

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**  
**COORDINACIÓN DE FORMACIÓN BÁSICA**  
**COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA**  
**PROGRAMA DE UNIDAD DE APRENDIZAJE**

**I. DATOS DE IDENTIFICACIÓN**

1. Unidad Académica: Facultad de Ciencias
2. Programa (s) de estudio: Lic. en Ciencias Computacionales, Lic. en Matemáticas Aplicadas
3. Vigencia del plan:
4. Nombre de la Unidad de Aprendizaje: Graficación                      5. Clave:
6. HC:   2      HL   3      HT         HPC         HCL         HE   2      CR   7
7. Ciclo Escolar:
8. Etapa de formación a la que pertenece: Disciplinaria
9. Carácter de la Unidad de Aprendizaje:            Obligatoria
10. Requisitos para cursar la Unidad de Aprendizaje:    Ninguna

Formuló: L.C.C. Miguel Riesgo Tirado  
Dra. María Victoria Meza Kubo

Vo. Bo. Dr. Alberto Leopoldo Moran y Solares

**Fecha:** Agosto de 2016

**Cargo:** Subdirector

## II. PROPÓSITO GENERAL DE LA UNIDAD DE APRENDIZAJE

La graficación tiene como propósito fundamental establecer los algoritmos y técnicas que permiten manipular escenas e imágenes en dos y tres dimensiones a través de un equipo de cómputo.

De tal forma, que esta unidad de aprendizaje tiene como propósito, que el docente:

- presente al estudiante las bases que generaron la graficación por computadora
- analice los distintos algoritmos que permiten la manipulación y despliegue de objetos en dos y tres dimensiones
- comparta con el estudiante nuevas técnicas de graficación
- desarrolle junto con el estudiante un sistema que integre los algoritmos estudiados en el curso

Se requiere que el estudiante que tome esta unidad de aprendizaje tenga conocimientos básicos de geometría vectorial y álgebra lineal, así como un gran manejo de estructuras de datos y algoritmos; esto aunado a la disposición del estudiante a investigar y trabajar en equipo. La asignatura se encuentra en la etapa disciplinaria y es de carácter obligatorio para la Lic. en Ciencias Computacionales y optativo para la Lic. en Matemáticas Aplicadas.

## III. COMPETENCIA (S) DE LA UNIDAD DE APRENDIZAJE

Desarrollar algoritmos de la teoría de la graficación mediante su codificación para manipular y visualizar elementos en dos y tres dimensiones con una actitud propositiva y creativa.

## IV. EVIDENCIA (S) DE DESEMPEÑO

Desarrolla un sistema de cómputo que genere una escena tridimensional aplicando los algoritmos de la teoría de la graficación.

## V. DESARROLLO POR UNIDADES

### COMPETENCIA

Identificar los distintos elementos de una imagen mediante el análisis de formato de archivos gráficos para el diseño de uno propio de forma innovadora.

### CONTENIDO

1. Conceptos básicos de graficación.
  - 1.1 Historia de la graficación.
  - 1.2 Sistemas de despliegue.
  - 1.3 Elementos de una imagen.
  - 1.4 Pixeles y colores.
  - 1.5 Formatos de archivo gráfico.

**DURACIÓN 6hrs**

### COMPETENCIA

Manejar los conceptos básicos del álgebra lineal y la geometría vectorial a través de la resolución de ejercicios gráficos que nos permita codificar una biblioteca matemática que sea óptima para un sistema de graficación con una alta capacidad de abstracción.

### CONTENIDO

2. Herramientas matemáticas para la graficación.
  - 2.1 Vectores en 2D y 3D.
  - 2.2 Producto punto y producto cruz.
  - 2.3 Operaciones con matrices de orden 2, 3, y 4.
  - 2.4 Representación de la línea recta.
  - 2.5 Distancia de un punto a una recta.
  - 2.6 Intersección entre 2 rectas.
  - 2.7 Definición de un plano y su intersección con la línea recta.
  - 2.8 Gradiente de una función.

**DURACIÓN 3 hrs**

**COMPETENCIA**

Analizar las funciones básicas de despliegue mediante la codificación de los mismos para programar algoritmos que permitan dibujar figuras geométricas en una computadora de forma creativa.

