

Planificador inteligente preliminar para sincronizar capacitación y producción basado en lenguajes de planeación basados en definición de dominios PDDL.

Preliminary intelligent planner to synchronize training and production based on PDDL Planning Domain Definition Language.

Diego Ramírez Hernández (1).
Estudiante Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez.
L22270523@tuxtla.tecnm.mx.

Mauricio Jair Reyes González (2). Estudiante Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez,
L22271100@tuxtla.tecnm.mx.

Héctor Guerra Crespo* (3). Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez. hector.gc@tuxtla.tecnm.mx.

Luis Tomás García Andrade (4). Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez,
luis.ga1@tuxtla.tecnm.mx.

Jorge Octavio Guzmán Sánchez (5). Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez,
jorge.gs1@tuxtla.tecnm.mx.

Elfer Isaías Clemente Camacho (6), Tecnológico Nacional de México/I. T. de Tuxtla Gutiérrez,
elfer.cc@tuxtla.tecnm.mx.

*corresponding author.

Artículo recibido en mayo 04, 2026; aceptado en junio 02, 2026.

Resumen.

Este artículo presenta el desarrollo de un planificador inteligente preliminar basado en el Lenguaje de Definición de Dominios para Planificación (PDDL) con el propósito de sincronizar los procesos de capacitación del personal y producción industrial en líneas de ensamble teóricos. La investigación surge de la necesidad de optimizar la asignación de operadores y garantizar que cada trabajador cuente con la capacitación necesaria antes de desempeñarse en una estación de trabajo específica. El estudio toma como referencia una línea modelo de una planta manufacturera de arneses, donde la complejidad de las operaciones exige personal certificado y una adecuada coordinación de recursos. Se revisan antecedentes teóricos de la planificación automática, incluyendo GPS, STRIPS, Graphplan y PDDL, destacando su importancia en la generación de planes eficientes. Posteriormente, se modela un dominio compuesto por dos líneas de producción, tres estaciones por línea, diez operadores y dos piezas a fabricar. El modelo establece restricciones de capacitación, asignación y operación, definiendo predicados, acciones, estados iniciales y metas mediante PDDL. Los experimentos realizados con la herramienta editor.planning.domains muestran que el planificador genera secuencias de acciones más eficientes que las propuestas inicialmente por un humano, reduciendo pasos y demostrando el potencial de la planificación automática para mejorar la productividad industrial.

Palabras claves: PDDL, Planificación automática, planificación industrial, sincronizar procesos.

Abstract.

This paper presents the development of a preliminary intelligent planner based on Planning Domain Definition Language (PDDL) to synchronize personnel training and industrial production processes on theoretical assembly lines. The research stems from the need to optimize operator allocation and ensure that each worker receives the necessary training before working at a specific station. The study uses a model line from a harness manufacturing plant as a reference, where the complexity of the operations demands certified personnel and proper resource coordination. The theoretical background of automated planning is reviewed, including GPS, STRIPS, Graphplan, and PDDL, highlighting their importance in generating efficient plans. Subsequently, a domain composed of two production lines, three stations per line, ten operators, and two parts to be manufactured is modeled. The model establishes training, allocation, and operational constraints, defining predicates, actions, initial states, and goals using PDDL. Experiments conducted with the editor.planning.domains tool show that the planner generates more efficient action sequences than those initially proposed by a human, reducing steps and demonstrating the potential of automated planning to improve industrial productivity.

Keywords: Automated planning, industrial planning, PDDL, process synchronization.

1. Introducción.

La incorporación de la inteligencia artificial a nuestra vida diaria ha dado lugar a que su implementación se traslade a diferentes entornos, incluyendo en esto el sector industrial. Dicho sector ha experimentado un avance significativo en la automatización y optimización de procesos, así como en el apoyo a la toma de decisiones. Bajo este enfoque, la planificación automática se ha consolidado como una disciplina fundamental para resolver múltiples problemas presentes en cualquier industria. Su aplicación resalta especialmente en sistemas de manufactura modernos, donde la variabilidad, la presión por la eficiencia y la necesidad de adaptación continua demandan soluciones inteligentes y flexibles capaces de superar métodos tradicionales.

