



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

## Universidad Nacional de Colombia

Ingeniería de Sistemas y Computación

2025969 Modelos estocásticos y simulación en computación y comunicaciones (01)

### Tarea 6

El problema de la programación académica en la UNAL

### Integrantes:

Javier Andrés Tarazona Jiménez	jtarazonaj@unal.edu.co
Yenifer	@unal.edu.co
Jefferson	@unal.edu.co

Abril 15 de 2024

# Tabla de Contenidos

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Marco Teórico</b>	<b>3</b>
2.1	Ejemplo Documentación . . . . .	3
2.1.1	Contextualización de la solución . . . . .	3
2.1.2	Descripción general de la solución . . . . .	3
2.1.3	Naturaleza de la solución . . . . .	3
2.1.4	Funcionamiento del sistema . . . . .	3
2.1.5	Resultados o impacto . . . . .	3
2.1.6	Referencia . . . . .	3
2.2	Bundled Probabilistic Serial (BPS) . . . . .	4
2.2.1	Contextualización del problema y la solución . . . . .	4
2.2.2	Descripción general de la solución . . . . .	4
2.2.3	Naturaleza de la solución . . . . .	4
2.2.4	Funcionamiento del sistema . . . . .	4
2.2.5	Propiedades adicionales y consideraciones . . . . .	5
2.2.6	Resultados del proceso . . . . .	5
2.2.7	Impacto . . . . .	5
2.2.8	Referencia . . . . .	5
2.3	Random Serial Dictatorship (RSD): Jefferson . . . . .	6
2.4	Probabilistic Serial Rule (PBS) . . . . .	6
2.4.1	Descripción general de la solución . . . . .	6
2.4.2	Naturaleza de la solución . . . . .	6
2.4.3	Funcionamiento del sistema . . . . .	6
2.4.4	Referencia . . . . .	7
2.5	Asignación de Cursos Universitarios mediante Mercados Competitivos . . . . .	7
2.5.1	Contextualización de la solución . . . . .	7
2.5.2	Descripción General de la Solución . . . . .	8
2.5.3	Naturaleza de la Solución . . . . .	8
2.5.4	Funcionamiento del Sistema . . . . .	8
2.5.5	Resultados o Impacto . . . . .	9
2.5.6	Referencia . . . . .	9
<b>3</b>	<b>Descripción y Justificación del Problema a Resolver</b>	<b>9</b>
3.1	Objetivo Principal . . . . .	10
<b>4</b>	<b>Diseño de la solución</b>	<b>10</b>
4.1	Metodología . . . . .	10
4.1.1	Preparación . . . . .	10
4.1.2	Modelo Estocástico: Proceso de Asignación . . . . .	12
4.1.3	Métricas . . . . .	13
<b>5</b>	<b>Código Fuente</b>	<b>14</b>
<b>6</b>	<b>Manual Usuario</b>	<b>14</b>
<b>7</b>	<b>Manual Técnico</b>	<b>15</b>
<b>8</b>	<b>Experimentación</b>	<b>15</b>
8.1	Análisis de resultados . . . . .	15
8.1.1	Escenario 1 . . . . .	15
8.1.2	Escenario 2 . . . . .	15
8.1.3	Escenario 3 . . . . .	15



# 1 Introducción

Aquí si empieza el contenido del artículo de la sección 1 y también hay una referencia [2].

## 2 Marco Teórico

### 2.1 Ejemplo Documentación

#### 2.1.1 Contextualización de la solución

Problema identificado: ¿Qué necesidad o dificultad buscaba resolver la universidad? (por ejemplo: congestión en cursos, planificación ineficiente, asignación no equitativa, etc.)

Entorno institucional: ¿Qué tipo de universidad es? ¿Cuántos estudiantes? ¿Qué tan complejo es su sistema académico?

Objetivos del sistema de asignación: (eficiencia, justicia, reducción de tiempos administrativos, etc.)

#### 2.1.2 Descripción general de la solución

Nombre o enfoque de la solución: (por ejemplo: algoritmo de colonia de hormigas, programación entera, RSD, etc.)

Tipo de herramienta: ¿Es una herramienta automatizada, un modelo matemático, un sistema web, un algoritmo implementado?

Actores involucrados: (¿Está orientado a estudiantes, administrativos, docentes?)

#### 2.1.3 Naturaleza de la solución

Aquí se refiere a la categoría técnica o metodológica de la solución:

¿Es un modelo de optimización matemática, una heurística, un sistema probabilístico, una metaheurística, un sistema de recomendación?

También puede implicar si es una solución centralizada (administración asigna) o descentralizada (los usuarios eligen con base en preferencias).

#### 2.1.4 Funcionamiento del sistema

¿Cómo se realiza la asignación o planificación?

¿Qué datos se necesitan?

¿Cómo se expresan las preferencias?

¿Qué lógica sigue el algoritmo (en términos generales)?

¿Qué tan automático es?

¿Cómo se gestionan los conflictos o empates?

#### 2.1.5 Resultados o impacto

¿Qué mejoras reportaron?

¿Se adoptó oficialmente?

¿Se continúa utilizando?

#### 2.1.6 Referencia

Nombre del artículo o proyecto, universidad, y si es posible, enlace o fuente bibliográfica.

