

**INSTITUTO DE INVESTIGACIÓN
CIENTÍFICA, CONTABLE Y FINANCIERA**
FACULTAD DE CIENCIAS CONTABLES, AUDITORÍA
SISTEMAS DE CONTROL DE GESTIÓN Y FINANZAS

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

NÚMERO DE GRUPO

34



**XIV FERIA
FACULTATIVA**
DE EMPRENDEDURISMO INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA

NOMBRE DEL PROYECTO: ALGORITMO DE PLANIFICACION DE SHORTEST JOB FIRST (SJF)

CATEGORÍA
EXPOSICIÓN

INTEGRANTES

LIMA YAHUITA CINTHIA ARIANY

MOSCOZO VERA KATTY LIZETH

TORREZ MARAÑÓN DANNA CAMILA

ZEBALLOS CABRERA JORGE ANDRES

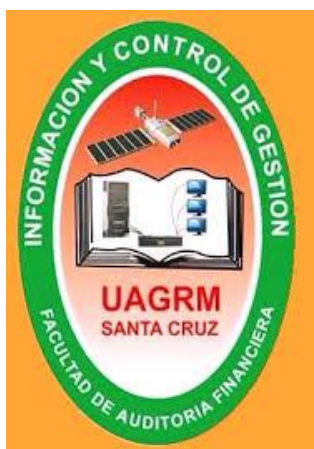
DOCENTE GUIA

PEINADO PEREIRA JUAN CARLOS



**FACULTAD DE CIENCIAS CONTABLES, AUDITORIA SISTEMAS DE
CONTROL DE GESTION Y FINANZAS**
UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO

FERIA DE EXPOSICIÓN



ALGORITMO DE PLANIFICACIÓN
SHORTEST JOB FIRST (SJF)

INTEGRANTES:

- Cinthia Ariany Lima Yahuita
- Katty Lizeth Moscoso Vera
- Danna Camila Torrez Marañón
- Jorge Andres Zeballos Cabrera

REGISTRO

217026613
221012532
223128414
214140105

Materia: Sistema Operativo

Docente: Peinado Pereira Juan Carlos

Grupo: "SIF 150 - M"

SANTA CRUZ - BOLIVIA
2025

DEDICATORIA

El presente proyecto de exposición estará dirigido a todos los estudiantes de la Universidad Autónoma Gabriel Rene Moreno y otras.

De igual manera a los empresarios, docentes del nivel superior, inversionistas y al público interesado en el proyecto.

AGRADECIMIENTO

A Dios porque él nos da la vida, la salud y por mantenernos firmes en nuestras metas académicas, para que de esa manera podamos lograr nuestros objetivos trazados.

A nuestros padres por la confianza y el apoyo incondicional, que nos inspira a continuar en el camino del aprendizaje y a la vez por inculcarnos principios y valores que dirigen nuestras vidas.

A Licenciado Juan Carlos Peinado por las enseñanzas, consejos y por compartir sus experiencias que nos servirán de mucho en nuestra vida laboral.

INDICE

1. INTRODUCCION	Pág. 1
1.1. Presentación del tema.....	Pág. 1
2. JUSTIFICACIÓN	Pág. 2
3. OBJETIVOS.....	Pág. 3
3.1. Objetivos General	Pág. 3
3.2. Objetivos Específicos	Pág. 3
4. DESARROLLO DEL TEMA.....	Pág. 4
4.1. Antecedentes / Contexto	Pág. 4
4.2. Conceptos claves	Pág.4
4.3. Análisis / Desarrollo del contenido	Pág.5
4.4. Problemática.....	Pág.6
4.5. Fundamentación del Algoritmo Shortest Job First (SJF)	Pág.7
4.6. Solución.....	Pág.7
4.7. Datos relevantes, ejemplos o casos prácticos	Pág.10
5. VENTAJAS Y DESVENTAJAS.....	Pág.11
5.1. Ventajas	Pág.11
5.2. Desventajas	Pág.11
6. COMPARACION DE SJF CON OTROS ALGORITMOS EN ESTE PROBLEMA.....	Pág.12
7. DIFERENCIAS CLAVE DE SJF CON LOS OTROS MÉTODOS EN ESTE PROBLEMA.....	Pág.12
7.1. SJF vs FCFS	Pág.12
7.2. SJF vs PRIORIDAD	Pág.13
7.3. SJF vs ROUND ROBIN	Pág.13
7.4. SJF vs COLAS MULTINIVEL.....	Pág.13
8. CONCLUSION COMPARATIVA.....	Pág.13
9. CONCLUSION Y RECOMENDACIÓN	Pág.14
10.RECOMENDACIÓN	Pág.15
11.BIBLIOGRAFIAS	Pág.16

