

Domina la Programación de Dispositivos Lógicos:

Soluciones Industriales con PLCC

Ingeniería | Ingeniería industrial | Aprendizaje Basado en Problemas

Descripción

Este plan de clase está diseñado para estudiantes de educación técnica y tecnológica en Ingeniería Industrial, con el propósito de desarrollar habilidades prácticas y teóricas en la programación de dispositivos lógicos programables (PLCC). A través de la metodología de Aprendizaje Basado en Problemas (ABP), los estudiantes analizarán y resolverán situaciones reales de la industria que requieren automatización y control mediante PLCC, reforzando así el pensamiento crítico y la capacidad de aplicar conocimientos en contextos laborales concretos.

Los estudiantes aprenderán a diseñar, programar y verificar sistemas automatizados con PLCC, entendiendo su lógica de funcionamiento y su aplicación en procesos industriales. Este aprendizaje es fundamental para mejorar la eficiencia y la calidad en la industria moderna, donde la automatización es clave para la competitividad y la innovación. Además, se fomenta la colaboración, la investigación y la autoevaluación, preparando a los estudiantes para enfrentar desafíos profesionales reales con confianza y conocimiento actualizado.

Objetivos de Aprendizaje

- Analizar problemas industriales que pueden ser automatizados mediante dispositivos lógicos programables (PLCC).
- Diseñar y programar soluciones lógicas utilizando software de programación de PLCC.
- Evaluar y verificar el funcionamiento correcto de los programas desarrollados en simuladores o dispositivos reales.
- Resolver problemas reales de automatización industrial aplicando los conceptos y técnicas aprendidas.
- Comunicar efectivamente el proceso y resultados de la programación de PLCC en contextos industriales.

Recursos Necesarios

- Computadoras con software de simulación y programación de PLCC (ejemplo: LogixPro, Siemens TIA Portal, o similar).
- Dispositivos lógicos programables PLCC físicos (mínimo 2 unidades para prácticas presenciales).
- Manual básico de programación de dispositivos lógicos programables (impreso o digital).
- Proyector y pantalla para presentaciones y demostraciones.
- Materiales impresos con casos reales de automatización industrial.
- Conexión a internet para acceso a tutoriales y recursos digitales.
- Elementos para elaboración de diagramas de flujo y esquemas lógicos (pizarras, marcadores, papelógrafos).

Requisitos Previos

- Conocimientos básicos de electrónica digital y lógica booleana.
- Familiaridad con conceptos de automatización industrial.
- Habilidades básicas en el manejo de software y computadoras.
- Experiencia previa en interpretación de diagramas eléctricos o lógicos.

Actividades

Plan de actividades para Programación de dispositivos lógicos (PLCC)

Sesión 1: Introducción y diagnóstico del problema industrial para automatización con PLCC

Fase de Inicio

Tiempo estimado: 15 minutos

Propósito de la sesión: Conectar con conocimientos previos y contextualizar la importancia de la programación de PLCC en la industria.

Activación de conocimientos previos:

- **Docente:** "¿Pueden mencionar qué tipos de procesos industriales conocen que requieren automatización? ¿Han visto o usado algún dispositivo que controle máquinas o procesos automáticamente?"
- **Estudiantes:** Comparten ejemplos breves y experiencias relacionadas con automatización o control.

Motivación y enganche:

- **Docente:** Presenta un video corto (3 min) que muestra una línea de producción automatizada usando PLCC, enfatizando el beneficio en eficiencia y seguridad.
- **Estudiantes:** Observan atentamente y anotan ideas o preguntas.

Contextualización:

- **Docente:** Explica cómo la programación de PLCC es una habilidad clave para solucionar problemas reales en la industria y mejorar procesos.
- **Estudiantes:** Relacionan el contenido con su futura labor profesional y plantean expectativas.

Fase de Desarrollo

Tiempo estimado: 95 minutos

Presentación del contenido:

El docente presenta el caso problema industrial: una planta que requiere automatizar el arranque y parada de un motor con condiciones específicas de seguridad y secuencia, sin usar programación tradicional sino PLCC.

