2. Marco teórico

En este primer capítulo se presenta la información de las herramientas que utilicé para el desarrollo de los proyectos que mencioné en el capítulo anterior, en los que participé en mi labor en el Departamento de Investigación y Desarrollo de la División de Educación continua y a Distancia de la Facultad de Ingeniería.

La mayor parte de la información que conforma este capítulo trata sobre XNA, entorno de trabajo en el que desarrollé el proyecto principal, que es la Herramienta de Visualización de Geometría Analítica, con ello busco explicar de forma sencilla su funcionamiento general. También incluí información sobre las herramientas y librerías con las que desarrollé los proyectos de Videoconferencia, el Paseo Virtual y la práctica 1 del nuevo Laboratorio Virtual de Geometría Analítica.

2.1.XNA

XNA es el *framework* utilizado para desarrollar la *Herramienta de Visualización de Geometría Analítica*.

2.1.1. Definición de XNA

XNA es un *framework* o entorno de desarrollo creado por Microsoft que incluye una serie de librerías .NET de Microsoft que se utilizan para crear juegos. Otro elemento que forma parte de la solución es XNA Game Studio, el cual es una *API* que permite desarrollar juegos y aplicaciones utilizando el lenguaje *C#* y el paradigma de programación orientado a objetos para las siguientes plataformas[2]:

- Windows
- Xbox 360
- Zune
- Windows Phone

La razón por la que elegí XNA para el desarrollo de la herramienta de geometría analítica es la facilidad para desarrollar aplicaciones multiplataforma mediante un conjunto de API's o interfaces de programación de aplicaciones, normalizadas entre las plataformas mencionadas.

2.1.2. Definición de Framework y API

Un *framework* es un conjunto de códigos o librerías que tienen una funcionalidad común y amalgamada para una diversa cantidad de aplicaciones. Un *framework* ahorra el tiempo al no tener que escribir la lógica usada comúnmente en diferentes aplicaciones que utilicen funciones similares. La diferencia principal entre un *framework* y una librería es que ésta únicamente aporta una función específica y mientras que el *framework* ofrece varias.[3]

Una *Aplication Program Interface* (API) o Interfaz de Programación de Aplicaciones es un conjunto de reglas y especificaciones que un programa puede hacer uso de los servicios y recursos provistos por un programa particular que implementa esa API. Básicamente es una interfaz entre distintos programas de manera que facilita su interacción. Una API puede contener uno o varios *framework*.[4]

2.1.3. Definición de .NET

El *framework*.NET es la plataforma de Microsoft para programar aplicaciones que consiste en lo siguiente[5]:

- Rutinas de lenguaje común que proveen una capa de abstracción sobre el sistema operativo.
- Librerías de clases base o códigos pre construidos para tareas de programación de bajo nivel.
- *Framework* y tecnologías de desarrollo reusables y personalizables para tareas de programación mayores.

.NET se emplea en diversos *framework* y es compatible con la mayoría de los lenguajes de Microsoft como Visual Basic, C, C++, C#, etcétera.

Con ello tenía un *framework* suficientemente robusto para elaborar una aplicación educativa de calidad.

2.1.4. Estructura de XNA

La estructura general de XNA está dividida en las siguientes capas[6]:

- Plataforma

El nivel más bajo y consiste en interfaces de bajo nivel nativas y administradas sobre las cuales se construye el framework de XNA. Algunas interfaces son Direct3D 9(encargado de los gráficos en 3D), XACT (administra el audio y los efectos de sonido), XInput (maneja la entrada del usuario), XContent (administra la carga del contenido restante).

- Núcleo

La primera capa de XNA y provee la funcionalidad básica que requieren las demás capas. Las áreas de funcionalidad son Gráficos, Audio, Entrada, Matemáticas y Almacenaje.

- Extensión

En esta capa nos encontramos con el Modelo de Aplicación y la Tubería de Contenido (del inglés *Content Pipeline)*.

- Juegos

Esta es la capa superior y consiste del código y contenido de la aplicación o juego. Aquí se manejan las plantillas y los componentes de juego.

2.1.5. Características de XNA

Las características principales de XNA son las siguientes[6]:

- Modelo de aplicación

El propósito de dicho modelo es hacer que el programador se olvide de la plataforma en la que ejecutará su juego para así concentrarse en la concreta tarea de programar la aplicación. Así mismo el framework provee al desarrollador de un Componente de Gráficos que facilita la creación y manejo del Dispositivo Gráfico utilizado para la *renderización* tanto para la computadora, el Xbox y los dispositivos móviles. Por su parte el modelo de componentes permite agregar a la aplicación de una manera sencilla y rápida Componentes de Juego lo cual otorga al programador la capacidad de crear una librería de componentes que puede reutilizar en diversos proyectos.

