

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

I. DATOS DE IDENTIFICACIÓN

1. **Unidad Académica:** Facultad de Ingeniería, Arquitectura y Diseño, Ensenada; Facultad de Ciencias Químicas e Ingeniería, Tijuana y Facultad de Ingeniería, Mexicali.
2. **Programa Educativo:** Ingeniero en Electrónica
3. **Plan de Estudios:** 2020-1
4. **Nombre de la Unidad de Aprendizaje:** Programación Visual
5. **Clave:** 36180
6. **HC:** 01 **HL:** 04 **HT:** 00 **HPC:** 00 **HCL:** 00 **HE:** 01 **CR:** 06
7. **Etapas de Formación a la que Pertenece:** Disciplinaria
8. **Carácter de la Unidad de Aprendizaje:** Optativa
9. **Requisitos para Cursar la Unidad de Aprendizaje:** Ninguno



Equipo de diseño de PUA

Abraham Flores Vergara
Susana Borrego Domínguez
Gabriela Camarena Clemente

Firma

**Vo.Bo. de Subdirectores de
Unidades Académicas**

Humberto Cervantes de Ávila
Rocío Alejandra Chávez Santoscoy
Alejandro Mungaray Moctezuma

Firma

Fecha: 19 de febrero de 2019

II. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

La programación visual es útil para el Ingeniero Electrónico debido a que permite instrumentar los sistemas electrónicos de automatización, así como elaborar interfaces gráficas de usuario en la industria. Los estudiantes adquieren las herramientas para programar, por lo tanto, la asignatura favorece el desarrollo del pensamiento lógico y creativo para la resolución de problemas relacionados con la programación.

La unidad de aprendizaje se encuentra ubicada en la etapa disciplinaria con carácter optativo y aporta al área de conocimiento de Ingeniería Aplicada.

III. COMPETENCIA DE LA UNIDAD DE APRENDIZAJE

Diseñar propuestas de solución con interfaz gráfica de usuario, implementando sistemas integrales de desarrollo de aplicaciones y considerando los lenguajes de programación más utilizados en el desarrollo de aplicaciones de los sistemas embebidos, PC y dispositivos móviles con base en el estado del arte de la ingeniería, para simplificar el diseño de sistemas de pruebas, medidas y control, de manera ordenada dentro de un contexto de valores en armonía con la sociedad.

IV. EVIDENCIA(S) DE DESEMPEÑO

Elabora un proyecto que integren las propuestas de solución, utilizando programación con interfaz gráfica de usuario, para aplicaciones que permitan simplificar sistemas de pruebas, medidas y control, así como su presentación ante grupo.

V. DESARROLLO POR UNIDADES

UNIDAD I. Programación visual

Competencia:

Distinguir los modelos de programación, a partir del análisis de los conceptos básicos de la programación visual, para establecer la interrelación con los nuevos paradigmas de programación, con actitud crítica, interés y compromiso.

Contenido:**Duración:** 4 horas

- 1.1. Funciones
 - 1.1.1. Estructura de una función
 - 1.1.2. Prototipos de las funciones
 - 1.1.3. Tipos de parámetros de una función
- 1.2. Interfaz gráfica de Usuario (GUI)
 - 1.2.1. Fundamentos de la Interfaz Gráfica
 - 1.2.2. Principios del diseño de interfaz gráfica de usuario
- 1.3. Lenguajes de programación para el desarrollo de interfaces gráficas de usuario
 - 1.3.1. Componentes de GUI
 - 1.3.2. Comparación de editores GUI en diversos ambientes de programación visual
- 1.4. Modelos de Programación
 - 1.4.1. Estructurada
 - 1.4.2. Orientada a Eventos
 - 1.4.3. Orientada a Objetos
- 1.5. Programación Orientada a Objetos
 - 1.5.1. Objetos
 - 1.5.2. Clase
 - 1.5.3. Propiedades
 - 1.5.4. Métodos
 - 1.5.5. Eventos
 - 1.5.6. Constructores y destructores
 - 1.5.7. Polimorfismo
- 1.6. Conceptos Fundamentales de Estructura de Datos
 - 1.6.1. Pilas
 - 1.6.2. Colas
 - 1.6.3. Listas encadenadas

1.6.4. Árboles Binarios

UNIDAD II. Programación orientada a eventos

Competencia:

Construir las bases de la programación orientada a objetos, a partir de la exploración de los componentes del software, para operar la interfaz gráfica, con seguridad y eficiencia.