1. INTRODUCCIÓN

1.1. Presentación del tema

El presente trabajo aborda los desafíos de planificación en sistemas de entrega a demanda, tomando como referencia la plataforma **Pedidos Ya**. Estos servicios requieren una organización eficiente para garantizar entregas rápidas, justas y económicamente sostenibles.

En el mundo del delivery, la eficiencia en la asignación y gestión de pedidos es crucial para la satisfacción del cliente y la rentabilidad de las plataformas. Este trabajo explora los **algoritmos de planificación de Shortest Job First (SJF)**, específicamente aplicados al contexto de **Pedidos Ya**, una de las principales aplicaciones de delivery en la región. Se analizará cómo estos algoritmos optimizan la distribución de pedidos a los repartidores disponibles, minimizando tiempos de espera y maximizando la capacidad operativa.

SJF (Shortest Job First) es un algoritmo de planificación que ejecuta primero la tarea con **menor duración estimada**.

En el contexto de delivery, la "duración" puede interpretarse como el **tiempo estimado de entrega** (considerando distancia, tráfico, etc.).

El problema que resolveremos es:

La empresa **PEDIDOS YA**, un servicio de delivery local en Santa Cruz, enfrenta problemas operativos en su logística de entregas múltiples.

2. JUSTIFICACIÓN

La eficiencia en la asignación de pedidos y rutas es crucial para satisfacer a los clientes, optimizar recursos y mantener la competitividad. Una mala planificación puede provocar demoras, pérdidas económicas y deterioro de la reputación de la empresa. Este análisis permite explorar qué algoritmos de planificación pueden adaptarse mejor al contexto logístico actual.

¿Por qué es importante tratar este tema?

El algoritmo de planificación tiene un impacto directo en la eficiencia operativa y la satisfacción del cliente. Tratar el SJF es importante porque:

- **Minimización del tiempo de espera y entrega:** El SJF está teóricamente diseñado para reducir el tiempo de respuesta promedio de las tareas. Aplicado al delivery, esto significa clientes que reciben sus pedidos más rápido.
- **Aumento de la productividad del repartidor:** Al enfocar la atención en pedidos rápidos, los repartidores pueden completar más entregas en un período determinado.
- **Optimización de recursos:** Una asignación eficiente de pedidos basada en la duración estimada puede llevar a un uso más eficaz de la flota de repartidores.
- **Reto de implementación:** Aunque atractivo en teoría, la implementación práctica del SJF en un entorno dinámico como el delivery presenta desafíos únicos que merecen ser explorados.
- **Optimización continua:** El proceso es dinámico y se reevalúa constantemente a medida que nuevos pedidos entran, repartidores se liberan o las condiciones cambian (tráfico, etc.).
- **Asignación:** Una vez tomada la decisión, el algoritmo asigna el pedido al repartidor y actualiza el estado del pedido y del repartidor.

3. OBJETIVOS

3.1. Objetivos General:

- Comprender el funcionamiento y la aplicación de los algoritmos de planificación de Shortest Job First (SJF) en la optimización de la gestión de pedidos de delivery en plataformas como Pedidos Ya.

3.2. Objetivos específicos:

- Identificar los conceptos clave asociados a la planificación de SJF y su relevancia en el contexto de Pedidos Ya.
- Identificar ventajas y desventajas de cada uno en el contexto de entregas a domicilio.
- Proponer una solución adecuada basada en criterios de eficiencia y realismo.
- Analizar los desafíos y problemáticas inherentes a la implementación de algoritmos de planificación de SJF en un entorno de delivery.
- Explorar posibles soluciones y estrategias para mejorar la eficiencia de la asignación de pedidos utilizando estos algoritmos.
- Presentar ejemplos o casos prácticos que ilustren la aplicación y los beneficios de dichos algoritmos.