En el ámbito del ensamblaje industrial, múltiples investigaciones han demostrado la versatilidad del uso de PDDL para optimizar secuencias de tareas y mejorar la utilización de recursos. "No obstante, la industria automotriz presenta desafíos particulares debido a la alta densidad de componentes eléctricos presentes en las plantas de fabricación de arneses a gran escala, donde la secuenciación de tareas resulta crítica para la productividad (Araújo & Silva, 2016)."

En casos como el de Yazaki, la operatividad se define por una estructura multilínea que abarca desde el subensamblaje y direccionamiento, hasta el encintado y la prueba final. Cada una de estas etapas posee niveles de complejidad técnica distintos; por ejemplo, un solo modelo de arnés puede derivar en cientos de variantes. Esta complejidad exige que el personal posea un alto grado de polivalencia y certificaciones específicas para cada estación de trabajo. Dentro del proceso de planificación, destaca el Lenguaje de Definición de Dominios de Planificación (PDDL), desarrollado inicialmente para estandarizar la representación de problemas dentro de la comunidad de inteligencia artificial. "Este estándar ha proporcionado una base sólida para la aplicación de algoritmos de planificación gracias a su capacidad expresiva, permitiendo el modelado de dominios que requieren un manejo preciso del tiempo y recursos numéricos (Fox & Long, 2003)". Asimismo, este lenguaje generó una gran oportunidad de integración en ámbitos industriales, puesto que la toma de decisiones y la ejecución de acciones bajo planes estratégicos son fundamentales en el día a día empresarial.

El desarrollo de este documento se inspira en el trabajo que realiza la planta Yazaki Tuxtla 2 (figura 2), dedicada a la manufactura de arneses automotrices (figura 1).



Figura 1. Arnés eléctrico para vehículos híbridos y eléctricos de las marcas GM y Subaru. Fuente sitio web empresa japonesa Yazaki. Recuperado diciembre 2024.



Figura 2. Planta Yazaki Tuxtla 2. fotografía diciembre 2024.

La planta cuenta con 6 líneas de producción.

Línea modelo. Estandariza la fabricación del arnés. Esta línea es la piloto en el desarrollo de este trabajo.

Cada línea cuenta con 4 estaciones subensamble, direccionado, encintado y línea final.

Cada línea tiene entre 12 y 20 estaciones de subensambles dependiendo de la complejidad del arnés.

2. Métodos.

La inteligencia artificial cuenta con diferentes ramas, a continuación, se citan algunas.

- Aprendizaje. (Redes neuronales)
- Aprendizaje automático. (Machine learning)
- Aprendizaje profundo. (Deep learning)
- Lenguaje natural.
- Visión.
- Robótica.
- Planificación.
- Lógica difusa.
- Sistemas expertos.

La planificación automática como rama de la inteligencia artificial se dedica a la generación de secuencias de acciones (planes) que transforman un estado inicial en uno o varios estados objetivo, teniendo en cuenta las restricciones del dominio y los efectos de las acciones. La planificación produce deliberadamente planes antes de su ejecución o durante la ejecución cuando es necesario replantearlos, lo que la hace esencial para sistemas autónomos, robótica, logística y optimización de procesos industriales (Ghallab, 2004).

General Problem Solver (GPS).

General Problem Solver (GPS) fue uno de los primeros sistemas de inteligencia artificial desarrollados por Allen Newell, Herbert A. Simon y J. C. Shaw (1959) a finales de la década de 1950. Su propósito era modelar el proceso de resolución de problemas humanos mediante una estrategia general que pudiera aplicarse a diferentes dominios. GPS representó un avance significativo en la historia de la inteligencia artificial al introducir mecanismos formales para analizar problemas, identificar diferencias entre el estado actual y el estado objetivo, y seleccionar acciones que permitieran reducir dichas diferencias de manera sistemática.

El funcionamiento de GPS se basa principalmente en la técnica denominada *Means-Ends Analysis*, mediante la cual el sistema compara continuamente la situación actual con la meta deseada y genera subobjetivos para disminuir las discrepancias encontradas. Este enfoque permitió demostrar que los procesos de razonamiento podían describirse mediante reglas y procedimientos computacionales, sentando las bases para el desarrollo posterior de sistemas de planificación automática, representación del conocimiento y algoritmos de búsqueda utilizados en la inteligencia artificial moderna.