Contenido:

- 2.1. Controles básicos
- 2.2. Estructuras de control
- 2.3. Listas y listas desplegadas
- 2.4. Casillas de verificación
- 2.5. Procedimientos Intrínsecos y Extrínsecos
- 2.6. Control Imagen

Duración: 4 horas

UNIDAD III. Plataforma y entorno de desarrollo para diseñar sistemas

Competencia:

Diseñar programas gráficos, a partir de la identificación y manipulación de datos, archivos y gráficas, por medio de la aplicación de software, para crear plataformas visuales amigables para el usuario, con actitud analítica y responsable.

Contenido:

Duración: 5 horas

- 3.1. Diseñar una interfaz de usuario
- 3.2. Tipos de datos
- 3.3. Bucles
 - 3.3.1. While
 - 3.3.2. For
- 3.4. Temporización
- 3.5. Estructura Case
- 3.6. Bases de programación modular (subrutinas)
- 3.7. Gráficos
- 3.8. Arreglos
- 3.9. Estructura de datos en aplicaciones “Cluster y cluster error”
- 3.10. Archivos
 - 3.10.1. Definición de Archivo
 - 3.10.2. Lectura/Escritura de Archivos
 - 3.10.3. Funciones para el manejo de Archivos
- 3.11. Manejo de fórmulas
- 3.12. Concatenar Cadenas
- 3.13. Registro acumulador
- 3.14. Simulación de señales y de datos
- 3.15. Adquisición de datos

UNIDAD IV. Usos de recursos de hardware

Competencia:

Diseñar un proyecto, utilizando la comunicación de puertos, para establecer la interconectividad entre el software y el hardware, con actitud, crítica, colaborativa y eficiente.

Contenido:

Duración: 3 horas

- 4.1. Definición de Puertos
- 4.2. Librerías para manipular puertos
 - 4.2.1. Puerto Serie
 - 4.2.2. Puerto Paralelo
 - 4.2.3. Puerto USB

VI. ESTRUCTURA DE LAS PRÁCTICAS DE LABORATORIO

No. de Práctica	Competencia	Descripción	Material de Apoyo	Duración
UNIDAD I				
1	Resolver ejercicios, a partir de la aplicación de los conceptos básicos de programación estructurada, para la solución de problemas en el área de ingeniería, con actitud crítica y analítica.	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Estructura secuencial: un programa para obtener el promedio de cuatro calificaciones. 2. Estructura selectiva: comparar tres números y que despliegue el mayor de ellos. 3. Estructura repetitiva: Desplegar los números pares hasta donde el usuario lo desee. 	Computadora, software de programación, pintarrón y proyector.	2 horas
2		<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Por medio de una función calcula X1 y X2 utilizando la fórmula general (a, b y c son parámetros de la función). Demuestra los resultados en la función formula. Valida raíz negativa y división entre cero. 2. Obtiene área de un círculo 3. Obtiene área de un trapecio. 4. Obtiene el área de cualquier polígono regular. 5. Utiliza un parámetro por referencia en una función, calcular el cuadrado de un número. Muestra el resultado en 	Computadora, software de programación, pintarrón y proyector.	4 horas

		la función main(). NO return.		
3	Interpretar la programación orientada a objetos, a partir del uso del software especializado, para esbozar el diseño de una programación gráfica, con orden, creatividad y disciplina.	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Realiza un ejemplo que detalle en los comentarios, la programación orientada a objetos. 2. Define una clase medición que tenga como atributo privado volt tipo entera así como los métodos públicos que se enlistan a continuación: IniciarV, SubirV, BajarV, VisualizarV. Crea un objeto voltaje que invoque las funciones SubirV y BajarV en varias ocasiones en la función principal. Muestra los resultados cada vez que ocurra un cambio, inicializar voltaje en 5. 3. Realiza un programa que permita resolver las ecuaciones de segundo grado. Para esto crea la clase 'Ecu2G' que tenga variables y métodos adecuados que permitan resolver la ecuación. 4. Realiza un programa que calcule el promedio de 2 alumnos (alumno1 y alumno2 serán objetos), cada alumno tendrá 2 calificaciones. La clase tendrá como atributos privados: matrícula, calif1, calif2, 	Computadora, software de programación, pintarrón y proyector.	4 horas