4. DESARROLLO DEL TEMA

4.1. Antecedentes / Contexto

La plataforma de delivery como Pedidos Ya ha transformado la forma en que los consumidores acceden a bienes y servicios. Detrás de la aparente simplicidad de pedir comida o productos desde una aplicación, existe una compleja red logística que debe gestionar miles de pedidos simultáneamente. Históricamente, la asignación de pedidos se realizaba de forma manual o con algoritmos rudimentarios, lo que llevaba a ineficiencias. Con el crecimiento exponencial del sector, la necesidad de algoritmos de planificación sofisticados se ha vuelto imperativa para manejar la escala y la complejidad de las operaciones.

Pedidos Ya es una plataforma de entrega a domicilio que conecta usuarios, restaurantes y repartidores. El proceso de entrega debe considerar factores como ubicación, tráfico, volumen de pedidos y tiempos de entrega. La planificación es esencial para decidir qué pedido debe ser entregado, por quién y en qué orden.

4.2. Conceptos clave

- SJF (Shortest Job First) es un algoritmo de planificación que ejecuta primero la tarea con **menor duración estimada**.
- En el contexto de delivery, la "duración" puede interpretarse como el **tiempo estimado de entrega** (considerando distancia, tráfico, etc.).
- **Planificación de procesos:** Técnicas que permiten organizar tareas (en este caso, pedidos) para maximizar eficiencia.
- **Algoritmos clásicos:**
 - FCFS: Atiende pedidos por orden de llegada.
 - SJF: Atiende primero los pedidos que requieren menos tiempo.
 - Prioridad: Atiende según nivel de importancia.
 - Round Robin: Atiende por turnos iguales.

- Colas Multinivel: Clasifica tareas en distintas colas y aplica un algoritmo distinto a cada una.

4.3. Análisis / Desarrollo del contenido

Los algoritmos de planificación de Shortest Job First (SJF) en Pedidos Ya buscan resolver el **problema de asignación de pedidos y enrutamiento de vehículos**. El objetivo es asignar de manera óptima los pedidos que llegan a la plataforma a los repartidores disponibles, considerando múltiples factores para minimizar costos y tiempos, y maximizar la satisfacción del cliente.

El Algoritmo SJF para Pedidos Ya operaría de la siguiente manera:

- **Ingreso de Pedidos:** Cada nuevo pedido entra en una cola, y el sistema calcula una **duración estimada** para cada uno. Esta estimación es clave y se basa en datos históricos (tiempos de preparación del restaurante, tiempos de viaje promedio en rutas similares), información en tiempo real (tráfico) y la ubicación del repartidor más cercano.
- **Mantenimiento de la Cola:** La cola de pedidos se mantiene ordenada por la duración estimada del "job" (del más corto al más largo).
- **Asignación de Repartidores:** Cuando un repartidor está disponible (liberado de un pedido anterior o recién conectado), el algoritmo le asigna el pedido **disponible y más corto** de la cola que el repartidor pueda atender.
- **Consideraciones Adicionales:**
 - **Proximidad del Repartidor:** Aunque el SJF prioriza la duración, el algoritmo también debe considerar la distancia del repartidor al restaurante para asegurar que la "duración estimada" siga siendo válida. Un pedido corto lejos de un repartidor no es realmente "corto" para ese repartidor.
 - **Pedidos Múltiples (Batching):** El SJF puede combinarse con estrategias de batching. Si un repartidor puede llevar dos pedidos, el

algoritmo podría buscar la combinación de "jobs" que, en conjunto, representen la duración total más corta o la ruta más eficiente.

- **Vencimiento de Pedidos:** Aún con SJF, los pedidos con una fecha de vencimiento crítica (ej. comida fría si no se entrega a tiempo) pueden necesitar una "prioridad" forzada para evitar que se queden atascados si hay muchos pedidos cortos.
- **Optimización continua:** El proceso es dinámico y se reevalúa constantemente a medida que nuevos pedidos entran, repartidores se liberan o las condiciones cambian (tráfico, etc.).

4.4. Problemática

La empresa **PEDIDOS YA**, un servicio de delivery local en Santa Cruz, enfrenta problemas operativos en su logística de entregas múltiples. Los pedidos son asignados sin un orden estratégico, simplemente por orden de llegada o ubicación aproximada, lo cual genera:

- Retrasos en entregas cortas y fáciles.
- Repartidores que recorren más distancia innecesariamente.
- Clientes molestos porque sus pedidos pequeños llegan tarde.
- Mala reputación en redes sociales y menor fidelización de clientes.