## 2.2 Bundled Probabilistic Serial (BPS)

### 2.2.1 Contextualización del problema y la solución

La asignación de horarios de cursos en instituciones educativas es un desafío complejo, especialmente cuando los estudiantes tienen preferencias sobre combinaciones de cursos que se ajusten a sus necesidades y restricciones personales. Los métodos tradicionales, como el orden de llegada (First-Come, First-Served - FCFS), suelen resultar en asignaciones ineficientes y generan envidia entre los estudiantes debido a la falta de equidad en el proceso. Para abordar estas limitaciones, se ha propuesto el mecanismo Bundled Probabilistic Serial (BPS), que busca asignar cursos de manera más justa y eficiente, considerando las preferencias de los estudiantes sobre paquetes de cursos.

### 2.2.2 Descripción general de la solución

El BPS es un mecanismo de asignación aleatoria que permite a los estudiantes expresar sus preferencias sobre combinaciones de cursos (paquetes) en lugar de cursos individuales. Este enfoque reconoce que los estudiantes valoran ciertos conjuntos de cursos que se ajustan a sus horarios y objetivos académicos. El BPS asigna probabilidades a cada estudiante para recibir diferentes paquetes de cursos, basándose en sus preferencias declaradas. Posteriormente, se realiza una asignación determinista mediante una lotería que respeta estas probabilidades, garantizando una distribución equitativa y eficiente de los cursos disponibles.

### 2.2.3 Naturaleza de la solución

El BPS es un mecanismo de naturaleza estocástica, ya que asigna probabilidades a los estudiantes para recibir ciertos paquetes de cursos. Sin embargo, la implementación final es determinista, ya que se utiliza una lotería para realizar la asignación concreta de cursos a cada estudiante. Este enfoque combina la equidad y eficiencia de los mecanismos aleatorios con la necesidad de asignaciones deterministas en la práctica. Además, el BPS satisface propiedades deseables como la eficiencia ordinal, la equidad (ausencia de envidia) y una forma débil de resistencia a la manipulación estratégica.

### 2.2.4 Funcionamiento del sistema

El algoritmo *Bundled Probabilistic Serial* (BPS) está diseñado para abordar el problema de asignación de paquetes de cursos, donde cada paquete representa una combinación de recursos indivisibles (por ejemplo, horarios de cursos, laboratorios o seminarios). A continuación, se describe detalladamente el proceso de asignación mediante BPS:

- **Elicitación de preferencias:** Los estudiantes deben expresar sus preferencias sobre los paquetes de cursos. Este proceso puede ser complejo, ya que el número de paquetes posibles crece exponencialmente con la cantidad de cursos disponibles.

Para simplificar la tarea, el sistema proporciona una interfaz que permite clasificar paquetes en función de atributos clave (como horarios compatibles, cursos obligatorios y electivos preferidos). Además, se pueden emplear técnicas de reducción dimensional, como la agrupación de cursos en categorías o la limitación del número de opciones permitidas.

- **Consumo simultáneo de paquetes:** El mecanismo se basa en un enfoque similar al proceso de “simultaneous eating” del algoritmo *Probabilistic Serial Rule* (PS).

Durante un intervalo continuo de tiempo, todos los estudiantes comienzan a “consumir” su paquete de cursos preferido al mismo ritmo. Cuando un paquete alcanza su capacidad (por ejemplo, si uno de sus cursos se llena), el estudiante detiene su consumo de dicho paquete y pasa al siguiente en su lista de preferencias. Este proceso se repite hasta que se agotan todas las capacidades disponibles en la oferta académica.

- **Asignación de probabilidades:** Al finalizar el proceso de consumo, cada estudiante acumula una fracción de probabilidad asociada a cada paquete consumido. Estas probabilidades representan el tiempo relativo dedicado a consumir cada paquete antes de que dejara de estar disponible.

- **Descomposición en asignaciones deterministas:** Una vez calculadas las probabilidades, el sistema convierte estas fracciones en una asignación determinista a través de un proceso de descomposición de matriz probabilística.
  - Cada fila de la matriz representa a un estudiante, y cada columna a un paquete de cursos.
  - Los valores en la matriz indican la probabilidad acumulada de que un estudiante reciba un paquete específico.
  - Para transformar esta matriz en asignaciones concretas, se utiliza el algoritmo de Birkhoff–von Neumann, que permite descomponer la matriz en una combinación convexa de matrices de permutación.
- **Lotería para asignación final:** Con base en la descomposición anterior, se realiza una lotería que asigna a cada estudiante un paquete de forma determinista, respetando las probabilidades acumuladas. Este procedimiento garantiza una asignación factible, acorde con las capacidades y preferencias declaradas por los estudiantes.

### 2.2.5 Propiedades adicionales y consideraciones

- **Velocidad uniforme:** En el proceso de consumo simultáneo, todos los estudiantes consumen a la misma velocidad. No obstante, esta puede ajustarse si se incorporan factores de prioridad (por ejemplo, año académico, urgencia de graduación, etc.).
- **Eficiencia y justicia:** BPS garantiza eficiencia ordinal, es decir, no se puede mejorar la asignación de un estudiante sin perjudicar a otro. Asimismo, promueve una distribución justa basada en preferencias declaradas.
- **Resistencia a la manipulación:** Aunque BPS no es completamente estrategia-proof, los incentivos para manipular preferencias son bajos. Las estrategias subóptimas tienden a resultar en asignaciones menos favorables, especialmente en comparación con mecanismos como RSD.