- **Actividad 1: Análisis del problema industrial**

- **Objetivo:** Analizar y comprender el problema para identificar requerimientos de automatización.
- **Instrucciones:**
 - **Docente:** Divide a los estudiantes en grupos de 4. Entrega el caso con especificaciones técnicas.
 - **Estudiantes:** Discuten en grupo y hacen una lista de requerimientos y condiciones para el sistema.
 - **Producto:** Listado de requisitos del sistema automatizado.
 - **Tiempo:** 30 minutos.
 - **Rol docente:** Observa grupos, formula preguntas como "¿Qué condiciones deben cumplirse para iniciar el motor?" o "¿Qué medidas de seguridad consideran?" para guiar el análisis.
- **Actividad 2: Diseño del diagrama lógico preliminar**
 - **Objetivo:** Crear un diagrama lógico que refleje el control necesario para el sistema.
 - **Instrucciones:**
 - **Docente:** Solicita a cada grupo diseñar un diagrama de lógica booleana que cumpla con los requisitos.
 - **Estudiantes:** Elaboran el diagrama en papel o pizarra.
 - **Producto:** Diagrama lógico preliminar.
 - **Tiempo:** 40 minutos.
 - **Rol docente:** Supervisa y retroalimenta, planteando preguntas para mejorar el diseño.
- **Actividad 3: Presentación grupal y discusión**
 - **Objetivo:** Comunicar y discutir las soluciones propuestas.
 - **Instrucciones:**
 - **Docente:** Cada grupo expone su diagrama y razonamiento en 5 minutos.
 - **Estudiantes:** Escuchan y aportan comentarios.
 - **Producto:** Presentación verbal y gráfica de la solución.
 - **Tiempo:** 25 minutos.
 - **Rol docente:** Facilita la discusión, destaca aciertos y áreas de mejora.

Diferenciación:

- Para estudiantes que terminan antes: Proponer que comiencen a explorar el software de simulación para familiarizarse con la interfaz.
- Para estudiantes con dificultades: El docente ofrece apoyo individual o en parejas para clarificar conceptos y guiar el diseño.

Transición: El docente conecta la lógica diseñada con la necesidad de programarla en un software PLCC en la próxima sesión.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis: Cada estudiante escribe en un papel 3 ideas clave aprendidas y una duda.

Reflexión metacognitiva:

- ¿Qué aspectos del análisis del problema me resultaron más claros o difíciles?
- ¿Cómo puedo aplicar este análisis en otros casos industriales?

Retroalimentación: El docente recoge las ideas y dudas, comenta los puntos comunes y aclara inquietudes.

Transferencia: Se anticipa la programación práctica en software PLCC para la siguiente sesión.

Tarea: Investigar ejemplos de aplicaciones reales de PLCC en la industria y traer al aula un caso para compartir.

Sesión 2: Programación básica en software PLCC y simulación del problema industrial

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión: Conectar la lógica diseñada con la implementación en software de programación PLCC.

Activación de conocimientos previos:

- **Docente:** Pregunta: "¿Qué elementos ven en el diagrama lógico que podrían traducirse en instrucciones dentro del software de programación?"
- **Estudiantes:** Responden y relacionan conceptos.

Motivación y enganche: Demo rápida de un programa simple en el software que controla una lámpara virtual, mostrando entrada, proceso y salida.

Contextualización: Se explica que el software es la herramienta que usaremos para crear programas que controlan maquinaria real.

Fase de Desarrollo

Tiempo estimado: 100 minutos

- **Actividad 1: Familiarización con el software de programación PLCC**

- **Objetivo:** Conocer la interfaz y funciones básicas del software.
- **Instrucciones:**
 - **Docente:** Guía paso a paso para abrir el software, crear un nuevo proyecto y ubicar herramientas principales.
 - **Estudiantes:** Siguen los pasos en sus computadoras.
 - **Producto:** Proyecto creado y guardado en el software.
 - **Tiempo:** 30 minutos.
 - **Rol docente:** Supervisa, resuelve dudas y sugiere atajos.

- **Actividad 2: Programación del diagrama lógico en el software**

- **Objetivo:** Traducir el diagrama lógico del problema a un programa funcional en el software.
- **Instrucciones:**
 - **Docente:** Indica a los grupos que comiencen a programar usando bloques de lógica (AND, OR, NOT, temporizadores, etc.) según su diseño.
 - **Estudiantes:** Programan colaborativamente y prueban cada bloque.
 - **Producto:** Programa funcional que simula el control del motor.
 - **Tiempo:** 50 minutos.
 - **Rol docente:** Observa, formula preguntas para corregir errores y apoya en la interpretación de funciones del software.

■ **Actividad 3: Simulación y pruebas del programa**

- **Objetivo:** Verificar el funcionamiento del programa y corregir errores.
- **Instrucciones:**
 - **Docente:** Explica cómo usar la herramienta de simulación para probar el programa.
 - **Estudiantes:** Ejecutan simulaciones, identifican fallos y ajustan el programa.
 - **Producto:** Programa corregido y validado mediante simulación.
 - **Tiempo:** 20 minutos.
 - **Rol docente:** Facilita la reflexión sobre errores comunes y estrategias de corrección.