El sistema actual **no toma en cuenta el tiempo estimado de entrega de cada pedido**, lo que afecta tanto a los clientes como a los repartidores.

- Algunos pedidos urgentes o de clientes prioritarios pueden quedar desatendidos si se usa solo FCFS.
- Repartidores pueden estar mal asignados si no se consideran distancias o prioridades.
- Mala planificación genera sobrecarga en algunos repartidores y ociosidad en otros.

4.5. Fundamentación del Algoritmo Shortest Job First (SJF)

El algoritmo **Shortest Job First (SJF)** resuelve esta situación porque:

- Atiende primero los pedidos que se pueden entregar en menos tiempo.
- Reduce el tiempo promedio total de entrega.
- Aumenta la satisfacción del cliente, ya que las órdenes rápidas se despachan primero.
- Optimiza el uso de las motos o bicicletas al evitar rutas extensas innecesarias.
- Mejora la imagen del negocio, ya que se ve como más “rápido”.

En un entorno de delivery con varios pedidos en espera, este algoritmo permite priorizar estratégicamente **entregas más rápidas** sin dejar de lado las más largas (que se irán atendiendo progresivamente).

4.6. Solución

Un repartidor de FLASH tiene 5 pedidos pendientes por entregar. Cada uno tiene un **tiempo estimado de entrega**, dependiendo de la distancia, dificultad del lugar y tipo de pedido (paquete, comida caliente, documento, etc.).

APROPIADO

PEDIDO	CPU (T. DE ENTREGA min)	T. LLEGADA (hora)	T. ESPERA	T. RETORNO	T. RESPUESTA
PEDIDO A	10	11:00	3	3	13
PEDIDO B	8	12:00	12	12	20
PEDIDO C	14	10:00	10	10	4
PEDIDO D	6	13:00	19	19	25
PROMEDIO					15.5

C	C	C	C	C	C	C	A	A	A	A	A	B	B	B	B	D	D	D	
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38

T. Retorno = 1 Ejecución - T. Llegada

PEDIDO A = 14 - 11 = 3

PEDIDO B = 24 - 12 = 12

PEDIDO C = 0 - 10 = 10

PEDIDO D = 32 - 13 = 19

T. Respuesta = Ult. Ejecución - T. Llegada

PEDIDO A = 24 - 11 = 13

PEDIDO B = 32 - 12 = 20

PEDIDO C = 14 - 10 = 4

PEDIDO D = 38 - 13 = 25

total = 62

Comparación con el orden original (por llegada):

- Sin algoritmo: Pedido A, Pedido B, Pedido C, Pedido D.
- Con algoritmo de Shortest Job First (SJF Apropriativo): Pedido C, Pedido A, Pedido B, Pedido D.

NO APROPIADO

PEDIDO	CPU (T. DE ENTREGA min)	T. LLEGADA (hora)	T. ESPERA	T. RETORNO	T. RESPUESTA
PEDIDO A	10	11:00	3	17	27
PEDIDO B	8	12:00	12	8	16
PEDIDO C	14	10:00	10	10	4
PEDIDO D	6	13:00	19	1	7
PROMEDIO					13.5

C	C	C	C	C	C	C	D	D	D	B	B	B	B	A	A	A	A	A	
0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38

T. Retorno	=	1 Ejecución	-	T. Llegada	
PEDIDO A	=	28	-	11	= 17
PEDIDO B	=	20	-	12	= 8
PEDIDO C	=	0	-	10	= 10
PEDIDO D	=	14	-	13	= 1

T. Respuesta	=	Ult. Ejecución	-	T. Llegada	
PEDIDO A	=	38	-	11	= 27
PEDIDO B	=	28	-	12	= 16
PEDIDO C	=	14	-	10	= 4
PEDIDO D	=	20	-	13	= 7
total					= 54

Comparación con el orden original (por llegada):

- Sin algoritmo: Pedido A, Pedido B, Pedido C, Pedido D.

- Con algoritmo de Shortest Job First (SJF No Apropiativo): Pedido C, Pedido D, Pedido B, Pedido A.

La solución a estos desafíos reside en el diseño e implementación de **algoritmos de planificación avanzados** que incorporen elementos de inteligencia artificial y aprendizaje automático.

