

GUIA DOCENTE

Objetivos de aprendizaje

Competencias

Prerrequisitos

Temario

Descripción de los temas

Metodología

Cronograma

Profesor

OBJETIVOS DE APRENDIZAJE

Se muestran los objetivos docentes específicos de aprendizaje de la asignatura según las categorías de objetivos **declarativos y procedimentales**.

I) Declarativos

- Comprender los campos operativos de la instrumentación virtual y clásica
- Decidir el instrumental a emplear en cada experimentación o medida.
- Introducir al alumno en el diseño y arquitectura de diversos montajes prácticos obtenidos del entorno industrial.
- Comprender los fundamentos en el manejo de la instrumentación virtual.
- Introducir al alumno en la instrumentación avanzada.
- Iniciar al alumnado en los sistemas de instrumentación virtual y remota.

II) Procedimentales (OA_P)

- Analizar y diseñar sistemas electrónicos de adquisición datos y su visualización en el computador mediante software de instrumentación
- Manejar la instrumentación virtual con destrezas y competencias profesionales.
- Capacitar la posibilidad de detectar problemas de instrumentación y solucionarlos.
- Aprender a manejar y seleccionar documentos técnicos de módulos de instrumentación virtual.
- Discriminar el instrumental tanto de campo como de panel a emplear en cada situación y buscar alternativas.
- Manejar el software necesario para trabajar con sistemas de instrumentación virtual.
- Manejar entornos de desarrollo de prototipos de test y medida aplicados a la instrumentación.
- Manejar y configurar las aplicaciones hardware asociadas a la instrumentación virtual.
- Diseñar proyectos con control de instrumentación tanto tradicional como virtual

COMPETENCIAS

En este apartado se describen la competencias **Específicas-Procedimentales /Instrumentales/(Saber/Hacer) (C_EP)** que se trabajan en el aprendizaje de esta asignatura

- Utilizar con precisión las herramientas y entornos de trabajo disponibles orientados al diseño e implementación de instrumentos virtuales en el campo de la ingeniería industrial.
- Recoger e integrar información de diversas fuentes sobre cómo, en el ámbito del control y monitorización de procesos industriales, existe hoy en día una tendencia creciente caracterizada por la inclusión de la instrumentación virtual y sus sistemas electrónicos asociados.
- Desarrollar y aplicar diversas técnicas de diseño e implementación de sistemas digitales basados en instrumentación virtual.
- Aplicar con criterios operativos los bloques funcionales basados en instrumentos virtuales como subsistemas básicos en la medida y control de procesos industriales.
- Asimilar activamente que en ingeniería saber es hacer.
- Trabajar con una adecuada estimación y programación del trabajo.
- Interpretar y redactar correctamente documentación técnica.

PRERREQUISITOS

Conocimiento aplicado de: Técnicas y circuitos analógicos aplicados, Técnicas digitales, Tecnología electrónica e Instrumentación electrónica.

Con objeto de practicar con el entorno de programación es imprescindible la instalación del programa LabVIEW. Es suficiente la instalación de la versión de estudiante de libre distribución. Todos los programas disponibles están realizado con la versión LabVIEW 2009.

TEMARIO.

TEMA 1. INTRODUCCION A LabVIEW

- 1.1 Instrumentos Virtuales. Entorno operativo.
- 1.2 Panel frontal. Barra de herramientas.
- 1-3 Diagrama de bloques. Barra de herramientas.
- 1.4 Programación de flujo de datos
- 1.5 Creación de un VI simple.
- 1.6 Técnicas de depuración
- 1.7 Comprobación y gestión de errores
- 1.8 Ejemplos

TEMA 2. IMPLEMENTACIÓN DE UN VI

- 2.1 Diseño del panel frontal
- 2.2 Tipos de datos de LabVIEW
- 2.3 Documentación del código
- 2.4 Bucles While
- 2.5 Bucles For
- 2.6 Temporización de los Vis
- 2.7 Transferencia de datos iterativa
- 2.8 Representación de datos
- 2.9 Estructuras CASE
- 2.10 Ejemplos