**CONTENIDO****DURACIÓN 6 hrs**

3. Raster y funciones básicas de graficación.
  - 3.1 Modelos de color.
  - 3.2 La línea recta (algoritmo DDA y de Bresenham).
  - 3.3 Círculos, elipses y polígonos.
  - 3.4 Algoritmo de relleno para polígonos.
  - 3.5 Curva de Bézier (algoritmo de Casteljaou).
  - 3.6 B-spline.

**COMPETENCIA**

Identificar los distintos elementos de un sistema gráfico para generar una escena bidimensional aplicando el *pipeline* de visualización de una manera creativa e innovadora.

**CONTENIDO****DURACIÓN 6 hrs**

4. Transformaciones geométricas y visualización 2D.
  - 4.1 Transformaciones en 2D (Escalamiento, traslación y rotación).
  - 4.2 Coordenadas homogéneas y su representación matricial.
  - 4.3 Composición de transformaciones.
  - 4.4 *Pipeline* de visualización.
  - 4.5 Ventana de recorte.
  - 4.6 Mapeo de coordenadas de mundo a coordenadas de pantalla.
  - 4.7 *Panning* y *Zoom*.
  - 4.8 Transformaciones en 3D (Escalamiento, traslación y rotación).

**COMPETENCIA**

Identificar los distintos elementos de un sistema gráfico para generar una escena tridimensional mediante la codificación de objetos de revolución de forma creativa.

**CONTENIDO****DURACIÓN 6 hrs**

- 5. Visualización 3D.
  - 5.1 Proyecciones.
  - 5.2 Algoritmos de profundidad (algoritmo del pintor, BSP, Z-Buffer)
  - 5.3 Superficies cuádricas.
  - 5.4 Objetos de revolución.
  - 5.5 Ray Casting.

**COMPETENCIA**

Generar una escena tridimensional realista utilizando distintas fuentes de luz por medio del trazado de rayos para el diseño de escenas en 3D de una manera creativa.

**CONTENIDO****DURACIÓN 5 hrs**

- 6. Iluminación y sombreado.
  - 6.1 RayTracing.
  - 6.2 Fuentes de luz.
  - 6.3 Intersección de un rayo con objetos geométricos.
  - 6.4 Modelos básicos de iluminación.(ambiental, difusa, atenuación)
  - 6.5 Reflexión y refracción.
  - 6.6 Sombras y texturización.

## VI. ESTRUCTURA DE LAS PRÁCTICAS

No. De Práctica	Competencia(s)	Descripción	Material de Apoyo	Duración
1	Manejar formatos de archivo gráfico para identificar los elementos de una imagen mediante la codificación de una función de forma creativa.	Desarrollar un programa que permita desplegar en pantalla una imagen en un formato de archivo gráfico estándar.	Bibliotecas de funciones graficas de C/C++, SDL2.	3 hrs
2	Manejar formatos de archivo gráfico para identificar los elementos de una imagen mediante la codificación de una función de forma creativa.	Hacer un programa que permita mostrar en pantalla información contenida en un formato gráfico propio.	Bibliotecas de funciones graficas de C/C++, SDL2.	3 hrs
3	Programar una biblioteca matemática para optimizar los cálculos realizados en un sistema de graficación con sentido de abstracción.	Programar las funciones básicas del algebra lineal y cálculo vectorial empleadas en un sistema de cómputo de graficación en lenguaje C/C++.	Bibliotecas de funciones matemáticas de C/C++, numerical recipes library.	3 hrs
4	Aplicar los modelos de color para codificar una función mediante un api gráfico de forma innovadora.	Codificar los algoritmos para la transformación entre los distintos modelos de color.	C/C++, SDL2	3 hrs