### 2.2.6 Resultados del proceso

Al concluir el proceso de asignación, se observan los siguientes resultados:

- Cada estudiante recibe una asignación determinista que refleja sus preferencias y cumple con las restricciones de capacidad.
- Los paquetes asignados optimizan el uso de los recursos académicos, minimizando cursos no utilizados y asignaciones ineficientes.
- Estudios realizados en la Universidad Técnica de Múnich demuestran que BPS mejora significativamente la equidad y la satisfacción estudiantil frente a métodos tradicionales como *first-come, first-served* (FCFS).

### 2.2.7 Impacto

La implementación del BPS en la Universidad Técnica de Múnich ha demostrado mejoras significativas en comparación con métodos tradicionales como FCFS. Se observaron asignaciones más equitativas, una mayor satisfacción de los estudiantes y una reducción en la envidia entre los participantes. Además, el BPS permitió una mejor utilización de la capacidad de los cursos y facilitó la gestión de las preferencias complejas de los estudiantes sobre combinaciones de cursos.

### 2.2.8 Referencia

Vease [1].

## 2.3 Random Serial Dictatorship (RSD): Jefferson

## 2.4 Probabilistic Serial Rule (PBS)

### 2.4.1 Descripción general de la solución

Propuesto por Bogomolnaia y Moulin (2001), el algoritmo Probabilistic Serial Rule, también llamado Probabilistic Serial Mechanism, es un mecanismo de asignación aleatoria para repartir bienes indivisibles (como cupos en asignaturas) respetando las preferencias de los participantes bajo un enfoque equitativo.

Algunos entornos donde el mismo se puede aplicar se encuentran en la asignación de cursos a estudiantes, como en el caso de la Universidad de París y en la Universidad de Tsinghua, reparto de turnos o recursos indivisibles, y asignación de cupos de movilidad internacional o servicios públicos.

Se podría decir entonces que se usa cuando un grupo de personas desea obtener uno (o varios) recursos limitados y los mismos no se pueden dividir, es decir, son discretos. Y se quiere una asignación que sea justa y eficiente en promedio, sin usar dinero ni intercambios, es decir, una solución equitativa que sea directa.

### 2.4.2 Naturaleza de la solución

Este algoritmo tiene una intuición muy práctica. Suponga un proceso en el cual se encuentran participantes de un buffet. En el mismo, cada uno empieza a comer de forma simultánea y de forma continua el plato que desee comer, es decir, su preferido. Una vez que dicho plato termina, los integrantes pasan a comer su segundo plato preferido y así sucesivamente hasta que se terminan los platos.

Esta solución cuenta con unas ventajas y otras desventajas.

#### Ventajas

- El método es eficiente, ya que maximiza el bienestar de los integrantes según preferencias. Ellos mismos buscan el "orden".
- Es justo y equitativo, ya que ningún integrante tiene prioridades.
- No es un método manipulable por los integrantes como el RSD, ya que, en principio, hacer una estrategia es una mala idea por parte de los participantes, pues no hay prioridad. Suponga que la mejor opción es seleccionar lo que más quieren, porque de lo contrario se pueden quedar sin dicho recurso una vez se realice la repartición simultánea (Weak Strategy-Proofness).

#### Desventajas

- La asignación que se realiza es netamente probabilística y no determinista, pues depende de las elecciones aleatorias de los integrantes y sus preferencias subjetivas. Usualmente, el método viene con una segunda etapa de ronda de sorteos (*lottery decomposition*).
- Su carácter de paralelismo y varias variables hace que sea más difícil de implementar que RSD o BPS.

### 2.4.3 Funcionamiento del sistema

De forma generalizada, tenemos  $n$  integrantes de un evento con  $m$  recursos para escoger, y cada uno con una disponibilidad de  $d_m$ . Cada integrante va a consumir, de forma paralela, un elemento de  $m$ , reduciendo su disponibilidad asociada  $d_m$  bajo una velocidad uniforme, es decir, todos a la misma velocidad. Cuando  $d_m$  es cero, los integrantes consumen su siguiente recurso  $m$  preferido. Y así se continúa hasta que todos los  $d_m$  sean iguales a 0.

Como lo puede notar, este proceso no determina una asignación exacta, sino que es una asignación probabilística. Otra forma de verlo es como una lotería justa, porque cada integrante tiene una probabilidad de obtener el recurso según sus preferencias y su comportamiento.

Al final del proceso, hay una interpretación probabilística y una fase de sorteo. En este punto, cada integrante tiene una fracción de cada recurso  $m$ ; este valor acumulado es la probabilidad de cada integrante de obtener el recurso en una asignación final. Aquí hay dos casos:

- **Caso 1: Los integrantes pueden recibir varios recursos**

Se asigna cada recurso de forma individual según la probabilidad registrada.

