windows 8”	9
ILUSTRACIÓN 12: “Tile de Windows 8”	10
ILUSTRACIÓN 13: “Aplicaciones metro completa”	10
ILUSTRACIÓN 14: “Mosaico o Title”	10
ILUSTRACIÓN 15: “Zoom de aplicaciones”	11
ILUSTRACIÓN 16: “Charms Windows 8”	11
ILUSTRACIÓN 17: “Charms Windows 8”	12
ILUSTRACIÓN 18: “Panel de control Windows 8”	12
ILUSTRACIÓN 19: “Resultado de búsquedas”	12
ILUSTRACIÓN 20: “Ajuste de pantalla”	13
ILUSTRACIÓN 21: “Pantalla Windows 8 estilo metro”	13
ILUSTRACIÓN 22: “Pantalla Windows 8 normal”	14
ILUSTRACIÓN 23: “Run-time Windows 8”	14
ILUSTRACIÓN 24: “Estados de la aplicación”	19
ILUSTRACIÓN 25: “Windows Store”	20
ILUSTRACIÓN 26: “Windows Store”	21

ILUSTRACIÓN 27: “Categorías”	21
ILUSTRACIÓN 28: “Distribución de la pantalla”	22
ILUSTRACIÓN 29: “Logo en estilo metro”	24
ILUSTRACIÓN 30: “Vistas de varias ventanas”	28
ILUSTRACIÓN 31: “Movimiento táctil en la pantalla”	29
ILUSTRACIÓN 32: “Logo en estilo metro”	30
ILUSTRACIÓN 33: “Acceso rápido”	30
ILUSTRACIÓN 34: “Animación de iconos”	31
ILUSTRACIÓN 35: “Api metro style”	32
ILUSTRACIÓN 36: “Elementos de la plataforma Windows 8”	32
ILUSTRACIÓN 37: “Task manager”	33
ILUSTRACIÓN 38: “Visual Studio 2012”	52
ILUSTRACIÓN 39: “Blend 2012”	54
ILUSTRACIÓN 40: “Vista de diseño”	55
ILUSTRACIÓN 41: “Mesa de trabajo”	56
ILUSTRACIÓN 42: “Panel de activos”	57
ILUSTRACIÓN 43: “Panel de proyectos”	58
ILUSTRACIÓN 44: “Panel propiedades”	60
ILUSTRACIÓN 45: “Panel de herramientas”	61
ILUSTRACIÓN 46: “Lenguaje táctil de Windows 8”	66
ILUSTRACIÓN 47: “Precisión con los objetos”	66
ILUSTRACIÓN 48: “Contenido explorado con gestos táctiles”	67
ILUSTRACIÓN 49: “Seguimiento de los movimientos”	67
ILUSTRACIÓN 50: “Contenido explorado con gestos táctiles”	68
ILUSTRACIÓN 51: “Iteración con varios dedos”	68
ILUSTRACIÓN 52: “Tamaño de objetos”	69
ILUSTRACIÓN 53: “Sistema jerarquico”	70

ILUSTRACIÓN 54: “Anatomía de navegación jerarquico”	72
ILUSTRACIÓN 55: “Navegación entre niveles”	72
ILUSTRACIÓN 56: ”Anatomía de la navegación plana”	73
ILUSTRACIÓN 57: “Diseño fluido y adaptable”	74
ILUSTRACIÓN 58: “Diseño estático y fijo”	74
ILUSTRACIÓN 59: “Visualización de los estados”	75
ILUSTRACIÓN 60: “Orientación del screen”	75
ILUSTRACIÓN 61: “Etiquetas de seccion”	76
ILUSTRACIÓN 62: “Casilla opcionales”	81
ILUSTRACIÓN 63: “Cuadro de mensajes”	82
ILUSTRACIÓN 64: “Caso de uso del negocio”	87
ILUSTRACIÓN 65: “Caso de uso del sistema”	88
ILUSTRACIÓN 66: “Caso de uso del escenario”	89
ILUSTRACIÓN 67: "Diagrama de actividades"	91
ILUSTRACIÓN 68: "Diagrama de secuencia"	92
ILUSTRACIÓN 69: "Diagrama de paquetes"	93
ILUSTRACIÓN 70: “Arquitectura del aplicativo”	93
ILUSTRACIÓN 71: “Elegir nueva solución”	95
ILUSTRACIÓN 72: “Página básica”	95
ILUSTRACIÓN 73: “Mensaje de solución en blanco”	96
ILUSTRACIÓN 74: “Vista general del proyecto”	96
ILUSTRACIÓN 75: “Archivo App.xaml”	97
ILUSTRACIÓN 76: “Accesibilidad de clases”	97
ILUSTRACIÓN 77: “Ejecutor del aplicativo”	98
ILUSTRACIÓN 78: “Vista en el simulador”	98
ILUSTRACIÓN 79: “Fragmento del código del aplicativo”	99
ILUSTRACIÓN 80: “Creación clase genéricos”	100

ILUSTRACIÓN 81: “Uso del método syndicationclient”	100
ILUSTRACIÓN 82: “Creación de la clase”	101
ILUSTRACIÓN 83: “Vista del formato de clase”	101
ILUSTRACIÓN 84: “Codificación de la clase”	101
ILUSTRACIÓN 85: “Archivo LectorMainView.xaml”	102
ILUSTRACIÓN 86: “Fragmento de código archivo LectoroMainview.xaml”	103
ILUSTRACIÓN 87: “Código del archivo LectoroMainview.xaml”	103
ILUSTRACIÓN 88: “Modelo del aplicativo”	104
ILUSTRACIÓN 89: “Interfaz base del aplicativo”	105
ILUSTRACIÓN 90: “Menú lateral de aplicativo”	105
ILUSTRACIÓN 91: “Pantalla inicio lector de noticias RSS”	106
ILUSTRACIÓN 92: “Pantalla del aplicativo lector de noticia RSS”	107

INDICE DE TABLAS

TABLA 1: “Programación Asíncrona”	36
TABLA 2: “Sistema plano”	71
TABLA 3: “Navegación con filtros”	77
TABLA 4: “Declaración del problema”	86
TABLA 5: “Stakeholders”	87
TABLA 6: “Lista de riesgos”	88
TABLA 7: “Escenario de casos de uso”	89
TABLA 8: “Escenario de noticias”	90

CAPITULO I

1 INTRODUCCIÓN

Estar al día y buscar lo nuevo es el desafío de los programadores, conocer y estar al tanto de los nuevos lenguajes de programación, buscar nuevas técnicas que mejore nuestra calidad en el software, el desarrollo ágil y la programación guiada por pruebas (TDD), o la Programación en pares obliga a todos los programadores a mantener los conocimientos actualizados.

Windows 8 es una plataforma nueva, que presenta varios desafíos a los desarrolladores.



ILUSTRACIÓN 1: Logo Windows 8”

Fuente: http://commons.wikimedia.org/wiki/File:Windows_8_logo_and_wordmark.svg

En particular hay 4 puntos clave que indican el tipo de desafío a enfrentar:

- **API de Windows 8:** WinRT agrupa toda la interfaz de programación de aplicaciones más conocidas por APIs y presta sus componentes de una manera simple pero muy diferente a las anteriores. la programación y los procesos asíncronos, son métodos no-bloqueantes y las animaciones desde código y la forma de acceder a los archivos de disco siguen un mundo diferente al ya acostumbrado.

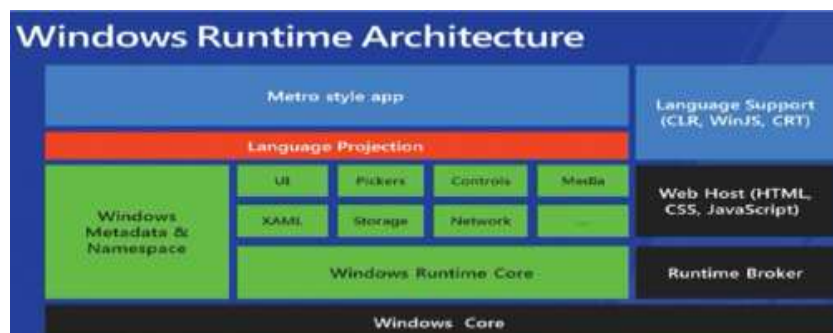


ILUSTRACIÓN 2: “Windows Runtime”

Fuente: <http://nblog.azurewebsites.net/?p=936>

- **Windows 8 conectado a la nube:** La movilidad es una razón fundamental de la existencia de Windows 8 y esto se lo puede entender con la experiencia de conectividad e integración perfecta de las aplicaciones con la nube



ILUSTRACIÓN 3: "Todo en la nube"

Fuente: <http://nblog.azurewebsites.net/?p=936>

- **La Experiencia de Usuario Windows 8:** Lo más destacado de Windows 8 es la llamada "Experiencia de Usuario" que reemplaza a lo ya conocido "Interfaz de Usuario" pues es una extensión del estilo al usar el sistema operativo. lo cual como programadores resultó un término desconocido, pero forma parte de lo ya mencionada experiencia Windows



ILUSTRACIÓN 4: "Experiencia de usuario"

Fuente: <http://nblog.azurewebsites.net/?p=936>

- **Dispositivos:** Este nuevo estilo de Windows 8, es necesario que no se limite solo en su presentación sino también debe ajustarse perfectamente en al menos 5 dimensiones de pantalla y varios tipos de dispositivos que se puede mencionar: Ultrabooks con pantalla táctil, Tablets, Computadoras con pantalla táctil, PCs entre otros.



ILUSTRACIÓN 5: “Transición de programas”

Fuente: <http://nblog.azurewebsites.net/?p=936>

“En la práctica, más de 350 millones de terminales con Windows 8 se estiman para este 2012 y 500 millones corriendo el sistema operativo para 2013. Estos números superan cualquier otra plataforma móvil, sistema operativo o teléfono vendido y garantiza un mercado infinito a nivel mundial”. (**Power by wordPress, 2014**)

- **Todas las aplicaciones que quieras:** En la nueva experiencia Windows todo está relacionado con la nube, es decir cada equipo está conectado entre si y pueden compartir información a toda la red, además servicios como los son redes sociales, mensajería, skydrive, también de permitir buscar y evaluar miles de aplicación en la tienda Windows conocida por Windows Store

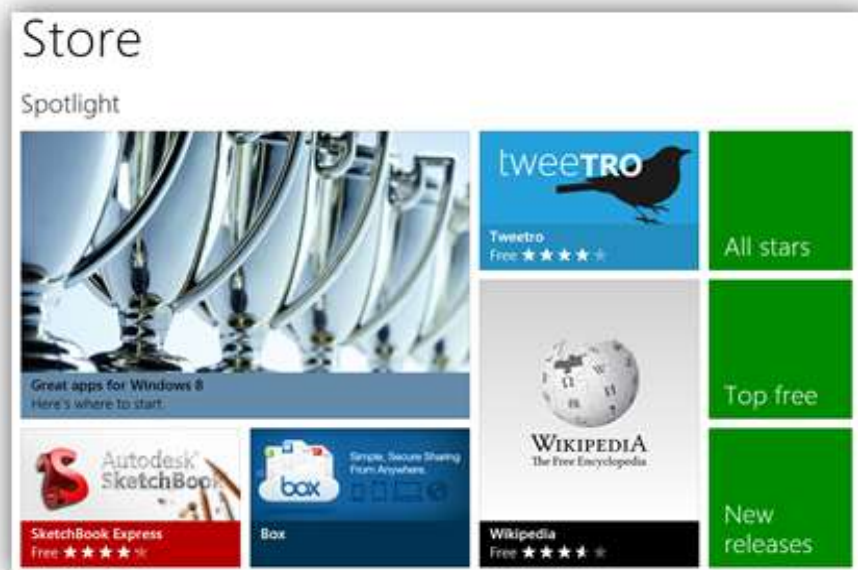


ILUSTRACIÓN 6: “Imagen tienda Windows 8”

Fuente <http://nblog.azurewebsites.net/?p=936>

1.1 PLANTEAMIENTO DEL PROBLEMA

Desarrollo de Aplicaciones Metro sobre plataforma Windows 8 utilizando IDE Visual Studio 2012

1.2 OBJETIVO GENERAL

Estudiar el desarrollo de las aplicaciones metro sobre un entorno Microsoft Windows 8 RTM, mediante la investigación y creación de un aplicativo simple a estilo metro utilizando IDE Visual Studio 2012.

1.3 OBJETIVOS ESPECÍFICOS

- ✓ Estudiar el nuevo estilo y estructura de las aplicaciones metro sobre plataforma Windows 8.
- ✓ Experimentar el Windows Store que es la tienda de aplicaciones para Windows 8.
- ✓ Desarrollar un lector RSS diseñado bajo el nuevo estándar de diseño marcado por Microsoft para Windows 8.
- ✓ Entregar documentación escrita que sirva de guía para los estudiantes y profesionales de la UTN en el desarrollo de las nuevas aplicaciones metro

1.4 ALCANCE DEL PROYECTO

En la presente tesis de grado, se estudiará las características que se usan para crear aplicaciones estilo Metro sobre plataforma Windows 8. Siguiendo el proceso de la creación de una aplicación de lector de RSS, se establecerá los conceptos fundamentales para el desarrollo con XAML, como:

- Diseño, controles, plantillas y enlace de datos
- Uso de la navegación y las plantillas de página que hay integradas en Microsoft Visual Studio Express 2012 para Windows 8, usadas para empezar a desarrollar las aplicaciones de forma rápida.
- Uso de estilos personalizados para alterar la apariencia de la aplicación y adaptar la interfaz de usuario a varios diseños y vistas.
- En los últimos capítulos de la tesis, se analizará cómo integrar nuestra aplicación con Windows 8 y cómo publicarla en la Tienda de Windows.

El alcance real del presente trabajo de tesis es lograr una familiarización con las aplicaciones metro sobre plataforma Windows 8 utilizando IDE Visual Studio 2012, para lograr finalmente obtener una guía de introducción y creación de aplicaciones metro para la tienda Windows que permita al estudiante de la UTN comenzar a crear sus propias aplicaciones.

El aplicativo que se desarrollará es un lector de Feeds el cual permite:

- Buscar y Mostrar datos de una fuente (RSS¹) se utilizará las fuentes del sitio The Windows Blog que servirá de ejemplo. Cuando finalice el desarrollo el aplicativo tendrá el siguiente aspecto.

¹ **RSS** son las siglas de *RDF Site Summary* or *Rich Site Summary*, un formato XML para **sindicar** o compartir contenido en la web.



ILUSTRACIÓN 7: “Aplicativo lector de feeds”

1.5 ANTECEDENTES

Las aplicaciones han revolucionado en forma imprevista desde hace un tiempo atrás, esto se debe a los grandes avances de hardware lo cual permite que las aplicaciones cambien en su forma de procesar como su forma de manifestación o interfaz gráfica.

Las primeras aplicaciones fueron básicas, trabajos simples y con una interfaz basada en línea de comandos. Luego surgieron aplicaciones con labores más complejas y con una interfaz gráfica no muy precisa.

Con el retoque de Windows se pudo definir una interfaz gráfica para las aplicaciones, que permite la interrelación con los diferentes elementos de presentación como son: menús desplegables, combos, textbox, entre otros, estos elementos se utilizan para crear cualquier tipo de aplicación sin distar el área que se las aplique.

Hoy se está cruzando a una nuevo avance en el desarrollo de aplicaciones que va orientada para el sistema operativo Windows 8 manejando la interfaz metro, en países extranjeros estas aplicaciones ya están siendo desarrolladas y puestas ya a la venta, lamentablemente esto no ocurre en Ecuador por la falta de información e interés que tienen los desarrolladores por el tema.

Microsoft permite la descarga gratuita de las herramientas necesarias para crear aplicaciones Windows 8. Lo cual permite proveer que en el futuro estas aplicaciones pueden llegar a ser una gran oportunidad de negocio.

1.6 JUSTIFICACIÓN DEL PROBLEMA

Contrapuestas a las tradicionales aplicaciones a las que se está acostumbrado, una aplicación metro en plataforma Windows 8 brinda una sola ventana con un contenedor visual que rellena toda la pantalla lo cual permite que no exista distracción para el usuario y apoya a la concentración de las actividades en el aplicativo.

Estas aplicaciones están pensadas para trabajar perfectamente bajo dispositivos táctiles así como también en dispositivos que utilizan mouse y teclado, las aplicaciones presentan una silueta definida que refiere qué partes deben tener contenido y qué partes deben tener controles, haciendo más escuetas y homogéneas las aplicaciones.

Las aplicaciones estilo metro para Windows 8 cuidan mucho las tipografías para facilitar una lectura más rápida y ayudar al usuario a encontrar la información que busca.

(Microsoft, 2014)

Esta investigación está directamente desarrollada y enfocada para que sirva de principal ayuda para los estudiantes de la facultad de Ingeniería Ciencias Aplicadas para futuras creaciones de sistemas basadas en el modelo de aplicaciones metro sobre una plataforma de características nuevas como es Windows 8, con el aplicativo presentado se verán beneficiados indirectamente todos los estudiantes de la Universidad Técnica del Norte accediendo al manual de desarrollo entregado para la presente tesis, podrán ver en un modelo a pequeña escala todas las bondades que tienen las nuevas aplicaciones a estilo metro.

CAPÍTULO II

2 LA PLATAFORMA DE WINDOWS 8 PARA LAS APLICACIONES DEL NUEVO ESTILO DE INTERFAZ DE USUARIO

2.1 INTRODUCCIÓN A LA EXPERIENCIA WINDOWS

Windows 8 es la adaptación actual del sistema operativo de Microsoft basado en su diseño anterior de sistema operativo Windows 7 , diseñado específicamente para su uso en tablets, PC táctiles y varios dispositivos domésticos, también se lo utiliza en servidores y centros multimedia.

Soporta microprocesadores ARM², además de los microprocesadores tradicionales x86 de Intel y AMD3, su interfaz ha sido modificada para usarla adecuadamente en pantallas táctiles, además de los tradicionales ratón y teclado.

Esta última versión de Windows proporciona la nueva experiencia informática desde su interfaz inicial:



ILUSTRACIÓN 8: "Inicio Windows 8"

Fuente: http://www.aulaclie.es/guia-windows8/t_1_1.htm

La nueva interfaz METRO. Es el pilar fundamental del nuevo Windows 8, pero no por esto signifique que no se pueda utilizarla como sus antecesores Windows 7, vista o xp, pero si realmente se desea conocer el alcance de la nueva experiencia se debe usarla bajo su estilo metro.

² ARM es una arquitectura RISC (Reduced Instruction Set Computer=Ordenador con Conjunto Reducido de Instrucciones) de 32 bits desarrollada por ARM Holdings. Se llamó *Advanced RISC Machine*



ILUSTRACIÓN 9: “Pantalla principal Windows 8”

Fuente: http://www.aulaalic.es/guia-windows8/t_1_1.htm

“Todas las aplicaciones instaladas en Windows 8 se encuentran en la START SCREEN que maneja la nueva interfaz METRO. Esta nueva interfaz adopta las características de WINDOWS PHONE 7 que es el sistema operativo para dispositivos móviles como los Smartphones. Es por ello que con Windows 8, Microsoft ha querido introducir el eslabón perdido entre las computadoras y los móviles.” (Jorge, 2012)



ILUSTRACIÓN 10: “Presentación Windows 8 para móviles”

Fuente: <http://windows8uleam.blogspot.com/2013/03/introduccion.html#.UktlqmaHddg>

WINDOWS STORE. Es tienda en línea de Microsoft, permite la descarga de software diseñado para METRO, algunos son de pago y otras gratuitas.

2.2 WINDOWS RE IMAGINADO



ILUSTRACIÓN 11: “Pantalla principal Windows 8”

Fuente: <http://tutoriales.com/?p=1823>

“Windows 8 ha sido re-imaginado con una nueva pantalla de inicio. Tiene entrada unificada utilizando el mouse, el teclado y/o un dispositivo de entrada táctil (por ejemplo un monitor táctil o la pantalla de una Surface), esto hace que se pueda diseñar una aplicación para un dispositivo táctil y podrá funcionar con el mouse y teclado. También puede diseñarse una aplicación pensando en un mouse y teclado y funcionará en un dispositivo táctil.” (Zapata, 2012)



ILUSTRACIÓN 12: “Tile de Windows 8”

Fuente: <http://tutoriales.com/?p=1823>

El gesto “Tap” el cual Permite activar una aplicación en la interfaz Metro con un solo clic en el mosaico.



ILUSTRACIÓN 13: “Aplicaciones metro completa”

Fuente: <http://tutoriales.com/?p=1823>

Ya no se tiene Marcos en la aplicación, se ejecutan en pantalla completa



ILUSTRACIÓN 14: “Mosaico o Title”

Fuente: <http://windows.microsoft.com/es-419/windows-8/tutorial>

- Los mosaicos dan vida a las aplicaciones Metro, tiene las siguientes actividades:
- Los Mosaicos pueden mostrar información sin que la aplicación esté activa.
- Los Mosaicos se pueden acomodar, intercambiar solamente arrastrando los mosaicos.

Las pantallas del estilo Metro permiten hacer zoom semántico. Lo cual admite ver con más o menos detalles los elementos del entorno Metro.

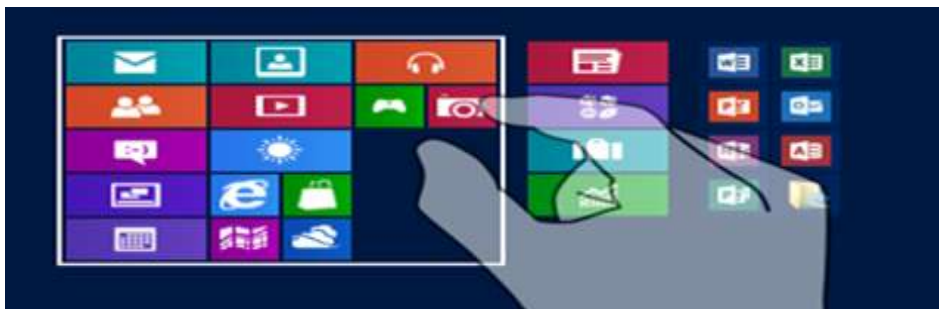


ILUSTRACIÓN 15: “Zoom de aplicaciones”

Fuente: http://www.lawebdelprogramador.com/cursos/Windows_8/index1.htm



ILUSTRACIÓN 16: “Charms Windows 8”

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wintr>

Los Charms (o encantos). Asoma a la derecha de la pantalla. Puede mostrarse arrastrando el Mouse y/o el dedo hacia la parte derecha de la pantalla o presionando las teclas Windows + C, y esta se dividen en: Search, Share, Start, Devices, Settings

Una aplicación puede implementar contratos y con él aprovechar los Charms.



ILUSTRACIÓN 17: “Charms Windows 8”

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wintr>

El Charm Search de Internet Explorer implementa un contrato de búsqueda. Para buscar nuevas aplicaciones que se integran a los contratos de búsqueda.

Para moverse entre aplicaciones se utiliza la combinación de teclas Alt + Tab (cómo en el modo tradicional)

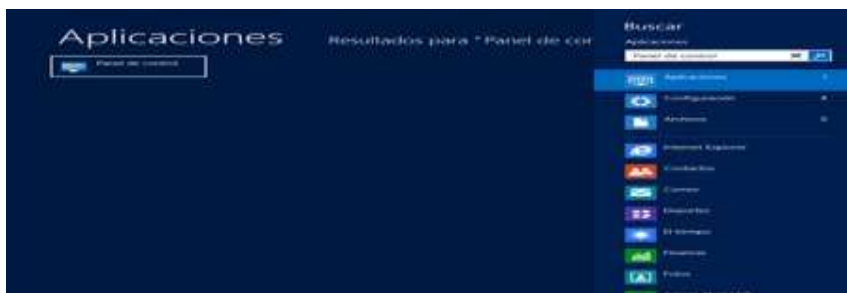


ILUSTRACIÓN 18: “Panel de control Windows 8”

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wintr>

Para ingresar al panel de control basta con escribir panel en los Charm Search o presionar la tecla Windows.

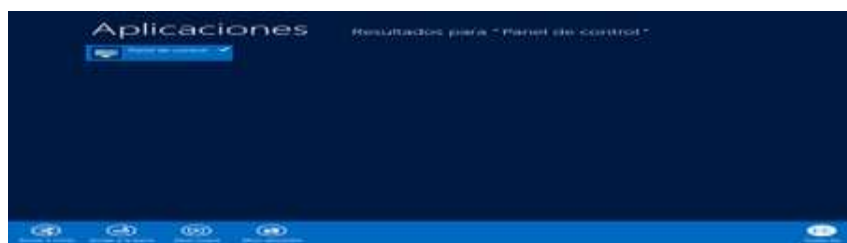


ILUSTRACIÓN 19: “Resultado de búsquedas”

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wintr>

Si se necesita poner una aplicación a la barra de inicio hay que realizar una búsqueda de las aplicaciones, hacer clic secundario sobre la aplicación y seleccionar la opción de Pin to start (Anclar al inicio).

De la misma forma, para desanclar aplicaciones, hacer clic secundario sobre ella y seleccionar la opción Un pin from Start (Desanclar de inicio)



ILUSTRACIÓN 20: "Ajuste de pantalla"

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wintr>

"Las aplicaciones generalmente se ejecutan en primer plano, pero si se tienen dos aplicaciones en primer plano se ajustando una de ellas a un extremo de la pantalla (Snapping). Desde el borde izquierdo arrastrar la aplicación que se necesita compartir en la pantalla. Cuando aparece la división soltar y la aplicación quedará ajustada" (Microsoft, 2013)

Las aplicaciones de la nueva interfaz Windows 8 detectan cambios en el tamaño de la pantalla y la orientación de los dispositivos.

En Windows 8 se puede trabajar en dos modos

Modo Metro.- Para ir al modo Metro apriete la tecla Windows



ILUSTRACIÓN 21: "Pantalla Windows 8 estilo metro"

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wintr>

Modo Escritorio.- Para ir al modo Escritorio presionar sobre el Tile Escritorio o presionar la combinación de teclas Windows + D



ILUSTRACIÓN 22: “Pantalla Windows 8 normal”

Fuente: Holguera Javier, (25 de Julio del 2012). Plataforma WinRT. Recuperado el 11 de febrero del 2013, de <http://www.genbetadev.com/javascript/programar-aplicaciones-metro-de-windows-8-estandares-y-wint>

Lo que funciona en Windows 7 seguirá funcionando en el Modo Escritorio de Windows 8

2.3 INTRODUCCIÓN AL RUN – TIME DE WINDOWS 8

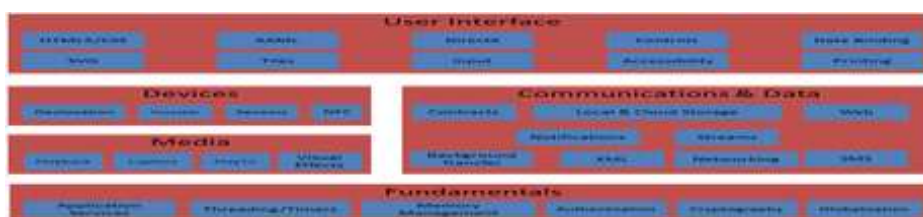


ILUSTRACIÓN 23: “Run-time Windows 8”

Fuente: Xatacala, (2012). Versiones Windows 8, Recuperado el 12 de Abril del 2013, de <http://www.xatakawindows.com/windows/windows-8-elige-entre-las-versiones-conocidas>

El Windows Runtime, o WinRT, que constituye el núcleo del nuevo sistema operativo y es quizás la concepción central más confuso en Windows 8, gracias a su posición o relación, con la pantalla de inicio y de estilo Metro Aplicaciones.

“Vale la pena señalar que esta es un área donde Microsoft ha sido muy comunicativo. Se ha criticado pero el silencio, pero con WinRT, la empresa ha sido transparente, aunque por supuesto el tema es complejo y confuso.” (Microsoft, 2013)

Como un sistema operativo moderno, Windows tenía un motor de tiempo de ejecución el cual ya lo disponía en sus antecesores, un entorno en el que se ejecutan las aplicaciones nativas.