Es un método heurístico que consiste en:

- Comparar el estado actual con el estado meta.
- Identificar las diferencias entre ambos.
- Seleccionar una operación (medio) que reduzca alguna diferencia (fin).
- Aplicar la operación para acercarse al objetivo.
- Repetir el proceso hasta alcanzar la meta.
- La técnica busca minimizar la diferencia entre el estado actual y el estado deseado paso a paso.

STRIPS.

Un salto importante en las técnicas de planificación fueron los STRIPS (Stanford Research Institute Problem Solver). Es uno de los modelos más influyentes en la historia de la planificación automática en inteligencia artificial. Fue desarrollado por Richard Fikes y Nils Nilsson en el SRI (Stanford Research Institute) a inicios de los años 1970 (Fikes y Nilsson, 1971).

STRIPS es un formalismo para describir problemas de planificación, es decir, para especificar:

- Estado inicial.
- Estado meta (objetivos).
- Acciones disponibles, cada una con:
- Precondiciones (lo que debe cumplirse para ejecutarla).
- Add list (efectos positivos: lo que añade al estado).
- Delete list (efectos negativos: lo que elimina del estado).

STRIPS se convirtió en el estándar para que los algoritmos de planificación puedan razonar y buscar planes.

STRIPS es importante porque introdujo ideas clave que definieron la planificación moderna como:

- Representación lógica de estados. Los estados se representan con predicados lógicos, lo que permite razonar formalmente.
- Efectos separados en *add* y *delete lists* lo que facilita la simulación de acciones y cálculo de estados sucesores.

- Precondiciones explícitas. Permite saber qué acciones son aplicables en un estado.
- Base del lenguaje moderno PDDL. El PDDL (Planning Domain Definition Language) usado hoy en competencias internacionales ICLP está basado directamente en STRIPS.
- Fundó la “planificación clásica”.

Graphplan.

A mediados de los 90 la planificación dio un salto importante con la introducción de Graphplan (Blum A. y Furst M., 1995), GraphPlan es un algoritmo de planificación automática que construye una estructura llamada grafo de planificación (planning graph) para encontrar planes eficientes de manera rápida.

Es importante por qué.

- Es más rápido que los planificadores anteriores
- Introduce el uso de mutuas exclusiones (mutex). Sirven para evitar combinaciones imposibles.
- Combina expansión estructurada y búsqueda eficiente.
- Genera planes mínimos en número de pasos.
- Se convirtió en la base de muchos planificadores modernos.
- PDDL adoptó su filosofía y muchas aplicaciones de planificación usan variantes de GraphPlan.

PDDL, lenguaje de definición de dominios para planificación.

La planificación en Inteligencia Artificial por lo general utiliza un lenguaje de modelación llamado PDDL (Planning Domain Definition Language), en la actualidad prácticamente es el lenguaje estándar para la codificación de los dominios de planificación dado a conocer por Drew McDermott en la International Planning Competition (IPC) de 1998 (Mcdermott, 1998).

PDDL está basado en STRIPS y cuenta con las versiones 2.1 (2002), 2.2 (2004), 3.0 (2006) hasta la actual 3.1 y última versión de 2011.

International Planning Competition (IPC).

La IPC (ICAPS, 2025) es una competencia académica internacional, organizada por la comunidad de planificación automática, que reúne a investigadores de todo el mundo para evaluar planificadores (planners) usando un conjunto estandarizado de dominios y problemas expresados en PDDL.

Se realiza aproximadamente cada dos años y está vinculada con el congreso ICAPS (International Conference on Automated Planning and Scheduling).

Objetivos principales de la IPC.

- Establecer un marco de evaluación común para comparar planificadores.
- Proveer métricas estandarizadas que permitan medir rendimiento.
- Evaluar eficiencia, escalabilidad, calidad del plan y tiempo de cómputo.
- Motivar el desarrollo de nuevos algoritmos y enfoques de planificación.
- Impulsar la evolución del lenguaje PDDL, que nació junto con la IPC.