- **Caso 2: Cada integrante debe recibir exactamente un recurso**

Hay que descomponer la matriz de probabilidades en una combinación convexa de matrices de permutación (Ej: Algoritmo de Birkhoff) para generar una asignación determinista a partir de las probabilidades.

### Consideraciones

- A pesar de que se señaló que la velocidad debe ser uniforme, se pueden plantear prioridades para ciertos integrantes aumentando su velocidad.
- Se asume que los agentes son neutralmente riesgosos. Es decir, que su satisfacción se mide por el valor esperado de la utilidad obtenida de la asignación final. Es decir, que el mecanismo se basa en las preferencias que expresan los agentes en vez de su actitud frente al riesgo.
- Hay una resistencia a la manipulación si un agente conoce las preferencias de los otros y encuentra satisfacción en el riesgo. Pero este beneficio está acotado a lo sumo al 50% de ganancia adicional, es decir, un factor de  $\frac{3}{2}$ .
- El método presenta equilibrio de Nash; hay perfiles de preferencias en los que ningún agente puede mejorar unilateralmente su resultado.
- Bajo comparación lexicográfica decreciente, se puede calcular una respuesta óptima en tiempo razonable a la manipulación estratégica. Pero cuando hay que maximizar la utilidad esperada para cada agente (si hay más de dos agentes), el problema se vuelve NP-Hard. Esto hace al método más robusto contra manipulaciones.

#### 2.4.4 Referencia

Vea [2].

## 2.5 Asignación de Cursos Universitarios mediante Mercados Competitivos

### 2.5.1 Contextualización de la solución

En contextos universitarios, cada semestre se asignan horarios de cursos a millones de estudiantes. Este entorno se caracteriza por la diversidad de preferencias de los alumnos y por restricciones severas derivadas de la capacidad limitada de los cursos, los horarios, los requisitos académicos y otros factores, como la prioridad basada en la antigüedad o la pertenencia a determinados departamentos o programas.

Los métodos tradicionales, tales como el *first-come-first-served* o los mecanismos simples de subasta, han demostrado ser ineficientes y pueden generar resultados poco equitativos, donde estudiantes con conexiones o ventajas estructurales obtienen todas las plazas deseadas, mientras que otros quedan con asignaciones subóptimas.

En respuesta a estos desafíos, Kornbluth y Kushnir proponen un novedoso mecanismo de asignación determinístico: el mecanismo de Pseudo-Mercado con Prioridades (PMP). Esta solución adopta la idea de equilibrio competitivo, pero adaptada a un entorno sin transferencias monetarias reales. El mecanismo emplea “dinero ficticio”, asignado casi equitativamente a los estudiantes, permitiendo traducir las preferencias cardinales declaradas en precios implícitos para cada curso.

Además, se incorpora explícitamente una estructura de prioridades para respetar elementos clave como la antigüedad o la pertenencia a ciertos grupos. Esto da lugar a un “corte” de prioridad: los estudiantes con mayor prioridad pueden acceder a ciertos cursos sin costo, mientras que quienes están por debajo deben “pagar” una fracción de su presupuesto ficticio. Así, el sistema busca minimizar el error de mercado, maximizar la utilidad global y garantizar un trato justo.



### 2.5.2 Descripción General de la Solución

El mecanismo PMP se compone de los siguientes pasos fundamentales:

- **Recolección de preferencias:** Se solicita a cada estudiante que informe sus preferencias sobre los horarios y combinaciones de cursos aceptables (respetando restricciones como traslapes o prerrequisitos), utilizando un lenguaje de reporte diseñado para evitar sobrecarga cognitiva.
- **Asignación de presupuestos ficticios:** Cada alumno recibe un presupuesto dentro de un intervalo casi uniforme, por ejemplo  $[1, 1 + \beta]$ , donde  $\beta$  es pequeño. Esto asegura equidad y refleja ligeras diferencias en la prioridad.
- **Cálculo del equilibrio competitivo aproximado:** Con base en las preferencias y presupuestos, se calcula una asignación de cursos junto con un vector de precios por curso y nivel de prioridad. Se utiliza una parametrización que reduce la dimensión efectiva del problema y facilita la búsqueda de un equilibrio donde el error de mercado no supere un valor teórico  $\alpha$ .
- **Optimización en dos fases:**
  - *Fase I:* Se ajustan los precios mediante un algoritmo inspirado en el proceso de tâtonnement (ajuste gradual de precios) para equilibrar oferta y demanda, minimizando el error de mercado.
  - *Fase II:* Una vez alcanzado el umbral teórico, se refina la asignación ajustando localmente los precios de los cursos sobreasignados para asegurar que no se exceda el límite de  $k - 1$  asignaciones adicionales (siendo  $k$  el número máximo de cursos que puede tomar un estudiante).

### 2.5.3 Naturaleza de la Solución

La propuesta PMP es determinística y se fundamenta en conceptos de equilibrio competitivo, adaptados a un entorno sin transacciones monetarias reales. Sus características principales incluyen:

- **Enfoque de pseudo-mercado:** El uso de dinero ficticio permite representar presupuestos casi iguales y calcular precios de equilibrio que respetan restricciones de capacidad y mejoran las asignaciones.
- **Integración de prioridades:** La estructura de prioridades (por ejemplo, por año académico o pertenencia a programas) influye directamente en los precios, permitiendo acceso preferente o gratuito a estudiantes con mayor prioridad.
- **Uso de preferencias cardinales:** A diferencia de mecanismos ordinales (como Gale–Shapley), PMP considera la intensidad de las preferencias, logrando asignaciones más eficientes y mayor satisfacción promedio.
- **Propiedades teóricas deseables:** El mecanismo garantiza estabilidad aproximada, eficiencia de Pareto y minimización de la envidia (limitada a un curso y entre estudiantes de similar prioridad).