El shell es la aplicación nativa especial del sistema operativo Windows 8, proporciona la interfaz con la que los usuarios interactúan con el tiempo de ejecución y el medio ambiente,

La pantalla de inicio es el Shell de explorer de Windows 8, es el nuevo shell de Windows 8, es el perfil visual de un nuevo motor de tiempo de ejecución que en este caso se llama WinRT.

WinRT adapta a Windows en un nivel muy profundo, Gracias al trabajo orquestado que comenzó en la previa de Windows Vista a mediados de los años 2000, Windows es ahora muy por componentes, una técnica de separación por módulos en el que los diferentes componentes, se pueden mezclar y combinar, como un rompecabezas con varias soluciones, para formar diferentes versiones de Windows.

En la base de este punto de vista arquitectónico de Windows es Windows Core, o lo que antes se llamaba MinWin. Esto incluye el núcleo de Windows, a modo de usuario, servicios, y otras cosas de bajo nivel. WinRT³, no se basa completamente sobre Windows Core. En cambio, gran parte de WinRT es parte de Windows Core. Esto, por supuesto, explica la afirmación de Microsoft de que el nuevo Windows en tiempo de ejecución "no es una cosa de nada" (**Microsoft, 2013**). Es el motor de tiempo de ejecución de Windows, puro y simple.

La ventana de tiempo de ejecución es también compuesta por capas que sirven para los niveles de funcionalidad para el Shell y para aplicaciones de estilo de Metro. En la parte inferior, conceptualmente, son los servicios de bajo nivel, tales como la gestión de la memoria, la criptografía y la globalización, los cuales se consideran parte del tiempo de ejecución de Windows. Por encima de ellos son capas relacionadas con dispositivos, medios de comunicación y las comunicaciones y los datos (contactos, almacenamiento local y la nube, web, notificaciones, y mucho más). Y por encima de eso es la capa de interfaz de usuario, con los diferentes modelos de desarrollo de aplicaciones WinRT apoyados: HTML5/CSS, XAML, DirectX, y similares.

Los programadores manejan las API para cada una de estas capas, que existen en el SDK para crear aplicaciones estilo Metro. Estas API son, por ahora, horriblemente documentado, así que es poco claro qué cantidad de aplicaciones de alta calidad se toman de las manos de Microsoft.

³ WinRT es la abreviación de Windows Runtime

“La aceptación de la afirmación de Microsoft de que WinRT es de hecho el nuevo motor de tiempo de ejecución de Windows, la pregunta obvia es: ¿Por qué? ¿Por qué Microsoft reemplaza algo que ha funcionado tan bien y con tanta eficiencia con algo que es tan nuevo, tan diferente, y por ahora, al menos esto no está probado, y para los desarrolladores, desconocido?”

Creo que la forma más sencilla de explicar esto es que Microsoft había chocado contra un muro de ladrillo con las tecnologías existentes. Para entender esto, hay que dar un pequeño rodeo en cómo el software de Windows se ha desarrollado durante los últimos 15 años.

Win32 fue creado para Windows NT a principios y mediados de 1990 - que ha sido usurpado por las API más modernas en varias ocasiones, empezando por los MFC basado (Microsoft Foundation Classes), y más recientemente el código administrado, NET C entornos basados incluido él. NET Framework, Windows Presentation Foundation y Silverlight”. (SANCHEZ, 2012)

Con la nueva API WinRT, los programadores pueden interactuar con las lenguas modernas como .NET y su código administrado C# y Visual Basic junto con el lenguaje de marcado XAML y lo familiar, limpio, y moderno gestionando APIs de estilo de código. El desarrollador se beneficia de más de una década de mejoras a estos tipos de interfaces de programación. Win32. El punto es que Microsoft está modernizando todo aquí: El motor de tiempo de ejecución y la interfaz para los usuarios, también las herramientas de desarrollo, lenguajes, APIs, frameworks, el medio ambiente, y más.

Los usuarios podrán ver cambios muy significativos no solo en la forma como se interactúa con Windows sino también en las áreas en que Windows y las aplicaciones se ejecutan, no es solo una capa de nueva interfaz, es un diseño nuevo completo, sin dejar la oportunidad de seguir manejando como los modelos anteriores de Windows.

“Como con cualquier evolución, habrá dolores de cabeza en su crecimiento. Esto es lo que pasó cuando se cambió de DOS / Windows 3.x para Windows 95 y NT. Durante la transición, el DOS migró de ser el núcleo del sistema operativo a ser un motor de tiempo de ejecución integrada y, a continuación, finalmente, a ser un simple intérprete de línea de comando que corrió en la parte superior del sistema operativo "real".” (González, 2014)

En el nuevo diseño de Windows 8, el WinRT y la pantalla de inicio se establecerá junto con Win32 y el escritorio Explorer. Cada uno tiene sus fortalezas y debilidades en diferentes contextos. Win32 sencillamente no está disponible en WinRT. A pesar de algunos pequeños cambios a la API de Win32 en derivación se han introducido en Windows 8, Win32 es en gran parte un modo de mantenimiento. Todos los grandes cambios y mejoras son, y seguirán ocurriendo en WinRT.

Se espera que el Explorer quede obsoleto y eliminado con el pasar del tiempo. Y esto sucederá paulatinamente por problemas de compatibilidad. Pero WinRT y su experiencia de usuario al estilo de Metro es el futuro.

“Uno de los puntos confusos sobre WinRT de Windows 8, puedo hacer la siguiente declaración general: WinRT (el nuevo tiempo de ejecución de Windows) no es un reemplazo para Silverlight o NET, es un reemplazo para Win32. Y eso significa que es el nuevo tiempo de ejecución nativo de Windows, y no una capa de código administrado que se encuentra más arriba en la pila. Y eso significa que más allá de las capas que se sientan en la parte superior de WinRT - XAML / C # (y otros idiomas), HTML / JavaScript y DirectX son mucho menos abstraído de código nativo que se pensaba”. (Ocu, 2012)

Usualmente las aplicaciones podrán no ser escritas como una aplicación WinRT. Muchos están imaginando aplicaciones muy simples, como HTML, y aunque es seguro de que habrá un montón de ellos, tendrá que restablecer sus expectativas para arriba. WinRT es increíblemente completo y no limitado a los servicios públicos.

2.3.1 ADQUISICIÓN DE APLICACIONES

La tienda Windows es el repositorio de las aplicaciones METRO, para instalar una aplicación el desarrollador no tendrá que hacer nada en los paquetes de instalación, Windows toma los byte desde la nube los valida y los instala, y el usuario tendrá una instancia almacenada del registro, la desinstalación lo hace desde la nube mismo sin dejar huellas de la instalación anterior.

2.3.2 LA ESTRUCTURA DE UN PAQUETE DE INSTALACIÓN DE APLICACIONES

Con Visual Studio se puede crear el paquete **.appx** para instalar/desinstalar una aplicación en Windows 8. Incluye los siguientes componentes:

2.3.2.1 MANIFIESTO DE UN PAQUETE DE INSTALACIÓN DE APLICACIONES

Comprende la información para distribuir la aplicación (por ejemplo Nombre de la aplicación, datos del publicador). Se indica también los contenidos de la aplicación (por ejemplo para acceder a la cámara, micrófono, red). Esto protege la integridad del sistema.

2.3.2.2 BLOCKMAP

Incluye los códigos Hash de todos los archivos dentro del paquete de la aplicación.

2.3.3 FIRMA

Valida la integridad del paquete de la aplicación.

Al crear una aplicación se crea una firma de desarrollador y mapas de bloques que permite verificar la integridad del paquete, la validación sucede mientras se descarga, en este paquete se encuentran los recursos necesarios para su funcionamiento.

2.3.3.1 AISLAMIENTO DE PROCESOS

En Windows y las aplicaciones Metro los procesos tienen su propio contexto local y no puede ser accedido por otro proceso. Solo pueden comunicarse a través de los contratos este contrato permite que los datos de una aplicación puedan ser expuestos a otras aplicaciones consumidoras. Estos procesos puede ser la consumidora de los datos que exponen otros procesos y puede acceder a los recursos del sistema mediante las APIs del Run-time.

2.3.4 TRANSICIÓN DE ESTADO DE LOS PROCESOS

Estados de la aplicación:

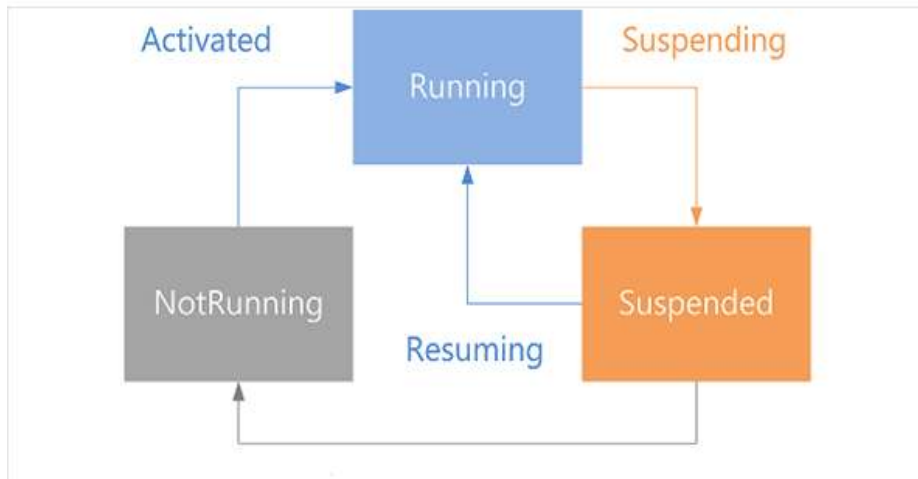


ILUSTRACIÓN 24: “Estados de la aplicación”

Fuente: Xatacala, (2012). Versiones Windows 8, Recuperado el 12 de Abril del 2013, de <http://www.xatakawindows.com/windows/windows-8-elige-entre-las-versiones-conocidas>

- No ejecución: Cuando se acaba de instalar, dejó de funcionar o fue suspendida pero no pudo mantenerse en memoria
- Activando: La aplicación se asegura que la interfaz está lista. Registra .controladores de eventos.
- Ejecución: La aplicaciones toma los recursos de memoria y permite la ... visualización del aplicativo
- Suspendiendo: Cuando el usuario cambia de aplicación o cuando Windows entra en un estado bajo de energía. Las aplicaciones detienen su ejecución. Si la aplicación se manda al fondo y transcurren 10 segundos de que el usuario no regresa a la aplicación, Windows pone la aplicación en estado de Suspensión. Se invoca el controlador de evento de “Suspendiendo” de la aplicación. El código que se ejecuta dentro del controlador de evento no debe durar más de 5 segundos, si esto sucede Windows finaliza la aplicación.
- Reanudando. Cuando pasa del estado suspendido al estado ejecución.
- Suspendido.- Windows pone en memoria la mayoría de las aplicaciones . suspendidas. Al mantenerlas en memoria se asegura que los usuarios puedan navegar rápidamente entre las aplicaciones suspendidas.

2.3.5 DATOS DE LA APLICACIÓN VS DATOS DEL USUARIO

Los datos de configuraciones, archivos, datos de sesión o cache local son exclusivos para cada aplicación Metro, no pueden acceder otras aplicaciones, pero los datos de usuarios pueden ser accedidos desde otras distintas aplicaciones, esto ayuda a sincronizar la información en la nube.

2.4 LA TIENDA DE WINDOWS 8

Windows 8 tiene su propia tienda de aplicaciones. La llamada Windows Store la cual se puso en marcha con la salida de la versión Consumer Preview del nuevo sistema operativo. Lo hizo en modo de prueba, distribuyendo aplicaciones gratuitas y dejando entender que será la fuente para distribuir las diferentes aplicaciones para nuestro uso personal y/o empresarial.

El equipo de Microsoft ha dicho que su mayor preocupación en el diseño de Windows Store era facilitar al máximo a los usuarios el descubrimiento y acceso a las aplicaciones.

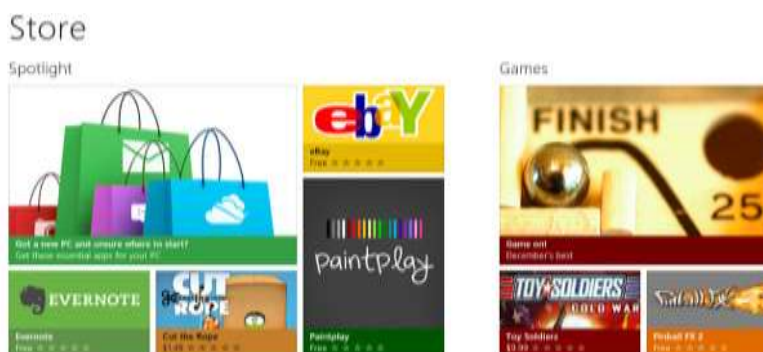


ILUSTRACIÓN 25: "Windows Store"

Fuente: http://www.aulaclie.es/guia-windows8/t_1_2.htm

La Windows Store es la fiel representación del modelo de aplicaciones al estilo metro, su forma y su distribución de las áreas reflejan la continuidad de su excelente administración de su espacio de aplicación con los menús, Title, Charms etc.

El resto de la portada está ocupado por las principales categorías que se distribuyen para poder navegar entre ellas mediante scroll horizontal. Existe la opción además de hacer zoom, out para ver más categorías de un simple vistazo. Cada categoría se agrupa mostrando unas pocas aplicaciones destacadas, de nuevo elegidas por el equipo editorial, botones para acceder a las listas de las aplicaciones más descargadas o mejor valoradas, y la opción de abrir la página de la propia categoría.



ILUSTRACIÓN 26: “Windows Store”

Fuente: http://www.aulaclie.es/guia-windows8/t_1_3.htm

2.4.1 CATEGORÍA O RESULTADO DE LA BÚSQUEDA

La distribución de las filas y columnas en pequeños rectángulos compensan el limitado número de aplicaciones visibles mostrando un elevado número de pantallas que es lo contrario a la portada.

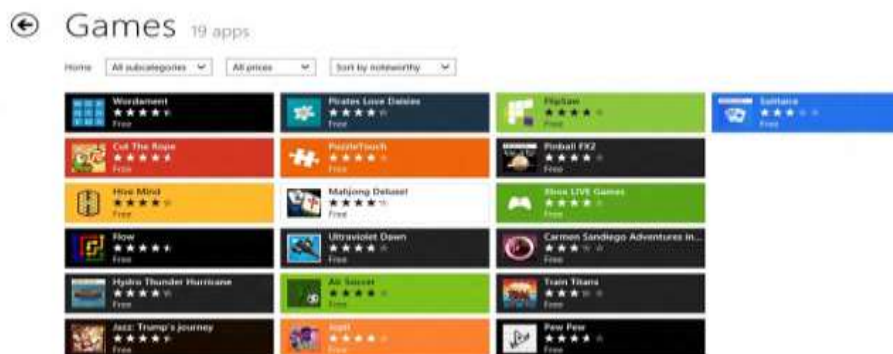


ILUSTRACIÓN 27: “Categorías”

Fuente: http://www.aulaclie.es/guia-windows8/t_1_2.htm

El scroll se mantiene en horizontal permitiendo aprovechar mejor las diagonales actuales de pantalla. Permite navegar entre todas las aplicaciones de las listas, se puede recurrir a los típicos filtros que se espera encontrar en una tienda de este tipo, diferenciando entre subcategorías, por precio y por novedad, votos, basta pulsar en la aplicación que se necesita para acceder a su página en la Windows Store.

2.4.2 LAS PÁGINAS DE APLICACIONES

Microsoft conoce que la parte que al usuario que más llama la atención es lo visual, y con esto pretende atraer a más usuarios a Windows Store, es por eso que sitúa imágenes de las aplicaciones en recuadros de considerable tamaño en la columna principal, es decir al momento de elegir una aplicación metro se abre de la misma página las capturas de la pantalla, con una breve descripción del producto y otras características de la aplicación,

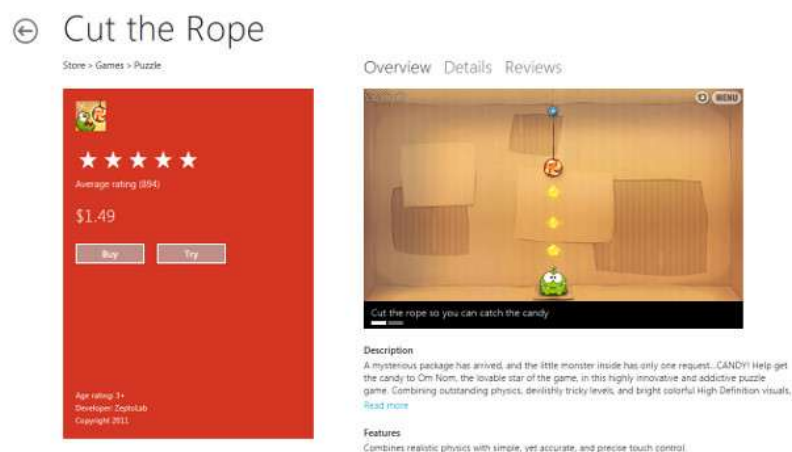


ILUSTRACIÓN 28: “Distribución de la pantalla”

Fuente: Desarrollo de Aplicaciones Windows 8. Recuperado el 15 de abril 2013, de <http://www.xatakawindows.com/windows/windows-8-la-windows-store-a-fondo/>

Pero lo significativo se encuentra en la columna de la izquierda. Seguido del nombre de la aplicación que corona la página y con un recuadro con el color de fondo identificativo de la misma presenta varias características del producto como: su icono, su puntuación, el precio, etc. También están los botones de instalación con la leyenda de si es gratuita o no. El compendio de información lo remata la esquina inferior izquierda del recuadro para consultar el nombre del desarrollador y la edad recomendada.

2.4.3 INSTALAR APLICACIONES

La estrategia que está utilizando Microsoft con su nuevo Windows 8 es la de transmitir su pureza y sencillez en forma visual, y de esta manera aumentar las compras y descargas de sus aplicaciones, apreciando la sencillez de su instalación sin los tediosos botones de siguiente y las condiciones de leído.

La instalación ya no tendrá los múltiples y simpáticos clics, todo está resumido en un elegante botón instalar ahora, y esta se descarga y automáticamente continúa su instalación, de tal manera de darle al usuario el mínimo trabajo.

Ya no será necesario ir paso a paso con cada instalación. Además todo el proceso de descarga e instalación se realiza en segundo plano con lo que se evita estar pendientes y dedicar nuestro tiempo a seguir navegando por la Store o continuar con tareas pendientes.

2.4.4 GUARDAR Y ACTUALIZAR APLICACIONES

Las actualizaciones son aún más sencillas, se asemeja en el modelo de los SmartPhones, cuando esté una actualización disponible simplemente notifica al usuario en el tile, y si lo desea automáticamente lo instala en segundo plano.

Para las aplicaciones que se adquiere, la Windows Store permitirá que se instalen en hasta cinco dispositivos con Windows 8 diferentes, sean PC, tableta, híbridos, o de cualquier tipo con el que sorprendan los fabricantes. Con nuestra cuenta de usuario estará asociada una lista de los dispositivos que se usará para cada aplicación, pudiendo quitar y añadir otro cuando se llegue al tope. Eso sí, todas nuestras aplicaciones estarán sincronizadas entre dispositivos para tenerlas siempre disponibles sin que importe el que se utilice

2.4.5 SEGURIDAD Y CONTROL PARENTAL

La Windows Store es configurable en su control parental, de tal forma de brindar seguridad a los usuarios menores de edad, para los cuales se podrá limitar las aplicaciones permitidas.

Para poder subir una aplicación a la Windows Store, debe pasar por un proceso de certificación de Microsoft, y de esta manera están asegurando que el software que estas instalado sea un producto bueno y no sea software maliciosos.

2.5 DISEÑO DE APLICACIONES CON LOS PRINCIPIOS Y PERSONALIDAD DEL NUEVO ESTILO DE INTERFAZ DE USUARIO

2.5.1 DISEÑO DEL NUEVO ESTILO DE INTERFAZ DE USUARIO

Metro Style es un nuevo tipo de interfaz que brinda Windows 8, esta interfaz permite su ejecución en diferentes dispositivos de forma elegante y atractiva para el usuario, siendo estas el núcleo de la nueva experiencia de usuario de Windows 8.

Metro es una interfaz que el usuario maneja para interactuar con el sistema operativo en pantalla

La idea de crear una nueva interfaz es para dar más funcionalidad y compatibilidad al Sistema Operativo con los diversos dispositivos portátiles, con la premisa de usar menos recursos y se cuide el medio ambiente, por eso la interfaz es más sencilla, pura y accesible diferente a sus anteriores versiones.

Metro contiene una tipografía atractiva y funcional, todo de acuerdo con un peso y posición exacta, para dar una jerarquía visual complementada con un contenido sin ningún tipo de adorno y siempre mostrando la información más objetiva.

El estilo metro, es sin duda la revolución de las interfaces, pues permite mayor fluidez y rapidez en la navegación del PC con el usuario además de ser muy interactiva.

Un claro ejemplo de esta revolución es la inclusión de este nuevo estilo a varias aplicaciones y empresas de renombre que han cambiado su logo para que lleve este estilo, entre las más destacadas están: Microsoft, Firefox, Office.



ILUSTRACIÓN 29: “Logo en estilo metro”

Fuente: MSDN, (2012). Principios Diseño Metro. Recuperado el 28 de octubre del 2012, de <http://msdn.microsoft.com/es-es/library/windows/apps/hh464920.aspx>

2.5.2 PILARES DEL DISEÑO METRO

Son 3 los fundamentos del estilo metro, el diseño moderno, el estilo Tipográfico Internacional y el diseño en movimiento. Con esta premisa Microsoft crea su estilo metro, añadiendo algunas directrices para su complementación con el mundo digital

El estilo metro tiene 3 pilares fundamentales:

- **El diseño moderno** .- Este estilo se ideó para la industrialización de productos y se centra en reducir todos los elementos decorativos innecesarios para darle funcionalidad
- **El estilo Tipográfico Internacional**.- Este estilo centra a la tipografía como su elemento central y además toma características del diseño moderno de Bauhaus
- **El diseño en movimiento**.- Este tipo de diseño ha ido creciendo en los últimos años y se centra en dar vida al resultado de diseño. Este estilo de diseño da dinamismo y movimiento a los elementos haciéndolos más reales

2.5.3 PRINCIPIOS Y PERSONALIDAD DEL ESTILO METRO

2.5.3.1 SACAR PROVECHO AL ESTILO DE DISEÑO MICROSOFT

- **El Contenido**.- Lo que Microsoft ha puesto más empeño es al contenido como lo más importante de la aplicación y muestra la información sustancial para el beneficiarios.
- **Presentación del contenido**.-Mostrar el contenido antes del contenedor visual es vital en las aplicaciones metro, ya que todo lo diferente al contenido son solo accesorios visuales que permiten mostrar el contenido.
- **Diseño limpio y abierto**.-Un diseño claro y sincero es el que predomina en metro, la pantalla debe de contener los elementos más relevantes, eliminando distracciones como líneas, cuadros y efectos gráficos para concentrarse en el contenido.
- **Jerarquía en la información**.-La comparación del contenido es muy importante en el contenido, ya que ayuda a que el usuario pueda identificar qué información es más relevante que otra, esto se logra con una tipología limpia, utilizando diferentes tamaños, colores y espesores en el texto.

- **Animaciones con Propósito** Las aplicaciones metro deben de responder a las acciones del usuario rápidamente y con fluidez para que mantenga el principio de este estilo.

Utilizar animaciones con un propósito y no solo como adorno visual hace que las aplicaciones tengan vida, ofreciendo al usuario una experiencia elegante.

Una aplicación estilo metro utiliza animaciones con el objetivo de mejorar su presentación visual brindando la usuario una verdadera experiencia.

- **Diseño táctil.**-Todas las aplicaciones deberían comúnmente ser desarrolladas con vista a lo touch para utilizar el lenguaje táctil del sistema operativo Windows 8.
- **Ajusta y cambia la escala con elegancia.**- Existe una variedad de tamaños y formas de pantallas en Windows 8 lo cual permite usar controles automáticos que configuran la resolución de la aplicación, hay controles que se debe poner mucha atención como lo son: ListView, GridView, FlipView.

Las aplicaciones deben ser funcionales que mantengan sin distorsionar las formas y tamaños de los objetos, las herramientas de Visual Studio permiten mantener este principio de las aplicaciones metro.

Las aplicaciones estilo metro también deben tener la posibilidad de ajustarse con otras aplicaciones y distribuir el tamaño, ya que el usuario puede tener una o más aplicaciones abiertas al mismo tiempo.

- **Usa los contratos adecuados**

Los contratos son una de las herramientas que ayudan a realizar las tareas más comunes como lo son: buscar, compartir, selectores de archivos.

- **Buscar.**-Busca el contenido de una aplicación desde Windows, se puede o se debe suministrar información para que la búsqueda filtre y busque más rápidamente.
- **Compartir.**- Este contrato comparte datos de una aplicación con otra.
- **Selector de Archivos.**- Brinda la oportunidad al usuario para acceder al contenido de una aplicación desde otra.
- **Invierte en un icono excepcional.**- El icono de las aplicaciones muestran el uso de la misma y debe tener información resumida de la aplicación, esta debe ser muy agradable a la vista del usuario.

Si se desea implementar un icono Visual Studio brinda las herramientas necesarias para hacerlo incluyendo plantillas y se las puede personalizar de acuerdo con la aplicación que se esté diseñando.

- **Notificaciones.-** Las notificaciones juegan un papel importante en las aplicaciones metro, reflejan un ambiente conectado y vivo a través de la creación y visualización de iconos dinámicos.
- **Iconos Dinámicos.-** Personaliza el icono con contenido notable y afanoso, que sea sencillo y a la vez importante.
- **Perfiles Móviles en la Nube.** Si es necesario dejar una aplicación anclada al escritorio y poder retomarla luego de un tiempo el perfil móvil nos permitirá hacerlo sin ningún contratiempo sacando provecho de la cuenta Microsoft y su almacenamiento en la nube.

2.5.3.2 APLICACIONES WINDOWS 8 ESTILO METRO

- **Adopta los principios de estilo de diseño Microsoft.**

Las aplicaciones metro se las desarrolla basándose en directrices que Microsoft brinda para este objetivo estos son:

- ✓ Muestra buen hacer.
- ✓ Haz más con menos
- ✓ Sé rápido y fluido
- ✓ Sé auténticamente digital
- ✓ Gana en equipo

Microsoft ha revolucionado su estilo en sistemas operativos con su estilo metro, el cual se considera muy diferente a lo presentado anteriormente, además este sistema operativo se abre campo en el desarrollo de aplicaciones para Windows 8, y así dar una forma de fuente de ingreso para desarrolladores.

En un futuro las aplicaciones para Windows 8 pueden ser las más usadas y descargadas en todo el mundo, por ello la importancia de ya ir conociendo las teorías de cómo desarrollar estas aplicaciones y así lograr que el país también proporcione aplicaciones de este tipo.

- **Oportunidades brindadas a los desarrolladores**

Si la idea es crear software para mejorar tus ingresos Microsoft brinda la oportunidad de hacerlo y con ventajas como:

- ✓ **Windows Store.-** La tienda online de Microsoft brinda la oportunidad de subir tus aplicaciones y venderlas a un costo razonable y además dar a conocer tus productos desarrollados.
- ✓ **Tamaño del Mercado.-** Windows es reconocida por sus estrategia de mercadeo la cual ha brindado la oportunidad de pasar a hacer la primera empresa en generar millones de dólares en venta de sus productos como principal venta son las licencias.
- ✓ **Ventas Windows 8.-** “Al tercer día de su salida se reveló que se han vendido 4 millones de licencias superando las expectativas tomando en cuenta que no se insertó en el informe las empresas privadas” (**xatakawindows, 2013**)

Es evidente por las estadísticas, que desarrollar aplicaciones metro para Windows 8 es una gran oportunidad de negocio y muy rentable.