3. Desarrollo.

La implementación del planificador es a través de la herramienta *editor.planning.domains* del proyecto Education.Planning.Domains (Education.Planning.Domains, 2025).

Etapas en la especificación del planificador.

1. Descripción del dominio.

El dominio considera dos líneas de producción L1 y L2.
 Tres estaciones por cada línea E1, E2 Y E3.
 10 operadores W1, W2, ..., W10.
 1 pieza a fabricar, la misma pieza en cada línea de producción.

2. Formalización del planificador.

Capacitación obligatoria antes de operar.
 Un empleado solo puede capacitarse en una estación (E1 o E2 o E3).
 Un empleado solo puede pertenecer a una línea (L1 o L2).
 Un empleado no puede operar estaciones de ambas líneas.
 Una pieza debe pasar secuencialmente por E1→E2→E3.
 La producción de L1 y L2 es independiente.
 La capacitación y la producción pueden coexistir porque existen empleados libres que pueden seguir capacitándose mientras otros ya producen.
 No se puede asignar un operador a una estación sin estar capacitado específicamente para esa estación.

3. Predicados (Predicates).

Predicado	Significado
(libre ?w - empleado)	Empleado sin capacitación ni asignación
(capacitar-e1 ?w - empleado)	Capacitación exclusiva por cada estación
(capacitar-e2 ?w - empleado)	
(capacitar-e3 ?w - empleado)	
(en-linea1 ?w - empleado)	Asignación del operador permanente a línea
(en-linea2 ?w - empleado)	
(op-e1 ?w - empleado ?l - línea)	Operadores instalados para producción.
(op-e2 ?w - empleado ?l - línea)	
(op-e3 ?w - empleado ?l - línea)	
(fabricar ?p - pieza)	Producción. Inicio de la fabricación de la pieza.
(en-e1 ?p - pieza)	Pieza en la estación actual.
(en-e2 ?p - pieza)	
(en-e3 ?p - pieza)	

(terminada ?p - pieza)

Pieza fabricada o terminada.

3. Operadores STRIPS (Acciones).

A continuación, se listan acciones STRIPS para cada proceso. Cada acción incluye: precondiciones y efectos.

--- Capacitación ---

Acción 1: capacitar en estación 1

Precondición.

```
(and
  (libre ?w)
  (not (capacitar-e1 ?w))
  (not (capacitar-e2 ?w))
  (not (capacitar-e3 ?w))
)
```

Efecto.

```
(capacitar-e1 ?w)
)
```

Capacitar en la estación 1 implica que el empleado, operador o trabajador está libre y no se encuentra capacitado en ninguna estación, al terminar esta acción el efecto es que el empleado está capacitado en la estación 1. Esta acción tiene la misma estructura para Acción 2: capacitar en estación 2 y Acción 3: capacitar en estación 3, la única diferencia sería el efecto.

--- Asignación a línea ---

Acción 4: Asignar a línea 1.

Precondición.

```
(and
  (not (en-línea1 ?w))
  (not (en-línea2 ?w))
  (or
    (capacitar-e1 ?w)
    (capacitar-e2 ?w)
    (capacitar-e3 ?w)
  )
)
```

Efecto.

```
(en-línea1 ?w)
)
```

Asignación en línea 1 implica que el operador no está asignado a una línea y está capacitado en cualquiera de las tres estaciones y el efecto es que estará asignado a línea 1, esta acción tiene la misma estructura que para la línea 2 Acción 5: Asignar a línea 2.

.

--- Operadores en línea ---

Acción 6: Instalar en estación 1 y en línea 1.

Precondición.

```
(and
  (capacitar-e1 ?w)
  (en-línea1 ?w)
)
```

Efecto.

```
(op-e1 ?w 11)
)
```

Instalar en estación 1 y en línea 1 implica que el operador está capacitado en la estación 1 y asignado a línea 1 y el efecto es que estará produciendo en estación 1 en línea 1, esta acción tiene la misma estructura que para las estaciones 2 y 3 en línea uno y línea generando las acciones.

- Acción 7: Instalar en estación 2 y en línea 1.
- Acción 8: Instalar en estación 3 y en línea 1.
- Acción 9: Instalar en estación 1 y en línea 2.
- Acción 10: Instalar en estación 2 y en línea 2.
- Acción 11: Instalar en estación 3 y en línea 2.