5	Codificar los algoritmos de líneas, círculos, elipses y polígonos para desplegar figuras	Programar los algoritmos para el trazado de primitivas gráficas en la computadora.	Bibliotecas graficas de C/C++, SDL2, OpenGL	3 hrs
---	--	--	---	-------

	geométricas básicas de forma creativa.			
6	Analizar los algoritmos de trazado de curvas en dos dimensiones para desplegar curvas en pantalla mediante la codificación de los mismos de una manera creativa.	Programar un algoritmo que permita trazar curvas en 2 dimensiones (Bézier, Hermit o B-spline)	Bibliotecas graficas de C/C++, SDL2, OpenGL	3 hrs
7	Diseñar un editor gráfico para realizar un sistema de despliegue básico mediante la codificación de las primitivas gráficas y relleno de color de forma innovadora y creativa.	Desarrollar un editor gráfico que implemente trazado de figuras geométricas básicas, relleno de color y lo guarde como un archivo gráfico propio.	Bibliotecas graficas de C/C++, SDL2, OpenGL, SDL2_image	3 hrs
8	Analizar las transformaciones graficas en 2D para realizar un programa que permita transformar objetos mediante la codificación del mismo en lenguaje C/C++ y OpenGL de forma creativa.	Hacer un programa que permita escalar, trasladar y rotar imágenes en 2D.	C/C++, SDL2, OpenGL	3 hrs
9	Identificar los distintos elementos que intervienen en el <i>pipeline</i> para diseñar aplicaciones gráficas de forma innovadora mediante la codificación de los algoritmos en lenguaje	Desarrollar una función que tome datos en coordenadas reales y despliegue la escena en coordenadas de pantalla, incluyendo <i>zoom</i> y <i>panning</i> .	C/C++, SDL2, OpenGL, SDL2_image.	3 hrs

	C/C++.			
10	Integrar las distintas transformaciones tridimensionales con un modelo de proyección para visualizar objetos en tres dimensiones de forma creativa.	Desarrollar un programa que visualice una escena tridimensional.	C/C++, SDL2, OpenGL	3 hrs
11	Analizar los métodos de sólidos de revolución para generar objetos tridimensionales a partir de transformaciones geométricas en 3D de forma creativa.	Programar una función que permita desplegar objetos por revolución.	C/C++, SDL2, OpenGL	3 hrs
12	Analizar las distintas fuentes de luz para construir una escena 3D fotorrealista utilizando OpenGL de una forma creativa.	Desarrollar un sistema que involucre reflexión y refracción en el proceso de iluminar un objeto.	C/C++, SDL2, OpenGL	3 hrs
13	Aplicar la texturización de un objeto para texturizar un polígono usando OpenGL y SDL2.	Desarrollar una función que texturice un polígono.	C/C++, SDL2, OpenGL	5 hrs
14	Realizar un proyecto final mediante la integración de los distintos algoritmos de una forma creativa e innovadora para obtener una escena tridimensional realista.	Desarrollar un proyecto de curso que involucre la manipulación de gráficos en 3D.	Bibliotecas graficas de C/C++, SDL2, OpenGL, SDL2_image.	7 hrs

## VII. METODOLOGÍA DE TRABAJO

### **Aprendizaje participativo**

Durante el desarrollo de la unidad de aprendizaje, el docente fomentará la participación activa de los estudiantes en actividades individuales y grupales. Mediante la discusión de las temáticas vistas en las clases teóricas y la asignación de ejercicios y prácticas en las sesiones de taller y laboratorio

### **Prácticas de laboratorio**

En las sesiones de laboratorio, se centrará en el análisis y codificación de los distintos algoritmos, de tal manera que permita al estudiante rediseñarlos o redescubrirlos; el estudiante debe de tener una actitud propositiva a investigar y actualizarse con respecto al estado del arte. Las prácticas se podrán realizar en distintas plataformas de desarrollo y en equipos de trabajo con sus compañeros, por lo que se requiere que el estudiante tenga una disposición a trabajar en equipo.

En las prácticas de computo, el alumno debe demostrar que:

- Analiza la función de los algoritmos.
- Comprende cuando emplear los algoritmos y que estructuras de datos utilizar.
- Programa los algoritmos en un lenguaje de cómputo óptimo para el desempeño gráfico

### **Investigación Bibliográfica**

Se sugiere solicitar investigación en diferentes fuentes bibliográficas sobre temas de actualidad o temáticas que serán discutidos posteriormente en clase. El propósito de estos trabajos es fomentar el autoaprendizaje y que el estudiante aprenda a realizar investigación en medios electrónicos (Internet), libros, y revistas sobre temas del área. Las fuentes serán tanto en el idioma inglés como español. Los reportes deberán contener las referencias que se utilizaron para la realización del trabajo y debe contar imprescindiblemente una conclusión personal acerca de la investigación. El maestro debe enfatizar a los estudiantes que los reportes escritos sean claros y bien redactados, recalcándoles también las faltas de ortografía.