2.5.3.3 CARACTERÍSTICAS

- **Las aplicaciones tienen una ventana que admiten varias vistas.**

Todas las aplicaciones permiten su presentación en diferentes diseños y vistas además de su forma y tamaño pues están diseñadas para diferentes dispositivos móviles debido a que la aplicación tiene una sola ventana sin contenedor visual que llena toda la pantalla y hace más fácil su interacción con el usuario.



ILUSTRACIÓN 30: “Vistas de varias ventanas”

Fuente: MSDN, (2012). Principios Diseño Metro. Recuperado el 28 de abril del 2013, de <http://msdn.microsoft.com/es-es/library/windows/apps/hh464920.aspx>

- **Las aplicaciones funcionan muy bien con entradas táctiles y de puntero**

Los periféricos de entrada son completamente funcionales con las aplicaciones metro así como lo son las dispositivos tradicionales que utilizan mouse y teclado.

Metro incorpora estilos predeterminados que garantizan que los elementos de la interfaz funcionen correctamente con escenarios táctiles.



ILUSTRACIÓN 31: “Movimiento táctil en la pantalla”

Fuente: MSDN, (2012). Principios Diseño Metro. Recuperado el 28 de Abril del 2013, de <http://msdn.microsoft.com/es-es/library/windows/apps/hh464920.aspx>

- **Las aplicaciones tienen nuevos controles y superficies de interfaz de usuario**

La barra de la aplicación y los botones de acceso a la aplicación, son controles exclusivos de las aplicaciones metro las cuales dan vida a la experiencia de los usuarios de la aplicación.

- **La barra de la aplicación**

Después de la ventana de aplicación, la barra de la aplicación supone la interfaz de comandos primordial de la aplicación. Esta barra es utilizada para exponer la navegación, los comandos y las herramientas a los usuarios.

La barra de la aplicación está oculta de manera predeterminada y se muestra cuando los usuarios deslizan un dedo desde el extremo superior o inferior de la pantalla. Cubre el contenido de la aplicación y el usuario puede descartarla al deslizar el dedo rápidamente por el extremo o al interactuar con la aplicación.



ILUSTRACIÓN 32: “Logo en estilo metro”

Fuente: MSDN, (2012). Principios Diseño Metro. Recuperado el 28 de abril del 2013, de <http://msdn.microsoft.com/es-es/library/windows/apps/hh464920.aspx>

- **Los botones de acceso**

Los botones de acceso son un conjunto específico y coherente de botones que son comunes a todas las aplicaciones: buscar, compartir, conectar, configuración e inicio. Las acciones básicas que los usuarios pueden realizar con estos botones son:

- ✓ Buscar contenido de la aplicación desde otra aplicación.
- ✓ Compartir contenido de la aplicación con personas o servicios.
- ✓ Ve directamente a la pantalla de inicio.
- ✓ Conectarse a dispositivos y enviar contenido, transmitir medios por secuencias e imprimir.
- ✓ Configurar la aplicación según sus preferencias.



ILUSTRACIÓN 33: “Acceso rápido”

Fuente: MSDN, (2012). Principios Diseño Metro. Recuperado el 28 de abril del 2013, de <http://msdn.microsoft.com/es-es/library/windows/apps/hh464920.aspx>

- **Las aplicaciones usan iconos dinámicos**

Los iconos dinámicos que admiten una mejor interacción con el usuario, permite el ingreso del contenido de la aplicación incluso cuando no se esta usando pero si se encuentre activa, presenta datos relevantes de la aplicación en una sola vista.



ILUSTRACIÓN 34: “Animación de iconos”

Fuente: MSDN, (2012). Principios Diseño Metro. Recuperado el 28 de abril del 2013, de <http://msdn.microsoft.com/es-es/library/windows/apps/hh464920.aspx>

2.6 DESARROLLO DE APLICACIONES CON EL NUEVO ESTILO DE INTERFAZ DE USUARIO CON XAML

2.6.1 WINRT PARA EL DESARROLLO .NET

2.6.1.1 PLATAFORMA WINRT

WinRT en su términos técnicos es “Windows Runtime” es un conjunto de biblioteca de software que facilita una API de gráficos, redes y todos los recursos necesarios para que las aplicaciones metro accedan al kernel del sistema brindando una velocidad extraordinaria en la ejecución de los procesos de una manera sencilla, cabe señalar que su acceso es de forma nativa.

Como WinRT no es un componente de .NET logra que la comunicación con los objetos de la plataforma sea más directa y sencilla logrando que los DLLIMPORT del win32 sean innecesarios.

El COM es el único componente del WinRT pero mejorado en lo referente a los objetos, y no en sus interfaces en los cual COM está basado.

La plataforma WinRT se ha desarrollado para trabajar sobre el sistema operativo Windows 8 y en futuros sistemas operativos de la empresa Microsoft por ello se define como la plataforma recomendada para las aplicaciones metro.

WinRT contiene 33 APIs nativas las cuales están divididas en 5 grupos:

- APIs Fundamentales para proporcionar el servicio a las aplicaciones.
- APIs Media para efectos visuales y multimedia
- APIs Devices para manejo de dispositivos de comunicación y sensores
- APIs para Comunicaciones y Datos
- APIs para Interfaz de Usuario

Con estas APIs se pueden acceder al core del sistema haciendo que esta plataforma sea una base sólida y eficiente para desarrollar Metro Style Apps.

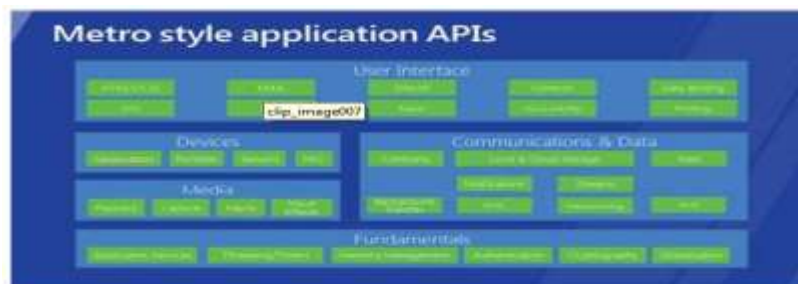


ILUSTRACIÓN 35: “Api metro style”

Fuente: <http://social.msdn.microsoft.com/Forums/windowsapps/enUS/home?categorywindowsapp>



ILUSTRACIÓN 36: “Elementos de la plataforma Windows 8”

Fuente: <http://social.msdn.microsoft.com/Forums/windowsapps/en-US/home?category=windowsapps>

Se puede describir el grafico anterior de la siguiente manera:

- La base o la parte inferior son los servicios fundamentales de W8 (Kernel)
- En la parte derecha están las aplicaciones de entorno del escritorio desarrolladas en los diferentes lenguajes de programación(HTML, c, C++), las cuales acceden a las API de win32
- La parte central le pertenece al WinRT y sirve para las APIs de comunicación, datos, multimedia, etc.

WinRT es la base de las aplicaciones metro y se puede utilizar lenguajes como C, C++, C#, VB para la lógica de la aplicación y XAML para implementar la interfaz. Para los desarrolladores web se utiliza JavaScript para la lógica y HTML/CSS para la interfaz.

WinRT utiliza Application Model que es un modelo de aplicaciones en el cual cada una de las aplicaciones es una clase quien contiene otras clases, las cuales son instanciadas por WWAHost.exe que es el que gestiona las funciones del S.O. como la suspensión.

Las aplicaciones utilizan un WWAHost para ser instanciadas y lo hacen a través del ejecutable

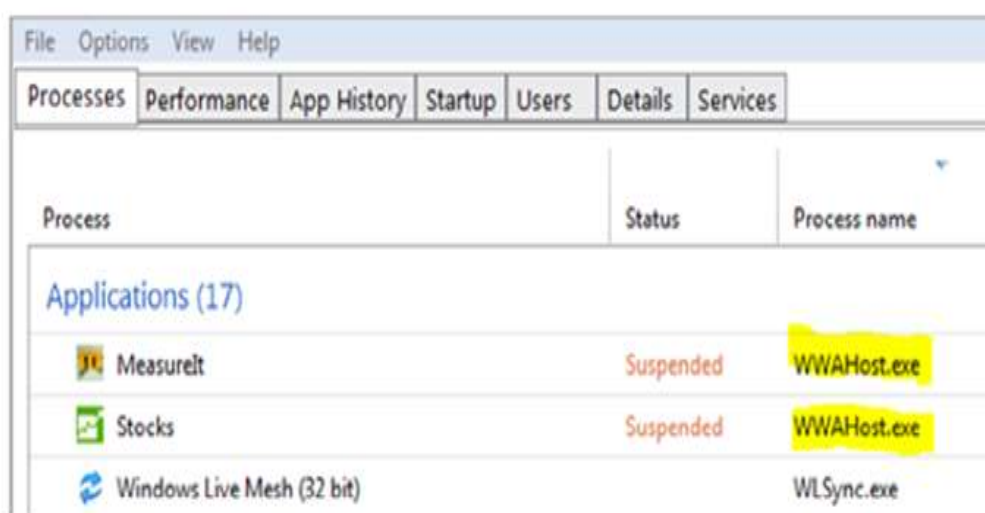


ILUSTRACIÓN 37: "Task manager"

Fuente: <http://social.msdn.microsoft.com/Forums/windowsapps/en-US/home?category=windowsapps>

Este nuevo modelo accede a los recursos del sistema y librerías de forma rápida y sencilla las cuales ayudan en el desarrollo de las aplicaciones metro, porque tienen conectadas todas las APIs directamente al kernel, y de esta manera permite el uso de más lenguajes de programación para la creación de aplicaciones metro.

2.6.1.2 LENGUAJES DE DESARROLLO SOPORTADO

La plataforma WinRT admite lenguajes de programación como C++, JavaScript y lenguajes .NET como VB.NET y C#. WinRT también permite que se pueda desarrollar aplicaciones para el chip Intel como para el chip ARM, este último para Tablets y demás dispositivos que tienen limitado sus recursos y energía.

Además en esta plataforma se ha ampliado la utilización de XAML ya que se puede crear aplicaciones en WPF20, en Silverlight2.1 y también HTML y JavaScript, así se abre nuevas opciones y posibilidades para programar.

2.6.1.3 WINRT Y METRO PARA DESARROLLO

WinRT se ampara de las aplicaciones metro en su diseño por su flexibilidad, el uso de dispositivos externos y no solo táctiles, de igual manera HTML5 está muy relacionado con la plataforma WinRT Windows 8, pues estas aplicaciones dependen mucho de HTML5 y cambiarlo a otros sistemas puede resultar muy complejo.

Una de las características principales de las aplicaciones metro es que son asincrónicas es decir que la comunicación entre el usuario y el App son rápidas, si cumple esta característica y entre otras la aplicación será certificada de forma correcta.

En tiempo de ejecución las aplicaciones metro son responsables de sus recursos, es decir cada aplicación tiene un espacio de memoria reservada para su necesidad, haciéndola independiente del resto de memoria, de esta forma cada aplicación no interfiere con otras aplicaciones que se estén ejecutando.

2.6.1.4 WINRT Y .NET

WinRT es independiente de .NET por lo cual no interfiere en ningún proceso de .NET haciendo que no sea una amenaza en sus procesos, pues cada una tiene sus propios papeles en interrelación con Windows 8.

Las APIs de win32 se mantienen en W8 pero ya no con la misma participación en sus anteriores versiones de Windows con esto se consigue que WinRT tiene camino libre para dar otra forma de desarrollo de aplicaciones, si se refiere al ahorro de recursos se puede decir que WinRT es la mejor manera de hacerlo, pero si se desea crear aplicaciones metro de escritorio se debe utilizar las APIs que proporciona el Win32 applications

2.6.2 PROGRAMACIÓN ASINCRÓNICA

La última versión de .NET Framework (4.5) y de los lenguajes de la plataforma de desarrollo Visual Studio (C# y Visual Basic) es la introducción del concepto de programación asincrónico.

La programación asíncrona se viene utilizando habitualmente, principalmente desde la llegada de Windows Forms en .NET.

La programación Asíncrona tiene como objetivo sacar provecho de los nuevos modelos de microprocesadores y de sus varios núcleos, esta programación es muy compleja y diferente al modelo sincrónico que habitualmente se lo usaba

El nuevo modelo de programación incluido en la última versión de C# (y Visual Basic) y de .NET Framework trata de simplificar y unir los dos modelos. Este nuevo modelo no es la única manera de afrontar la problemática de la programación asíncrona en .NET, pero el modelo propuesto es el más sencillo y el más recomendable en la mayoría de los casos.

Los métodos asincrónicos son la base del nuevo .NET frameworks por lo que es necesario incorporarlos para que las aplicaciones desarrolladas invoquen los recursos necesarios para su utilización. Recordar que no es una opción sino es una necesidad

2.6.2.1 ¿QUÉ ES MÉTODO ASINCRÓNICO?

La programación Asíncrona permite ejecutar las aplicaciones sin pérdida de tiempo, por ejemplo : si una aplicación necesita descargar un archivo y luego leerlo, si esta aplicación fuera diseñada con métodos sincrónicos la aplicación tomaría sus recursos y los utilizaría en su propósito haciendo que el usuario del aplicativo se frustrara con el tiempo de demora pues la aplicación no responderá a otras peticiones, es decir se bloquea hasta recuperar la información, es por eso que las aplicaciones metro utilizan métodos asincrónicos de tal manera que la App sigue ejecutándose y respondiendo a la interfaz de usuario mientras espera que la otra petición se termine por ejecutar.

Para aquellos métodos que posiblemente demanden mucho tiempo en finalizar, debe usarse la programación asíncrona y no la excepción en Windows en tiempo de ejecución. JavaScript, C#, Visual Basic y C++ proporcionan compatibilidad de idioma para los métodos asíncronos.

2.6.2.2 PROGRAMACIÓN ASÍNCRÓNICA EN WINDOWS EN TIEMPO DE EJECUCIÓN

Cuando se usan API asíncronas en la aplicación de la Tienda Windows, el código realizará llamadas de desbloqueo de manera coherente. Cuando se implementa estos modelos asíncronos en la API, los autores de las llamadas podrán comprender y usar código de manera predecible.

Algunas tareas comunes que requieren llamar a la API de Windows en tiempo de ejecución asíncronas pueden ser:

- Mostrar un cuadro de diálogo de mensaje
- Trabajar con el sistema de archivos
- Enviar datos a Internet y recibirlos

Con las API asíncronas de Windows en tiempo de ejecución, no necesita administrar subprocesos de manera explícita ni interactuar directamente con la implementación subyacente.

Cada lenguaje de programación admite el modelo asíncrono a su manera:

TABLA 1: “Programación Asíncrona”

Lenguaje de programación	Representación asíncrona
JavaScript	objeto de promesa, función then
C#	objetos futuros, operador await
Microsoft Visual Basic .NET	objetos futuros, operador Await
Visual C++	clase task , método .then

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh464924.aspx>

2.6.2.3 MODELOS ASINCRÓNICOS EN APLICACIONES DE LA TIENDA WINDOWS CON JAVASCRIPT

Los métodos asincrónicos son utilizados frecuentemente en el estándar de Javascript en los eventos de promesa, estas se usan en Windows en tiempo de ejecución como en sus bibliotecas.

Los objetos promesa es un valor que no se ve reflejado en ese momento sino hay que esperar un tiempo para su presentación, estos normalmente tienen sus nombres que finalizan en async en muchos casos controlar un método o llamar un método asincrónico es bastante sencillo, a diferencia de cuando necesitas usar las funciones then o done para signar los controladores para comunicar errores o para iniciar operaciones.

2.6.2.4 MODELOS ASINCRÓNICOS EN APLICACIONES DE LA TIENDA WINDOWS CON C# Y MICROSOFT VB

Un fragmento de código escrito en C# o Visual Basic se ejecuta sincrónicamente, significa que cuando se ejecuta una línea, finaliza antes de que se ejecute la siguiente.

Con los años se han usado diversos modelos de programación como:

- Modelo de programación asincrónica
- Modelo asincrónico de eventos

Esto para permitir a los desarrolladores escribir código asincrónico. Si bien estos modelos han permitido la ejecución asincrónica, el código resultante tiende a enfatizar la mecánica de la ejecución del código asincrónico en lugar de centrarse en la tarea que el código está tratando de realizar. Esto ha hecho que sea difícil escribir, comprender y mantener el código asincrónico. Los compiladores de Windows en tiempo de ejecución, .NET Framework y C# y Visual Basic tienen características adicionales que separan la mecánica asincrónica del código para poder escribir código asincrónico que se centre en lo que el código realiza en lugar de en el modo en que lo hace. Ahora se puede escribir código asincrónico cuya apariencia sea similar a la del código sincrónico.

2.6.2.5 MODELOS ASINCRÓNICOS EN APLICACIONES DE LA TIENDA WINDOWS CON C++

Visual C++ aplica la Programación Asíncrona basada en Task class derivada de su función then. El concepto es similar a las promesas de JavaScript, esta clase task class ofrece la función de cancelación y administración de los subprocesos.

La create_async es compatible con las APIs asíncronas para ser consumidas desde js o cualquier otro lenguaje admitido por Windows en tiempo de ejecución.

2.6.2.6 CONTROL DE ERRORES

Cuando se utiliza el operador **await** para recuperar resultados de un método asíncrono, si en una aplicación necesitas capturar el error ocasionado o producido por el operador await en un método asíncrono se debe o puede usar el bloque try/catch de la misma forma como se lo haría en métodos sincrónicos, cuando existen métodos asíncronos anidados en bloque try/catch deberá ir en el método externo para capturar los errores de todos. Una vez más, esto es similar al modo a capturar excepciones para los métodos sincrónicos.

2.6.2.7 RESUMEN Y NEXT STEPS

Convencionalmente, a los métodos asíncronos se les asignan nombres terminados en "Async".

El operador **await** debe tener la declaración marcada con la palabra clave **async**, que diferencia de los demás métodos.

Cuando la aplicación encuentra el operador **await**, la comunicación de la aplicación con el usuario no se ve interrumpida y se sigue ejecutando como si estuviera en segundo plano

Un método asíncrono devuelve un objeto que contiene su resultado. En la mayoría de los casos, lo útil es el resultado del valor devuelto, no el valor devuelto en sí mismo.

2.7 DESARROLLO DE APLICACIONES DEL NUEVO ESTILO DE INTERFAZ DE USUARIO CON HTML Y JAVASCRIPT

2.7.1 HTML 5.0

HTML 5 es el nuevo concepto de la elaboración de proyectos web trabaja satisfactoriamente en ambientes de dispositivos móviles y en la nube, su forma de comunicación es más segura y sencilla de aplicar

El proyecto comenzó hace mucho tiempo atrás con una versión sin complejos componentes estrictamente diseñada para formar la parte básica de las páginas web, organizar los contenidos y su información, todo para poder presentar información de texto en la web.

Como HTML era muy limitado varias empresas se enfocaron a mejorar este producto y desarrollar nuevos lenguajes y programas para agregar nuevos componentes al HTML y con eso nuevas características a la web, de esta manera comenzó una carrera sin final y se crearon varios y nuevos accesorios para la web.

Algunos juegos y animaciones simples se convirtieron en fantásticas aplicaciones que mejoraron significativamente la experiencia y el concepto web, de estas opciones se destacaron Java y Flash las dos fueron rápidamente adoptadas y consideradas el futuro de la web, sin embargo como todo paso muy rápido y el futuro de la web iba mucho más allá de tal forma que se convirtió en un proceso determinante para estos dos componente que les toco reinventarse.

El mayor inconveniente de Java y Flash puede describirse como una falta de integración. Ambos fueron concebidos desde el principio como complementos (plug-ins), algo que se inserta dentro de una estructura pero que comparte con la misma solo un espacio en la pantalla. No existía comunicación e integración alguna entre aplicaciones y documentos.

“La falta de integración resultó ser crítica y preparó el camino para la evolución de un lenguaje que comparte espacio en el documento con HTML y no está afectado por las limitaciones de los plug-ins. JavaScript, un lenguaje interpretado incluido en navegadores, claramente era la manera de mejorar la experiencia de los usuarios y proveer funcionalidad para la web. Sin embargo, después de algunos años de intentos fallidos para promoverlo y algunos malos usos, el mercado nunca lo adoptó plenamente

y pronto su popularidad declinó. Los detractores tenían buenas razones para oponerse a su adopción. En ese momento, JavaScript no era capaz de reemplazar la funcionalidad de Flash o Java.

A pesar de ser evidente que ambos limitaban el alcance de las aplicaciones y aislaban el contenido web, populares funciones como la reproducción de video se estaban convirtiendo en una parte esencial de la web y solo eran efectivamente ofrecidas a través de estas tecnologías.” (Rendon, 2012)

Con el tiempo java declino, pues su compleja estructura del lenguaje su lenta evolución y la poca capacidad de integrarse con otros lenguajes disminuyeron su importancia al punto de no ser utilizado en grandes proyectos web.

Los navegadores también debieron adquirir cambios significantes sobre todo en sus intérpretes de Javascript, esto trajo consigo mejores días para los lenguajes web.

Con el pasar del tiempo surgieron nuevos retos para los desarrolladores web a tal punto que java y flash no abastecían sus necesidades, sobre todo por el gran número de usuarios que ingresaban a sus aplicaciones, de esta manera nace el concepto de Web 2.0 para suministrar en gran medida JavaScript a sus aplicaciones.

En los últimos años los programadores implementaron varios e increíbles componentes para satisfacer los límites de java y flash, así como sus deficiencias iniciales en portabilidad, de esta manera JavaScript, HTML y CSS se convirtieron pronto en la más perfecta combinación para la necesaria evolución de la web

HTML 5 es una tecnología mejorada mezclando las dos tanto java como flash, con estándares que ayudan al aspecto de la web, es decir desde hoy en adelante HTML integra elementos estructurales, donde los CSS serán los encargados de la elegancia y crear un ambiente atractivo a la vista y java script lo volverá dinámico.

2.7.2 CONCEPTOS BÁSICOS

Las tres características básicas de HTML son en esencia de su relación con las APIs, estas son estructura, estilo, y funcionalidad. HTML es considerado la combinación perfecta de HTML, CSS, JAVASCRIPT, HTML 5 las convirtió dependiente una de otra y actúan como una sola unidad organizada y bajo especificaciones de HTML5 donde HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto.

La estructura fue y es la parte esencial de un documento web, la cual provee los componentes necesarios para la correcta ubicación del contenido sea este estático o dinámico, y es la base de las aplicaciones, considerando que en estos tiempos existen varios dispositivos móviles es necesario considerar que la estructura se vuelve vital para el documento, esta provee organización, flexibilidad, y forma y debe ser fuerte pues es la base de la aplicación.

2.7.3 ESTRUCTURA DEL CUERPO

Dentro del cuerpo del documento siempre se exige organizar en áreas la presentación, HTML ofrece un elemento para cumplir este objetivo la `<table>` lo cual permite acomodar toda la información que se a presentar como: datos, texto, imágenes a través de su distribución de filas y columnas.

Los primeros pasos del diseño web, las tablas fueron el logro más grande, era un paso hacia el futuro y aún queda más por explorar y era la única forma de dar formato a la visualización de los documentos y la presentación hacia el usuario, con el pasar de los tiempos nuevos y mejorados elementos fueron reemplazando las tablas en sus funciones y reduciendo código escrito en el documento y facilitando la creación y organización de los contenidos , permitiendo la portabilidad y el proceso de mantenimiento de los sitios web.

Las Webs cada día fueron creciendo en contenido y se volvieron interactivas esto gracias a su integración con HTML,CSS, Javascript de ahí que se necesita su organización del código, para lo cual se utiliza los `<div>` que en práctica es lo mismo que una `<table>` su problema es no tener información acerca de cómo está organizado las partes del cuerpo del documento que está presentando, todos los elementos necesarios para la creación del sitio web puede ir dentro de una etiqueta `<div>` `</div>` el `</div>` es una división del cuerpo parecida o semejante a la celda en una `<table>` pero no especifica el propósito o que contiene.

Para el usuario este proceso es transparente, no le interesa que sucede dentro de la estructura del sitio, pero para los navegadores es y juega un papel importante dentro del documento, la organización de su contenido puede dar uno u otro significado al proceso, con el avance tecnológico de dispositivos móviles y las diferentes formas de acceso a la web la identificación y ubicación de cada parte del cuerpo del documento es una tarea realmente importante y relativa.

Para este motivo HTML 5 incorpora los nuevos elementos para la identificación y ubicación de la sección y áreas que organizan el contenido del cuerpo y su organización,

De aquí en adelante la forma de usar estos elementos queda a criterio del desarrollador, pero su uso adecuado ayudan a que sus funciones permitan una mayor y mejor utilización de sus áreas y por ende de su contenido, pues normalmente un sitio web está dividido por áreas la cual se presenta al usuario para facilitar la interactividad, las palabras claves en HTML es la que representa con la áreas.

2.7.4 NUEVOS Y VIEJOS ELEMENTOS

La idea principal de HTML 5 y su creación son con el propósito de simplificar, y organizar el código, con esta premisa fue creado y agregados nuevos elementos.

Los cuales son completamente integrados a CSS y JavaScript, a pesar que estas mejoras no son solo para los nuevos elementos sino también para los elementos que ya están en funcionamiento.

- **<Mark>**

La etiqueta de código HTML 5 <Mark> fue agregada para dar el efecto de resaltado de una parte del texto el cual no se lo consideraba como importante anteriormente, pero que con el tiempo se ha vuelto relevante para el usuario, como ejemplo se puede citar la búsqueda o filtrado de un texto en un consulta el código podría ser el siguiente:

```
<Spam>Mi <Mark>coche</Mark> es rojo</Spam>
```

Usando el elemento <Mark> para resaltar la palabra **Coche** se puede conceptualizar lo siguiente: que en la búsqueda el texto coche se resaltará dando al usuario la oportunidad de verificar su consulta de forma más rápida, en algunos navegadores esta palabra se colorea de color amarillo.

En el pasado, se obtenía similares resultados usando el elemento . El agregado de <Mark> tiene el objetivo de cambiar el significado y otorgar un nuevo propósito para éstos y otros elementos relacionados:

- <En> es para indicar énfasis (reemplazando la etiqueta <i> que anteriormente se usaba).

· es para indicar importancia. · <Mark> es para resaltar texto que es relevante de acuerdo con las circunstancias.

· debería ser usado solo cuando no hay otro elemento más apropiado para la situación.

- **<small>**

El elemento <small> es otro componente al cual se lo ha integrado en HTML 5 esta etiqueta realizaba la función de presentar un texto con letra pequeña el elemento referenciaba el tamaño del texto muy independiente de lo significaba, pero ahora en HTML 5 su función es presentar la letra pequeña como impresiones legales.

Ejemplo: <Small>Derechos Reservados © 2011 MinkBooks</small>

Inclusión de información legal con el elemento <small>

- **<cite>**

El siguiente elemento es <cite>, fue modificado de su anterior propósito para desde hoy volverse más específico, su función ahora es encerrar los títulos, como: título de un libro, de una película, de una obra de teatro, una canción etc. Ejemplo:

<Spam>Mi canción Favorita es <cite>Te Conozco</cite></spam>

- **<article>**

Es otra de los nuevos elementos de HTML 5 está en la categoría de los elementos que definen secciones, como el header, hgroup entre otros, su valor es considerado semántico.

Solo se utilizará cuando tenga un contenido que sea independiente que se pueda reusar o compartir, es decir que cuando este contenido sea copiado hacia otro lugar no afecte el contenido original de la página, Ejemplo:

<article>

<Header>

<h1>La más taquillera </h1>

</Header>

Este es el texto del mensaje

```
<Footer>
```

```
<Address>
```

```
<a href="http://www.guaricha.com">Cinemark</a>
```

```
</Address>
```

```
</Footer>
```

```
</Article>
```

Agregando información detallada <article>.

```
<article>
```

```
<Header>
```

```
<h1>Estrenos</h1>
```

```
<time datetime="2014-09-03" pubdate>public ado en 1945 </time>
```

```
</Header>
```

Este es el texto del mensaje

```
</Article>
```

- **Fecha y hora usando el elemento <time>.**

Existen muchas razones por las cuales de una o de otra forma se debe ingresar un formato fecha en una página, estas pueden ser: fecha de creación de la página, fecha del último ingreso, fecha de modificación, mostrar la fecha de un comentario, reseñar un comentario, etc.