TEMA 3. RELACIONAR DATOS

- 3.1 Arrays
- 3.2 Clusters
- 3.3 Definiciones de tipo
- 3.4 Ejemplos

TEMA 4. ALMACENAMIENTO DE DATOS DE MEDIDA

- 4.1 Comprender la E/S de ficheros
- 4.2 E/S de ficheros de alto nivel
- 4.3 Strings
- 4.4 Comprender la E/S de ficheros de bajo nivel
- 4.5 Ejemplos

TEMA 5. DESARROLLO DE APLICACIONES MODULARES

- 5.1 Comprensión de la modularidad
- 5.2 Creación de iconos y conectores
- 5.3 Uso de un subVI
- 5.4 Ejemplos

TEMA 6. ADQUISICIÓN DE DATOS

- 6.1 Introducción
- 6.2 Conceptos básicos de los sistemas de adquisición de datos.
- 6.3 Funciones generales de acondicionamiento de señal
- 6.4 Tarjetas de adquisición de datos. Tipos
- 6.5 Configuración del sistema de adquisición. Measurement & Automation
- 6.6 Explorer MAX. Tareas.
- 6.7 Librerías de adquisición de datos.
- 6.8 Entradas analógicas
- 6.9 Salidas analógicas
- 6.10 Uso de contadores
- 6.11 E/S digital
- 6.12 Ejemplos

TEMA 7. CONTROL DE INSTRUMENTOS

- 7.1 Uso del control de instrumentos
- 7.2 Uso de GPIB
- 7.3 Comunicación por puerto serie
- 7.4 Uso de otras interfaces
- 7.5 Arquitectura del software
- 7.6 Instrument I/O Assistant
- 7.7 Uso de VISA
- 7.8 Controladores de instrumento
- 7.9 Ejemplos.

TEMA 8. TÉCNICAS Y MODELOS DE DISEÑO COMUNES

- 8.1 Programación secuencial
- 8.2 programación de estado.
- 8.3 Máquinas de estado.
- 8.4 Paralelismo
- 8.5 Ejemplos

TEMA 9. TÉCNICAS DE DISEÑO TÍPICAS

- 9.1 Arquitecturas de bucle simple
- 9.2 Paralelismo
- 9.3 Arquitecturas de múltiples bucles
- 9.4 Eventos
- 9.5 Temporizar un patrón de diseño
- 9.6 Ejemplos.

TEMA 10. COMUNICACIÓN ENTRE MÚLTIPLES BUCLES

- 10.1 Variables
- 10.2 Variables globales funcionales
- 10.3 Condiciones de carrera
- 10.4 Sincronización de transferencia de datos
- 10.5 Ejemplos

TEMA 11. PROGRAMACIÓN DE EVENTOS

- 11.1 Programación orientada a eventos
- 11.2 patrones de diseño basados en eventos
- 11.3 Ejemplos

TEMA 12. CONTROL DE LA INTERFAZ DE USUARIO

- 12.1 Arquitectura del VI Server
- 12.2 Nodos de propiedad
- 12.3 Referencias de control
- 12.4 Nodos de invocación
- 12.5 Ejemplos

TEMA 13. MEJORA DE UN VI EXISTENTE

- 13.1 Refactorización de un código heredado
- 13.2 Problemas típicos de la refactorización
- 13.3 Ejemplos.

TEMA 14. CREACIÓN Y DISTRIBUCIÓN DE APLICACIONES

- 14.1 Características de LabVIEW para el desarrollo de proyectos
- 14.2 Preparación de la aplicación
- 14.3 Construcción de la aplicación y el instalador.
- 14.4 Ejemplos.

DESCRIPCIÓN DE LOS TEMAS.