### 2.5.4 Funcionamiento del Sistema

El funcionamiento del PMP se desarrolla en varias etapas:

#### 1. Reporte de preferencias y presupuestos:

- Los estudiantes registran sus preferencias sobre combinaciones válidas de cursos, indicando la utilidad asociada a cada opción.
- Se les asigna un presupuesto ficticio (e.g., entre 1 y  $1 + \beta$ ) que representa la cantidad de “créditos” disponibles para “pagar” por los cursos deseados.

## 2. Cálculo de precios por prioridad:

Se define un vector de precios por curso y nivel de prioridad mediante la fórmula:

$$p_{c,r}(t) = \max(t_c - (r - 1)b, 0)$$

donde  $t \in [0, Rb]^M$ ,  $R$  es el número de niveles de prioridad y  $b$  es el incremento de precio. Esta estructura genera un punto de corte (*cutoff*) que determina qué estudiantes acceden sin costo y quiénes deben pagar, según su prioridad.

## 3. Fase I – Ajuste de precios y asignación inicial:

Se realiza un ajuste iterativo de precios vía tâtonnement, con el objetivo de igualar la demanda con la capacidad y minimizar el error de mercado.

## 4. Fase II – Refinamiento y control de sobreasignaciones:

Se ajustan los precios de cursos con sobreasignación para que no excedan el límite permitido de asignaciones adicionales ( $\leq k - 1$ ). Si no se obtiene mejora, puede reiniciarse la fase con nuevos precios iniciales.

### 2.5.5 Resultados o Impacto

Las simulaciones realizadas con datos reales de asignación de cursos demuestran que el mecanismo PMP ofrece mejoras significativas:

- **Mayor utilidad promedio:** PMP logra la utilidad media estudiantil más alta en todas las categorías académicas, especialmente en primer y segundo año (mejoras del 9% y 7%, respectivamente).
- **Reducción de la desigualdad en la satisfacción:** Se observa una disminución considerable en la desviación estándar de las utilidades, con una mejora del 11.16% en estudiantes de primer año, indicando mayor equidad.
- **Menor envidia:** Solo el 1.26% de los estudiantes manifiesta envidia (deseo por el horario de otro estudiante con la misma o menor prioridad), frente a porcentajes superiores al 7% en mecanismos como DA o RSD.
- **Control efectivo del error de mercado:** El error promedio se mantiene en torno a 14 asientos, muy por debajo del umbral teórico ( $\alpha \approx 43.5$ ), demostrando la eficiencia del mecanismo.
- **Robustez y resistencia a manipulaciones:** En mercados grandes, PMP es prácticamente inmune a manipulaciones estratégicas, mostrando que es estrategia-proof en el largo plazo.

### 2.5.6 Referencia

Vease [3].

## 3 Descripción y Justificación del Problema a Resolver

La Universidad Nacional de Colombia, como uno de sus fines misionales, tiene por objetivo impartir conocimiento al país, en particular a su comunidad de estudiantes mediante la docencia. Para realizarlo, se ofrecen cursos (asignaturas) como actividades académicas; sin embargo, dicha oferta requiere recursos y la sincronización de horarios, docentes y aulas. Y al mismo tiempo, dicho proceso se ve entorpecido por la falta de recursos, la pésima administración, las garantías académicas que se usan con poca conciencia por integrantes de la comunidad estudiantil, como la libre inscripción y cancelación, o el aprovechamiento del sistema mediante estrategias de manipulación.

Ahora bien, dicho problema se puede abordar mediante modelación estocástica, planteando una simulación que emplee algoritmos y las condiciones de la Universidad para encontrar una estrategia y solución óptima al problema real.

### 3.1 Objetivo Principal

- Formular *un modelo estocástico* que describa el problema de planificación académica en la Universidad Nacional.
- *Simular* ese modelo para encontrar al menos *una solución óptima* que contribuya a mitigar el caos descrito.

#### Variables que afectan:

- Mala administración.
- Insuficiencia de recursos (humanos, físicos, logísticos).
- Comportamiento estratégico o irresponsable de estudiantes (inscripción y cancelación masiva).

## 4 Diseño de la solución

Para abordar la solución, se presentan las áreas y abstracciones que se van a abordar.

- La oferta.
  - Cursos.
  - Docentes.
  - Salones (y laboratorios).
  - Horarios.
- Demanda fluctuante y mal gestionada.
  - Preferencias no explícitas.
  - Manipulación.
  - Cancelaciones y sobreinscripción.
- Solución.Problemas
  - Falta de coordinación.
  - Falta de mecanismos eficientes de asignación.
- **Objetivo esperado.**
  - Obtener una planificación justa, óptima y automatizada.