Para ellos HTML 5 creo el elemento time que su estructura es la siguiente:

```
<time datetime="fecha">hoy es (fecha a mostrar) </time> o podría ser
```

```
<time datetime="2012-10-08T12:00-07:00">12h del 8 de octubre de 2012 en UTC -  
07</time> (San Francisco)
```

2.8 HTML 5.0 E INTERNET EXPLORER Y MICROSOFT DEVELOPER TOOLS

La W3C es una organización conformada por funcionarios propios y particulares así como entidades privadas que invierten recursos humanos y económicos en ayudar al crecimiento y definir todo lo referente de la Web, ellos entre otras cosas delimitan las especificaciones técnicas que conforman HTML 5, la W3C, está conformada por grupos de trabajo que se encargan de un proyecto de especificaciones relacionadas, las especificaciones pueden y son presentadas por cualquier miembro del grupo y el grupo entero pondrá en marcha este proyecto si es aceptado.

Microsoft con su aporte es un miembro cotizado y respetado en W3C el cual desempeña uno de los papeles más importantes y activos en el proceso de las especificaciones aportando su intelecto y contingencia en busca de las mejores normas para el buen funcionamiento de HTML 5.

Microsoft ha sido el aporte más significativo y ha invertido fuertemente en la construcción de HTML 5 y trabajó conjuntamente con W3C y otros proveedores para garantizar que sus programadores estén con las herramientas de alta y última tecnología y de esta manera ofrecer HTML 5 como un producto confiable e interoperable en todos los navegadores principales.

En el contexto de Microsoft como proveedor de navegador, el enfoque tiene cuatro vertientes:

- 1) Entregar el mejor sitio listo para HTML5 hoy vía Internet Explorer 9.
- 2) Difundir entre los desarrolladores futuras características vía la Plataforma de vistas previas de Internet Explorer.
- 3) Invertir en interoperabilidad por medio de pruebas presentadas al W3C.
- 4) Hacer prototipos de estándares inestables vía HTML5 labs.

Microsoft ha definido con un término muy particular a su producto HTML 5 "Site-Ready HTML5" con la premisa de que hoy día cuenta con un amplio y basto soporte en los navegadores más conocidos. Las tecnologías y sus nuevas etiquetas como lo son:

Cancs, audio y video, geolocalización, Scalable Vector Graphics, almacenamiento Web y varios módulos desarrollados para CS3 todo se encierra en el área y se implementan en Internet Explorer 9 y se puede decir que también en el Explorer 10, y en la mayoría de los navegadores que hoy en día los usuarios están acostumbrados a utilizar, la idea es que todo sea transparente para el usuario.

Las notificaciones son otro pilar fundamental, Microsoft dispone vistas previas de plataforma general o pública la cual está destinada a informar al programador y usuario que está cerca una nueva versión del navegador, de igual manera esto sirve para recabar información adicional. En lo que respecta a Internet Explorer 9, Microsoft lanza sus notificaciones cada seis a ocho semanas, notificando las nuevas versiones y sus mejoradas características que mejoraran el rendimiento HTML 5 para que los programadores lo evalúen y lo prueben, internet Explorer 9 fue lanzado en marzo del 2012 y 4 meses más tarde fue lanzada sus dos plataforma de vista previa para internet Explorer 10, notificando que se lleva una liberación regular de vistas previas en internet Explorer.

La interoperabilidad es uno de los puntos en que Microsoft ha invertido mayor esfuerzo, con la idea de demostrar que HTML 5 trabaja sistemáticamente con todos los navegadores utilizados por la mayoría de programadores y usuarios, Microsoft ha presentado el mayor conjunto de pruebas y casos relacionados con HTML 5 al W3C, y esta empresa será la encargada de verificar la compatibilidad de HTML 5 con cada navegador. El resultado es que los programadores puedan aplicar las nuevas tecnologías de HTML 5 y estar seguro de que funcionará cómodamente en todos los navegadores.

Ahora es más que conocido que esta tecnología ya viene aplicándose en internet Explorer 9. Y otras nuevas están siendo promocionadas para las nuevas versiones de Internet Explorer 10 en su plataforma de vistas previas de internet Explorer, muchos de sus componentes más conocidos como especificaciones están siendo perfeccionadas por parte de W3C y sus proveedores de navegadores para que estén a punto de aplicarlos en nuestros proyectos, uno de los elementos que ha llamado la atención es la Web-socket un componente que permite abrir canales bidireccionales con servidores de servicios back-end con lo que se permitirá una conectividad en tiempo real que antes no era posible en aplicaciones Web, esto provocara que los programadores se imaginen incontables usos del Web sockets para sus aplicaciones, esto aún no está perfeccionado pero está cambiando a un ritmo muy acelerado. Solo habrá que esperar y muy pronto será muy fácil aplicar esta especificación.

2.9 CSS3

La experiencia en internet día a día debe ser más placentera y funcional sin dejar atrás la idea de facilidad, rapidez y confianza, el diseño web lo está logrando por medio de nuevas y bastas tecnologías de desarrollo web, y su componente estrella son los CSS3.

HTML ayuda a organizar y armar la estructura de los contenidos y de la información de nuestro documento Web, y por medio de las hojas de estilo llamados CSS3 darle forma y formato a sus contenidos.

2.9.1 HISTORIA

Cascading Style Sheets siglas en ingles que significa hojas de estilo en cascada y con su abreviatura CSS es el lenguaje utilizado por los desarrolladores para la presentación de sus documentos HTML o XML por su extensión XHTML, donde la W3C es la encargada de normalizar las especificaciones en un estándar comprendida para el agente de los usuario y/o navegadores.

El objetivo de los CSS3 es separar la estructura del contenido con la presentación para el usuario y el navegador.

Los CSS pueden estar dentro del documento estructurado o separado de él, en el segundo caso este debe definirse en la cabecera del documento con la etiqueta <style> la cual es muy usada para mejorar las presentaciones

En el año de 1996 vio por primera vez la luz el CSS en su primera versión, la cual logra una rápida aceptación por los programadores web, con el pasar del tiempo fue saliendo nuevas y mejoras versiones y adquiriendo nuevas funcionalidades hasta llegar a la conocida versión de CSS 2.1 que es casi el 100% compatibles con la mayoría de los navegadores.

Así CSS ha logrado cumplir su objetivo principal que era independizar el contenido estructural HTML del diseño o la estética del programa, y lograr un control entero de los elementos del documento Web.

La idea es que ya no haya que modificar la estructura HTML cada vez que se desea modificar, eliminar o actualizar un aspecto de algún elemento del documento, ya que las hojas de estilo de cascada CSS se encargara de dicho proceso, independiente del contenido.

Con la llegada de las nuevas tecnologías las aplicaciones requieren y necesitan nuevas funcionalidades de navegación y representación para sus dispositivos en sus documentos Web, para ese entonces más o menos por el año 2005 se comenzó a definir el sucesor de la versión de CSS 2.1, al cual se le denominaría CSS3. Este ofrece elementos más avanzados y varias funcionalidades en el diseño Web, lo cual mejoraría y se adaptaría muy fácilmente a los dispositivos móviles y táctiles.

Estas posibilidades van desde la creación de efectos visuales asombrosos, hasta la generación de animaciones avanzadas, y todo desarrollado con unas pocas líneas de código CSS3.

2.9.2 ¿PARA QUÉ SIRVE CSS3?

Una de las características de CSS es que se puede escribir su código independiente de la estructura de contenido de la página web, de tal forma que con tan solo modificar el archivo CSS se ejecutara y se actualizar nuestra aplicación con las nuevas características que se modificó en el archivo CSS, sin cambiar absolutamente nada del código de la aplicación HTML que componen la aplicación web.

Este lenguaje permite dar formato al contenido de las páginas Web como: colores del texto, tamaño, fuentes de los elementos que componen el sitio, los CSS establecen reglas que indicaran y guiaran al navegador de cómo debe presentarse un documento, entre los aspectos más elementales son: color de la letra, el tamaño de la letra, el tipo, si es negrita, si es arial, pero no significa que solo haga eso con las letras CSS abarca todos los elementos de la página, fondo de la página, tamaño de imágenes, alta u ancho de una tabla entre otros.

2.9.3 CARACTERÍSTICAS PRINCIPALES

Atributo gradiente de colores en borde con CSS y Firefox: Se puede definir el color de los bordes de los elementos algo que no era normal ni posible en Firefox.

Bordes redondeados en CSS 3: Anteriormente para poder hacer un borde con un aspecto redondeado y muy elegante era muy complicado con CSS3 este atributo es tan fácil como poner border-radius, que define la curvatura que debe tener el borde del elemento.

Múltiples imágenes de fondo con CSS: CSS3 permite a una página tener varias imágenes de fondo de una sola vez.

Colores RGBA en CSS 3: Son los colores RGBA que se manifiestan en las nuevas características CSS3.

Word-wrap en CSS 3: CSS# permite encajar palabras que son demasiado largas y que no se podía encajar en la anchura de una celda, ahora esta funcionalidad la resuelve CSS3 rompiendo la palabra y lograr insertarla en la caja.

Textos multi-columna con CSS 3: CSS3 permite la automática organización del texto sin multicolumna o varias columnas sin tener que ordenar la ejecución, automáticamente ve la necesidad y lo hace.

Bordes con imágenes en CSS 3: Permite insertar imágenes en los bordes de las páginas, sin mayor código html, solo con simple código css3.

Sombras en CSS 3 con box-shadow: Con atributos css3 ahora será posible poner sombras a los elementos de las páginas sin usar JavaScript ni nada adicional solo con el atributo box-shadow

Resplandor exterior con CSS3: Cómo realizar un elemento que tenga un resplandor exterior con CSS3 y la propiedad box-shadow.

2.9.4 VENTAJAS CSS3

“Se obtiene un mayor control de la presentación del sitio al poder tener todo el código CSS reunido en uno, lo que facilita su modificación.

Al poder elegir el archivo CSS que se intenta mostrar, puede aumentar la accesibilidad ya que se asigna un código CSS concreto a personas con deficiencias visuales, por ejemplo. Esto lo detecta el navegador web”. (FLOREZ, 2012)

Por la técnica que utiliza CSS3 los Sprites logra aliviar la carga de las aplicaciones y de los sitios web.

El ahorro de tiempo y de trabajo al poder disminuir el código en simples especificaciones y que permita maravillosas presentaciones sin necesidad de usar un editor de gráficos es una de las ventajas principales.

2.9.5 VENTAJAS CSS3

“Existen limitaciones que CSS 2.x todavía no permite, por ejemplo, la alineación vertical de capas, las sombras, los bordes redondeados...”

El uso de las tablas permitía crear diseños complejos de forma mucho más sencilla que utilizando CSS, aunque CSS3 está intentando facilitar dicho trabajo”. (FLOREZ, 2012)

A veces, dependiendo del navegador la página que ha sido maquetada con CSS puede verse distinta (Aunque, si se sigue los estándares web de forma correcta, el problema es del navegador).

CAPÍTULO III

3 CONSTRUYENDO UNA INTERFAZ DE USUARIO CON EL NUEVO ESTILO DE WINDOWS 8

3.1 HERRAMIENTAS DE DESARROLLO PARA APLICACIONES METRO SOBRE WINDOWS 8

Dos de las herramientas básicas y principales que se utilizará en este proyecto son Visual Studio 2012 para el desarrollo y Blend para el diseño en su defecto código nativo XAML.

VS2012 propone todo lo referente a la escritura de código, localizar, implementar, y ejecutar una aplicación estilo metro gracias a su firmeza en su estructura y su adecuada presentación.

Blend es la segunda herramienta que se utilizará para la creación del aplicativo lector de Feeds RSS con estilo metro, Blend agrupa las características necesarias y su compatibilidad es eficaz para diseñar una interfaz de usuario muy elegante y atractivo que es la base fundamental del concepto de aplicaciones metro,

Tanto VS2012 como Blend fueron creados para ser dependientes uno del otro, ya que permiten que su comunicación sea la mejor.

3.1.1 VISUAL STUDIO 2012

Visual Studio se ha convertido en la herramienta perfecta para los programadores en el desarrollo de sus aplicaciones metro ya que no solo es un grupo de varias aplicaciones con la capacidad de configurar sino que esta nueva herramienta de desarrollo de aplicaciones posee una gama de beneficios para sus seguidores que harán que la Programación sea una verdadera experiencia además trae consigo las ediciones express de forma gratuita.

Con el pasar del tiempo Visual Studio se ha consolidado en el desarrollo de software con tecnologías Microsoft, con sus herramientas consolidadas ha decidido mejorarlas integrando una versión de su producto, dedicada a la creación de aplicaciones metro sobre cualquiera de los conocidos lenguajes de programación.

3.1.2 INTERFAZ A ESTILO METRO

Un cambio significativo y por qué no decir el mejor e impactante cambio de VS2012 en su diseño de la interfaz gráfica, pues es muy diferente a sus antecesores VS2005 hasta el popular VS2010, esta versión abarca las aplicaciones metro.

Un aspecto más simple es un paso al estilo metro, reducir líneas, iconos que en ciertas circunstancias eran molestos han vuelto su diseño en algo muy dinámico, respetando las jerarquías y tipografías para darle una forma más llamativa a las ventanas y a su contenido.

Los iconos VS 2012 contienen estilo metro, siendo más planos y más esquematizados, el cual hace un cambio importante de las versiones anteriores causando problemas para los programadores que estaban acostumbrados al anterior estilo, aunque el tiempo de adaptación es corto.

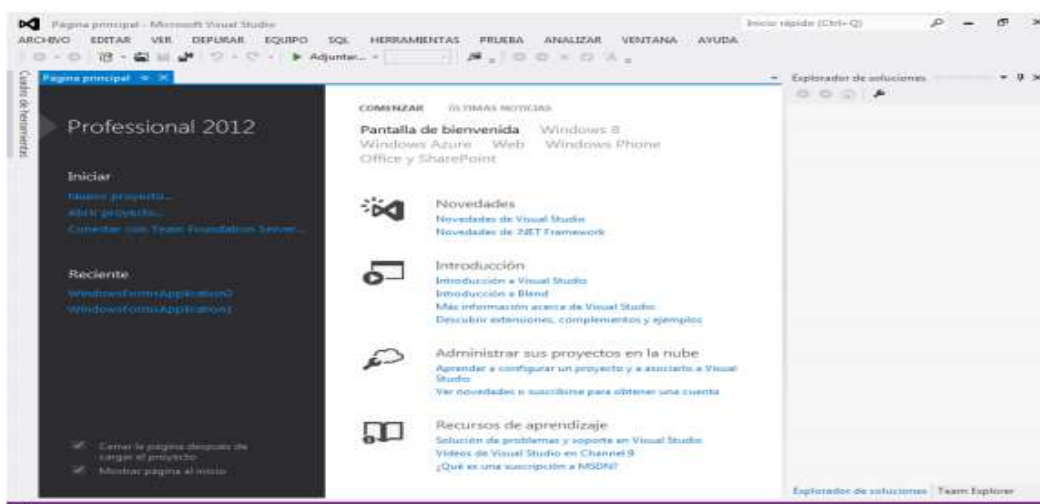


ILUSTRACIÓN 38: “Visual Studio 2012”

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

3.1.3 INTELLISENSE DE JAVASCRIPT

Una funcionalidad que permite ayuda contextual a los programadores es el IntelliSense, el cual brinda información simultánea en línea de los métodos, APIs, tipos, resultados, excepciones entre otros eventos que en el momento de programar puede ser muy importantes tenerlos en consideración y a la mano, esto ayudaría a escribir el código de forma correcta y reducir los errores de compilación. Además VS2012 se relaciona y contiene Javascript que es muy utilizado en creación de aplicaciones metro.

3.1.4 PLANTILLAS DE PROYECTO

Vs2012 presenta nuevas y elegantes plantillas para los proyectos Metro, estas contienen elementos HTML/JS que ayuda en el aprendizaje del nuevo estilo, entre las plantillas principales:

Blank App: Esta plantilla origina un proyecto con una página única, la que cargará el sistema operativo al lanzar la aplicación. Esta plantilla se utiliza para conocer los ficheros indispensables que todo proyecto de aplicación metro necesita.

Grid App: Es la plantilla más completa, la cual se compone de 3 páginas, una muestra el resumen de la información, otra muestra la información específica de cada grupo y por último la tercera mostrará información de cada ítem.

Split App: Esta plantilla se compone de 2 páginas, una muestra los grupos de información y la otra muestra los elementos de cada grupo.

FixedLayout App: Similar a Blank App, Esta plantilla origina una aplicación con un layout fijo, la cual está diseñada para juegos.

Navigation App: La plantilla sirve como ejemplo del tipo de navegación que se compone las aplicaciones metro.

3.1.5 ARCHIVOS IMPORTANTES DE UN PROYECTO METRO EN VISUAL STUDIO 2012

Al momento de crear un nuevo proyecto VS2012 crea de forma automática varios archivos y páginas XAML necesarias para el sistema operativo, incluye una sistema de navegación entre páginas además de datos de ejemplo, si se observa el área del explorador de soluciones se puede diferenciar el contenido de la carpeta raíz del proyecto en la cual tiene 4 archivos principales:

App.xaml. Este archivo representa a la aplicación y sus recursos.

GroupDetailPage.xaml. Este archivo representa la página de detalle de un grupo.

GroupedItemsPage.xaml. Este archivo representa la página de inicio de la Aplicación.

ItemDetailPage.xaml. Este archivo representa la página de detalle de un elemento.

StandardStyles.xaml Contiene recursos XAML utilizados para aplicar estilos a la aplicación.

La carpeta Assets del proyecto, se caracteriza por disponer de las imágenes utilizadas para personalizar la aplicación.

3.1.6 BLEND

Blend es una herramienta importante en la creación de aplicaciones metro, esta herramienta está diseñada para crear estilo utilizando código XAML, pero se actualizó la versión para que trabaje con VS1012 en el desarrollo de aplicaciones, para poder utilizar interfaces basadas en HTML5, JavaScript, CSS3.

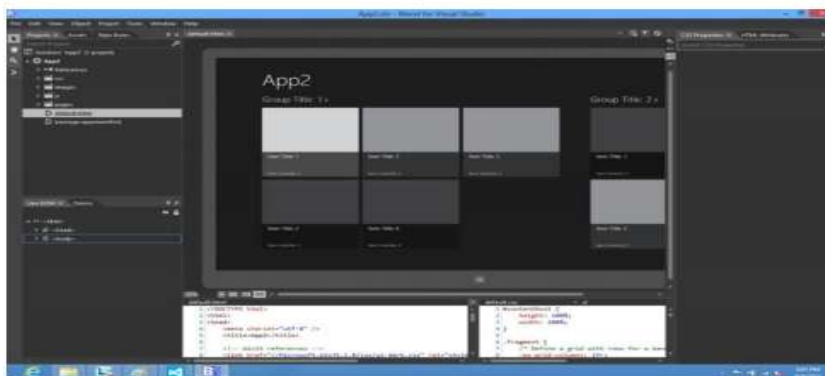


ILUSTRACIÓN 39: “Blend 2012”

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

Para la creación de interfaces Blend ofrece una gama interesante de presentaciones de las cuales sus principales características son:

- Diseño visual, de tipo WYSIWYG (WhatYouSeelsWhatYouGet) Lo que ves es lo que obtienes. Tipo de aplicación o herramienta aplicable especialmente a los editores de texto que permiten hacer un documento viendo directamente el resultado final, además de permitir emular distintos tamaños y orientación del dispositivo para la aplicación creada.
- Modo interactivo, esto permite que se pueda ejecutar las aplicaciones dentro de Blend para poder crear diseños para los estados que sólo se visualizan cuando están siendo ejecutados.
- Blend permite arrastrar y soltar controles y propiedades para un mejor manejo de la herramienta.

- Blend contiene un generador de código que se basa en los elementos que se agregan visualmente, además genera código eficaz cuidando de toda la sintaxis y creando código preciso y confiable.
- Editor de HTML, CSS para poder crear diseños específicos para estilo metro y con ello poder visualizar lo que los usuarios verán en la aplicación en Windows 8.

3.1.7 AREA DE TRABAJO

El área de trabajo proporcionada por Blend visualiza todos los elementos de la interfaz, la mesa de trabajo, configuraciones de la mesa de trabajo, paneles, menús, etc. El área de trabajo tiene una propiedad de personalizar y escalar de forma exclusiva, se puede cambiar la orientación de los paneles además acoplarlos o en su defecto ponerlos como flotantes configurar su tamaño con un simple arrastre y hale de sus límites ajustar la mesa alejando a acercándola y de la misma manera configurar el tamaño de los objetos.

Blend permite trabajar en dos vistas: vista de diseño y en vista de editor de código, o a sus vez se puede trabajar con las dos vistas conjuntamente, la manipulación del proyecto obviamente es más fácil desde la vista de diseño, recomendado para principiantes, pero si se conoce la estructura XAML es mejor hacerlo en vista de código, pues de esa manera tienes el control sobre todo el componente.

3.1.8 VISTA DE DISEÑO



ILUSTRACIÓN 40: “Vista de diseño”

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

1. Pestañas de documentos.- Presenta los elementos que componen los proyectos XAML y que se encuentran es estado activo: archivos, páginas, controles, etc.

2. Archivos activos.- Presenta la mesa o área de trabajo del documento seleccionado además provee una vista de todos los documentos abiertos en su pestaña de documento.

3.- Vistas:

- La vista Diseño crea documentos mediante el asistente gráfico o wizard visual en la mesa de trabajo.
- La vista Código para crear o editar código XAML.
- Vista en dos paneles Muestra la vista Código y la vista Diseño. También puede cambiar la orientación de las ventanas mediante la opción Orientación de la vista en dos paneles del menú.

3.1.9 MESA DE TRABAJO

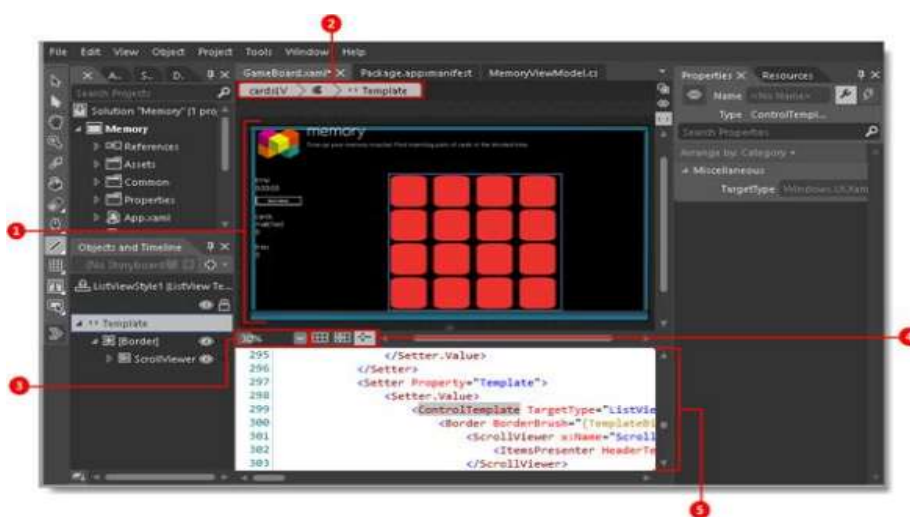


ILUSTRACIÓN 41: "Mesa de trabajo"

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

1. Vista Diseño.- Presenta visualmente el documento para su modificación en la mesa de trabajo.

2. Ruta de navegación.- Brinda la oportunidad de desplazamiento rápido entre los modos de edición de plantillas, entre los más conocidos se tiene: el modo de edición de estilos y el ámbito de edición de objetos para un objeto seleccionado.

3. Zoom.- Permite acercar y alejar la mesa de trabajo, y los objetos que contiene.

4. Controles de la mesa de trabajo.- Los controles como: Mostrar cuadrícula de ajuste, Ajustar a líneas de cuadrícula y Activar ajuste a las guías de alineación permiten establecer las opciones de ajuste, con lo cual se pueda alinear los objetos unos con los otros con líneas de separación uniforme.

5. Editor de código.- Editar el código XAML, C#, C++ o Visual Basic de forma manual en el Editor de código.

3.1.10 PANEL DE ACTIVOS

Este panel permite presentar todos los elementos XAML, los cuales se pueden añadir a la mesa de trabajo, además lista todos los controles que se pueden utilizar en el proyecto de Blend distribuido en Vs2012, los elementos utilizados recientemente podrán visualizarse en este panel.



ILUSTRACIÓN 42: "Panel de activos"

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

1. Cuadro de búsqueda.- Permite la búsqueda de elementos y admite la filtración de búsqueda como escribir dentro del cuadro de texto.

2. Modo de cuadrícula y Modo de lista.- Brinda la oportunidad de pasar de un modo cuadrícula a un modo de vista de lista de forma rápida y sencilla.

3.-Categorías de activos.- Presenta una lista de categorías con sus respectivas subcategorías y con un simple clic las se puede elegir un elemento.

4. Todo.- Muestra todos los controles disponibles.

5. Común.- Muestra estilos disponibles del diccionario de Microsoft y sus recursos adicionales.

6. Descripción de categoría.- Visualiza información adicional de las categorías y subcategorías seleccionada.

3.1.11 PANEL DE PROYECTOS



ILUSTRACIÓN 43: "Panel de proyectos"

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

1.-Cuadro de búsqueda.- Permite la búsqueda de elementos y admite la filtración de búsqueda como escribir dentro del cuadro de texto

2.- Referencias.- Presenta todos los archivos DLLs que son utilizados como referencias en los proyectos.

3.-Activos.- Contiene y visualiza los recursos del proyecto. Como las imágenes, las clases, archivos config etc.

4.- Común.- Contiene y muestra clases y estilos XAML comunes que permiten un desarrollo de la aplicación más rápida.

5.- Propiedades.- Contiene información general acerca del ensamblado.

6.-TemporaryKey.pf.- Al momento de crear una nueva solución de proyecto automáticamente VS2012 crea una clave de aplicación diferente para cada una. Todos los manifiestos ClickOnce están firmados digitalmente mediante un certificado.

7.- LayoutAwarePage La clase LayoutAwarePage proporciona lo siguiente:

Métodos GoBack, GoForward y GoHome, asignación de un estado de vista de aplicación a un estado visual métodos abreviados de teclado y mouse para la navegación

Modelo de vista predeterminado Administración de estado para controlar la duración del proceso y la navegación.

8.- StandardStyles.- Repositorio de Recursos que contiene los estilos recomendados de Tienda Windows.

9.- DataModel.- Contiene datos de ejemplo para usarlos durante el desarrollo de la Aplicación.

10.- App.xaml.- Este elemento es muy importante ya que contiene los recursos compartidos y además controla los eventos globales en el nivel de aplicación. App.xaml es necesario para mostrar la interfaz de usuario.

11.- Package.appxmanifest.- Presenta los metadatos de la aplicación como el nombre que se mostrará, logotipos, descripción de la aplicación, declaraciones, características y el paquete de implementación entre otros.