- Gráficos

Los API's están basados en Direct3D 9, pero a diferencia de programar en su totalidad en Direct3D, uno de los beneficios es el uso de una tubería de renderización programable con *shaders* en lugar de una tubería de funcionalidad fija.

Un componente de XNA de suma importancia es el Efecto Básico o *BasicEffect*, el cual es un efecto de fácil uso que posee propiedades como luces, texturas, transparencia, etc. Utilizar el shader o efecto básico del framework permite a los programadores mostrar diversos elementos en pantalla sin tener que escribir complejos shaders.

- Audio

La interfaz de audio que utiliza XNA está realizada sobre XACT el cual es una API de Microsoft multiplataforma. La ventaja de utilizar XACT radica en que la herramienta permite crear paquetes de efectos de sonido con propiedades como volumen, bucles, canales, etc. Para posteriormente cargarlos y agregarlos al proyecto sin tener que inicializar los buffers o administrar los datos.

- Entrada

Las API's de Entrada están construidas a partir de XInput el cual es un API multiplataforma que funciona como driver del control de Xbox 360. La interfaz de Entrada ofrece un modo inmediato que no requiere inicialización, por lo que no hay que preocuparse por el reconocimiento o liberación del dispositivo, el modo para compartirlo, etc. Así mismo para la entrada específica por plataforma tenemos el teclado para Xbox y computadora, el mouse para computadora y para los dispositivos móviles existe la entrada táctil y los acelerómetros.

- Matemáticas

La interfaz de Matemáticas provee datos de programación de juegos como vectores en dos, tres y hasta cuatro dimensiones, matrices y planos. Dentro de la librería de las matrices podemos encontrar útiles métodos de creación de vistas, transformaciones, proyección, etc.

Ahora bien las características que ofrece XNA me permitieron realizar las geometrías sin más complicaciones que los cálculos matemáticos más óptimos para generarlas. Y realmente lo que fue más tardado de elaborar fue: la aplicación, el control táctil para la versión de Windows Phone 7 y el diseño.

Otro elemento importante que debo mencionar después de haber trabajado con XNA es el API de gráficos que utiliza el *framework*, el cuál es el más utilizado para crear gráficos en computadoras con sistemas Windows.

2.1.6. Definición de Direct3D

Direct3D de Microsoft es un API de gráficos de bajo nivel que permite manipular modelos visuales de objetos tridimensionales y hace uso de la aceleración por hardware, como las tarjetas de video.[7]

El término de aceleración por hardware básicamente es incrementar la velocidad de un proceso de software con la ayuda del hardware. Esto se logra al incluir parte del código de software en el hardware.

XNA utiliza Direct3D para crear gráficos mediante pequeños códigos o programas denominados *shader*.

2.1.7. Definición de Shader

Un *shader* es cualquier código escrito en un lenguaje de sombreado que se compila independientemente. La traducción literal de *shader* es "sombreador", la cual no se usa comúnmente, en cambio se usa la palabra efecto para describirlos debido al hecho que se utilizan principalmente para crear efectos especiales.[8]

Los lenguajes de sombreado son lenguajes de programación de alto nivel que se encargan de realizar cálculos de iluminación como indica su nombre. La principal ventaja de utilizarlos es que permiten la independencia de hardware, es decir liberan al procesador ya que se procesan en las tarjetas de video.

2.1.8. Definición de HLSL

El lenguaje de sombreado que utiliza XNA es *HLSL* o *High Level Shader Language* (Lenguaje de Sombreado de Alto Nivel) de Microsoft que utiliza *DirectX* también de Microsoft que es una implementación de Direct3D.[9]

Los tipos de *shader* que existen se ejecutan en 3 distintos procesadores que se encuentran en las tarjetas de video, estos shaders son:

- *Shader* de vértices. Realiza transformaciones en coordenadas, color, textura, vector normal, vector binormal, vector tangente, entre otros, de un vértice.
- *Shader* de geometrías. Encargado de generar dinámicamente o modificar primitivas geométricas o figuras geométricas básicas como puntos, líneas y triángulos.
- *Shader* de pixeles y fragmentos. El *pixel* es la unidad homogénea en color más pequeña de una imagen digital o *raster*. Es el que realiza transformaciones relacionadas con la profundidad, trabajar con las unidades de textura denominadas *texel*, calcular efectos de iluminación por pixel de alta precisión. Todos estos cálculos determinan el color que deberá ser aplicado a cada *pixel*.