### 4.1 Metodología

Para el desarrollo se abordará el problema en fases. Y el programa se realizará a través de módulos bajo una arquitectura de componentes.

#### 4.1.1 Preparación

##### Módulo de Oferta

Este módulo tiene como objetivo generar una oferta académica simulada de cursos a partir de una base de datos inicial, seleccionando aleatoriamente docentes, aulas y horarios bajo una distribución uniforme. Esta etapa corresponde a una primera aproximación estocástica del sistema, sin aún aplicar criterios de optimización o restricciones complejas.

Constantes del programa (Se pueden tomar de una base de datos o emularlas):

- **Cursos:** Lista de diccionarios, donde cada elemento representa un curso con los siguientes atributos:
  - `codigo` (int): identificador único del curso.
  - `programado` (bool): si ya se programó el curso.
  - `numGrupos` (int): numero de grupos que el curso puede tener.
- **Docentes:** Lista de diccionarios, cada uno con:
  - `cedula` (int): identificador del docente.
  - `programado` (bool): si ya se agendó a un curso.
- **Aulas:** Lista de diccionarios, cada uno con:
  - `edificio` (int): número del edificio.
  - `aula_id` (str): identificador del aula.
  - `programado` (bool): si ya se agendó a un curso.
- **Horarios disponibles:** Lista de diccionarios con:
  - `ID`
  - `dia` (list[int]): día(s) de la semana, de lunes a sábado.
  - `franja` (str): intervalo de tiempo disponible (e.g., 7-9, 9-11, 11-13, 14-16, 16-18, 18-20, 7-13, 14-18).

Para cada curso (hasta que los grupos que puede tener sean cero), el módulo va a realizar lo siguiente:

1. **Selección de docente:** Se elige aleatoriamente un docente disponible (que no esté asignado a otro curso en el mismo horario) y se marca como programado.
2. **Selección de aula:** Se selecciona aleatoriamente un aula disponible (que no esté ocupada en el horario asignado) y se marca como ocupada.
3. **Selección de horario:** Se selecciona aleatoriamente un horario. Para garantizar una distribución balanceada de clases a lo largo de la semana, se propone el siguiente esquema de asignación adaptativa:
  - Llevar un conteo de asignaciones por cada día de la semana.
  - Definir una función de probabilidad que asigne mayor peso a los días menos cargados.
  - Aplicar una probabilidad ponderada inversamente a la carga actual de cada día:

$$P(dia_i) \propto \frac{1}{1 + asignaciones : dia_i}$$

- Excluir los días viernes y sábado inicialmente. Estos se incluirán únicamente cuando reste el 5% de los cursos por asignar.
4. **Creación de la clase:** Se registra la asignación como un nuevo diccionario en la colección `clases`, la cual contiene los siguientes campos: `codigo_clase` generado aleatoriamente, `codigo_curso`, `cedula_docente`, `horario` y `aula` (identificada como `edificio+aula_id`). Además, se asigna el número de `cupos` como un valor aleatorio entre 30 y 60, calibrado de modo que el promedio de cupos se acerque al valor proporcionado por el usuario. También se asigna el número de grupo `grupo` con base a la clase, y se resta uno a la cantidad de grupos del curso.

Es así que la clase queda de la siguiente forma:

**Clase:** Lista de diccionarios con:

- `codigo_clase` (int)
- `codigo_curso` (int): código del curso asociado
- `cedula_docente` (int): código del docente asociado
- `horario` (Tuple[str, str]): horario asociado Dia: str, franja Horaria: str.
- `edificio+aula_id` (str): aula asociada
- `grupo` (int): número de grupo
- `cupos` (int): cupos asociados

### Módulo de Estudiantes

Aquí se definirán los deseos de los estudiantes.

- **Estudiante:** Lista de diccionarios con
  - Cédula estudiante (int)
  - P.A.P.I estudiante (int)
  - Numero de materias que va inscribir (int), menor o igual al tamaño de la lista de desos.
  - Tamaño de la lista de deseos (int)
  - Lista de deseos/preferencias de tamaño n donde cada elemento es un código de clase.
  - Lista de asignaciones, las asignaturas que tiene inscritas

**\*Nota:** El estudiante quiere inscribir 5 materias, pero puede tener 10 opcionadas. 10 va a ser el tamaño de la lista de preferencias.

Para cada estudiante se define un P.A.P.I aleatorio siguiendo una distribución Gaussiano con valor medio 3.8 y una desviación estandar acorde.

Para cada estudiante, se define el número de materias que va a inscribir de forma aleatoria entre 3 y el número del parámetro de entrada. Se crea una lista de deseos de un tamaño que se consigue tomando un Numero aleatorio del parámetro de entrada y seleccionando aleatoriamente, bajo una distribución uniforme, un código de clase y agregándolo a dicha lista (entre 3 y el parámetro de entrada).

Ahora se deben asignar clases a la lista de deseos. Para esto en un ciclo for desde 0 hasta *tamanoListaDeseos*. Tomar n curso aleatorio del pool, guardar sus clases en una lista de elegibles. Se toma una clase de esa lista aleatoriamente si presenta conflicto de horario con las ya seleccionadas, elimina la clase de elegibles y toma otra clase. Si definitivamente tiene conflictos con todas las clases de ese curso, descarta esta elección. Independientemente de si selecciono o no una materia como deseada, el contador se reduce.