3.1.12 PANEL DE PROPIEDADES

El panel propiedades permite ver y modificar las propiedades de un objeto XAML seleccionado en la mesa de trabajo o en el panel Objetos y escala de tiempo. Si modifica un objeto directamente en la mesa de trabajo, los cambios de propiedad se reflejarán en el panel propiedades

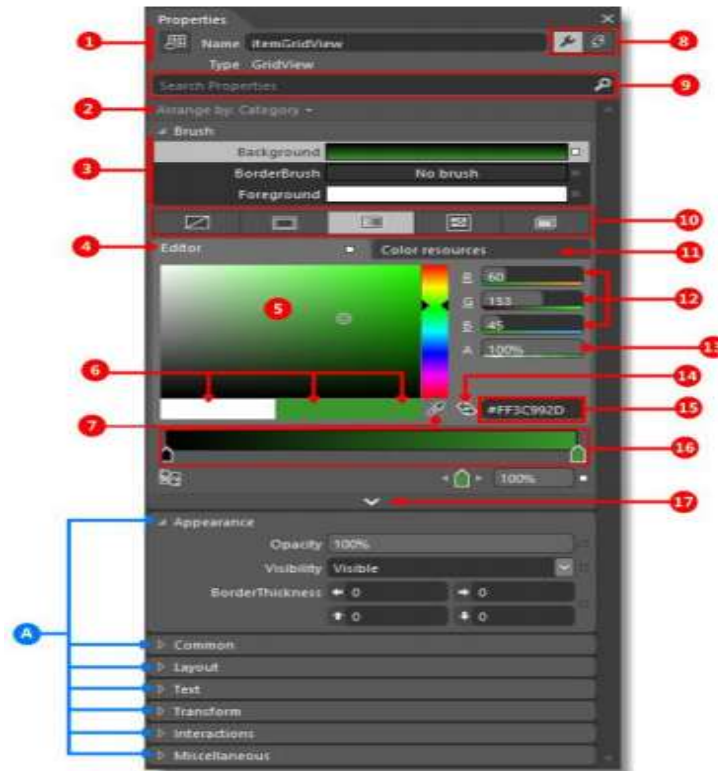


ILUSTRACIÓN 44: “Panel propiedades”

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

- 1.- **Nombre y tipo.**- Muestra el icono, nombre y tipo del objeto seleccionado.
- 2.- **Organizar por.**- Se usa para organizar propiedades alfabéticamente por nombre, origen o categoría, categorías de propiedades que se pueden expandir y contraer.
- 3.- **Propiedades del pincel.**-Muestra las propiedades visuales de pinceles como: Fill, Stroke y Foreground.
- 4.- **Editor de colores.**- Sirve para los pinceles de color sólido y degradado.
- 5.- **Selector de colores.**- Use el control deslizante de color para seleccionar un color.
- 6.- **Chips de color.**- Muestra los colores inicial, actual y último.
- 7.- **Cuentagotas.**- Cuentagotas de color está disponible cuando se selecciona pincel de color sólido, cuentagotas de degradado está disponible cuando se selecciona pincel de degradado.
- 8.- **Propiedades y eventos.**- Muestra las propiedades o los controladores de eventos de un elemento seleccionado.

9.- Cuadro de búsqueda.- Permite la búsqueda de controles filtrando sus propiedades.

10.- Pestañas del editor de pinceles.- Permite seleccionar un editor de pinceles. La propiedad seleccionada en pinceles se puede establecer en sin pincel, pincel de color sólido, pincel de degradado, pincel de diseño en mosaico o bien en un recurso de pincel.

11.- Recursos de color.- Permite aplicar el mismo color exacto a distintas propiedades. La pestaña Recursos de color incluye Recursos locales y Recursos del sistema.

12.- Espacio de color RGB.- Permite modificar el color ajustando los valores para R, G, o editores del número B (rojo, verde, azul).

13.- Canal alfa.- Modifica el valor alfa mediante el editor de números situado al lado de A.

14.- Convertir color en recurso.- Muestra categorías de propiedades que se usan con menos frecuencia. Puede convertir el color seleccionado en un recurso de color. Los recursos de color están disponibles al hacer clic en la pestaña Recursos de color.

15.- Valor hexadecimal.- Muestra el valor hexadecimal del color presentado.

16.- Control deslizante de degradado.- Control deslizante de degradado con delimitadores de degradado, que solo aparece si se selecciona un pincel de degradado.

17.- Mostrar propiedades avanzadas.- Muestra categorías de propiedades que se usan con menos frecuencia

3.1.13 Panel de herramientas

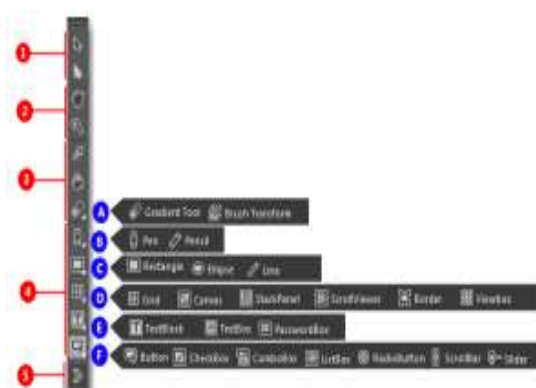


ILUSTRACIÓN 45: “Panel de herramientas”

Fuente: <http://msdn.microsoft.com/es-es/library/jj153219.aspx>

Con el panel de herramientas se puede crear y modificar objetos de la aplicación

1.-Herramientas de selección.-Se usan para seleccionar elementos. La herramienta selección directa, situada bajo la herramienta selección, se utiliza para seleccionar elementos anidados.

2. - Herramientas de vista.- Se usa para ajustar la vista de la mesa de trabajo; por ejemplo, para desplazarla lateralmente y hacer zoom.

3.- Herramientas de pincel.-Sirve para trabajar con los atributos visuales de un objeto; por ejemplo, transformar un pincel, pintar un objeto o seleccionar los atributos de un objeto para aplicarlos a otro.

4. -Herramientas de objeto.-Utilícelas para dibujar en la mesa de trabajo los objetos más habituales, como trazados, formas, paneles de diseño, texto y controles.

5. -Herramientas de activos.- Utilícelas para obtener acceso al panel Activos y mostrar el último activo de la biblioteca que se ha utilizado.

3.1.14 LICENCIA DE DESARROLLADOR PARA WINDOWS 8

Una estrategia que usa Microsoft en pro de crear aplicaciones seguras y confiables es la obtención de la licencia de desarrolladores para Windows 8, con esta licencia se puede instalar, depurar, probar y evaluar nuestras aplicaciones, para que luego sea más fácil obtener la certificación Windows que aprueben que nuestra aplicación posee el estándar de una aplicaciones metro consideradas por Microsoft, es necesario obtener esta licencia ya que si no se la obtiene Windows bloquea la aplicación que intentas ejecutar, las licencia son gratuitas y fáciles de conseguir sin necesidad de tener una cuenta Microsoft y puede tener cuantas licencias deseen y sean necesarias.

La licencia dura aproximadamente entre 30 a 90 días, además se la puede extender o adquirir otra. Para obtenerla existen dos formas mediante Visual Studio o mediante símbolo del sistema.

3.1.14 OBTENCIÓN DE LA LICENCIA PARA DESARROLLO DE APLICACIONES METRO SOBRE WINDOWS 8

Vs2012 provee un análisis de ejecución en la primera vez que se ejecuta la herramienta la cual verifica que se disponga de la licencia de desarrollador, esta licencia es gratuita y no necesita tener una cuenta Microsoft para conseguirla. Si se desea actualizar la licencia se lo consigue con los siguientes pasos:

En Microsoft Visual Studio Express 2012 para Windows 8, clic en Tienda > Adquirir licencia de desarrollador.

Si es una versión no Express de Visual Studio 2012, clic en Proyecto > Tienda > Adquirir licencia de desarrollador.

3.1.16 OBTENCIÓN DE LA LICENCIA PARA PROGRAMADOR MEDIANTE SÍMBOLO DEL SISTEMA

Si lo que se necesita es adquirir la licencia de desarrollador por medio del símbolo del sistema se tiene que seguir estos pasos:

- Show-WindowsDeveloperLicenseRegistration: Es el acceso al cuadro de diálogo que permite obtener la licencia e instalar, esto sólo es permitido si se ingresa con una cuenta Microsoft y permisos administrativos en el sistema.
- Sintaxis: C:\> Show-WindowsDeveloperLicenseRegistration
- Get-WindowsDeveloperLicense: Con este código de comando se ayuda a visualizar la obtención de la fecha y hora de expiración de la licencia.
- Sintaxis: C:\>Get-WindowsDeveloperLicense
- Unregister-WindowsDeveloperLicense: Si se necesita eliminar la licencia este código es el correcto, se necesita tener permisos de administrador.
- Sintaxis: C:\>Unregister-WindowsDeveloperLicense

3.2 DIRECTRICES PARA EL DESARROLLO DE APLICACIONES METRO

Para el desarrollo de aplicaciones se debe tener en cuenta ciertas directrices o especificaciones diseñadas por Microsoft para las aplicaciones de estilo metro, estas directrices son estrictamente apuntadas al diseño, a la interacción táctil, directrices de texto, directrices de contenido, las directrices tiene 5 principios fundamentales dadas por Microsoft estas son:

3.2.1 MUESTRA BUEN HACER

Los pequeños detalles merecen mayor énfasis en la aplicaciones metro, ya que son una experiencia para el cliente final, no descuidarse que esto suceda en todas las etapas de la aplicación.

3.2.2 ESFUÉRZATE EN LOS DETALLES

Despliega seguridad y confianza al usuario del aplicativo.

Utiliza sencillez, equilibrio, integridad, simetría para ofrecer al usuario una experiencia nueva y confiable.

3.2.3 HAZ MÁS CON MENOS

Las distracciones son completamente eliminadas en las aplicaciones metro, pero se activa la oportunidad de que el usuario explore y descubre su alcance, visualizar lo necesario para que el usuario del aplicativo se concentre en su actividad.

Mostrar solo pantallas con los elementos relevantes para no interrumpir con su trabajo con elementos que posiblemente no los va a usar en ese momento, dar prioridad a la interfaz y su contenido sacará de la mediocridad las aplicaciones.

3.2.4 PON EL CONTENIDO ANTES QUE EL CONTENEDOR VISUAL

Es visualmente centrado y directo, y permite que el usuario se concentre en lo que le gusta.

3.2.5 INSPIRA CONFIANZA AL USUARIO

- Sé rápido y fluido
- El objetivo es que el usuario sea el relacionado directo del contenido y que el aplicativo responde a las acciones rápidamente.
- Crear una sensación de continuidad y que cuente una historia mediante el uso significativo del movimiento para dar vida a la experiencia.
- Diseñar para una interacción táctil e intuitiva.
- Permite el movimiento del usuario dentro de la aplicación.
- Diseñar una interfaz de usuario atractiva
- Deberá ser auténticamente digital
- Deberá aprovechar todas las ventajas de los medios digitales.

- Eliminará los límites físicos para crear experiencias que sean más eficientes y naturales que la realidad.
- Aceptará el hecho de que las aplicaciones son píxeles en una pantalla. Creará diseños con imágenes y colores intensos, vibrantes y llamativos que traspasen los límites de los materiales del mundo real.
- Deberá conectarse a la nube para que los usuarios estén conectados entre sí.
- Usará la belleza de la tipografía. Usará el movimiento con significado, gana en equipo, saca provecho del ecosistema y trabaja con otras aplicaciones, dispositivos y el sistema para completar los escenarios para el usuario.
- Ajustará al modelo de interfaz de usuario para reducir la redundancia. Aprovechará lo que la gente ya sabe para ofrecer una sensación de familiaridad, control y confianza.
- Usará las herramientas y plantillas de Microsoft disponibles para promover la coherencia

3.3 DIRECTRICES PARA INTERACCIÓN DEL USUARIO CON LA APLICACIÓN

Las directrices presentadas por Microsoft en relación a las aplicaciones metro, conlleva que el usuario visualice una aplicación amigable, con funcionalidades que son independientes del dispositivo que esté usando.

Lo táctil es el enfoque de las aplicaciones metro por lo tanto todas sus Apps deben ser creadas pensando en estos dispositivos sin descuidar que también podrán ser intervenida por componentes como el mouse y el teclado, el lápiz óptico, por esto hay que tomar muy en cuenta las directrices propuestas dejando otros elementos que Windows 8 ya los insertará automáticamente.

3.4 DIRECTRICES DE INTERACCIÓN TÁCTIL

Los dispositivos táctiles son aquellos que permiten la manipulación de una aplicación en dispositivos touch, las directrices permiten que se diseñe interacciones táctiles correctas y adecuadas para cada aplicación metro.

El sistema operativo Windows 8 ya posee un conjunto de interacciones táctiles en todo su estructura lo que permite que las aplicaciones metro sean fáciles de implementar y de usar, esto puede confundir al usuario de la aplicación ya que se mezclan o se duplican las interacciones propias del sistema versus las aplicaciones creadas, estas interacciones táctiles normalizadas de Windows 8 son:

1. Presión continua.
2. Pulsar para acción principal.
3. Deslizar movimiento panorámico
4. Deslizar borde para seleccionar, ordenar o mover.
5. Girar.
6. Zoom.
7. Deslizar borde inferior o superior para comandos de la aplicación,
8. Deslizar borde derecho para comandos del sistema



ILUSTRACIÓN 46: “Lenguaje táctil de Windows 8”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

1. **Precisión en la selección de objetos.**- La Selección de objetos en las interacciones táctiles necesitan como principal requerimiento la precisión en la pantalla touch, los dedos por su grosor no son muy precisos a la hora de seleccionar un objeto, el lápiz óptico y el mouse son perfectos para esta actividad, si los objetos son grandes tiene una mayor precisión lo cual permite mejor manipulación.

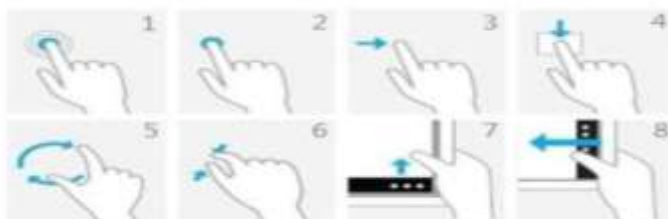


ILUSTRACIÓN 47: “Precisión con los objetos”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

2. **Contenido examinado mediante gestos táctiles.**- Permite realizar movimientos panorámicos y zoom a las páginas de la aplicación para una mejor exploración con gestos táctiles

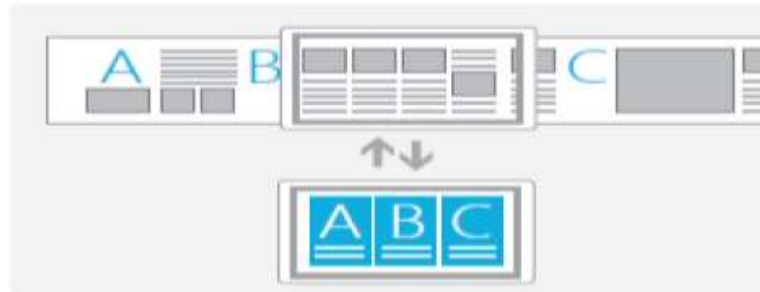


ILUSTRACIÓN 48: “Contenido explorado con gestos táctiles”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

Cada acción del usuario necesita respuesta visual, los elementos de la aplicación son los encargados de esta respuesta ya sea interactuando con diversas formas o simplemente con animaciones de sus objetos, cambio de color, aumento y disminución de tamaño de sus objetos de tal forma que le den la impresión al usuario que está teniendo respuesta.

3. **El contenido sigue al dedo.**- Los elementos que son guiados con el dedo del usuario en la aplicación deben seguir su trayectoria mientras permanece en movimiento o presionado. Cuando el usuario termine su actividad estos elementos deberán volver a su estado de inicio.



ILUSTRACIÓN 49: “Seguimiento de los movimientos”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

4. **Interacciones sean reversibles.**- todo objeto debe volver a su estado normal luego de una interacción del usuario con la aplicación, para que esta actividad sea más sencilla y segura para el usuario deberá presentar una información visual indicando que es lo que sucederá si levanta o mantiene presionado la pantalla touch.

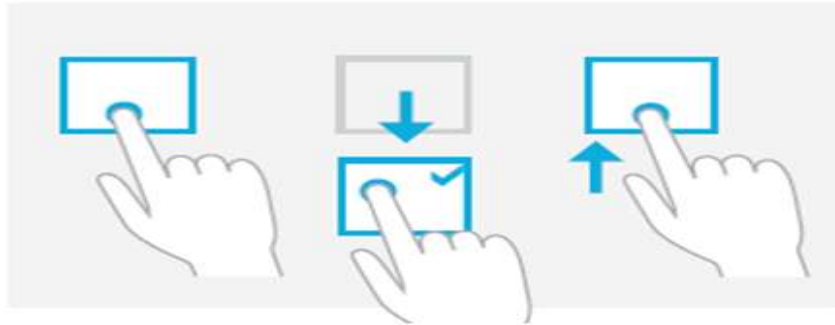


ILUSTRACIÓN 50: “Contenido explorado con gestos táctiles”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

5. **Cantidad de dedos.**- La cantidad de dedos que se usen en la aplicación debe ser permitida e independiente de su actividad, ya que es costumbre del usuario colocar varios de sus dedos en el dispositivo, el resultado debe ser el mismo si usa un dedo o si usa varios dedos esto dará una aire de seguridad de que la aplicación hará lo que usuario lo solicita.



ILUSTRACIÓN 51: “Iteración con varios dedos”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

6. **Tiempo en interacciones.**- Eventualmente para poder ejecutar una acción en dispositivos normales hay que dar doble click con se da la orden de ejecutar dicha acción, lo mismo sucede en los dispositivos móviles con aplicaciones metro, pero para estas situaciones la aplicación debe dar un límite de tiempo para el doble click o para la presión continua del objeto, con esto se evitará que la aplicación realice una actividad sin consentimiento del usuario final.

3.5 DIRECTRICES PARA EL TAMAÑO DE OBJETOS

A continuación las directrices para determinar el tamaño de los objetos de una aplicación.

<p>Tamaño Mínimo</p> <p>El tamaño mínimo para un objeto es de 7x7 mm.</p>	
<p>Tamaño para Precisión</p> <p>El tamaño para mejor precisión en acciones como cerrar, eliminar es de 9x9 mm.</p>	
<p>Tamaño cuando lo objetos no caben</p> <p>Si los objetos no caben y están amontonados el tamaño a usar es de 5x5 mm.</p>	

ILUSTRACIÓN 52: “Tamaño de objetos”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh465415.aspx>

3.6 DIRECTRICES DE NAVEGACIÓN

La navegación dentro de la aplicación debe ser fácil y se debe adelantar al requerimiento del usuario, esto se alcanza con la organización del contenido y delimitando los controles visuales en las pantallas del aplicativo, las directrices de navegación cumplen con estos requerimientos y se dividen en dos tipos:

3.6.1 SISTEMA JERÁRQUICO

Las aplicaciones metro utilizan un sistema de navegación jerárquico que brinda a la aplicación rapidez y fluidez en su manipulación, para establecer la jerarquía existen 3 tipos de niveles jerárquicos: páginas de concentrador, páginas de sección y páginas de detalles.

- **Páginas de Concentrador** Son el primer nivel de jerarquía y representan el espacio de ingreso a la aplicación, muestra una vista panorámica horizontal del contenido en el que el usuario podrá observar una portada general de la aplicación. El concentrador divide el contenido en distintas categorías y asignadas a páginas de sección ofreciendo funcionalidades y diferentes destinos dentro de la aplicación.
- **Páginas de sección** Son el segundo nivel de jerarquía, muestra el contenido de la sección con mejor escenario y consta de elementos individuales, cada uno con su propia página de detalles.

- **Páginas de detalles** Tercer nivel de jerarquía, muestra a detalle los elementos individuales de la sección, estos elementos pueden contener mucha información o simplemente un solo objeto, como un video

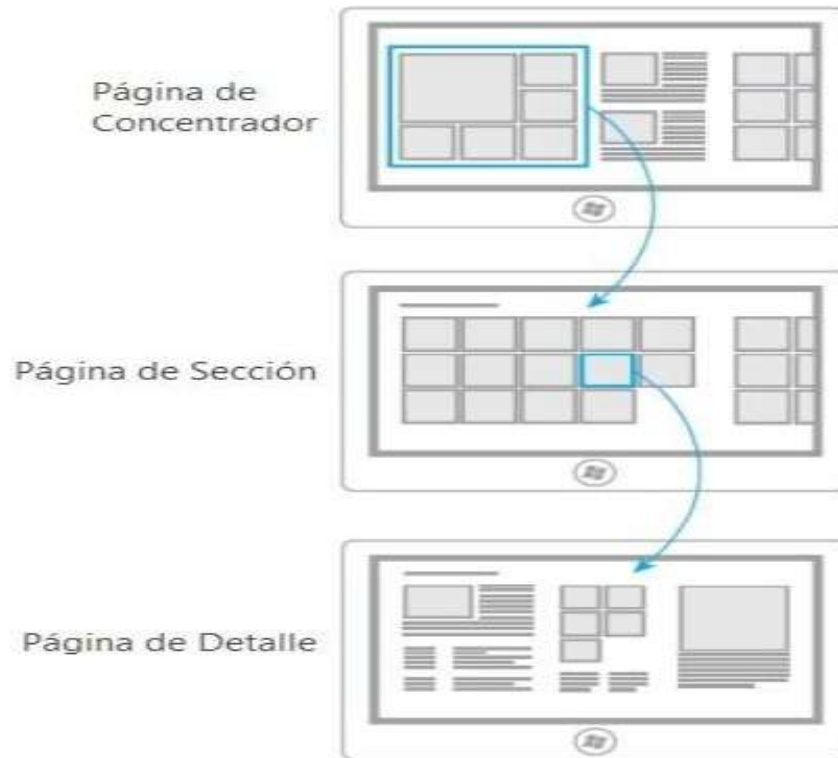


ILUSTRACIÓN 53: “Sistema jerárquico”

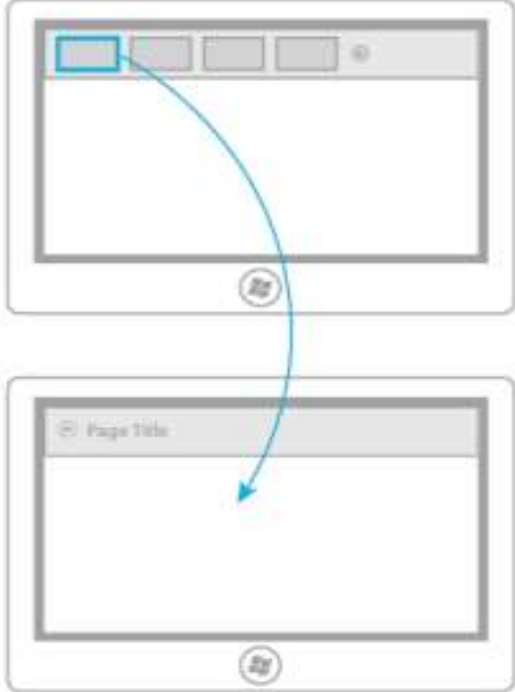
Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.6.2 SISTEMA PLANO

El diseño de navegación plano es el más utilizado por las aplicaciones metro, este es más común en los juegos, en aplicaciones de manejo de documentación, donde las aplicaciones permite desplazarse por las páginas y las pestañas a los nodos del mismo nivel de jerarquía,

Este diseño se lo usa cuando la pantalla principal necesita rapidez en el cambio de su pantalla hacia otras páginas e incluso varias páginas consecuentes.

TABLA 2: “Sistema plano”

<p>Barra de la aplicación superior</p> <p>La barra de la aplicación superior es perfecta para cambiar entre varios contextos. Entre los ejemplos, se incluyen pestañas, documentos y sesiones de mensajería o de juegos.</p> <p>Esta barra es un elemento transitorio que se encuentra en la parte superior de la pantalla, y está visible cuando los usuarios deslizan el dedo rápidamente desde el borde superior o inferior. Aunque el formato de los elementos de la barra puede variar, un tratamiento típico es el uso de una miniatura simple.</p>	 <p>El diagrama ilustra la transición de una barra de aplicación superior oculta a una barra de aplicación superior visible. En la parte superior, se muestra un dispositivo con una barra de aplicación superior oculta, representada por un recuadro azul en el borde superior. Una flecha curva indica la transición a la parte inferior, donde se muestra el mismo dispositivo con la barra de aplicación superior visible, mostrando un título de página "Page Title".</p>
<p>Cambiar</p> <p>A diferencia del sistema jerárquico, en el sistema plano no suele haber un botón atrás permanente ni una pila de navegación, por lo que para moverse por las páginas se usan vínculos directos dentro del contenido o de la barra de la aplicación superior.</p> <p>Se puede elegir incluir otras funcionalidades dentro de la barra de la aplicación superior, como agregar un botón '+' para crear una nueva pestaña, una página o una sesión, etc.</p>	

Fuente: Autor

3.6.3 ANATOMÍA DE NAVEGACIÓN

A continuación se muestra la anatomía de la navegación entre las secciones de una aplicación, entre los diferentes niveles de la jerarquía y dentro de una única página de la aplicación.

- **Encabezado y botón Atrás** El encabezado es la etiqueta que muestra la página actual. El botón Atrás permite regresar a la página anterior de forma rápida.

- **Página de concentrador** El concentrador visualiza en la pantalla principal la información de distintas áreas de la aplicación, lo cual permite al usuario tener una vista general y completa de lo disponible en la aplicación.
- **Secciones de contenido o categorías** Permite aplicar formato a las secciones de contenido y administración los espacios de los elementos.



ILUSTRACIÓN 54: “Anatomía de navegación jerárquico”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

- **Zoom semántico:** El zoom semántico permite examinar una vista y moverse por ella de manera rápida y fluida sobre todo si la vista es una larga lista panorámica.

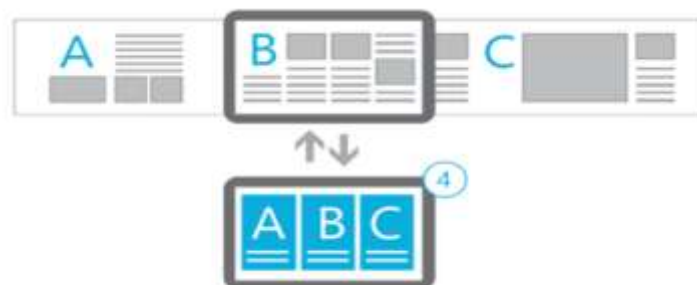


ILUSTRACIÓN 55: “Navegación entre niveles”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

- **Barra de la aplicación superior.-** Permite ver las opciones de navegación que posee la aplicación.
- **Menú de encabezado.-** Permite interactuar cada sección de la aplicación y vincularse a otra que esté disponible en las diferentes parte de la aplicación.
- **Vínculo de página principal.-** Es un acceso rápido a la raíz de la aplicación y está ubicada en la parte inferior.

- **Barra de la aplicación inferior.-** Son accesos a comandos importantes para una vista en particular.
- **Ver/Ordenar/Filtrar.-** Interactúan los modos es que se puede mostrar el contenido, están situadas en la barra de la aplicación.
- **Borde.-** Muestra la barra de aplicación cuando se desliza el dedo por este borde

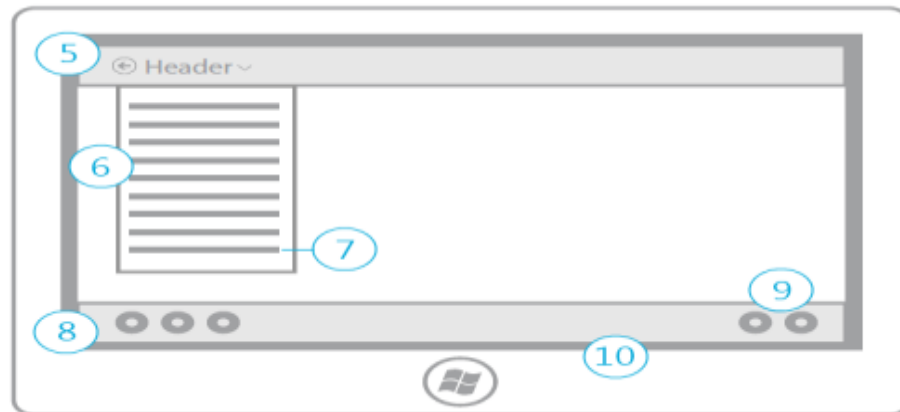


ILUSTRACIÓN 56: "Anatomía de la navegación plana"

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.7 DIRECTRICES DE DISEÑO

Son las normas para el correcto diseño de la aplicación metro, conservando los factores importantes como son: la forma, el tamaño, la visualización y su presentación elegante y atractiva para el usuario.