		<p>promedio. Tres métodos públicos: fijarDatos, promedio, mostrarDatos. Lee las calificaciones mostrar la matrícula, calificaciones y promedio.</p>		
4	<p>Manipular software de interfaz gráfica, para programar objetos, utilizando constructoras y destructoras, con orden, responsabilidad y creatividad.</p>	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Simulando una estación de Radio, crea un objeto MiEstacion que inicializa con una constructora los atributos frecuencia=119 y volumen=31, utilizando sus métodos Subir/Bajar frecuencia y volumen. Muestra cada cambio, libera memoria. 2. Crea dos objetos Hoy y Cumpleaños, inicializa con la función constructora la fecha de hoy y la fecha de su cumpleaños. Si es su cumpleaños mostrar ¡Feliz Cumpleaños! si no ¡Feliz día! 3. Realiza un programa que el alumno mediante el uso de un vector realice las siguientes opciones: <ol style="list-style-type: none"> 3.1. Alta 3.2. Borrar un elemento 3.3. Sacar un elemento 3.4. Vaciar 3.5. Mostrar 	<p>Computadora, software de programación, pintarrón y proyector.</p>	4 horas

		3.6. Salida 4. Utiliza la función constructora para inicializar el índice del vector y función destructora, basarse en el concepto la pila (LIFO).		
UNIDAD II				
5	Analizar los componentes de un programa orientado a eventos, a partir del uso de estructuras de control, para ubicar y comprender el funcionamiento de sus comandos, con curiosidad, cautela y orden.	El docente indica las instrucciones para que el alumno realice los siguientes ejercicios: 1. Realiza un programa que por medio de botones muestre sus datos en una caja de texto. 2. Realiza una calculadora que ejecute las operaciones de suma, resta, multiplicación y división de 2 números enteros, utilizando botones para cada operación y muestra su resultado en una caja de texto.	Computadora, software de programación, pintarrón y proyector.	4 horas
6	Manipular el software especializado, para desarrollar gráficos, a partir del uso de estructuras de control de un programa orientado a eventos, con actitud atenta, metódica y creativa.	El docente indica las instrucciones para que el alumno realice los siguientes ejercicios: 1. Utiliza la instrucción Select Case, realiza una calculadora aritmética. 2. Captura un número e imprime sí es impar o par. 3. Utiliza el ciclo Do.... Loop Until, captura un número y muestra sí es un número primo o no. 4. Utiliza el ciclo Do While. Loop calcula el Máximo Común Divisor. 5. Utiliza el ciclo For dibujar un pino de “*”	Computadora, software de programación, pintarrón y proyector.	4 horas

7	Manipular el software especializado, a partir del uso de listas, para el desarrollo de programas de aplicación, con actitud metódica y creativa.	1. El alumno elabora un programa que convierta un número decimal a cualquiera de las bases (binario, octal, hexadecimal y decimal) de una lista y una lista desplegable de datos.	Computadora, software de programación, pintarrón y proyector.	4 horas
8	Desarrollar gráficos, para elegir una opción o varias, utilizando el software de programación, con actitud metódica y creativa.	El docente indica las instrucciones para que el alumno realice los siguientes ejercicios: 1. Se elabora un programa que simule un editor de texto utilizando los controles Casilla de Verificación (CheckBox) y Botón de Opción (OptionButton) que tenga las siguientes opciones: 1.1. Casilla de verificación: 1.1.1. Tipo de letra con casilla de verificación: Negrita, Cursiva y cambiar de mayúsculas a minúsculas. Los siguientes grupos con OptionButton: 1.3. Tipo de letra: Arial y Times New Roman. 1.4. Tamaño: 12,16 y 20. 1.5. Color de letra: Negro, Rojo y Azul.	Computadora, software de programación, pintarrón y proyector.	4 horas
9	Desarrollar gráficos, para el uso de funciones, utilizando el software de programación, con actitud metódica y creativa.	El docente indica las instrucciones para que el alumno realice los siguientes ejercicios: 1. Diseña un programa que cree el CURP y RFC y después lo despliegue en pantalla. 2. Realiza un programa que simule un dado. 3. Crea un programa donde se genere un número aleatorio entre	Computadora, software de programación, pintarrón y proyector.	4 horas

		<p>20 y 30 (ambos valores incluidos) que debe almacenarse en una variable llamada 'alumnos'. Crea un número aleatorio entre 5 y 12 (incluir el 5, excluir el 12) que debe almacenarse en una variable llamada 'afectados'. Crea una variable llamada 'temperatura' que almacene un valor aleatorio entre 37 y 40 (ambos incluidos).</p> <p>A continuación en pantalla se despliega el siguiente mensaje: "El total de alumnos en el aula es xxxx, de ellos están afectados yyyy por la enfermedad, y la temperatura que están sufriendo es zzzz grados centígrados". Donde xxxx: valor de la variable 'alumnos', yyyy: valor de la variable 'afectados', zzzz: valor de la variable 'temperatura'.</p>		
UNIDAD III				
10	<p>Implementar técnicas de diseño en programación visual, por medio del manejo del software especializado, para crear una interfaz gráfica que lea y almacene información desde y hacia un archivo en memoria, con actitud proactiva, y sistemática.</p>	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Diseña una interfaz gráfica que permita capturar la información personal de alumnos y almacenarla en un archivo de texto. 2. Lee el archivo de texto para desplegar en pantalla un listado con la información personal de los alumnos capturados. 	<p>Computadora, software de programación, pintarrón y proyector.</p>	4 horas