#### 4.1.2 Modelo Estocástico: Proceso de Asignación

Ahora bien, una vez que la oferta de clases ya está construida y los estudiantes también están definidos junto con sus preferencias, sigue el proceso de asignación y selección de asignaturas. Se usará el algoritmo Probabilistic Serial Rule porque la teoría indexa que es Pareto-Eficiente.

### Probabilistic Serial Rule (PS)

#### • Inicialización

- Para cada estudiante, se crea una lista vacía de asignaciones (inicialmente fraccional).
- Cada estudiante dispone de una lista ordenada de preferencias de asignaturas.
- Para cada asignatura, se conoce la capacidad (número de cupos disponibles).
- Se inicializa un diccionario de consumo  $C[e, a] = 0$ , donde  $e$  denota al estudiante y  $a$  a la asignatura.

#### • Consumo continuo

- Se simula un avance en el tiempo continuo. Al inicio, cada estudiante comienza a "consumir" su asignatura más preferida que aún disponga de cupos, a una tasa unitaria.
- Mientras la asignatura no se agote (es decir, mientras su capacidad sea mayor que cero), el estudiante sigue consumiéndola.
- Cuando una asignatura se consume completamente (cupos = 0), todos los estudiantes que la estaban consumiendo pasan a consumir su siguiente asignatura preferida con cupos disponibles.
- Durante este proceso se actualiza el diccionario de consumo, de modo que, para cada asignatura  $a$ , la suma de los consumos de todos los estudiantes satisface:

$$\sum_e C[e, a] = \text{capacidadInicial} : a.$$

#### • Finalización y redondeo

- Al finalizar el proceso, cada entrada  $C[e, a]$  representa la fracción de la asignatura  $a$  "consumida" por el estudiante  $e$ . Dichas fracciones se interpretan como las probabilidades de asignación.
- Se aplicará una técnica de *rounding* (Greedy Fraccional) para convertir la asignación fraccional en asignaciones definitivas, asignando a cada estudiante una o más asignaturas según se requiera.
- Para ese método se van a ordenar a los estudiantes por P.A.P.I.

#### 4.1.3 Métricas

En este punto ya se tiene la preparación del entorno y la simulación de asignación de asignaturas, es decir, los estudiantes ya pudieron inscribir las materias que estuvieron a su alcance. Ahora queda por determinar si dicha solución fue óptima o no. Con este objetivo, se analizarán tres métricas: porcentaje de estudiantes que obtuvieron su primera, segunda y tercera opción; equidad (medida como la desviación estándar de la satisfacción) y eficiencia de Pareto.

##### Porcentaje de estudiantes que obtuvieron su primera, segunda, tercera opción:

Esta métrica evalúa la calidad de la asignación desde la perspectiva de la preferencia. Para cada estudiante, se determina en qué posición de su lista de preferencias quedó la asignatura asignada (por ejemplo, si se le asignó la asignatura que figura en la posición 1, eso equivale a haber recibido su primera opción). Lo anterior calculado mediante los siguientes pasos:

- Por cada estudiante, y por cada asignatura en su lista de deseos, tomar las que en efecto fueron asignadas y tomar su ranking (indica el número de opción/preferencia, si la 1, 2, 3...).
- Contar los estudiantes que en efecto tuvieron la primera, segunda y tercera.
- Calcular los porcentajes. Por ejemplo:

$$\text{porcentaje} : \text{primeraOpcion} = \frac{\text{numEstudiantesPrimeraOpcion}}{\text{totalEstudiantes}}$$

##### Equidad (Desviación estándar de la satisfacción)

La equidad se puede medir calculando la desviación estándar de la "satisfacción" entre los estudiantes. Para ello, asignamos un valor a cada asignatura recibida, basado en la posición de la preferencia:

Sea  $aS_i$  las asignaturas que el estudiante pudo inscribir y  $m$  dicha cantidad:

$$\text{satisfaccion} = \sum_{i=0}^{m-1} \left( \frac{1}{\text{posicion}(aS_i, \text{listaPreferencias}) + 1} \right) * \frac{1}{\text{maximoPosibleEstudiante}}$$

Donde *maximoPosibleEstudiante* es el grado de satisfacción que corresponde a las primeras n materias que el estudiantes piensa inscribir.

### Explicación

La satisfacción de cada estudiante se obtiene sumando los valores correspondientes a cada asignatura que le fue asignada. Por ejemplo, si un estudiante obtiene sus primeras cinco opciones, su satisfacción sería

$$1.0 + 0.75 + 0.5 + 0.25 + 0.10 = 2.6.$$

En cambio, otro que reciba asignaciones desde la sexta opción en adelante tendrá una satisfacción mucho menor, digamos, 0.3.

Ahora bien, para hacer comparables las satisfacciones entre estudiantes con distintos números de asignaturas deseadas, se puede normalizar la satisfacción dividiendo la suma obtenida entre el valor máximo posible para ese estudiante. Por ejemplo:

- Si un estudiante deseaba 1 asignatura, el valor máximo posible es 1.
- Si deseaba 5 asignaturas, el máximo sería 2.6.