Diferenciación:

- Estudiantes adelantados pueden explorar funciones avanzadas del software como temporizadores y contadores.
- Estudiantes con dificultades reciben apoyo guiado y ejemplos adicionales para programación básica.

Transición: El docente comenta que en la siguiente sesión se continuará con la programación avanzada y la aplicación en hardware real.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis: Rápido resumen grupal donde cada grupo menciona un desafío al programar y cómo lo resolvieron.

Reflexión metacognitiva:

- ¿Qué parte de la programación me pareció más difícil y por qué?
- ¿Cómo aseguro que el programa cumpla con los requisitos del problema?
- ¿Qué aprendí sobre la relación entre el diagrama lógico y la programación?

Retroalimentación: El docente valora la participación, puntualiza aprendizajes y aclara dudas.

Transferencia: Se anticipa la conexión con hardware real y pruebas físicas en la siguiente sesión.

Tarea: Practicar en casa la programación de un pequeño circuito lógico simple en el software.

Sesión 3: Programación avanzada y puesta en marcha de dispositivos PLCC

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión: Repasar la sesión anterior y preparar la conexión con el hardware real.

Activación de conocimientos previos: Pregunta grupal: "¿Qué funciones del software consideran esenciales para un control industrial confiable?"

Motivación y enganche: Demostración en vivo de un dispositivo PLCC ejecutando una tarea sencilla.

Contextualización: Explicación de cómo la programación se materializa en dispositivos físicos para controlar procesos reales.

Fase de Desarrollo

Tiempo estimado: 100 minutos

- **Actividad 1: Introducción a funciones avanzadas (temporizadores, contadores, bloques de función)**

- **Objetivo:** Incorporar funciones avanzadas para optimizar el control.

- **Instrucciones:**

- **Docente:** Explica y ejemplifica el uso de temporizadores y contadores en el software, relacionando con ejemplos industriales.

- **Estudiantes:** Observan y replican ejemplos.

- **Producto:** Ejercicios de función avanzada realizados.

- **Tiempo:** 30 minutos.

- **Rol docente:** Responde preguntas técnicas, ilustra con diagramas.

- **Actividad 2: Programación y conexión con dispositivo PLCC físico**

- **Objetivo:** Transferir el programa al dispositivo y realizar pruebas físicas.

- **Instrucciones:**

- **Docente:** Divide grupos para programar y cargar el programa en el PLCC físico.

- **Estudiantes:** Ejecutan carga, configuran entradas y salidas, y prueban el sistema con señales reales o simuladas.

- **Producto:** Programa operativo en dispositivo PLCC.

- **Tiempo:** 50 minutos.

- **Rol docente:** Supervisa conexiones, verifica seguridad y guía solución de problemas.

■ **Actividad 3: Registro y análisis de resultados**

■ **Objetivo:** Evaluar el desempeño del sistema y documentar mejoras.

■ **Instrucciones:**

■ **Docente:** Indica a los estudiantes que registren observaciones sobre funcionamiento y posibles fallas.

■ **Estudiantes:** Elaboran reporte breve con conclusiones y propuestas.

■ **Producto:** Informe de resultados y mejoras.

■ **Tiempo:** 20 minutos.

■ **Rol docente:** Revisa informes y fomenta discusión crítica.

Diferenciación:

■ Para estudiantes avanzados: Proponer optimizaciones del programa usando funciones adicionales.

■ Para estudiantes con dificultades: Apoyo personalizado para manejo de hardware y conceptos técnicos.

Transición: El docente prepara al grupo para abordar problemas más complejos y casos de estudio en la siguiente sesión.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis: Preguntas rápidas: ¿Qué aprendí hoy? ¿Cómo se aplica esto en la industria?

Reflexión metacognitiva:

■ ¿Qué funciones avanzadas incorporé y por qué?

■ ¿Qué dificultades tuve con el dispositivo físico y cómo las resolví?

Retroalimentación: Comentarios individualizados y grupales por parte del docente.

Transferencia: Preparación para un reto de diseño de sistema PLCC completo en próximas sesiones.

Tarea: Investigar fallas comunes en sistemas PLCC y posibles soluciones.

Sesión 4: Diagnóstico y solución de problemas en sistemas PLCC industriales

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión: Introducir la importancia del diagnóstico y la solución de problemas en programación PLCC.