Para el diseño de una aplicación metro existen dos opciones: diseños fluidos y estáticos, cabe recalcar que el recomendado para utilizarlos es el diseño fluido.

3.7.1 DISEÑO FLUIDO Y ADAPTABLE

Este diseño es el más utilizado por muchos diseñadores, ya que estos diseños reducen código e incrementan el flujo para adaptación del área visual del dispositivo, el diseño usa elementos y controles flexibles en sus aplicaciones uno de ellos es el ListViewGrid por lo que este control distribuye el contenido en forma automática. Los diseños fluidos se encuentran en aplicaciones de contenido, administración y creación.



ILUSTRACIÓN 57: “Diseño fluido y adaptable”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.7.2 DISEÑO ESTÁTICO O FIJO

El diseño estático es el menos utilizado por los desarrolladores, ya que su visualización no es la más óptima este diseño puede estirarse, perder su tamaño en dispositivos cortos, o reducirse, no se ajusta dinámicamente lo que conlleva un fracaso a la hora de presentar el diseño al usuario final.

Este diseño se lo utiliza en juego con visualización fija, para mostrar imágenes o contenido fijo es decir que no se vea nada más que lo que se ve en ese momento.

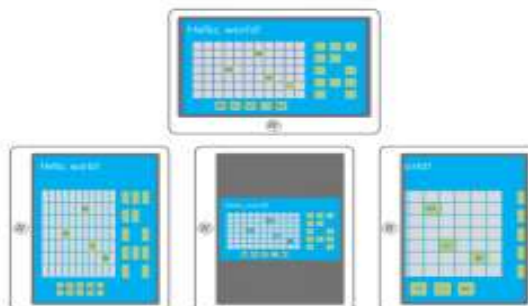


ILUSTRACIÓN 58: “Diseño estático y fijo”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.7.3 RESOLUCIÓN DE PANTALLA

Las resoluciones de las aplicaciones metro se basan de dos estados similares. La resolución mínima requerida es la de 1024X768 aquí funcionan perfectamente cualquier aplicación metro, garantizando que todos los elementos se ajusten a la pantalla sin cortes. La Resolución adecuada u óptima requerida para obtener todas las características de Windows 8 es la de 1366X768, asegurando que todos los elementos se ajusten a la pantalla sin regiones en blanco.

No es adecuado ni seguro ejecutar en pantalla con resoluciones inferiores a 1024X600 o 1280X720 con esto se perdería todo el diseño de la aplicación.

3.7.4 ESTADO DE VISUALIZACIÓN

Existen tres tipos de estado de visualización en las aplicaciones que el usuario puede utilizar en su aplicación.




<p>Pantalla Completa</p> <p>La aplicación utiliza toda la pantalla.</p>	
<p>Acoplada</p> <p>La aplicación se acopla a una parte específica de la pantalla.</p>	
<p>Rellena</p> <p>La aplicación ocupa la zona del resto de la pantalla que no está ocupada por la aplicación acoplada.</p>	

ILUSTRACIÓN 59: “Visualización de los estados”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.7.5 COMPATIBILIDAD CON LAS ORIENTACIONES DE LA PANTALLA

Todas las aplicaciones deben adaptarse a la orientación constante de la pantalla en dispositivos móviles, el usuario podrá girar, voltear, inclinar su dispositivo para lo cual la aplicación no debe perder su forma sea esta vertical u horizontal.

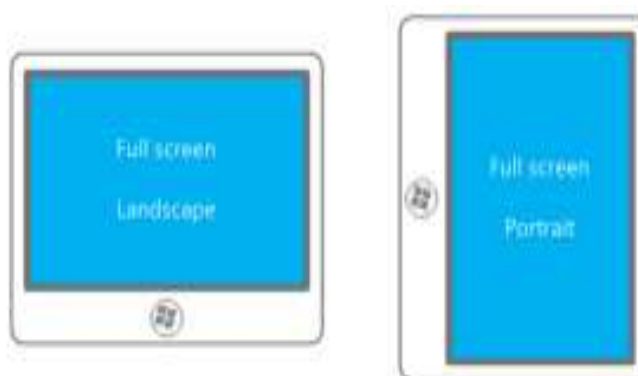


ILUSTRACIÓN 60: “Orientación del screen”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.7.6 NAVEGAR CON LAS ETIQUETAS DE LA SECCIÓN Y LOS MENÚS DE ENCABEZADOS

Este es un ejemplo Genérico de una aplicación en la cual se utiliza un diseño para usar las secciones de contenido, posee un menú de encabezado y el botón de retroceso, para navegar por la aplicación que podría estar en la Windows Store.

El menú del encabezado permite vincularse con las páginas de la sección siguiente, (L2) y posee un vínculo con la página principal o concentrador (L1), esto logra que la navegación sea rápida, el menú es creada como un master page, es decir estará en cada página que se navegue, lo que hace también que las aplicaciones sea confiable y eficaz para su movilización.

Los usuarios pueden pulsar la etiqueta de la sección para rastrear la página correspondiente de dicha sección. Proporciona una indicación visual, como ver todos los (+), para indicar a los usuarios que existen más elementos en esta sección que los que se muestran en el concentrador. Con este diseño se evita la necesidad de usar un espacio de icono o colocar un vínculo dentro del contenido.

En vista general del ejemplo se puede obtener esta vista del aplicativo, además el diagrama de navegación. Considerando que es un aplicativo básico, adecuados de elementos de navegación, empleados como distintivos de todo lo que es interactivo

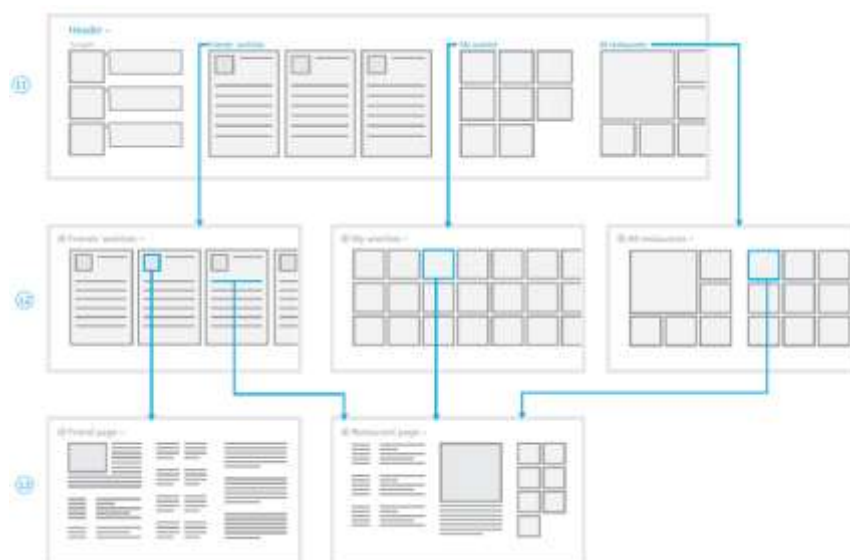


ILUSTRACIÓN 61: “Etiquetas de sección”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.7.7 NAVEGAR CON FILTROS, TABLAS DINÁMICAS, ORDENACIONES Y VISTAS

Una objetivo muy interesante de las aplicaciones metro es saber determinar el estado de la navegación es decir saber dónde, cuándo, y como permitir al usuario más control sobre el modo de como explorar el contenido de su aplicativo.

Tanto los filtros como las tablas dinámicas, los selectores y las ordenaciones deben ser prioridad en la visualización del diseño de la aplicación.

TABLA 3:"Navegación con filtros"

Periodo	Definición	Ejemplo
Filtro	Quitar u ocultar contenido dentro de un conjunto de datos, sobre la base de ciertos criterios.	Si se quiere buscar un juego probablemente eliges ver solo aquellos juegos dentro de una categoría definida y/o personalizada.
Tabla dinámica	Reorganizar el contenido dentro de un conjunto de datos, sobre la base de ciertos criterios.	Permitir organizar una colección de información por los diferentes filtros como por ejemplo organizar una colección de videos musicales por intérprete, álbum o género.
Ordenación	Cambiar el orden en que se muestra el contenido dentro de un conjunto de datos.	Apoyar para que la información sea más fácil de buscar mediante los filtros de ordenamiento de más recientes, o más populares , etc.
Vista	Cambiar el estilo o método en que se muestra el contenido.	Presentar diferentes modos de visualizar la información ya sea por medio de una lista o un gráfico descriptivo.

Fuente: Autor

Las aplicaciones metro tiene la facilidad de presentar sus controles en el canvas de tal forma que permiten filtrar, ordenar, y hasta crear tablas dinámicas, esto cuando el objetivo de la aplicación son búsquedas de información.

Todos los controles están dentro de la barra de aplicación si la aplicación tiene la finalidad de búsquedas de información, como una aplicación de venta y compra de artículos de ferretería, o una blog de noticias etc. La ordenación y filtración de la información es muy importante.

Existen varios comandos de filtro y/u ordenación de contenido, depende del diseño para saber el lugar adecuado de donde deben ir los controles.

3.7.8 BARRA DE APLICACIÓN SUPERIOR

En la parte superior de la pantalla esta la barra de aplicaciones cuyo objetivo final es permitir la navegación entre las secciones y las páginas que conforman la aplicación que usan el diseño de navegación plano, a pesar que no es estrictamente del diseño plano este también puede utilizarse en el diseño de navegación jerárquico sustituyendo al menú del encabezado, como una forma de presentar una navegación global, como un master page la barra se muestra en todas las páginas de la aplicación, de forma adecuada para el buen y fácil uso del usuario.

3.7.9 CAMBIO DE VISTA DE LA BARRA DE APLICACIÓN

La barra de aplicación tiene características de una mesa de comandos pero puede usarse para modificar el modo de cómo se visualiza la información. También la barra de aplicación ayuda al cambio de vista, la ordenación, y el filtrado, es preferible no usar esta barra para desplazarnos entre las páginas.

3.8 DIRECTRICES DE TEXTO

Esta directriz norma las fuentes, el tamaño, el grosor, colores de fuente, forma, pequeños detalles que dan vida al texto en las aplicaciones.

- **Fuente**

El estilo metro utiliza 3 tipos principales de fuentes para texto y cada una destinada para diferentes partes de la aplicación. Las fuentes que se utilizan son:

- **Fuente Segoe UI:** Usado para botones y selectores de fecha.
- **Fuente Calibri:** Usado para el texto que el usuario escribe y lee, como correo electrónico y chat.
- **Fuente Cambria:** Usado para bloques de texto grandes, como un lector RSS25 o de revistas.
- **Grosor de la Fuente**

La fuente principal que está compuesta las aplicaciones metro es la Segoe UI, y con ella se han establecido diferentes grosores:

- ✓ Segoe UI Light: Segoe UI con un grosor de 200.
- ✓ Segoe UI Semilight: Segoe UI con un grosor de 300.
- ✓ Segoe UI: Segoe UI con el grosor normal, 400.
- ✓ Segoe UI Semibold: Segoe UI con un grosor de 600.
- ✓ Segoe UI Bold: Segoe UI con un grosor de 700.

En las hojas de estilo estos valores se establecen según la siguiente sintaxis:

```
<p style="font-family: 'Segoe UI Semibold'"> texto A Resaltar</p>
```

• **Tamaño de la Fuente**

El tamaño y el grosor de la fuente se relacionan en el estilo metro y se establecen los siguientes tamaños:

- ✓ Segoe UI Semilight de 11 ptos utilizado para la mayor parte del texto.
- ✓ Segoe UI de 9 ptos utilizado para elementos de texto cortos, como títulos de botones.
- ✓ Segoe UI Light de 20 ptos utilizado para texto corto que cabe en una sola línea y que debe de atraer la atención del usuario.
- ✓ Segoe UI Light de 42 ptos para elementos muy destacados que tengan una o dos palabras en una sola línea.
- ✓ Calibri utiliza un tamaño de 13 ptos.
- ✓ Cambria utiliza los siguientes tamaños: 9 ptos, 11 ptos y 20 ptos.

• **Color de Fuente**

Metro no establece un color determinado para la fuente siempre y cuando se utilice la opacidad adecuada.

- ✓ Opacidad del 100% para texto principal.
- ✓ Opacidad del 60% para texto secundario.
- ✓ Opacidad del 80% para texto en estado de desplazamiento.
- ✓ Opacidad del 40% para texto en blanco y negro estado presionado
- ✓ Opacidad del 60% para texto de otro colores.

3.9 DIRECTRICES PARA BOTONES

El diseño de los botones son los que dan vida a una aplicación si estos son elegantes y llamativos para el usuario, es por eso que las directrices de los botones son una prioridad para los desarrolladores, se debe elegir el mejor control de botones para su aplicación, según su necesidad.

• Tipos de Botones

Existen 3 tipos de botones para las aplicaciones metro, estos son: Enviar, Restablecer, Normal.

- ✓ **Botón Enviar.**- Sirve para realizar una acción que envíe la entrada del usuario a un servidor.
- ✓ **Botón Restablecer.**- Sirve para borrar la entrada del usuario restableciendo el formulario.
- ✓ **Botón normal.**- Realiza una acción personalizada por el usuario.

• Recomendación para los Botones

- ✓ El texto dentro del botón debe ser concreto y corto, identificando la actividad del botón.
- ✓ Los botones es recomendable utilizar imágenes para diferenciar el botón enviar del aceptar.
- ✓ El texto no debe cambiar de color al reestablecer a menos que sea para localizar, No debe intercambiar los estilos predeterminados de los botones Enviar, Restablecer y normal.

3.10 DIRECTRICES PARA CASILLAS

Son las casillas donde el usuario puede seleccionar o deseleccionar una acción de la aplicación.

Selección de una opción entre dos Usa una sola casilla para elegir entre sí y no.	<input checked="" type="checkbox"/> I agree
Una o más opciones que no son exclusivas mutuamente Utiliza un grupo de casillas cuando los usuarios puedan seleccionar cualquier combinación de opciones.	Pizza Toppings <input type="checkbox"/> Pepperoni <input checked="" type="checkbox"/> Beef <input type="checkbox"/> Mushrooms <input checked="" type="checkbox"/> Onions
Opciones mixtas Si la opción se aplica a varios objetos, se utiliza una casilla para indicar si se aplica a todos, a alguno o a ningún objeto.	Pizza Toppings <input checked="" type="checkbox"/> All <input type="checkbox"/> Pepperoni <input checked="" type="checkbox"/> Beef <input type="checkbox"/> Mushrooms <input checked="" type="checkbox"/> Onions

ILUSTRACIÓN 62: “Casilla opcionales”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

- **Recomendación para Casillas**

- ✓ Es recomendable poner las casillas dentro de una etiqueta con esto se logra aumentar el área de aplicación de la selección.
- ✓ Si una casilla está en estado indefinido indica que es posible definirla en la selección.
- ✓ No es recomendable usar dos grupos de casillas juntas, porque darían un significado de confusión al usuario.

3.11 DIRECTRICES PARA BOTONES DE RADIO

Un control de selección unitaria son los botones de radio los cuales permiten al usuario seleccionar una sola opción de un grupo de opciones, las recomendaciones para este tipo de control son:

- ✓ Utiliza el botón de radio para llamar la atención de las opciones de selección.
- ✓ No Utilizar los botones de radio si solo hay dos opciones mutuamente exclusivas que puedes combinar en una única casilla.
- ✓ El botón de radio debe de estar dentro de un elemento de etiqueta para hacer fácil su selección.

- ✓ Coloca el texto de la etiqueta detrás del control de botón de radio.
- ✓ No utilizar más de 8 opciones de botones de radio, utiliza un promedio de 2 y 7 botones de radio.

3.12 DIRECTRICES PARA VÍNCULOS

Permite normalizar los controles de vínculos en las aplicaciones, es recomendado colocar texto sobre los controles de vínculos de esta manera se evita que el usuario no sepa que es lo que está siguiendo cuando ponga su dedo sobre el link.

Es preferible colocar el domino de un sitio web si en enlace es externo, pues se brinda información sobre la herramienta, y el usuario sabrá hacia donde se está enlazando.

El texto del vínculo debe ser igual que el de los botones corto y concreto.

3.13 DIRECTRICES PARA CUADROS DE DIALOGO

- **Información Urgente.-** Los cuadros de dialogo son manipulados para dar un mensaje de información apremiante para el usuario, en ello se podrá confirmar y denegar la acción.
- **Errores.-** Los mensajes de error utilizan cuadros de diálogos tipo mensaje, por lo general usan colores fuertes para su diseño.
- **Preguntas.-** Utiliza cuadros de diálogos para preguntas que puedan bloquear la aplicación y así hacer que el usuario los confirme o rechacé. Las preguntas de bloqueo se muestran a usuarios de alto nivel.

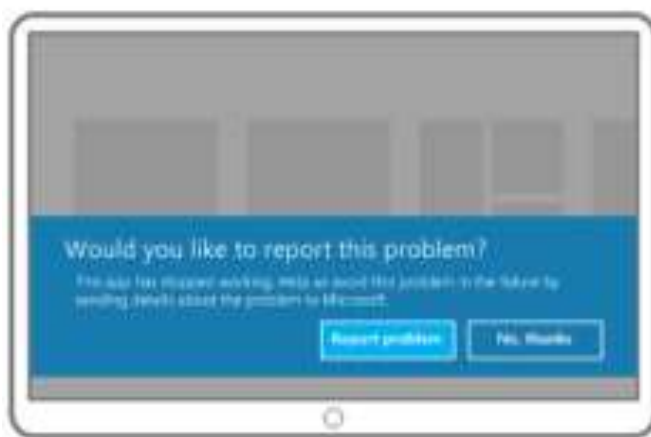


ILUSTRACIÓN 63: “Cuadro de mensajes”

Fuente: <http://msdn.microsoft.com/es-es/library/windows/apps/hh761500.aspx>

3.14 DIRECTRICES PARA ICONOS

En estas directrices se detallan procedimientos recomendados para la creación y actualización de iconos. El icono se encontrara en el menú inicial y será la representación gráfica de la aplicación así que se pretende que sea sencillo pero atractivo.

- **Filosofía de diseño de los iconos**

El objetivo de un icono es atraer al usuario a que abra la aplicación, debe de ser dinámico con contenido nuevo y seductor, No se debe de exceder en el uso de colores fuertes, un diseño limpio y elegante trae el éxito deseado. Las características para un icono atractivo son:

- ✓ Contenido nuevo y actualizado, así los usuarios sentirán a la aplicación activa.
- ✓ Actualizaciones personalizadas, el icono contendrá información de acuerdo con los gustos del usuario.
- ✓ Contenido relevante según el contexto actual del usuario

- **Tamaño de Icono**

El icono es imprescindible en una aplicación y se debe de elegir el tipo de icono de acuerdo a la necesidad. Existen 2 tipos de iconos, cuadrados y anchos.

- ✓ **Icono Cuadrado.**-El icono cuadrado se utiliza solamente cuando la aplicación no utilizara notificaciones y cuando la aplicación envía actualización que no deben mostrarse en la pantalla de inicio. Los iconos cuadrados muestran menos contenido que los iconos anchos, así que prioriza el contenido.
- ✓ **Icono Ancho.**-El icono ancho siempre tiene que actualizar frecuentemente y se utiliza si la aplicación tiene contenido nuevo e interesante para mostrar al usuario
- ✓ **Iconos predeterminados** El icono predeterminado es estático, cuadrado o ancho y la utilizan algunas aplicaciones cuando no está determinado un icono en especial a la aplicación. Utiliza el icono predeterminado con su imagen para reflejar el estilo de la aplicación. Centra el logotipo de icono horizontalmente. Mantener verticalmente el logotipo en iconos cuadrados y anchos. Coloca el nombre de la aplicación en la parte inferior del logotipo

CAPÍTULO IV

4 METODOLOGÍA

La metodología es el conjunto de procedimientos interrelacionados y usados para alcanzar un fin en particular que establece una investigación o un proceso que requieren habilidades, conocimientos o cuidados específicos.

4.1 PROCESO DE INVESTIGACIÓN

4.1.1 ANÁLISIS

Este proyecto está directamente diseñado para la creación de aplicaciones estilo metro la cual será un pilar fundamental en el desarrollo de aplicaciones para los alumnos de la Universidad Técnica del Norte, esta investigación permitirá que los alumnos puedan tener conocimientos fundamentales los nuevos diseños de programación de interfaces de esa forma aventurase un poco más en la creación de software para dispositivos móviles.

Esta investigación será transmitida por los diferentes medios web, para que esté al alcance de todas las personas que necesiten de esta investigación.

4.1.2 TIPO DE INVESTIGACIÓN

El proyecto es de investigación por lo tanto documental, la cual permite la consulta de la información de las diferentes fuentes digitales, artículos, libros, consultas Web, etc. Para conseguir la más eficiente información para cumplir con el objetivo de la investigación, además proveer un material claro y preciso.

Y con la ayuda de la técnica y el instrumento de la metodología de la investigación se estima cumplir con el propósito y objetivos de este proyecto.

4.1.3 MÉTODO

El método deductivo será lo que se utilizará para este proyecto, pues permite ir de algo general hacia algo más específico y particular.

De esta forma ir desde lo general que sería Windows 8 hasta el objetivo específico que son las aplicaciones metro, de esta manera conseguir una guía concreta y fácil de entender sobre el desarrollo de aplicaciones metro sobres Windows 8.

4.1.4 TÉCNICA

La información proporcionada para este proyecto de sustentará en las lecturas realizadas en recursos en la Web, y/o en revistas, artículos, libros de la biblioteca de la Universidad Técnica del Norte, de la cual se presentará en forma resumida, lo cual consiste en una lectura general de la información recopilada sacar los puntos más importantes y sobresalientes acerca del tema de investigación que será lo cual sustente la teoría de la investigación.

Como punto importante es conveniente realizar varias lecturas del texto para encontrar los puntos importantes de su contenido.

4.1.5 INSTRUMENTO

El instrumento óptimo para este proyecto de investigación es la documentación para lo cual se desarrollará un documento en el cual se recopila toda la información más relevante de todas las búsquedas realizadas con anterioridad, la misma que será la base de toda la teoría documental, cabe señalar que esta información será entregada con las fuentes de consulta y sus respectivos autores.

4.2 METODOLOGÍA INFORMÁTICA

4.2.1 PROCESO UNIFICADO DE DESARROLLO

El proceso unificado de desarrollo es una metodología que consta de distintas actividades y que se puede adoptar a un proyecto ya sea de software o investigativo.

4.2.2 CARACTERÍSTICAS

- Ordenado y disciplinado para la distribución de tareas
- Está dividido en 4 fases concretas de desarrollo
- Dirigido por caso de uso
- Orientado a la identificación de riesgos

4.2.3 FASES DEL PROCESO UNIFICADO DE DESARROLLO

4.2.3.1 FASE INICIO

En esta fase se define los objetivos y el alcance del proyecto, incluyendo los patrocinadores, además de incluir los riesgos asociados al proyecto, identificar posibles interesados y relacionados con el proyecto y proponer el documento visión del mismo.

El proyecto se enfoca en el desarrollo de aplicaciones a estilo metro sobre Windows 8, para lo cual se elaborará documentación pertinente que demuestre la investigación, adicionalmente y para la demostración se diseñará una pequeña aplicación estática un Lector RSS

4.2.3.2 DECLARACIÓN DEL PROBLEMA

TABLA 4: “Declaración del problema”

El problema	Un total desconocimiento del desarrollo de aplicaciones para Windows 8 con estilo metro y las directrices que son base de su desarrollo.
Afecta	Usuarios de la aplicación, y a los estudiantes de la Universidad Técnica del Norte en la Carrera de Ingeniería en Sistemas, además para todos los desarrolladores de software
Impacto	Dificultad en desarrollar con nuevas herramientas que permitan la creación de software con estilo metro, con esto se evitará que los usuarios no queden inconformes y no busquen otras opciones en ningún otro lado
La solución podría	Desarrollar un documento que sirva de guía base para los programadores de la UTN y de esta manera hacer conocer el desarrollo de aplicaciones metro sobre Windows 8.
Posicionamiento del producto	
Para:	Estudiar y analizar toda la base de desarrollo de la plataforma Windows 8 para aplicaciones con estilo metro.