Tema 1. En este primer tema se muestra cómo navegar por el entorno LabVIEW. Describe el uso de menús, barras de herramientas, paletas, herramientas, ayuda y cuadros de diálogo comunes de LabVIEW. También se aprenderá a ejecutar un VI (Virtual Instrumente) y conceptos generales de un panel frontal y diagramas de bloques. En esta lección también se un VI simple que adquiera, analice y represente datos.

Para ejecutar un VI, se deben cablear todos los subVIs, funciones y estructuras con los tipos de datos correctos para los terminales. A veces, un VI produce datos o se ejecuta de un modo inesperado. Puede utilizar el entorno de LabVIEW para configurar cómo se ejecuta un VI y para identificar problemas con la organización del diagrama de bloques o con los datos que pasan por el diagrama de bloques.

En el **Tema 2** se explica cómo implementar código en LabVIEW. Se explicará cómo diseñar una interface de usuario, elegir un tipo de datos, documentar el código, usar estructuras de bucles como los bucles for y while, añadir temporización por software al código implementado, representar los datos en una o varias gráficas y tomar decisiones en el código utilizando una estructura *case*.

Tema 3. En ocasiones es conveniente agrupar datos relacionados entre sí. Se emplean *arrays* y *clusters* para agrupar datos relacionados en LabVIEW. Los arrays combinan datos del mismo tipo en una estructura de datos, los clusters combinan datos de diferentes tipos en una estructura de datos. Se emplean definiciones de tipo para definir clusters y arrays personalizados. En este tema se explican arrays, clusters y definiciones de tipo, así como aplicaciones donde su uso puede resultar beneficioso.

Hasta el momento se ha aprendido a adquirir datos y visualizarlos, pero el almacenamiento de los datos suele ser muy importante en cualquier aplicación. En este **Tema 4** aprenderá a almacenar datos. LabVIEW presenta una gran variedad de funciones para tratar diferentes tipos de operaciones con ficheros. Estas funciones se encuentran en la paleta FILE I/O (entrada/salida de ficheros) del menú Functions. Para guardar datos en un fichero nuevo se han de seguir tres pasos: I) crear el fichero nuevo, II) escribir los datos en el fichero y III) cerrar el fichero. También se puede abrir uno ya existente y añadir la nueva información al final de ese fichero o sobrescribir su contenido.

En el entorno LabVIEW podemos escribir cualquier tipo de datos en el fichero que hemos abierto. Si otros usuarios o aplicaciones necesitan acceder a él, se deben escribir strings de datos, lo cual crea un fichero ASCII. Si la prioridad está en la rapidez de acceso y en lo compacto, se debe escribir la información con formato binario. Se crea un fichero con formato binario, por ejemplo, cuando se escriben datos no-string (como un array de números) en un fichero.

En el **Tema 5** se describe cómo desarrollar aplicaciones modulares. La potencia de LabVIEW reside en la naturaleza jerárquica del VI. Tras crear un VI, se puede utilizar en el diagrama de bloques de otro VI. No existen límites en el número de capas o niveles de la jerarquía. El uso de la programación modular ayuda a gestionar cambios y depurar el diagrama de bloques rápidamente.

Tema 6. Un sistema de adquisición de datos (DAQ) usa un dispositivo de adquisición de datos para pasar una señal eléctrica previamente acondicionada a un ordenador para su análisis y registro de datos. Se puede elegir un dispositivo de adquisición de datos que use un bus PCI, un bus PCI Express, un bus PXI, o el puerto USB o IEEE 1394 del ordenador. En este tema se

explica el hardware empleado en un sistema de adquisición de datos, cómo configurar los dispositivos y cómo programar las entradas y salidas analógicas, contadores y las entradas y salidas digitales.

La interface gráfica de LabVIEW ofrece una gran potencia de representación de señales y además dispone de librerías de procesado para el tratamiento de las señales adquirida. Para que todo esto sea posible, LabVIEW ofrece una librería de adquisición de datos que proporciona al usuario una herramienta de trabajo de fácil uso y que permite disponer de una mayor flexibilidad en cuanto al manejo de las DAQ se refiere.