### **Ejercicios y exámenes de conocimientos**

El maestro deberá aplicar al menos 2 exámenes de conocimientos durante el periodo, que permitan identificar la obtención de competencias de los estudiantes. Los exámenes podrán ser de varios tipos, tales como: de preguntas abiertas, opción múltiple y solicitud de programas. Así como, la asignación de ejercicios para ser realizados en la sesión de taller o extra clase, de tal manera que refuercen los conocimientos aprendidos durante la clase teórica. Se solicitará la entrega oportuna y formal de tareas y trabajos de investigación.

## VIII. CRITERIOS DE EVALUACIÓN

### Criterios de acreditación

- Para la acreditación del curso se atenderá al Estatuto Escolar Vigente, artículos 70-71, por lo que el estudiante deberá contar un mínimo de 80% de asistencias en el periodo. Tener un mínimo aprobatorio de 60 en su calificación final y para tener derecho a examen extraordinario es necesario haber asistido en un 40% de las clases del curso.
- Para tener derecho a ser evaluado el alumno debe cumplir con las prácticas y tareas.

### Criterios y medios de evaluación

Medios	Criterios de evaluación	Valor
Prácticas	<p>Asistir por lo menos al 90% de las prácticas. Las prácticas deberán estar estructuradas de la siguiente manera:</p> <ul style="list-style-type: none"> <li>• Realizar un reporte escrito de los resultados de la práctica que incluya:                             <ul style="list-style-type: none"> <li>○ Número de práctica</li> <li>○ Título de la práctica</li> <li>○ Resumen de los resultados obtenidos</li> <li>○ Conclusión</li> </ul> </li> <li>• El código fuente deberá ser enviado vía correo electrónico a la dirección que se proporcione durante el encuadre.</li> </ul>	50%
Tareas	<p>Deberá estar conforme a los propósitos establecidos al momento de su asignación. Las tareas deben cumplir el siguiente esquema:</p> <ul style="list-style-type: none"> <li>• Portada principal, que debe incluir:                             <ul style="list-style-type: none"> <li>○ Título de la tarea.</li> <li>○ Número de tarea.</li> <li>○ Nombre y matrícula el estudiante</li> <li>○ Fecha de entrega</li> </ul> </li> <li>• El contenido deberá estar estructurado de la siguiente manera:                             <ul style="list-style-type: none"> <li>○ Índice</li> <li>○ Antecedentes</li> <li>○ Marco teórico</li> <li>○ Conclusiones</li> <li>○ Bibliografía</li> </ul> </li> </ul>	20%
Exámenes	Aplicar dos exámenes parciales y obtener un promedio de 60.	30%

## IX. BIBLIOGRAFÍA

Básica	Complementaria
<p>Dunn, F., &amp; Parberry, I., (2011). 3D Math Primer for Graphics and Game Development, 2nd Edition.</p> <p>Guha, S. (2014). Computer Graphics Through OpenGL: From Theory to Experiments, Second Edition.</p> <p>Hughes, J. F., Van Dam, A., Foley, J. D., &amp; Feiner, S. K. (2013), Computer Graphics: Principles and Practice, 3rd Edition.</p>	<p>Gortler, S. J. (2012). Foundations of 3D computer graphics. MIT Press.</p> <p>June, F. (2011). An Introduction to 3D Computer Graphics, Stereoscopic Image, and Animation in OpenGL and C/C++.</p> <p>Shreiner, D., Sellers, G., Kessenich, J. M., &amp; Licea-Kane, B. (2013). OpenGL programming guide: The Official guide to learning OpenGL, version 4.3. Addison-Wesley.</p> <p><a href="http://lodev.org/cgtutor/">http://lodev.org/cgtutor/</a></p> <p><a href="http://www.scratchapixel.com/">http://www.scratchapixel.com/</a></p>

## X. PERFIL DOCENTE

El docente de esta asignatura deberá ser un profesionalista con formación en el área de computación o áreas afines; con experiencia en docencia y los conocimientos en teoría de la graficación y el uso de herramientas vigentes de visualización por computadora.