--- Producción ---

Acción 12. Producir estación 1.

Precondición.

```
(and
  (fabricar ?p)
  (op-e1 ?w 1)
)
```

Efecto.

```
(and
  (not (fabricar ?p))
  (en-e1 ?p)
)
```

Producir en estación 1 implica que el operador está instalado en la estación 1 y la pieza esta por fabricar o libre, el efecto es que la pieza no está por fabricar y la pieza se encuentra en la estación 1 disponible para la siguiente estación. La misma estructura es para las acciones.

Acción 13. Producir estación 2.

Acción 14. Producir estación 3.

Acción 15. Terminar. Pieza fabricada.

Precondición.

```
(en-e3 ?p)
```

```
Efecto.  
(and  
  (not (en-e3 ?p))  
  (terminada ?p)  
)
```

Pieza terminada o fabricada implica que el operador ya pasó por la estación 3 y el efecto es que la pieza queda fabricada y libre de la estación 3.

4. Estado inicial.

```
(libre w1)  
(libre w2)  
(libre w3)  
(libre w4)  
(libre w5)  
(libre w6)  
(libre w7)  
(libre w8)  
(libre w9)  
(libre w10)
```

```
(fabricar pieza-11)  
(fabricar pieza-12)
```

Cada operador disponible y las piezas libres para línea 1 y línea 2.

5. Objetivo (Goal).

```
(and  
  (terminada pieza-11)  
  (terminada pieza-12)  
)
```

Las piezas terminadas en línea 1 y línea 2.

6. Secuencia de planificación esperada (Plan).

El plan óptimo STRIPS generado sería:

Fase 1. Capacitar personal para Línea 1.

1. capacitación e1 w1.
2. capacitación e2 w2.
3. capacitación e3 w3

4. asignar lineal w1.
5. asignar lineal w2.
6. asignar lineal w3.

7. instalar e1 w1 11.

- 8. instalar e2 w2 l1.
- 9. instalar e3 w3 l1.

Fase 2. Capacitar personal para Línea 2.

- 10. Capacitación e1 w4.
- 11. Capacitación e2 w5.
- 12. Capacitación e3 w6.

- 13. Asignar linea2 w4.
- 14. Asignar linea2 w5.
- 15. Asignar linea2 w6.

- 16. Instalar e1 w4 l2.
- 17. Instalar e2 w5 l2.
- 18. Instalar e3 w6 l2.

Ya está sincronizada producción y capacitación.

Línea	E1	E2	E3
L1	W1	W2	W3
L2	W4	W5	W6

Fase 3. Producción en Línea 1.

- 19. Producir e1 p1 l1.
- 20. Producir e2 p1 l1.
- 21. Producir e3 p1 l1.

Fase 4. Producción en Línea 2.

- 22. Producir e1 p1 l2.
- 23. Producir e2 p1 l2.
- 24. Producir e3 p1 l2.

Estado final alcanzado.

Terminada p l1.
Terminada p l1.

Código PDDL del planificador.

Problema.

```
(define (problem fabrica)
(:domain produccion-capacitacion)
(:objects
```

```
w1 w2 w3 w4 w5 w6 w7 w8 w9 w10 - empleado
l1 l2 - línea
pieza-11 pieza-12 - pieza
)
(:init
  (libre w1)
  (libre w2)
  (libre w3)
  (libre w4)
  (libre w5)
  (libre w6)
  (libre w7)
  (libre w8)
  (libre w9)
  (libre w10)

  (fabricar pieza-11)
  (fabricar pieza-12)
)
(:goal
  (and
    (terminada pieza-11)
    (terminada pieza-12)
  )
)
)
```