Para concluir, de manera sencilla un shader es un código con un conjunto de instrucciones que serán ejecutadas por procesadores que contienen las tarjetas de video para obtener efectos en tiempo real y liberar al procesador de dichas operaciones y así se pueda encargar de otros procesos.

Debo aclarar que realicé dos versiones de la aplicación, la principal razón de esto fue que la versión casi terminada no era compatible con los dispositivos móviles, debido a la mayoría de los cambios aplicados a la sintaxis del lenguaje C# en la nueva versión de XNA, sin embargo esa no fue la única razón del pequeño contratiempo. La otra razón fue el hecho que se utilizaron *shaders* ajenos al efecto básico incluido en ambas versiones de XNA con las que se trabajó y debido a las restricciones de *hardware* de los dispositivos celulares y multimedia, se tuvo que trabajar con el efecto predeterminado.

Para continuar con la parte teórica de los lenguajes de sombreado es necesario definir la palabra *renderizar,* que viene del inglés *render*.

2.1.9. Definición de renderización

Renderizar o *rasterizar* es el proceso de convertir un modelo tridimensional generado por computadora en una imagen en el espacio de dos dimensiones. El resultado del proceso de renderizado se denomina *raster* o imagen digital.[1011]

Existen 2 tipos de renderizado:

- Pre-renderizado. Es el proceso computacionalmente más intensivo, se utiliza generalmente para películas y tiene como característica ser de muy alta calidad. Los cálculos son realizados por el procesador. Para la animación de escenas en 3 dimensiones, los movimientos son fijados antes de la renderización y se calculan durante ella.
- Renderizado en tiempo real. Utilizado en juegos de video. Se procesa en las tarjetas de video o aceleración 3D. A diferencia del otro método, los cambios se calculan en tiempo real.

2.1.10. XNA Game Studio

Si bien para elaborar la aplicación de geometría programé en XNA, un componente importante de la solución es XNA *Game Studio* que se define como:

"Entorno de programación que permite usar Visual Studio para crear juegos para Windows Phone, la consola Xbox 360 y equipos basados en Windows. XNA Game Studio incluye XNA Framework, que es un conjunto de bibliotecas administradas diseñadas para el desarrollo de juegos basado en Microsoft .NET Framework 2.0" – Definición de la página oficial de XNA Game Studio 4.0 en MSDN en español.[12]

La versión de XNA Game Studio 4.0 es la más reciente pero la versión 3.1 aún se encuentra vigente.

Las aplicaciones que son ejecutadas en los celulares con Windows Phone 7 son las creadas en la versión 4.0. Ahora bien estas aplicaciones también son ejecutadas en Windows y en la Xbox 360 así como las aplicaciones de XNA Game Studio 3.1. Los dispositivos multimedia Zune únicamente ejecutan programas creados en la versión 3.1

La totalidad de versiones que han existido son[13]:

- XNA Game Studio Express
- XNA Game Studio 2.0
- XNA Game Studio 3.0
- XNA Game Studio 3.1
- XNA Game Studio 4.0
- XNA Game Studio 4.0 Refresh

2.1.10.1. Requisitos de Sistema

Para comenzar con la programación de la aplicación se tuvieron que cumplir los siguientes requisitos:

- Windows XP (con la excepción que no se pueden desarrollar aplicaciones para Windows Phone 7).
- Windows Vista (el cual debe ser cuando menos Service Pack 1).
- Windows 7

Es importante mencionar que en Windows Vista y 7 se requieren tener permisos de administrador para poder instalar el entorno.

Los requisitos de hardware para ejecutar y programar juegos de XNA Framework se requiere mínimo una tarjeta de video que soporte *Shader Model 1.1* y *DirectX 9.0c* pero se recomienda que la tarjeta soporte *DirectX 10* para poder programar aplicaciones para Windows Phone.[14]

El software que se requiere para poder programar en XNA Game Studio 4.0 es cualquiera de las versiones de *Microsoft Visual Studio 2010*:

- Visual Studio 2010 Express for Windows Phone.
- Visual Studio 2010 Ultimate
- Visual Studio 2010 Premium
- Visual Studio 2010 Professional
- Visual C# 2010 Express

Para programar aplicaciones con la versión anterior, es decir XNA Game Studio 3.1 es cualquiera de las versiones de *Microsoft Visual Studio 2008*:

- Visual C# 2008 Express Edition
- Visual Studio 2008 Standar Edition
- Visual Studio 2008 Professional Edition

Los requisitos mínimos de hardware son:

Computadora con sistema operativo Windows con tarjeta aceleradora de gráficos que soporte como mínimo Shader Model 1.1 y DirectX *9*.