En el **Tema 7** se describe cómo controlar instrumentos independientes con una interface GPIB o serie. Se puede emplear LabVIEW para controlar y adquirir datos de instrumentos con el Instrument I/O Assistant, la API VISA y controladores del instrumento.

Lo primero para desarrollar un proyecto en LabVIEW es explorar las arquitecturas que existen en el propio entorno. Las arquitecturas son esenciales para crear un buen diseño software. Las arquitecturas más comunes se suelen agrupar en modelos de diseño. En el **Tema 8** se estudian estos aspectos.

A medida que un modelo de diseño va ganando aceptación, resulta más fácil reconocer cuándo se ha utilizado uno. Este reconocimiento ayuda a leer y modificar Vis que se basan en modelos de diseño.

Existen numerosos modelos de diseño para los VIs. La mayoría de las aplicaciones usan al menos uno.

Tema 9. Se pueden desarrollar mejores programas en LabVIEW y en otros lenguajes si se siguen técnicas y arquitecturas de programación sistemáticas. En este tema se describen dos tipos de arquitecturas programación: bucles simples y bucles múltiples. En su conjunto, estas arquitecturas se conocen como patrones de diseño.

Las arquitecturas de bucle simple incluyen los patrones de diseño del VI simple, del VI general y de la máquina de estados.

Las arquitecturas de múltiples bucles son los patrones de diseño del VI de bucles paralelos, de maestro/esclavo, y de productor consumidor.

En este contexto comprender el uso apropiado de cada patrón de diseño ayuda a crear Vis de LabVIEW más eficientes.

En el tema anterior *Técnicas de diseño típicas*, se solventan las dificultades de transferir datos entre múltiples bucles mientras se mantiene la ejecución paralela en los bucles. En este **Tema 10**, se describen técnicas de comunicación para transferir datos entre múltiples bucles. Estas técnicas de comunicación incluyen variables, notificadoros y colas. También se aprenderá a resolver los problemas de programación mediante la utilización de las referidas técnicas y métodos.

Tema 11. Los patrones de diseño basados en eventos permiten desarrollar aplicaciones más eficientes y flexibles. Los patrones de diseño basados en eventos usan la estructura *Event* para responder directamente a los eventos de usuario o de otro tipo. Este tema describe la programación orientada a eventos mediante la estructura *Event* y los patrones de diseño que usan esta estructura.

Tema 12. Al editar programas, a menudo se deben modificar los atributos de los objetos del panel frontal programáticamente. Por ejemplo, quizá sea necesario hacer un objeto invisible hasta cierto momento de la ejecución del programa. En LabVIEW se puede utilizar VI Server

para acceder a las propiedades y métodos de objetos del panel frontal. En este tema se explica el VI Server, Nodos de Propiedad, Referencias de Control y Nodos de Invocación.

Un problema común cuando se accede a Vis de otros desarrolladores es que pueden haberse añadido características sin prestar atención al diseño, lo que dificulta cada vez más añadir nuevas características durante la vida operativa del VI. Esto se conoce como decadencia del software. Una solución a la referida decadencia es refactorizar el software. En el **Tema 13** se afrontan y proponen soluciones a este problema. Refactorización es el proceso de volver a diseñar software para hacerlo más legible y mantenible para que el coste del cambio no aumente con el tiempo. La refactorización cambia la estructura interna de un VI para hacerlo más legible y mantenible sin cambiar su operatividad.