Fragmento del dominio.

```
(define (domain produccion-capacitacion)
  (:requirements :strips :typing :negative-preconditions)
  (:types
    empleado
    linea
    pieza
  )
  (:predicates
    ;; empleado sin capacitación ni asignación
    (libre ?w - empleado)

    ;; capacitación exclusiva
    (capacitar-e1 ?w - empleado)
    (capacitar-e2 ?w - empleado)
    (capacitar-e3 ?w - empleado)

    ;; asignación permanente a línea
    (en-lineal ?w - empleado)
  )
)
```

```

(en-linea2 ?w - empleado)

;; operadores instalados
(op-e1 ?w - empleado ?l - linea)
(op-e2 ?w - empleado ?l - linea)
(op-e3 ?w - empleado ?l - linea)

;; producción
(fabricar ?p - pieza)

(en-e1 ?p - pieza)
(en-e2 ?p - pieza)
(en-e3 ?p - pieza)

(terminada ?p - pieza)
)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; CAPACITACION
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(:action capacitacion-e1

:parameters (?w - empleado)

:precondition
(and
  (libre ?w)
  (not (capacitar-e1 ?w))
  (not (capacitar-e2 ?w))
  (not (capacitar-e3 ?w))
)

:effect
(capacitar-e1 ?w)
)

(:action capacitacion-e2

:parameters (?w - empleado)

:precondition
(and
  (libre ?w)
  (not (capacitar-e1 ?w))
  (not (capacitar-e2 ?w))
  (not (capacitar-e3 ?w))
)

:effect
(capacitar-e2 ?w)
)

```

Plan generado. Modelo BFWS—FF-parser versión.

- 1-(capacitacion-e1 w10)
- 2-(asignar-linea2 w10)
- 3-(instalar-e1-l2 w10)
- 4-(producir-e1 pieza-l2 w10 l2)
- 5-(capacitacion-e2 w9)
- 6-(producir-e1 pieza-l1 w10 l2)
- 7-(asignar-linea2 w9)
- 8-(capacitacion-e3 w6)
- 9-(instalar-e2-l2 w9)
- 10-(asignar-linea1 w6)
- 11-(instalar-e3-l1 w6)
- 12-(producir-e2 pieza-l2 w9 l2)
- 13-(producir-e3 pieza-l2 w6 l1)
- 14-(terminar pieza-l2)
- 15-(producir-e2 pieza-l1 w9 l2)
- 16-(producir-e3 pieza-l1 w6 l1)
- 17-(terminar pieza-l1)

Plan de manera gráfica.

Línea	E1	E2	E3	
	C1 1-w10	C2 5-w9	C3 8-w6	
L1	E1 6-producir l1 w10	E2	E3 10-asignar w6 11-instalar w6 16-producir l1 w6	17-terminar l1
L2	E4 2-asignar w10 3-instalar w10 4-producir l2	E2 7-asignar w9 9-instalar l2 w9 12-producir l2 w9 15-producir l1 w9	E3 13-producir l2 w6	14-terminar l2

Se puede apreciar que el plan desarrollado por el planificador tiene menos pasos que el esperado por un humano, el resultado es prometedor, aun considerando solo un modelo de planificación de varios existentes. Hay que mencionar que el paso 15 no es parte de las reglas formales, sin embargo, no va en contra del objetivo final, puede considerarse incluso eliminar de las reglas formales que un mismo operador pueda estar en más de una línea de producción, esto va a depender de las condiciones reales de la fábrica.

Conclusiones.

El desarrollo de un planificador inteligente basado en PDDL permitió demostrar que la planificación automática puede emplearse eficazmente para sincronizar los procesos de capacitación del personal y producción industrial dentro de un entorno de manufactura. A partir de la modelación formal de operadores, estaciones de trabajo, líneas de producción, restricciones operativas y metas de fabricación, fue posible generar planes que garantizan que cada trabajador adquiera previamente las competencias necesarias antes de participar en las actividades productivas. Los resultados obtenidos evidencian que el uso de técnicas de planificación automática no solo facilita la representación estructurada de procesos complejos, sino que también permite encontrar secuencias de acciones más eficientes que las propuestas inicialmente de forma manual. El plan generado por el sistema logró alcanzar los objetivos de producción utilizando menos pasos

y una mejor coordinación de recursos, lo que confirma el potencial de PDDL como herramienta para apoyar la toma de decisiones en contextos industriales. Asimismo, el estudio demuestra la viabilidad de integrar conceptos de inteligencia artificial en problemas reales de manufactura, contribuyendo a mejorar la productividad, la utilización del personal y la organización de las operaciones. Como trabajo futuro, se recomienda ampliar el modelo incorporando restricciones temporales, múltiples productos, indicadores de desempeño y escenarios reales de planta que permitan evaluar la escalabilidad y robustez del planificador en ambientes industriales de mayor complejidad.