Se recomienda una tarjeta aceleradora de gráficos que soporte el modelo de shader 2.0 (*Shader Model 2.0*).

Para ejecutar y depurar aplicaciones de Windows Phone 7 con el emulador, se requiere una tarjeta aceleradora de video que soporte como mínimo *DirectX 11*.

2.1.10.2. Herramientas empleadas

Para elaborar la herramienta de superficies geométricas hice uso de los siguientes elementos:

- Sistema operativo Windows 7 de 64 bits con permisos de administrador.
- Visual Studio Professional 2008 y 2010, para la versión de Xbox360 / PC y Windows Phone 7 respectivamente.
- XNA Game Studio 3.1 para la versión de Xbox360 / PC y 4.0 para programar para Windows Phone 7.
- Photoshop CS5 para editar las imágenes.
- Crazy Bump para obtener los mapas de normales y mapas de especularidad utilizadas por los *shaders* en la versión de XNA Game Studio 3.1.
- Softimage que es un software de modelado 3D.
- Software multimedia Zune para transferir la aplicación al celular con Windows Phone 7. Una gran ventaja es que estando conectado permite la ejecución en modo de depuración de errores.
- Software XNA Game Studio Connect para la consola Xbox 360 para poder transferir la aplicación a la consola, instalarla y ejecutarla en modo de depuración de errores.

Otros elementos que considero de gran importancia para la elaboración de ambas aplicaciones son:

- Consola de videojuegos Xbox 360.
- Control alámbrico de Xbox 360 / Windows para realizar pruebas tanto en la consola como en la computadora.
- Celular con sistema operativo Windows Phone 7 para realizar pruebas.
- Cuenta de estudiante en la página de internet de MSDN para obtener acceso a la licencia de Visual Studio Professional y a la licencia de desarrollador de XNA Game Studio.
- Licencia de desarrollador de juegos con XNA para Xbox 360, PC, Windows Phone 7 y Zune.

2.1.10.3. Perfiles de XNA

Además de los cambios obvios de sintaxis que se realizaron entre las versiones de XNA, un aspecto muy importante que se añadió al *framework*, es la posibilidad de modificar la configuración del proyecto desde las propiedades para seleccionar uno de dos perfiles de juego[15]:

- *Reach*. Para dispositivos móviles. Alcance.
- *HiDef*. Para Xbox 360 y computadora. Alta Definición.

A continuación listo las principales diferencias entre estos dos perfiles de aplicación.

Tabla 2-1. Tabla comparativa de perfiles de XNA 4.0

	Reach	HiDef
Plataformas Soportadas	Windows Phone 7, Xbox 360, cualquier computadora con Windows y una tarjeta aceleradora de video que soporte DirectX 9 con *shader model* 2.0.	Xbox 360, cualquier computadora con Windows y una tarjeta aceleradora de video que soporte DirectX 10.
Modelo de Shader (*Shader Model*)	2.0. Windows Phone no soporta *shaders* personalizados.	3.0 en adelante. (Xbox 360 maneja *shaders* con tipo vfetch pero no son soportados por Windows.
Tamaño máximo de textura	2048 pixeles	4096 pixeles
Tamaño máximo de mapa cúbico (*cubemap*)	512 pixeles	4096 pixeles
Tamaño máximo de textura volumétrica	No hay.	256 pixeles
Texturas no cuadradas	Si, sin embargo no se puede utilizar *wrap*, *mipmaps*, o compresión DXT.	Sí
Cubemaps no cuadrados	No	Sí
Texturas volumétricas no cuadradas	No	Sí
Primitivas máximas por llamada de dibujado	65535	1048575
Formatos de buffer de índices	16 bits	16 y 32 bits
Formatos de elementos de vértices	Color, Byte4, Single, Vector2, Vector3, Vector4, Short2, Short4, NormalizedShort2,	Todos los formatos de Reach más *HalfVector2* y *HalfVector4.*

	NormalizedShort4	
Formatos de textura	Color, Bgr565, Brga4444, NormalyzedByte2, NormalizedByte4, Dxt1, Dxt3, Dxt5	Todos los formatos de Reach más *Alpha8, Rg32,* Rgba64, Rgba1010102, Single, Vector2, Vector4, HalfSingle, HalfVector2, HalfVector4. Las texturas de punto flotante no soportan filtrado.
Formatos de vértices texturizados	Texturizado de vértices no está soportado	Single, Vector2, Vector4, HalfSingle, HalfVector2, HalfVector4
Formatos de objetivos de rasterizado	Variable	Variable
Objetivos de rasterizado múltiples	No	Hasta 4. Deben de tener la misma profundidad en bits. Soporta mezclado de transparencia y máscaras de escritura independientes por objetivo.
Consulta de oclusión	No	Sí
Mezcla de transparencia separada	No	Sí
Blend.SourceAlphaSaturation	Solo para SourceBlend y no DestinationBlend	Sí
Flujo de vértices máximo (*vertex streams*)	16	16
Alcance máximo del flujo (*stream stride*)	255	255

HiDef hereda de *Reach* por lo tanto, si se corre una aplicación de este último perfil en una plataforma de alta definición el *framework* seguirá aplicando las reglas de *Reach*.