Quien:	Estudiantes UTN, Facultad de Ingeniería en Sistemas
Nombre del Producto	Desarrollo de aplicaciones metro sobre Windows 8
El proyecto	Es una guía completa de desarrollo de aplicaciones a estilo metro sobre plataforma Windows 8

4.2.3.3 STAKEHOLDERS

Fuente: Autor

TABLA 5: "Stakeholders"

Actor	Actividad
Desarrollador	Crear aplicaciones
Usuario	Acceder a la aplicación y poder manipular

Fuente: Autor

4.2.3.4 CASO DE USO DE NEGOCIO

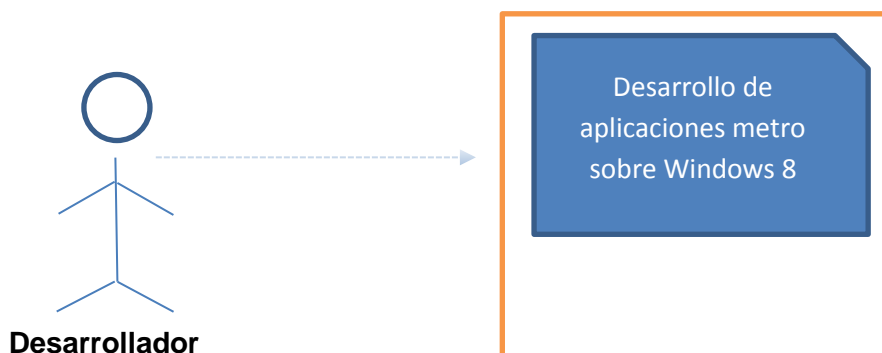


ILUSTRACIÓN 64: "Caso de uso del negocio"

Fuente: Autor

4.2.3.5 LISTA DE RIESGO

TABLA 6: " Lista de riesgos"

Problema	Descripción	Prioridad
Desconocimiento del estilo metro	Falta de información y material en el tema de desarrollo de aplicaciones con estilo metro	Alta
Desconocimiento de la plataforma de desarrollo	Poca información de la plataforma de desarrollo por ser una nueva versión de Windows	Alta
Desconocimiento de las herramientas de desarrollo	No se conoce las herramientas básicas y necesarias para el desarrollo de aplicaciones con estilo metro	Medio
Poca producción de aplicaciones con estilo metro	Hasta el momento no se han creado muchas aplicaciones siguiendo el patrón de desarrollo	Medio

Fuente: Autor

4.2.4 FASE DE ELABORACIÓN

4.2.4.1 CASO DE USO DE SISTEMA

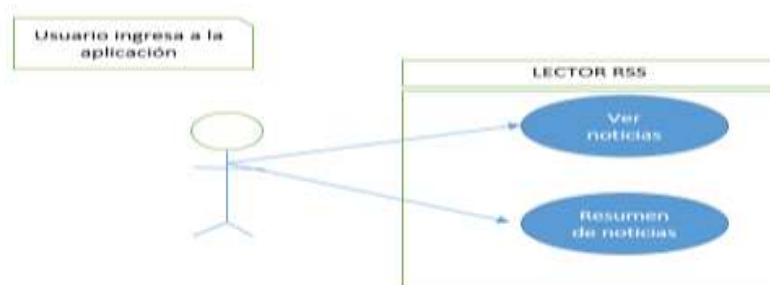


ILUSTRACIÓN 65: "Caso de uso del sistema"

Fuente: Autor

4.2.4.2 ESCENARIO CASOS DE USO

TABLA 7: “Escenario de casos de uso”

Caso de uso	LECTOR RSS	
Objetivo	Visualizar las noticias y tener acceso a un resumen de la noticias	
Actor	Usuario	
Descripción	Pasos	Acción
	1.	Visualizar noticias
	2.	Acceder a resumen de noticias
Frecuencia	Cada vez que el usuario ingrese al aplicativo	

Fuente: Autor

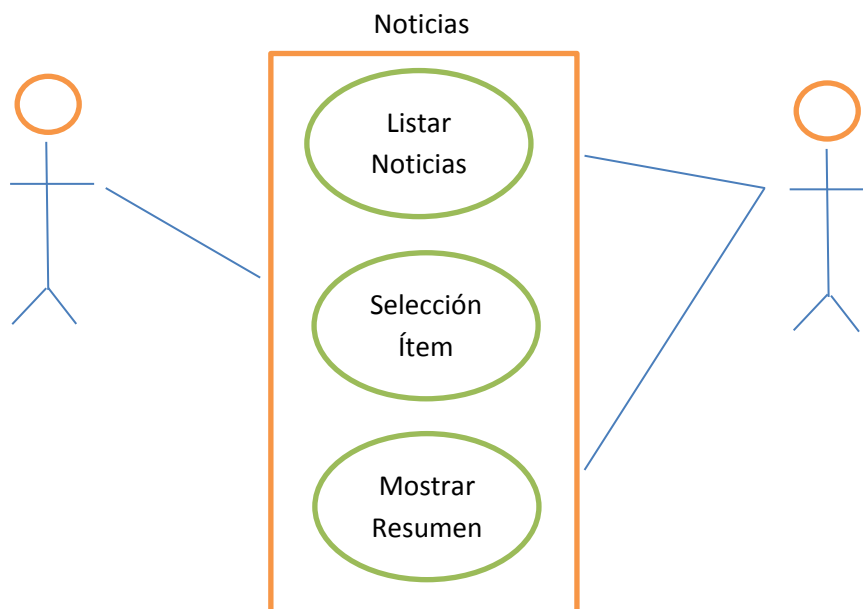


ILUSTRACIÓN 66: “Caso de uso del escenario”

Fuente: Autor

TABLA 8: “Escenario de noticias”

Caso de uso	Noticias	
Objetivo	Dar a conocer al usuario las noticias actualizadas y permitirle elegir una noticia para ver su resumen	
Actor	Usuario, Aplicativo	
Descripción	Pasos	Acción
	1.	El aplicativo visualiza las noticias actualizadas
	2.	El usuario mira las noticias de la lista y selecciona una para ver su resumen
	3	El aplicativo toma lo seleccionado y presenta al usuario
Frecuencia	Cada vez que el usuario ingrese al aplicativo y desee revisar las noticias	

Fuente: Autor

4.2.4.3 DIAGRAMA DE ACTIVIDADES

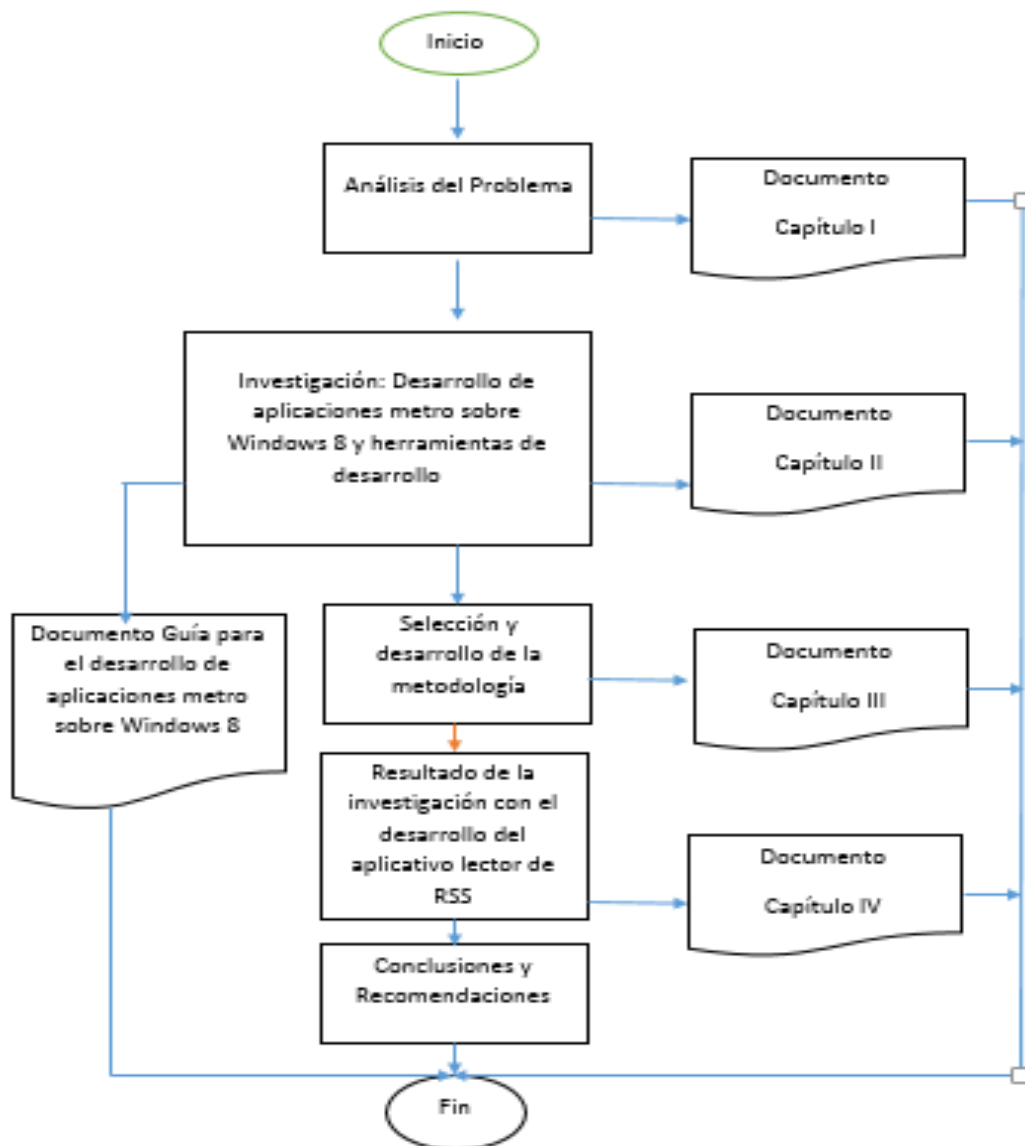


ILUSTRACIÓN 67: "Diagrama de actividades"

Fuente: Autor

El diagrama de actividades demuestra los pasos realizados para la culminación de este proyecto, en la cual se representa la investigación, la redacción y desarrollo de los distintos capítulos y el desarrollo del aplicativo Lector de RSS como resultado de la investigación.

4.2.4.4 DIAGRAMA DE SECUENCIA

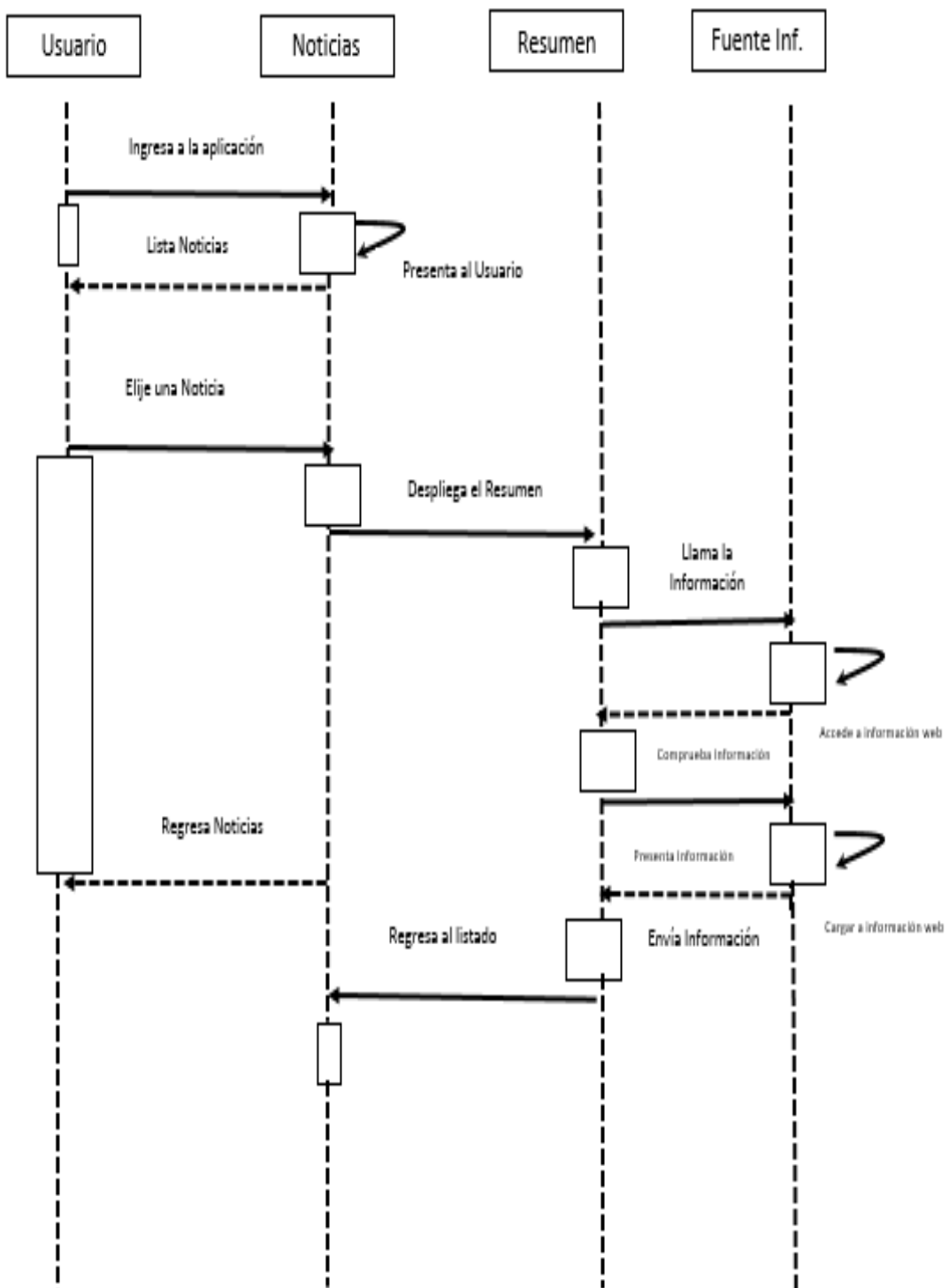


ILUSTRACIÓN 68: "Diagrama de secuencia"

4.2.4.5 DIAGRAMA DE PAQUETES

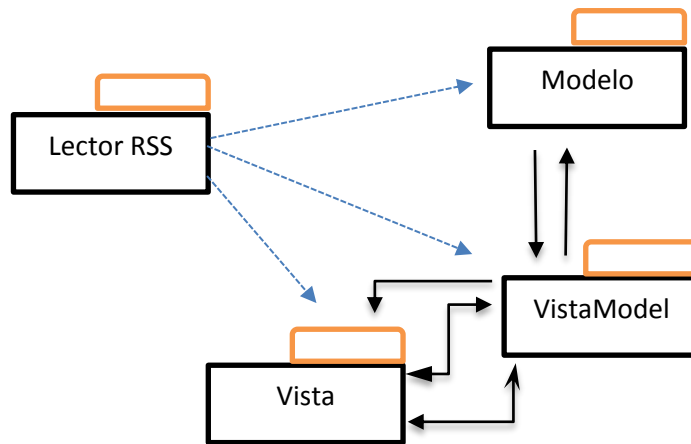


ILUSTRACIÓN 69: "Diagrama de paquetes"

Fuente: Autor

4.2.4.6 ARQUITECTURA DEL APLICATIVO

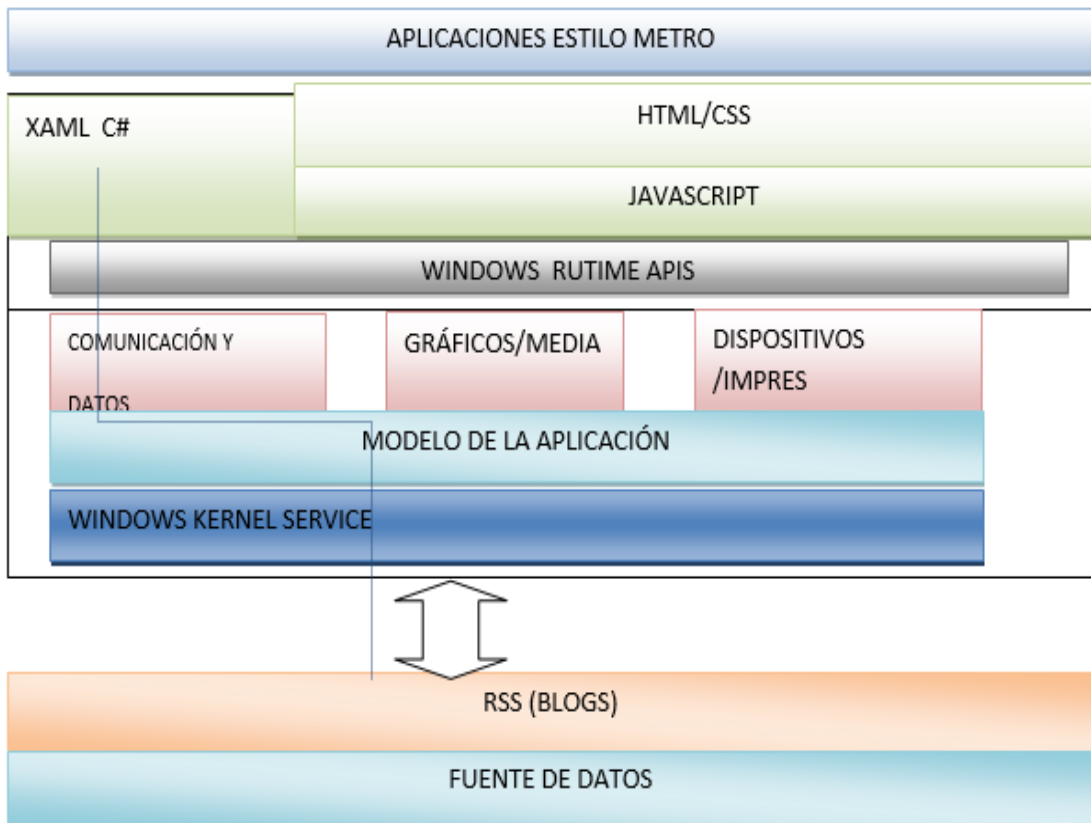


ILUSTRACIÓN 70: "Arquitectura del aplicativo"

Fuente: Autor

4.2.5 FASE DE CONSTRUCCIÓN

El propósito de esta fase es completar la funcionalidad del proyecto, obtener la construcción del software, administrar cambios de acuerdo a las evaluaciones realizadas por el usuario y se realizará las mejoras para el proyecto.

4.2.5.1 DESARROLLO DEL APLICATIVO “LECTOR DE RSS”

4.2.5.2 ESTRUCTURA

Si bien diseñar un lector RSS no llegará a demostrar con certeza el verdadero nivel y alcance de las aplicaciones metro, será un inicio bastante didáctico que permita dar un vistazo preciso y fácil a la nueva experiencia.

Definición de los alcances reales del aplicativo:

- Se conectará a una fuente de contenido RSS rica en contenido: Imágenes, Texto, Audio etc.; pues es conocido que una fuente RSS no es solo texto pues esto lo volvería sencillamente perezoso mirarlo para el usuario lo cual no se busca.
- La representación de la información que visualizará el beneficiario debe ser de forma agradable, siguiendo las directrices de las aplicaciones metro, y de la plataforma WinRT para tener una estética homogénea, fácil de usar incluso para un usuario novato.
- Se usará complementos de WinRT, la plataforma nativa suministra interfaces y lineamientos para las funcionalidades comunes, que permite acoplarse al sistema, para no tener que crear funcionalidades.
- Se puede utilizar la funcionalidad Settings.
- Como las aplicaciones metro comúnmente van a hacer usadas por dispositivos móviles y táctiles hay que fijarse bien en las diferentes orientaciones de la pantalla, esta deberá presentar consistencia para cada una de sus posibles orientaciones.
- Debe tener planificada para su uso bajo el fallo, como por ejemplo que sucede si no tiene internet, en este caso la aplicación debe informar al usuario este inconveniente sin necesidad de caerse la App.

Creación del aplicativo

Para crear una solución tienda Windows en VS (Visual Studio 2012) elegir file-> new-> project en las plantillas seleccionar Windows Store. Y nombrer BlogUtn

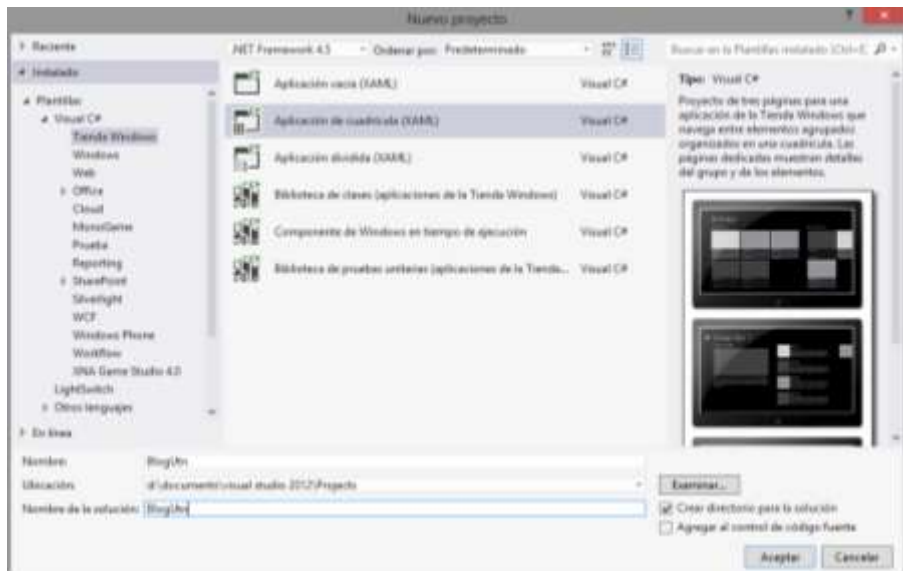


ILUSTRACIÓN 71: “Elegir nueva solución”

añadir un nuevo ítem a la solución, no importa el nombre solo se lo hace para poder extraer archivos necesarios para nuestra aplicación.

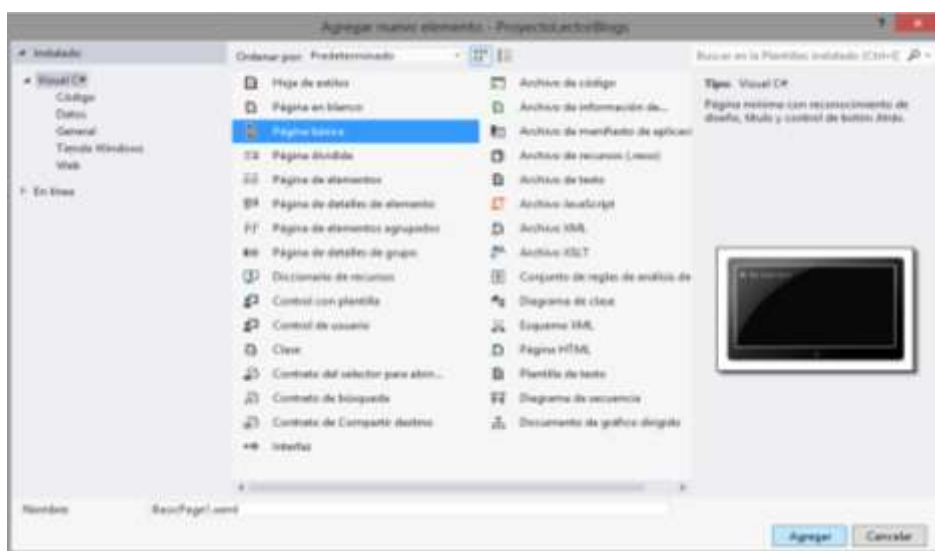


ILUSTRACIÓN 72: “Página básica”

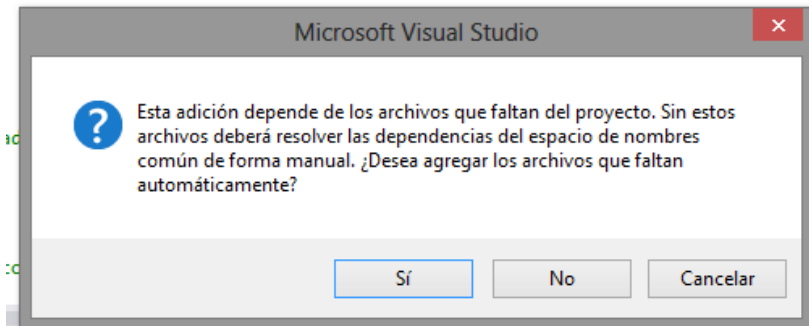


ILUSTRACIÓN 73: “Mensaje de solución en blanco”

Si se le aparece este mensaje aceptarlo.

Hay que eliminar el mainpage.xaml, el motivo es porque es nueva solución y no se la utiliza.

Se utilizará y creará 4 carpetas independientes para el modelamientos de datos.

Las carpetas son

Modelo, Vista, Vistamodelo, Útil

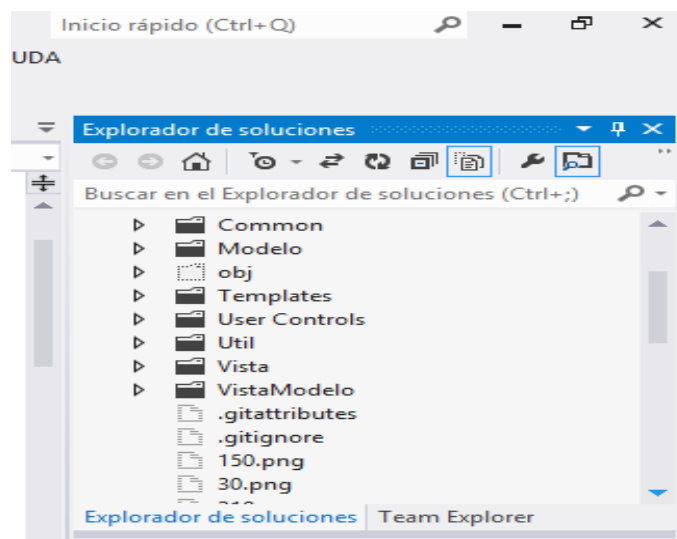


ILUSTRACIÓN 74: “Vista general del proyecto”

Dentro de la carpeta vista crear una página básica cuyo nombre será LectorMainView.xaml.

Abrir el archivo `app.xaml.cs` y poner en `rootFrame(typeof(LectorMainView.xaml))` que es donde se direcciona a la página principal con la que comenzará a ejecutarse al aplicativo.

```
if (rootFrame.Content == null)
{
    // Cuando no se restaura la pila de navegación para navegar a la primera página,
    // configurar la nueva página al pasar la información requerida como parámetro
    // parámetro
    if (!rootFrame.Navigate(typeof(LectorMainView), args.Arguments))
    {
        throw new Exception("Failed to create initial page");
    }
}
// Asegurarse de que la ventana actual está activa.
Window.Current.Activate();
}
```

ILUSTRACIÓN 75: “Archivo App.xaml”

Escribir la biblioteca de clase `BlogUtn.Vista` en el área de using

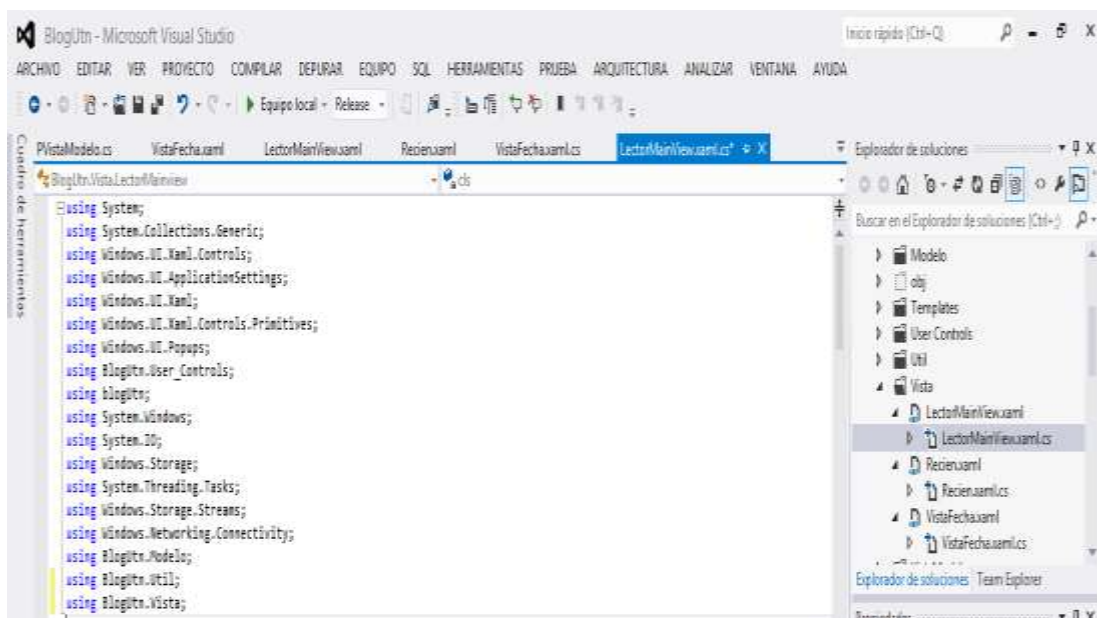


ILUSTRACIÓN 76: “Accesibilidad de clases”

En este momento realizar una prueba con el emulador si no dispone de Windows 8 o con el run local si tiene instalado Windows 8

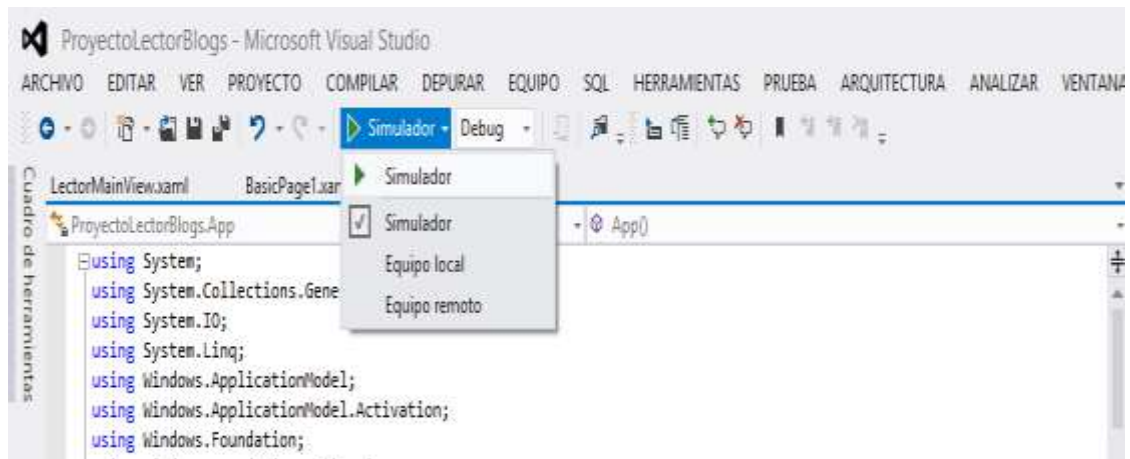


ILUSTRACIÓN 77: “Ejecutor del aplicativo”

Se obtendrá el inicio del aplicativo.