Créditos.

Los autores agradecen al Tecnológico Nacional de México por el financiamiento del proyecto 25453.26-P “Planificador inteligente para sincronizar capacitación y producción industrial basado en lenguaje de definición de dominios (PDDL)”, de la convocatoria de apoyo a proyectos de desarrollo tecnológico e innovación 2026 y las facilidades del Instituto Tecnológico de Tuxtla Gutiérrez para la realización de este trabajo.

Referencias bibliográficas.

Araújo, A., & Silva, C. A. (2016). Task scheduling in automotive harness assembly lines using automated planning. *International Journal of Production Research*, 54(12), 3500-3515.

Blum A. y Furst M. (1995). “Graphplan algorithm,” 1995.

Education.Planning.Domains (2025). A central source for educational resources for planning domains, <http://planning-domains-education.haz.ca/>

Fikes y Nilsson (1971). “Strips: A new approach to the application of theorem proving to problem solving,” 1971.

Fox, M., & Long, D. (2003). PDDL2.1: An extension to PDDL for expressive domain modeling. *Journal of Artificial Intelligence Research*, 20, 61-124.

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory and Practice*. Morgan Kaufmann. https://api.pageplace.de/preview/DT0400.9780080490519_A25022382/preview-9780080490519_A25022382.pdf.

ICAPS (2025). International Conference on Automated Planning and Scheduling, <https://www.icaps-conference.org/>.

McDermott, D. M., (1998). The 1998 AI planning systems competition, *AI magazine*, 2, pags. 35–55, 2000.

Newell y Simon (1959). “Report on a general problem-solving program,” 1959.

Información de los autores.



Diego Ramírez Hernández, futuro Ingeniero en Sistemas Computacionales (ITTG), realiza su tesis en Lenguaje de Planificación de Definición de Dominios. Actualmente cursa el último semestre de la Carrera.



Mauricio Jair Reyes González estudiante de Ingeniería en Sistemas Computacionales (ITTG) realizó su tesis en proyecto de un planificador inteligente para sincronizar capacitación y producción basado en lenguaje de definición de dominios (PDDL), actualmente labora como desarrollador backend para creación de agentes de IA funcionales para Doctores.



Héctor Guerra Crespo es egresado del I. T. de Mérida (Yucatán, México) de la carrera de Ingeniería en Sistemas Computacionales en 1994, es Doctor en Sistemas Computacionales por la Universidad del Sur (Chiapas, México) en 2011. Es profesor en el área de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez y en el área de Licenciatura en Sistemas Computacionales de la Universidad Autónoma de Chiapas, en ambas desde 1995.



Luis Tomás García Andrade, Ingeniero en Electrónica por el Instituto Tecnológico de Tuxtla Gutiérrez, Maestro en Administración de Redes y Telecomunicaciones por el Instituto de Estudios Superiores de Chiapas, Docente del Departamento de Eléctrica y Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez.



Jorge Octavio Guzmán Sánchez es docente de la carrera de Ingeniería en Sistemas Computacionales en el Tecnológico Nacional de México (TecNM), Campus Tuxtla Gutiérrez. Cuenta con una destacada trayectoria en la publicación de libros y artículos científicos, así como en el desarrollo de proyectos de investigación con y sin financiamiento institucional. Además de su labor docente, se desempeña como representante institucional ante el PRODEP, funge como asesor de estudiantes en competencias académicas y participa activamente en la gestión administrativa de la institución.



Elfer Isaías Clemente Camacho es Ingeniero en sistemas computacionales por el Tecnológico de Tuxtla Gutiérrez, cuenta con Maestría en desarrollo de software por el Instituto de Estudios Superiores de Chiapas. Doctor en Tecnología Educativa por la Universidad del País INNOVA. Docente desde 2019, en la Carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Tuxtla Gutiérrez. Docente en la Universidad Autónoma de Chiapas desde 2013. Docente en la Universidad Pablo Guardado Chávez desde 2020. Profesor con Perfil Deseable (PRODEP) desde 2025.