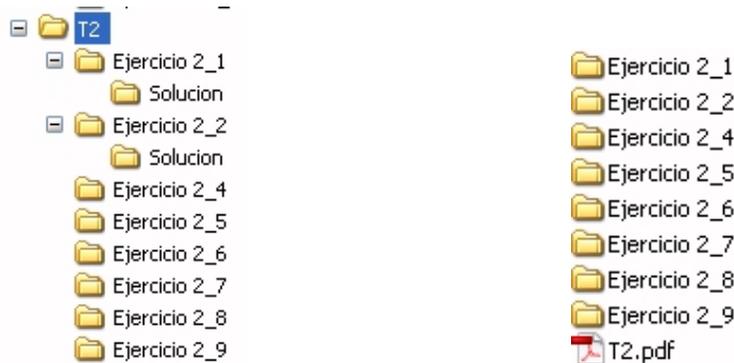
Tema 14. En este se describe el proceso de crear un instalador y una aplicación independiente para los proyectos de LabVIEW.

METODOLOGÍA

Debido a su carácter práctico, la exposición de los temas se realiza mediante ejemplos numéricos y gráficos. Los problemas elevan el nivel de los ejercicios numéricos resueltos en teoría. En este sentido se está empujando una estrategia formativa de probada eficacia como es la de aprender haciendo.

Material del curso

El curso está integrado en 14 temas. Para cada tema se proporciona la correspondiente información teórica en formato pdf junto a los ejercicios y ejemplos asociados para lo cual se necesita el programa LabVIEW en versión 2009 o superior. A modo de ejemplo se muestra la estructura del material docente correspondiente al *Tema 2: Implementación de un VI*, al extraer el contenido del fichero T2.rar



Siguiendo un desarrollo modular, aquellos Vis que son necesarios para que se ejecute otro se encuentran en la carpeta SUB VIs.

Como material de estudio complementario se adjunta:

- 1.- Un ANEXO_A sobre Análisis y Procesamiento de datos numéricos. Fichero ANEXO_A.pdf
- 2.- Un ANEXO_B sobre Fundamentos de Medida. Fichero ANEXO_B.pdf
- 3.- Un curso de LabVIEW editado en euskara. Fichero LabVIEW EUSKARA.pdf
- 4.- Links a páginas web sobre aplicaciones con LabVIEW. Fichero

Actividades

Las actividades emanan de la estrategia docente propuesta, aprender haciendo. En este contexto se plantea la asimilación de los ejercicios propuestos, realizar modificaciones

operativas en los mismos y finalmente desarrollar e implementar otro distinto. Con objeto de trabajar con una DAQ es posible su utilización previa solicitud al profesor.

Trabajo fin de curso

El trabajo fin de curso estará relacionado con la instrumentación virtual. Se estructurará en:

- Título
- Descripción
- Desarrollo e implementación del instrumento virtual
- Aplicaciones
- Resultados obtenidos
- Conclusiones

CRONOGRAMA

Temporalmente, el curso se distribuye en quince sesiones con un total de 45 horas. En la tabla adjunta se muestra la distribución temporal estimada.

Empleo del ordenador personal en la instrumentación de panel		
Sesión	Tema	Horas
1	Tema 1: Introducción a LabVIEW	2
2	Tema 2: Implementación de un VI	2
3	Tema 3: Relacionar datos	1
4	Tema 4: Almacenamiento de datos de medida	2
5	Tema 5: Desarrollo de aplicaciones modulares	2
6	Tema 6: Adquisición de datos	3
7	Tema 7: Control de instrumentos	1
8	Tema 8: Técnicas de modelos y diseños comunes	2
9	Tema 9: Técnicas de diseño típicas	2
10	Tema 10: Comunicación entre múltiples bucles	2
11	Tema 11: Programación de eventos	2
12	Tema 12: Control de la interfaz de usuario	2
13	Tema 13: Mejora de un VI existente	2
14	Tema 14: Creación y distribución de aplicaciones	1
15	Trabajo final	4

PROFESOR



Iñigo J. Oleagordia Aguirre

Centro: *Escuela Universitaria de Ingeniería Técnica Industrial de de Bilbao, UPV/EHU*

Despacho: *Laboratorio de Tecnología Electrónica*

Teléfono: *34 94 601 4304*

E-mail: ij.oleagordia@ehu.es