11	<p>Utilizar los elementos de la programación de interfaces gráficas de usuario, por medio del uso del software, para el diseño y desarrollo de programas de aplicación, con actitud responsable y organizada.</p>	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Se tienen 4 hornos de laboratorio en una empresa. Dos de ellos llegan a una temperatura máxima de 1000°C y los otros dos a 1315°C. Se necesita desarrollar una interfaz gráfica para controlar el encendido y apagado de dichos hornos. Cuando se encienden, en la interfaz se muestra la temperatura en °C, pero el usuario también tiene la opción de elegir mostrar la temperatura en °F o K. 2. Desarrolla una interfaz gráfica que simula el llenado gradual de 3 tanques. El primer tanque tiene una capacidad de 52 litros, el segundo tanque una capacidad de 100 litros, el tercer tanque una capacidad de 150 litros. 	<p>Computadora, software de programación, pintarrón y proyector.</p>	4 horas
12	<p>Utilizar los elementos de la programación de interfaces gráficas de usuario, por medio del uso del software, para el diseño y desarrollo de programas de aplicación, con actitud responsable y organizada.</p>	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <ol style="list-style-type: none"> 1. Se tienen dos números de entrada (A y B) dados por el usuario. Indicar (usar indicador booleano) si A es mayor a B, si A es menor a B o si A es igual a B. 2. Se tienen cinco controladores numéricos de entrada, en los cuales se vacían calificaciones, 	<p>Computadora, software de programación, pintarrón y proyector.</p>	4 horas

		<p>que van de un rango de 0 a 100. Dependiendo del promedio de dichas calificaciones, aparecerá un texto que diga “Excelente”, “Bien” o “Reprobado”, de igual manera, también se desplegará el promedio de las calificaciones.</p> <p>3. Se tiene un sensor programable que mide la temperatura cada 0.5 segundos y muestra los valores en tiempo real. Al apagar el sensor se muestra la temperatura máxima y mínima de las lecturas. La temperatura varía aleatoriamente entre un rango de 0°C a 100°C.</p> <p>4. Se tiene un tanque de agua, el cual su capacidad máxima es de 500 litros. El objetivo es crear una interfaz gráfica donde le indique al usuario si el tanque está a punto de desbordarse o si está a punto de quedarse sin agua.</p>		
13	<p>Manipular los elementos de las funciones booleanas, a partir del lenguaje de programación de interfaces gráficas de usuario, para el diseño y desarrollo de programas de aplicación, con actitud responsable, analítica y organizada.</p>	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <p>1. Diseña una interfaz gráfica que verifique el estado de interruptores de enclavamiento de seguridad antes de activar un mecanismo.</p> <p>2. Diseña una interfaz gráfica que seleccione entre obtener el logaritmo natural o el logaritmo</p>	<p>Computadora, software de programación, pintarrón y proyector.</p>	4 horas

		base 10 de un número entero. 3. Diseña una interfaz gráfica para calcular el volumen de un líquido en un tanque cilíndrico.		
UNIDAD IV				
14	Implementar técnicas de diseño en programación visual, utilizando las herramientas especializadas, para crear una interfaz gráfica que lea y escriba datos mediante el puerto de comunicación serial y realice el despliegue en pantalla de los mismos, con actitud organizada, sistemática y responsable.	El docente indica las instrucciones para que el alumno realice los siguientes ejercicios: 1. Diseña una interfaz gráfica que realice la lectura de datos mediante la implementación del módulo de comunicación serial para su despliegue en pantalla y su posterior procesamiento. Los datos provienen de un sistema simulado en MATLAB o Proteus para generar valores numéricos suponiendo que es un sistema embebido enviando datos por comunicación serial.	Computadora, software de programación, pintarrón y proyector.	6 horas
15	Implementar técnicas de diseño en programación visual, utilizando las herramientas especializadas de lectura y escritura mediante el puerto paralelo, para crear una interfaz gráfica que lea y escriba datos que permitan controlar el encendido y apagado de 3 leds, con actitud proactiva y responsable.	El docente indica las instrucciones para que el alumno realice los siguientes ejercicios: 1. Crea una interfaz gráfica donde existan 3 LEDs. Enviar un dato booleano, si este es verdadero los LEDs permanecen apagados, en caso contrario permanecen encendidos. 2. Mientras se presione un botón el LED1 y LED3 encienden, LED2 se apaga. Cuando no se presione el botón el LED2 enciende y los otros dos LEDs permanecen apagados.	Computadora, software de programación, pintarrón y proyector.	4 horas