La satisfacción normalizada,  $S_{\text{norm}}$ , se calcula como:

$$S_{\text{norm}} = \frac{\text{satisfaccionObtenida}}{\text{maximoPosibleEstudiante}}$$

De este modo, la satisfacción normalizada tendrá siempre un rango entre 0 y 1, independientemente de la cantidad de asignaturas esperadas.

La cantidad del maximo posible de estudiante e calcularía así:

$$\text{maximoPosibleEstudiante} = \sum_{i=1}^n \left(\frac{1}{i}\right)$$

Donde n es la cantidad de materias que el estudiante va a inscribir.

Finalmente, se calcula la desviación estándar de los valores  $S_{\text{norm}}$  de todos los estudiantes:

- Un valor cercano a 0 (por ejemplo, 0 a 0.1) indicaría **alta equidad**, pues todos tienen niveles de satisfacción similares.
- Valores intermedios (por ejemplo, de 0.2 a 0.5) indican una variabilidad moderada.
- Valores mayores (por encima de 0.5) señalarían una gran disparidad en la asignación, es decir, **baja equidad**.

## 5 Código Fuente

## 6 Manual Usuario

### Datos de entrada

- Número de cursos.
- Número de docentes, debe ser mayor o igual al número de cursos.
- Número de aulas, debe ser mayor o igual al número de cursos.
- Número de estudiantes.

- Máximo número de materias deseadas de un estudiante (Predeterminado=9).
- Máximo número de materias que un estudiante va a inscribir (mayor o igual a 3).
- Promedio de número de cupos de las clases (Predeterminado 40).
- Tiempo de simulación

## 7 Manual Técnico

## 8 Experimentación

### 8.1 Análisis de resultados

#### 8.1.1 Escenario 1

Resultados del escenario sin modificaciones.

#### 8.1.2 Escenario 2

En este escenario se evalúa el impacto de otorgar prioridad de inscripción a un subconjunto específico de estudiantes. Para ello, se selecciona aleatoriamente un grupo que representa el 20% del total de estudiantes. Este grupo realiza primero el proceso de asignación de materias mediante el mecanismo base (PS) y su respectivo redondeo.

El 80% restante de estudiantes entra al proceso de asignación únicamente una vez que el grupo prioritario ha completado su selección y ya no tiene materias adicionales por consumir. De este modo, se simula una política de inscripción diferenciada en dos etapas, en la que se garantiza prioridad efectiva para un subconjunto determinado.

#### 8.1.3 Escenario 3

En este escenario se simula la cancelación temporal de un subconjunto de asignaturas. Se define una probabilidad de cancelación por clase (por ejemplo, un 30%), y se selecciona aleatoriamente el conjunto de clases que serán excluidas inicialmente.

Durante la primera ejecución de la simulación, dichas clases tienen cupo igual a cero (aunque en realidad conservan su disponibilidad), lo que las hace no seleccionables. Una vez finalizado este primer proceso de asignación, se vuelve a ejecutar la simulación incluyendo ahora las clases previamente canceladas (volviendo desde la primera prioridad), manteniendo las decisiones y variables que se hayan modificado en la primera fase (por ejemplo, asignaciones ya efectuadas, preferencias agotadas, etc.).

Solo en esta segunda etapa se realiza el redondeo final, considerando la asignación con el universo completo de clases. Este escenario permite analizar el impacto de cancelaciones temporales sobre la equidad, eficiencia y estabilidad del proceso de asignación.

Tabla 1: Ejemplo de tabla 4x3 con contorno marcado

Encabezado 1	Encabezado 2	Encabezado 3	Encabezado 4
Celda 1	Celda 2	Celda 3	Celda 4
Celda 5	Celda 6	Celda 7	Celda 8
Celda 9	Celda 10	Celda 11	Celda 12



Tabla 2: Nivel de Agua en los Embalses

Fecha	Estado	Nivel de los embalses/Capacidad
Abril 2024	Inicio Racionamiento	14%
Junio 2024	Condiciones Actuales Escasez	30%
Ideal	Capacidad Completa	100%

Así mismo hay una tabla 1.

Tabla larga y personalizable en 3

Tabla 3: Análisis Stakeholders

Stakeholder	Necesidades	Intereses	Impacto
CD1	CD2	CD3	CD4
CD5	CD6	CD7	CD8

Y hay una figura 1.



Figura 1: Descripción de la figura.

Por último una lista:

- Item 1
- Item 2
- Item 3

Use la siguiente página para documentar las referencias en formato IEEE: <https://www.citethisforme.com/>

## 9 Referencias

- [1] M. Bichler, S. Merting, and A. Uzunoglu, “Assigning Course Schedules: About Preference Elicitation, Fairness, and Truthfulness,” arXiv preprint arXiv:1812.02630, 2018. [En línea]. Disponible en: <https://arxiv.org/abs/1812.02630>
- [2] A. Bogomolnaia and H. Moulin, “A new solution to the random assignment problem,” Journal of Economic Theory, vol. 100, no. 2, pp. 295–328, 2001, doi: 10.1006/jeth.2000.2710.
- [3] Kornbluth, D. & Kushnir, A. (2023). *Undergraduate Course Allocation through Competitive Markets*. Carnegie Mellon University / Harvard University.