4.7. Datos relevantes, ejemplos o casos prácticos

- **Análisis de Datos Históricos:** Pedidos Ya recopilaría datos sobre el tiempo promedio de preparación de cada restaurante, la duración de viajes entre diferentes puntos de la ciudad en distintas horas del día y la velocidad promedio de los repartidores. Estos datos serían la base para la "duración estimada del job".
- **Ejemplo de Estimación:** Un pedido de un café de una tienda a 5 minutos del repartidor, con un tiempo de preparación de 2 minutos y 3 minutos de viaje al cliente, se consideraría un "job" de ~10 minutos. Un pedido de una pizzería a 15 minutos, con 20 minutos de preparación y 10 minutos al cliente, sería un "job" de ~45 minutos. El algoritmo SJF priorizaría el café.
- **Caso de Inanición (Problemática):** Si constantemente llegan pedidos de café de 10 minutos, un pedido de pizza de 45 minutos podría esperar una hora o más en la cola, lo que sería inaceptable para el cliente. Aquí es donde el "envejecimiento" o la "prioridad forzada" entran en juego.
- **Implementación Híbrida (Solución):** En lugar de un SJF puro, Pedidos Ya podría usar un algoritmo que, cada 5 minutos, primero revise si hay pedidos a punto de exceder su tiempo límite de entrega, y luego aplique SJF a los pedidos restantes que no están en riesgo.

5. VENTAJAS Y DESVENTAJAS:

5.1. Ventajas:

- **Reducción del tiempo promedio de entrega:** Al atender primero los pedidos con menor tiempo estimado, se optimiza el tiempo total del sistema.
- **Mayor eficiencia para pedidos cortos:** los pedidos cercanos o de preparación rápida llegan más rápido, lo cual mejora la experiencia de muchos usuarios.
- **Aprovechamiento del tiempo del repartidor:** Se evitan demoras largas por pedidos complejos, manteniendo al repartidor en movimiento constante.
- **Simplicidad en la lógica de programación:** Es fácil de implementar si se tiene un sistema que calcule tiempos estimados de entrega con precisión.

5.2. Desventajas:

- **Injusticia para pedidos lejanos o grandes:** Pedidos con tiempos largos pueden quedar “esperando” indefinidamente si siempre hay pedidos más cortos (problema de inanición).
- **No considera urgencia ni prioridad del cliente:** Un pedido de un cliente VIP o urgente puede ser ignorado si otro pedido es más corto.
- **Requiere estimaciones muy precisas:** Si el cálculo del “tiempo estimado de entrega” es incorrecto, puede causar errores en la asignación.
- **No es reactivo a cambios en tiempo real:** si el tráfico cambia, el sistema no se adapta automáticamente, ya que SJF depende de valores iniciales.

6. COMPARACION DE SJF CON OTROS ALGORITMOS EN ESTE PROBLEMA

Algoritmo	¿Cómo funciona?	Ventajas en delivery	Desventajas en delivery
FCFS (First Come First Serve)	Atiende en orden de llegada	Simple y justo en teoría	Ignora urgencia, puede generar largas esperas
SJF (Shortest Job First)	Atiende pedidos con menor tiempo estimado	Reduce tiempo promedio de entrega	Injusto con pedidos largos o lejanos
PRIORIDAD	Atiende pedidos con mayor prioridad asignada	Considera urgencia y tipo de cliente	Puede causar inanición de pedidos de baja prioridad
ROUND ROBIN	Atiende por turnos iguales en tiempo (cuotas)	Justo para sistemas compartidos	Poco útil en delivery físico, genera muchas pausas
COLAS MULTINIVEL	Clasifica pedidos por tipo y aplica diferentes algoritmos	Flexible, realista, combina lo mejor de varios métodos	Complejo de implementar y mantener

7. DIFERENCIAS CLAVE DE SJF CON LOS OTROS MÉTODOS EN ESTE PROBLEMA

7.1. SJF vs FCFS

- **SJF** es más eficiente porque no espera innecesariamente por pedidos largos.
- **FCFS** puede ser justo pero lento si un pedido largo llega antes que otros rápidos.

7.2. SJF vs PRIORIDAD

- **SJF** se basa en tiempo estimado de entrega, no en importancia del pedido.
- **Prioridad** puede hacer que se atienda un pedido urgente aunque tarde más en completarse.