		<p>3. Utiliza dos botones (B1 y B2). Si se presiona B1 los tres LEDs encienden de forma ascendente con un parpadeo constante. Si se presiona el B2 los tres LEDs encienden de forma descendente con un parpadeo que va de lo más lento a lo más rápido. Si no se presiona nada, los 3 LEDs encienden y apagan cada 1.5 segundos.</p>		
16	<p>Implementar técnicas de diseño en programación visual, utilizando las herramientas especializadas, para crear una interfaz gráfica que controle el encendido y sentido del giro de un motor, con actitud proactiva e innovadora.</p>	<p>El docente indica las instrucciones para que el alumno realice los siguientes ejercicios:</p> <p>1. Crea una interfaz gráfica para controlar: El encendido de un motor para que gire a la izquierda. El encendido de un motor para que gire a la derecha. Mantiene apagado el motor. Controla la velocidad a la que gira el motor.</p> <p>2. Al finalizar la práctica el alumno deberá dar una aplicación práctica al motor.</p>	<p>Computadora, software de programación, pintarrón y proyector.</p>	4 horas

VII. MÉTODO DE TRABAJO

Encuadre: El primer día de clase el docente debe establecer la forma de trabajo, criterios de evaluación, calidad de los trabajos académicos, derechos y obligaciones docente-alumno.

Estrategia de enseñanza (docente)

El instructor expone el tema, genera discusión, modela y proporciona ejemplos de aplicación de interfaces gráficas en el área de electrónica, funge como un facilitador y orientador en la construcción de las competencias de los estudiantes.

Estrategia de aprendizaje (alumno)

El estudiante trabajará en computadora, programará en algún lenguaje de programación visual para practicar la teoría impartida por el docente. Al finalizar el curso el alumno desarrollará un proyecto con aplicaciones electrónicas que involucre el uso de interfaz gráfica, durante el desarrollo de las sesiones el aprendizaje se desarrolla de forma autónoma y colaborativa.

VIII. CRITERIOS DE EVALUACIÓN

La evaluación será llevada a cabo de forma permanente durante el desarrollo de la unidad de aprendizaje de la siguiente manera:

Criterios de acreditación

- Para tener derecho a examen ordinario y extraordinario, el estudiante debe cumplir los porcentajes de asistencia que establece el Estatuto Escolar vigente.
- Calificación en escala del 0 al 100, con un mínimo aprobatorio de 60.

Criterios de evaluación

- Evaluaciones parciales (2)..... 40%
 - Laboratorio..... 40%
 - Evidencia de desempeño..... 20%
- (Proyecto de integración de interfaz de usuario)
- Total..... 100%

IX. REFERENCIAS

Básicas

- Ceballos, F. (2013). *Microsoft C#: curso de programación* (2ª ed.). México: Alfaomega;RA-MA.
- Ceballos, F. (2014). *Enciclopedia de Microsoft Visual C#: interfaces gráficas y aplicaciones para internet con windows forms y ASP.NET* (4ª ed.) México: Alfaomega; Ra-Ma.
- Yang, Y. (2014). *LabVIEW Graphical Programming Cookbook*. U.K.: Packt Publishing. Retrieved from: <http://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=690400&lang=es&site=ehost-live>

Complementarias

- Microsoft. (2017). *Visual Studio: Getting started tutorial*. Recuperado de: <https://visualstudio.microsoft.com/vs/getting-started/>
- National Instruments. (2014). *NI LabVIEW 101: Instrucciones en Video para Estudiantes*. Recuperado de: <http://www.ni.com/academic/students/learnlabview/esa/>

X. PERFIL DEL DOCENTE

El docente que imparta esta asignatura debe contar con título en Ingeniero en Electrónica, Computación, Sistemas Computacionales o carrera afín, deseable con posgrado en ciencias de la tecnología o ciencias computacionales. Se sugiere tener experiencia laboral como docente a nivel licenciatura mínima de un año. Además debe tener facilidad de palabra, fomentar el trabajo crítico, analítico y colaborativo, ser responsable y demostrar respeto a los alumnos.