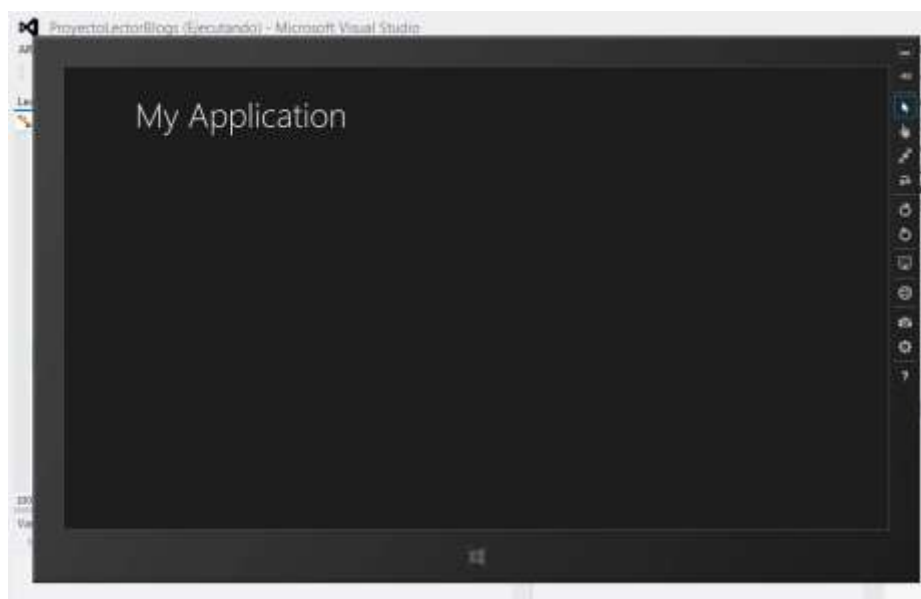


ILUSTRACIÓN 78: “Vista en el simulador”

Modelamientos de Datos.

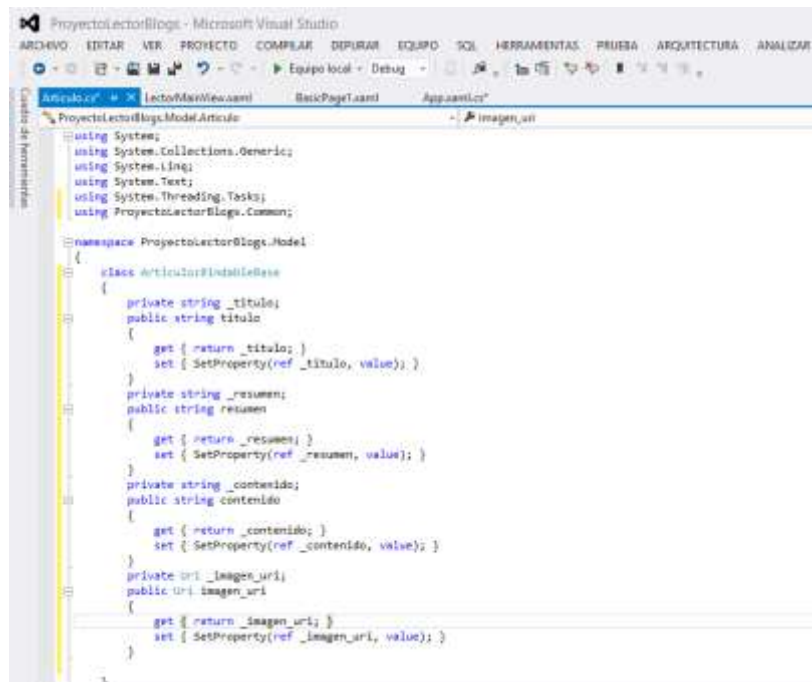
El objetivo del lector es publicar noticias desde la web, por lo que se modela la clase de los artículos o noticias.

Entre todos los parámetros que se pueden ingresar solo se elige 4 que son o se considera los más importantes:

- Título
- Imagen
- Resumen
- Contenido
- Fecha de Publicación

Se creará la clase Artículos en la carpeta de modelamiento o Modelo.

Ahora identificar el UI (User Interface), en XAML existe el concepto de Binding que permite asociar elementos de la UI a un modelo de datos, aquí hay que adicionar características adicionales para su comunicación.



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ProyectoLectorBlog.Common;

namespace ProyectoLectorBlog.Model
{
    class Articulo
    {
        private string _titulo;
        public string titulo
        {
            get { return _titulo; }
            set { SetProperty(ref _titulo, value); }
        }
        private string _resumen;
        public string resumen
        {
            get { return _resumen; }
            set { SetProperty(ref _resumen, value); }
        }
        private string _contenido;
        public string contenido
        {
            get { return _contenido; }
            set { SetProperty(ref _contenido, value); }
        }
        private Uri _imagen_uri;
        public Uri imagen_uri
        {
            get { return _imagen_uri; }
            set { SetProperty(ref _imagen_uri, value); }
        }
    }
}

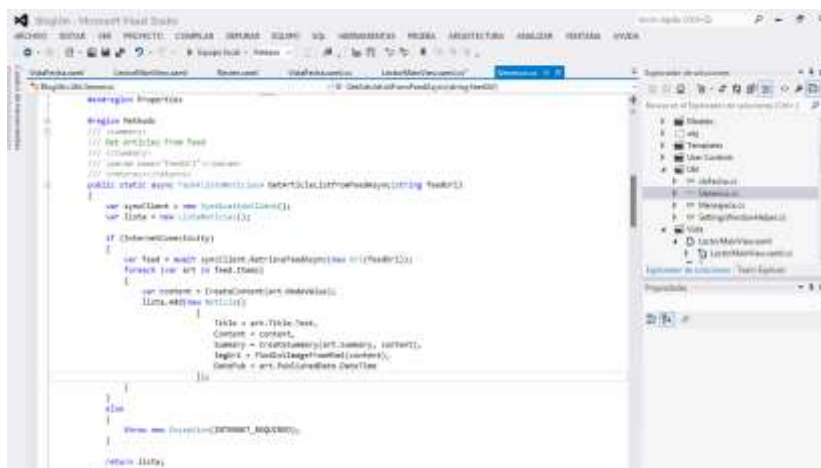
```

ILUSTRACIÓN 79: “Fragmento del código del aplicativo”

Como se presentará varios artículos. Para lo cual se necesita una colección de clases de tipo artículos. Es importante que esta colección notifique a la UI cuando se le agreguen, borren, o se le modifiquen elementos.

Con la ayuda de Vs2012 se adicionará en la carpeta Modelo una clase llamada *listaArticulos* y se tendrá el siguiente código.

En la carpeta útil se creará una clase llamada Genéricos



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Threading;

namespace Useful
{
    public class Genéricos
    {
        // Comentarios
        // Get Articulos from Feed
        // Comentarios
        // Comentarios
        // Comentarios

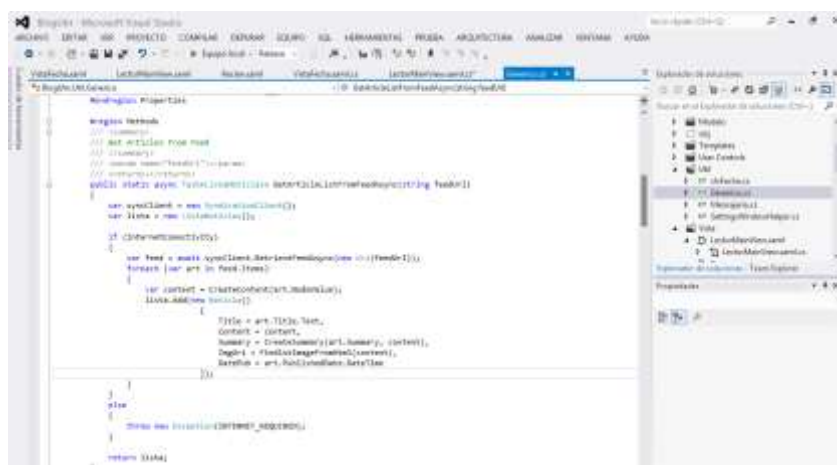
        public static async Task<IEnumerable<Article>> GetArticulosFromFeedAsync(string feedURL)
        {
            var syndClient = new SyndicationClient();
            var lista = new List<Article>();

            if (CheckNetworkConnectivity())
            {
                var feed = await syndClient.GetFeedAsync(new Uri(feedURL));
                foreach (var art in feed.Items)
                {
                    var content = DataContractSerializerDataContract();
                    lista.Add(new Article()
                    {
                        Title = art.Title.Text,
                        Content = content,
                        Summary = DataContractSerializerDataContract(),
                        ImageURL = FindImageUrlFromFeedItem(content),
                        DatePublished = art.PublishDate.DateTime
                    });
                }
            }
            return lista;
        }
    }
}

```

ILUSTRACIÓN 80: “Creación clase genéricos”

Implementar en el método un objeto SyndicationClient para extraer los datos del feed, y este traerá los artículos expuestos en el RSS, para listarlos uno a uno.



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Threading;

namespace Useful
{
    public class Genéricos
    {
        // Comentarios
        // Get Articulos from Feed
        // Comentarios
        // Comentarios
        // Comentarios

        public static async Task<IEnumerable<Article>> GetArticulosFromFeedAsync(string feedURL)
        {
            var syndClient = new SyndicationClient();
            var lista = new List<Article>();

            if (CheckNetworkConnectivity())
            {
                var feed = await syndClient.GetFeedAsync(new Uri(feedURL));
                foreach (var art in feed.Items)
                {
                    var content = DataContractSerializerDataContract();
                    lista.Add(new Article()
                    {
                        Title = art.Title.Text,
                        Content = content,
                        Summary = DataContractSerializerDataContract(),
                        ImageURL = FindImageUrlFromFeedItem(content),
                        DatePublished = art.PublishDate.DateTime
                    });
                }
            }
            return lista;
        }
    }
}

```

ILUSTRACIÓN 81: “Uso del método syndicationclient”

Se necesita crear objeto delegado de orquestar las funcionalidades que se correspondería tener creado para servir las a la interfaz gráfica.

Hay que darle funcionalidad para que exista y consuma el modelo de datos y llenarlo de información, para luego completar con los requerimientos para comunicarnos con la UI.

Para consumir el modelo de datos. Agregar una clase llamada VistaModelo no estática y va a derivar bindableBase de la siguiente forma

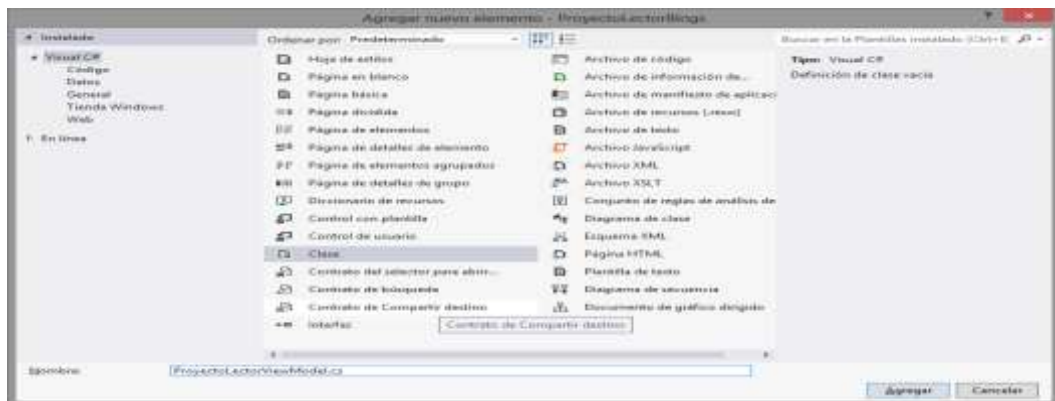


ILUSTRACIÓN 82: "Creación de la clase"

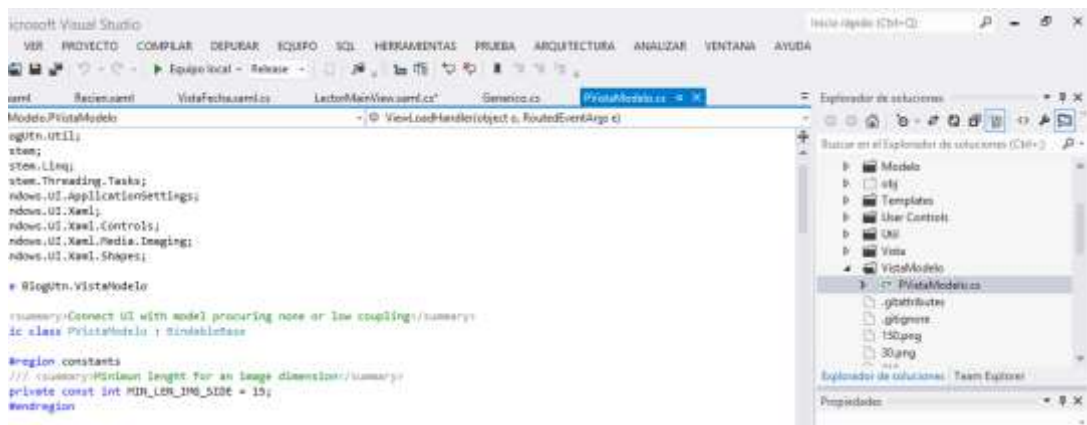


ILUSTRACIÓN 83: "Vista del formato de clase"



ILUSTRACIÓN 84: "Codificación de la clase"

Dentro del *VistaModelo* se adiciona la fuente de datos, se tendrán algo como lo antes descrito

Se observa la instancia del objeto lo cual permitirá facilidad del Binding desde la UI

Por el momento la colección está vacía, hay que llenarla con los datos de una fuente RSS para lo cual se usará la clase *Genericos*.

Al momento de pasar la clase *VistaModelo* hacia la UI, evitar boquearla porque eso daría la idea de lentitud de la App, lo cual sería molesto para el usuario, para evitar esto se utiliza un método *async* que no se llame desde el constructor, para evitar el bloqueo, el método *async* se encargará de inicializar y cargar los datos del feed en la lista.



ILUSTRACIÓN 85: “Archivo *LectorMainView.xaml*”

El objeto de extracción de datos sería *VistaModelo* el cual indicará a la vista que se ha creado un objeto de contexto de datos de donde debe tomar la información.

Esto se hace desde el XAML en el *VistaModelo* y se lo declarará como un objeto global y se utilizará para todo el proyecto.

Para indicarle a la Vista que fuente deberá utilizar se usa el *loaded* y aquí le asignar el *dataContext*.

4.2.5.3 INTERFAZ

Se ubica en el grid que por defecto se crea en el aplicativo.

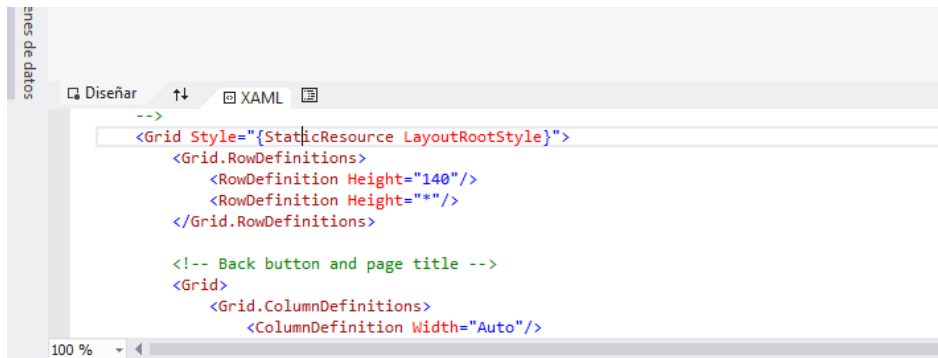


ILUSTRACIÓN 86: “Fragmento de código archivo LectoroMainview.xaml”

Agregar el siguiente código.

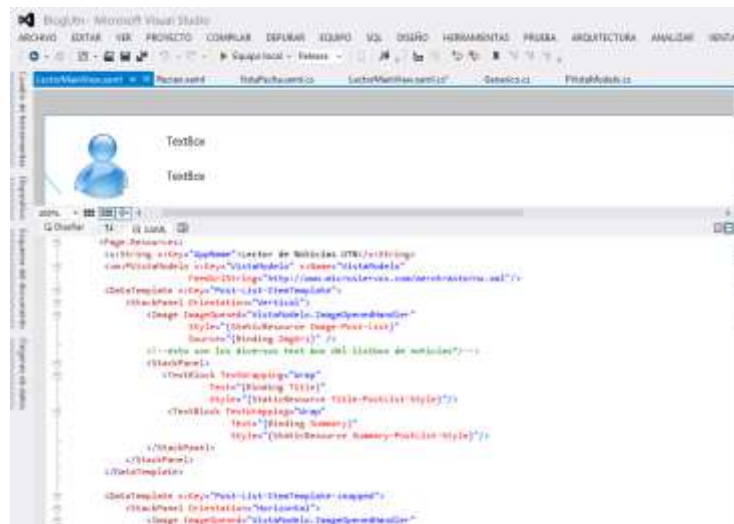


ILUSTRACIÓN 87: “Código del archivo LectoroMainview.xaml”

La interfaz gráfica se puede hacer de tres formas

- Editando el código XAML
- Usando el diseñador de Visual Studio
- Usando Expression Blend (incluido también gratis con Visual Studio 2012)

Si se lo hace desde el editor XAML se debe considerar que se debe conocer muy bien los códigos de creación ya que esto es mucho más tedioso escribir cada línea de código para su representación, es importante saber XAML pues es así como se logra tener un control absoluto de la UI pero sin lugar a dudas utilizar otras herramientas será más productivo.

Expresión Blend es la herramienta más productiva para este tipo de trabajo es más flexible no se tiene el acceso total del código pero lo que se necesita es tener muy claro de cómo va quedando la interfaz, así que sería lo más óptimo utilizarlo.

Esquema principal de la App

Esta es la forma en que se distribuirá el espacio de la App.

	Col 1	Columna 2	Columna 3
fila 1	icono	Nombre App	Titulo Articulo Actual
		RSS ITEM	Contenido Articulo
		RSS ITEM	
fila 2		RSS ITEM	
		RSS ITEM	
		RSS ITEM	
		RSS ITEM	

ILUSTRACIÓN 88: “Modelo del aplicativo”

Fuente: Autor

Si se lo hace por medio del editor tienes lo siguiente:

```
<GRID.ROWDEFINITIONS>
```

```
  <ROWDEFINITION HEIGHT="140"/>
```

```
  <ROWDEFINITION HEIGHT="*/>
```

```
</GRID.ROWDEFINITIONS>
```

Y las columnas así:

```
<GRID.COLUMNDEFINITIONS>
```

```
  <COLUMNDEFINITION WIDTH="100"/>
```

```
  <COLUMNDEFINITION WIDTH="500"/>
```

```
  <COLUMNDEFINITION WIDTH="*/>
```

```
</GRID.COLUMNDEFINITIONS>
```

Para lograr una interfaz similar a la siguiente

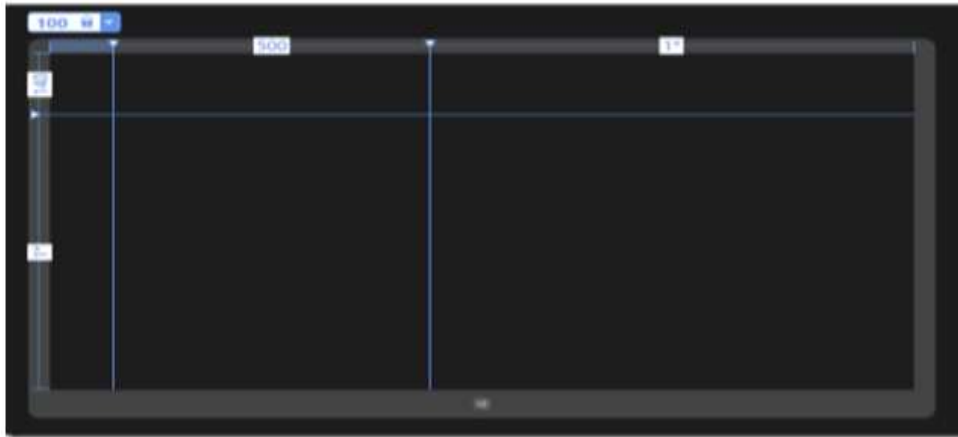


ILUSTRACIÓN 89: “Interfaz base del aplicativo”

De esta manera se tendría



ILUSTRACIÓN 90: “Menú lateral de aplicativo”

4.2.6 FASE DE TRANSICIÓN

Comprueba que la App esté en correcto funcionamiento y se realizar lar respectivas pruebas de validación, ajustar errores y defectos que se puedan ocasionar al momento de la comunicación con los modelos de datos. Verifica que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

4.2.6.1 REQUERIMIENTOS TÉCNICOS

Para el funcionamiento correcto de aplicaciones metro Windows 8 se consideran los siguientes requisitos:

- Sistema Operativo Windows 8 en cualquiera de sus versiones
- Procesador Core dual de 2.0 GHZ o superior
- 2 GB de memoria RAM
- Tarjeta gráfica compatible con DirectX9 Funcionalidad Táctil de la PC (Opcional)

4.2.6.2 INTERFAZ LECTOR RSS

Ventana de inicio



ILUSTRACIÓN 91: “Pantalla inicio lector de noticias RSS”

La pantalla de inicio es la imagen que por default permite la herramienta de desarrollo presentar como un constructor del aplicativo.

Menú Principal



ILUSTRACIÓN 92: “Pantalla del aplicativo lector de noticia RSS”

- **Información usuario:** Muestra la información relacionada con el usuario del equipo donde se está ejecutando el aplicativo.
- **Selector Vertical:** Muestra un listado de noticias para que el usuario pueda seleccionar
- **Contenedor:** Muestra la información relacionada con la selección del selector vertical
- **Menú del aplicativo:** Permite elegir las opciones que tiene el Aplicativo

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

La utilización de este documento facilitará a los desarrolladores la creación de aplicaciones metro y de esta manera aportando a la difusión del tema dentro del país.

En Conclusión a la presente investigación desarrollada y la redacción del documento se puede concluir:

Las aplicaciones Windows 8 con el nuevo estilo metro buscan revolucionar la perspectiva de desarrollo de aplicaciones, creando aplicaciones cómodas, sencillas, funcionales con una facilidad de uso inmejorable para el usuario y que se las puede utilizar en distintos dispositivos táctiles y de escritorio.

Las directrices del estilo metro permiten el correcto diseño y desarrollo de la interfaz de usuario, ya que están basadas en los principios dados por creador de este estilo Microsoft.

Las aplicaciones metro son desarrolladas con herramientas y lenguajes de programación que ya se manejan desde hace muchos años atrás, con ello facilita la programación de este tipo de aplicaciones.

5.2 RECOMENDACIONES

Una vez terminada la tesis se considera atractivo seguir la investigación sobre la plataforma Windows con temas relacionados a los siguientes:

- Con la documentación proporcionada en el presente trabajo de grado es recomendable realizar una investigación sobre las aplicaciones metro style basada en estilo de escritorio, pues el aplicativo está destinado al desarrollo en metro style para la tienda Windows.
- Extender los alcances del aplicativo al punto que se vuelva dinámico para cualquier fuente RSS y Atom
- Profundizar la investigación con Visual Studio 2013 y Windows 8.1 que son las nuevas herramientas proporcionadas por Microsoft para el desarrollo de aplicaciones metro style.
- Crear otros aplicativos basados en el Lector RSS para nuevos dispositivos móviles como las tables y los Celulares pero que tenga otro sistema operativo como por ejemplo Windows Mobile o el mismo Android.

5.3 BIBLIOGRAFÍA

FLOREZ, J. R. (2012). TRABAJO COLABORATIVO 2. Mexico: Universidad Nacional Abierta y Distancia .

Guia de producto (2012) Windows8_ReleasePreview_ProductGuide_Final_Spanish

Jorge, S. (2012). WinRT. USA.

Microsoft. (2013). <http://windows.microsoft.com/es-419/windows-8/getting-around-tutorial>. Obtenido de <http://windows.microsoft.com/es-419/windows-8/getting-around-tutorial>: <http://windows.microsoft.com/es-419/windows-8/getting-around-tutorial>

Jerry Honeycutt(Library of Congress Control Number (PCN): 2012950240

ISBN: 978-0-7356-7050-1)Introducing Windows 8

Morrison, S. (febrero de 1929). <http://investigacioncientifica.org>. Obtenido de <http://investigacioncientifica.org>

Rendon, D. (27 de Septiembre de 2012). Creando una app Windows 8 con HTML5 y JavaScript. Obtenido de <http://daverndn.com/2012/09/27/creando-una-app-windows8-con-html5-y-javascript/>

Paul Thurrott,Rafael Rivera (12 de enero del 2012) windows 8 Secrets idioma Ingles <http://books.google.es/books>

Bruce Johnson(23 de Octubre de 2012) Professional Visual Studio 2012 Obtenido www.pcworld.com.mx/Articulos/19627.htm

Kraig Brockschmidt(ISBN: 978-0-7356-7261-1) Programming Windows 8 Appswith HTML, CSS, and JavaScript

Rendon, D. (27 de Septiembre de 2012). Creando una app Windows 8 con HTML5 y JavaScript. Obtenido de <http://daverndn.com/2012/09/27/creando-una-app-windows8-con-html5-y-javascript/>

SANCHEZ, M. (1 de Agosto de 2012). Desarrollo de aplicaciones con windows 8 Modern UI. Obtenido de slidershare: <http://www.slideshare.net/rodolfoeduardofinochietti/desarrollo-de-aplicaciones-metro-en-windows-8>

Zapata, G. (2012). Gustavo.Zapata@stdntpartners.onmicrosoft.com. Obtenido de <http://tavo8.wordpress.com/>: <http://tavo8.wordpress.com/>

MSDN, (18 de julio del 2012). Introducción al desarrollo de aplicaciones Windows 8. Recuperado el 16 de octubre 2012, <http://blogs.msdn.com/b/esmsdn/archive/2012/07/18/introducci-243-n-al-desarrollo-de-aplicaciones-metro-para-windows-8.aspx>

Asmal Tenesaca, H. S. (2013). ANÁLISIS DE ASPECTOS TÉCNICOS, HERRAMIENTAS Y PLATAFORMA PARA EL DESARROLLO DE APLICACIONES WINDOWS 8 INTERFAZ METRO (Doctoral dissertation).

Vargas Ávila, C. (2014). Aplicación móvil para Windows 8 utilizando Metro UI.

Tanenbaum, A. S., & Seone, J. (1985). Organización de computadoras. Prentice Hall.

López, Z. P. (2014). Introducción a la informática. Edición 2013. Anaya Multimedia.

Gauchat, J. D. (2012). El gran libro de HTML5, CSS3 y JavaScript. Marcombo.

Iglesias, F. (2011). CSS3 Diseña con estilos. Macworld España: la revista para los usuarios del Macintosh., (211), 41-48.

Van Lancker, L. (2013). HTML5 y CSS3: Domine los estándares de las aplicaciones Web [2ª edición]. Ediciones ENI.

Morris, T., & Felke-Morris, T. (2011). Basics of web design: HTML5 and CSS3. Addison-Wesley Publishing Company.

Casado, F., Navarrete, T., Sayago, S., & Blat, J. (2005). Una herramienta para la creación de interfaces multiplataforma con UIML. In Proceedings on the VI Congreso de Interacción Persona Ordenador (INTERACCIÓN'2005). Editores: Ángel R. Puerta, Miguel Gea. Granada, Spain. ISBN (pp. 84-9732).

Van der Vlist, E., Ayers, D., Bruchez, E., Fawcett, J., & Vernet, A. (2007). Programación Web 2.0.

Lubbers, P., Albers, B., Salim, F., & Pye, T. (2011). Pro HTML5 programming (pp. 107-133). New York, NY, USA.: Apress.

Johnson, B. (2012). Professional Visual Studio 2012. John Wiley & Sons.

Rossberg, J., & Olausson, M. (2012). Pro Application Lifecycle Management with Visual Studio 2012. Apress.

Lizasoain, L., & Joaristi, L. (2001). SPSS para Windows: Versión 8 en castellano. Paraninfo.