El perfil de alta definición no utiliza DirectX10 directamente sino que requiere una tarjeta aceleradora gráfica con memoria de video y procesador de gráficos superior a una que soporte DirectX9.

Estos perfiles se configuran en las propiedades del proyecto dentro de Visual Studio y se explicará la manera en la que se realizó la configuración.

2.2.Unreal Development Kit

Unreal Development Kit es la versión gratuita del poderoso motor de juegos Unreal Engine 3, como dice el nombre es la tercera iteración del motor Unreal Engine.[16]

Este es un framework completo de desarrollo profesional de videojuegos y contiene todas las herramientas que se puedan necesitar para crear juegos, simulaciones, recorridos virtuales y demás aplicaciones virtuales para la PC y dispositivos móviles como *iPhone* o *iPad*.

El objetivo principal de *UDK* permitir el uso gratuito de *UnrealEngine3* para crear contenido por estudiantes, desarrolladores independientes, universidades, investigadores, etc.

Esta gran herramienta es actualizada mes con mes con el fin agregar nuevos elementos, mejorar los existentes y remover los que por estándares de la industria se dejan de utilizar.

Siendo un motor de juegos cuenta con todo lo necesario para desarrollar un videojuego sin mayor esfuerzo. Entre las características que incluye están:

- *Unreal Editor*. Un ambiente muy completo de edición de ambientes o niveles.
- *Unreal Content Browser*. Un navegador de contenido, que puede ser: texturas, modelos, música, etc.
- *Unreal Matine*. Animación artística de los modelos así como cinemáticos y escenas visualmente impactantes.
- *Unreal Script*, el cual es un lenguaje de programación de alto nivel.
- Física real.
- Iluminación y sombreado avanzado.
- Creación de terreno.
- Conectividad *LAN* y por *IP*.
- *Shaders* en tiempo real.
- Audio tridimensional y soporte de gran cantidad de formatos.
- Efectos de partículas.
- Inteligencia artificial.
- Ambientes destruibles.

2.3.Flex y ActionScript

Flex es un *framework* gratuito de *Adobe* utilizado en la creación de aplicaciones web tanto para sistemas operativos de computadoras, como para dispositivos móviles, por lo tanto tiene la ventaja de ser soportado en varias plataformas, reduciendo el tiempo de elaboración de dichas aplicaciones.[17]

ActionScript es un lenguaje de programación orientado a objetos de *Adobe* para *Flash* también de *Adobe*. Actualmente es utilizado en *Adobe Flash* y *Flex*.[18]

2.3.1. Flash Media Server

Flash Media Server es un servidor de medios de *Adobe*, el cual permite almacenar y ejecutar aplicaciones realizadas en otras plataformas de *Adobe*, estas plataformas pueden ser *Flash* y *Flex* entre otras.[19]

De igual manera este servidor permite la transmisión de audio y video en tiempo real por lo que puede ser utilizado para aplicaciones de videoconferencia o video chat así como chat sencillo entre otras.

2.4.Unity 3

Unity 3 es una herramienta de desarrollo de videojuegos muy completa que no exige demasiados recursos de *hardware*. Si bien es un motor de juegos utilizado para crear videojuegos y aplicaciones interactivas al igual que otros productos, este se caracteriza por ser más sencillo.[20]

Esta herramienta permite desarrollar videojuegos para PC, consolas de videojuegos, dispositivos móviles con sistema operativo *iOS*, *Android*, así como a diferencia de otros maneja un reproductor web que permite ejecutar las aplicaciones directamente en los navegadores de internet.

Las características de Unity son:

- Crear y editar niveles.
- Cambio instantáneo para desarrollo entre distintas plataformas.
- Método de iluminación mediante el mapeado de luces y oclusión de caras.
- *Scripting* con *C#* y *JavaScript*.
- Inteligencia artificial.
- Física real.
- Conectividad de red.
- Creación de terreno.
- Shaders en tiempo real.

La desventaja que presenta utilizar *Unity* es que la versión gratuita tiene desactivadas varias características, incluyendo la más remarcable, el sombreado.