7.3. SJF vs ROUND ROBIN

- **SJF** se adapta mejor al delivery porque se enfoca en eficiencia.
- **Round Robin** no tiene sentido práctico aquí porque no se pueden "dividir" entregas por turnos.

7.4. SJF vs COLAS MULTINIVEL

- **Colas multinivel** puede combinar **SJF dentro de una de las colas**, pero también puede aplicar prioridad o FCFS según el tipo de pedido.
- Es más flexible, pero requiere más diseño y control.

8. CONCLUSION COMPARATIVA

En resumen...	¿Es bueno para delivery?	¿Requiere adaptación?
SJF	Sí, pero con precaución	Sí (evitar inanición)
FCFS	No recomendable solo	No
Prioridad	Sí, muy realista	Sí (definir criterios)
Round Robin	No útil en delivery real	No
Colas Multinivel	La mejor opción combinada	Sí (más complejo)

9. CONCLUSION

El algoritmo **SJF aplicado al servicio de delivery Pedidos Ya** demostró ser una solución práctica y efectiva para mejorar la eficiencia del reparto. Al priorizar los pedidos más rápidos:

- Se entrega más en menos tiempo.
- Se mejora la experiencia del cliente.
- Se optimiza el uso del personal de reparto.
- Se generan mejores opiniones en redes y más fidelidad.

Es una herramienta útil para quienes trabajan en gestión operativa, logística o administración.

El **Algoritmo de Planificación SJF** ofrece un enfoque atractivo para maximizar la cantidad de entregas y reducir el tiempo de respuesta promedio en el delivery. Sin embargo, su aplicación directa presenta desafíos significativos, principalmente el riesgo de inanición para pedidos más largos y la necesidad de una estimación precisa de la duración de los "jobs".

10. RECOMENDACIÓN

- Usar SJF en combinación con sistemas de prioridad.
- Establecer límites máximos de espera para evitar inanición
- Invertir en un buen sistema de estimación de tiempos de entrega para que SJF sea efectivo
- **Adoptar un SJF Híbrido:** Es crucial no implementar un SJF puro, sino una variante modificada que incluya mecanismos de envejecimiento, prioridades de tiempo de entrega y, posiblemente, clasificaciones de pedidos para mitigar la inanición.
- **Invertir en Modelos de Predicción de Duración:** La precisión de la "duración estimada del job" es el factor crítico. Se debe invertir en modelos de aprendizaje automático robustos que utilicen datos en tiempo real (tráfico, preparación del restaurante) y datos históricos para hacer estas estimaciones lo más precisas posible.
- **Monitoreo Constante y Adaptación:** El algoritmo debe ser dinámico, con la capacidad de recalcular las duraciones estimadas y reajustar las prioridades a medida que cambian las condiciones del mundo real.
- **Considerar la Experiencia del Repartidor:** Implementar métricas que aseguren una distribución justa de pedidos entre los repartidores, evitando que algunos queden atrapados con solo "jobs largos" y potencialmente menos rentables.
- **Balancear Eficiencia y Equidad:** Buscar un equilibrio entre la eficiencia global del sistema (minimizada por SJF) y la equidad para todos los clientes y repartidores.

11. BIBLIOGRAFIA

- Daganzo, C. F. (2007). *Logistics Systems Analysis*. Springer. (Para principios de sistemas logísticos y enrutamiento).
- Laporte, G., & Semet, F. (2002). Routing problems in transportation networks. *Transportation Research Part B: Methodological*, 36(6), 461-482. (Para la complejidad del enrutamiento).
- Liu, Z., Niu, X., & Zhang, J. (2018). On-demand food delivery: An operations management perspective. *International Journal of Operations Research and Information Systems (IJORIS)*, 9(4), 1-18. (Para una visión general de la gestión de operaciones en el delivery).
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson. (Para conceptos de planificación, heurísticas y machine learning).
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems*. Pearson. (Para la teoría de algoritmos de planificación de CPU como SJF, que puede ser adaptada).
- Información pública de Pedidos Ya (sitio web, comunicados de prensa sobre operaciones y tecnología).
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2020). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.
- Pedidos Ya. (s.f.). *¿Cómo funciona Pedidos Ya?* Recuperado de <https://www.pedidosya.com>
- Kumar, S. (2014). *Scheduling Algorithms in Real Time Systems*. *International Journal of Computer Applications*.