Activación de conocimientos previos: Pregunta: "¿Qué pasos seguirían si un sistema automatizado no funciona como se espera?"

Motivación y enganche: Presentación de un caso real donde un fallo en programación causó pérdidas en una planta.

Contextualización: Discusión sobre el impacto económico y técnico de errores en sistemas PLCC.

Fase de Desarrollo

Tiempo estimado: 100 minutos

■ **Actividad 1: Análisis de casos de fallas comunes**

■ **Objetivo:** Identificar causas típicas de fallas en programación y hardware.

■ **Instrucciones:**

■ **Docente:** Proporciona ejemplos escritos y simulados de fallas.

■ **Estudiantes:** En grupos, analizan causas y proponen soluciones.

■ **Producto:** Lista de problemas y soluciones.

■ **Tiempo:** 40 minutos.

■ **Rol docente:** Guía la identificación y asegura comprensión.

■ **Actividad 2: Simulación de diagnóstico y reparación en software**

■ **Objetivo:** Practicar la identificación y corrección de errores en programas PLCC.

■ **Instrucciones:**

■ **Docente:** Entrega programas con errores intencionales para corregir.

■ **Estudiantes:** Detectan y corrigen errores en simuladores.

■ **Producto:** Programa corregido y funcional.

■ **Tiempo:** 40 minutos.

■ **Rol docente:** Proporciona pistas y retroalimentación oportuna.

■ **Actividad 3: Presentación de estrategias preventivas**

■ **Objetivo:** Desarrollar estrategias para evitar fallas en programación y operación.

■ **Instrucciones:**

■ **Docente:** Solicita que los grupos elaboren un listado de buenas prácticas.

■ **Estudiantes:** Presentan y discuten las listas.

■ **Producto:** Documento con estrategias preventivas.

■ **Tiempo:** 20 minutos.

■ **Rol docente:** Complementa con recomendaciones profesionales.

Diferenciación:

■ Estudiantes adelantados profundizan en diagnóstico de hardware.

■ Estudiantes con dificultades reciben apoyo para interpretación de errores y corrección.

Transición: Se prepara para diseño final y aplicación integral en las siguientes sesiones.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis: Creación colectiva de un mapa conceptual sobre diagnóstico y solución de problemas en PLCC.

Reflexión metacognitiva:

- ¿Cómo puedo identificar rápidamente un error en un sistema PLCC?
- ¿Qué estrategias me ayudan a corregir fallas sin afectar la producción?

Retroalimentación: Comentarios orales y escriturales por parte del docente.

Transferencia: Vinculación con la próxima actividad de diseño integral.

Tarea: Preparar una propuesta de mejora para un sistema PLCC real o simulado.

Sesión 5: Diseño integral y solución de un caso real con PLCC

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión: Motivar a los estudiantes a integrar todo lo aprendido en un proyecto completo.

Activación de conocimientos previos: Revisión rápida de los conceptos clave y aprendizajes anteriores mediante preguntas dirigidas.

Motivación y enganche: Presentación del reto final: automatizar una línea de envasado con varias condiciones y sensores.

Contextualización: Explicación de la relevancia industrial y expectativas del proyecto.

Fase de Desarrollo

Tiempo estimado: 100 minutos

■ Actividad 1: Planificación y diseño del sistema

- **Objetivo:** Elaborar el diseño lógico y secuencial para el sistema completo.
- **Instrucciones:**
 - **Docente:** Entrega el caso con especificaciones detalladas y recursos.
 - **Estudiantes:** En grupos diseñan diagramas y planifican la programación y pruebas.
 - **Producto:** Plan de trabajo y diseño lógico.
 - **Tiempo:** 40 minutos.
 - **Rol docente:** Orienta y valida el plan.

■ Actividad 2: Programación y simulación del sistema

- **Objetivo:** Programar y validar la solución en software y simulador.
- **Instrucciones:**

- **Docente:** Supervisa la ejecución y resolución de problemas.
- **Estudiantes:** Programan, simulan y ajustan su sistema.
- **Producto:** Programa funcional y simulado.
- **Tiempo:** 40 minutos.
- **Rol docente:** Provee retroalimentación continua.

■ **Actividad 3: Presentación y discusión del proyecto**

- **Objetivo:** Comunicar resultados y recibir retroalimentación.
- **Instrucciones:**
 - **Docente:** Organiza presentaciones de 5 minutos por grupo.
 - **Estudiantes:** Exponen y responden preguntas.
 - **Producto:** Presentación y discusión crítica.
 - **Tiempo:** 20 minutos.
 - **Rol docente:** Evalúa y motiva la autoevaluación.

Diferenciación:

- Estudiantes avanzados pueden agregar funciones extra o mejoras.
- Estudiantes con dificultades reciben apoyo en programación y diseño.

Transición: El docente prepara para la validación en hardware y cierre del proceso en la siguiente sesión.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis: Reflexión grupal sobre aprendizajes y retos superados.

Reflexión metacognitiva:

- ¿Qué parte del diseño o programación fue más retadora y cómo la superé?
- ¿Cómo puedo aplicar este proyecto en un entorno real?

Retroalimentación: Comentarios específicos del docente y reconocimiento del esfuerzo.

Transferencia: Anuncio de la siguiente sesión para pruebas en dispositivos físicos.

Tarea: Preparar documentación y manual de usuario para el sistema.

Sesión 6: Puesta en marcha final, evaluación y reflexión sobre la programación de PLCC

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión: Organizar el trabajo final y motivar la disposición para la evaluación práctica.

Activación de conocimientos previos: Breve repaso oral de conceptos clave y aprendizajes previos.

Motivación y enganche: Presentación de la importancia de la certificación y competencias en PLCC para la industria.

Contextualización: Enlace con oportunidades laborales y desarrollo profesional.

Fase de Desarrollo

Tiempo estimado: 100 minutos

■ **Actividad 1: Puesta en marcha y pruebas en dispositivo PLCC físico**

■ **Objetivo:** Validar funcionamiento del programa en hardware real.

■ **Instrucciones:**

■ **Docente:** Organiza estaciones de trabajo con dispositivos PLCC físicos.

■ **Estudiantes:** Cargan programa, configuran y realizan pruebas funcionales.

■ **Producto:** Sistema operativo en hardware real.

■ **Tiempo:** 60 minutos.

■ **Rol docente:** Supervisa seguridad, precisión y funcionamiento.

■ **Actividad 2: Evaluación práctica y autoevaluación**

■ **Objetivo:** Evaluar el desempeño individual y grupal en la programación y aplicación.

■ **Instrucciones:**

■ **Docente:** Aplicar lista de cotejo para evaluación y guiar autoevaluación.

■ **Estudiantes:** Completar autoevaluación y discutir aprendizajes.

■ **Producto:** Informe de autoevaluación y resultados de evaluación docente.

■ **Tiempo:** 40 minutos.

■ **Rol docente:** Proporciona retroalimentación constructiva y orienta mejoras.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis: Elaboración colectiva de un mural o cartel con las 5 mejores prácticas para programación y aplicación de PLCC.

Reflexión metacognitiva:

■ ¿Cómo aplicaría lo aprendido en un trabajo real?

■ ¿Qué habilidades debo reforzar para ser un profesional competente en PLCC?

Retroalimentación: Evaluación final oral y escrita, con recomendaciones para futuro desarrollo.

Transferencia: Invitación a explorar cursos avanzados o certificaciones en automatización industrial.

Tarea final: Preparar un portafolio digital con todos los proyectos y evidencias obtenidas durante el curso.

Evaluación

Tipo de evaluación:

- **Diagnóstica:** Sesión 1, para conocer conocimientos previos y expectativas.
- **Formativa:** Durante todas las sesiones en actividades prácticas, simulaciones, correcciones y discusiones.
- **Sumativa:** Sesión 6, evaluación práctica en hardware y evaluación de portafolio final.

Criterios de evaluación:

- Capacidad para analizar y definir requerimientos de automatización (Objetivo 1).
- Habilidad para diseñar y programar soluciones lógicas funcionales en software PLCC (Objetivos 2 y 4).
- Competencia para verificar y validar programas mediante simulación y pruebas en hardware (Objetivo 3).
- Destreza para resolver problemas y corregir fallas en sistemas PLCC (Objetivo 4).
- Claridad y efectividad en la comunicación técnica del proceso y resultados (Objetivo 5).

Instrumentos sugeridos:

- Lista de cotejo para evaluación práctica en software y hardware.
- Rúbrica para evaluación de presentaciones y reportes.
- Observación directa durante actividades de laboratorio.
- Portafolio digital con evidencias de programación, diseños y reflexión.
- Autoevaluación y coevaluación entre pares para fomentar reflexión crítica.

Evidencias de aprendizaje:

- Diagramas lógicos y listados de requisitos elaborados en grupo.
- Programas funcionales en software y dispositivos PLCC físicos.
- Reportes de pruebas y análisis de fallas con propuestas de solución.
- Presentaciones orales y escritas de proyectos finales.
- Portafolio con documentación y reflexión personal.