

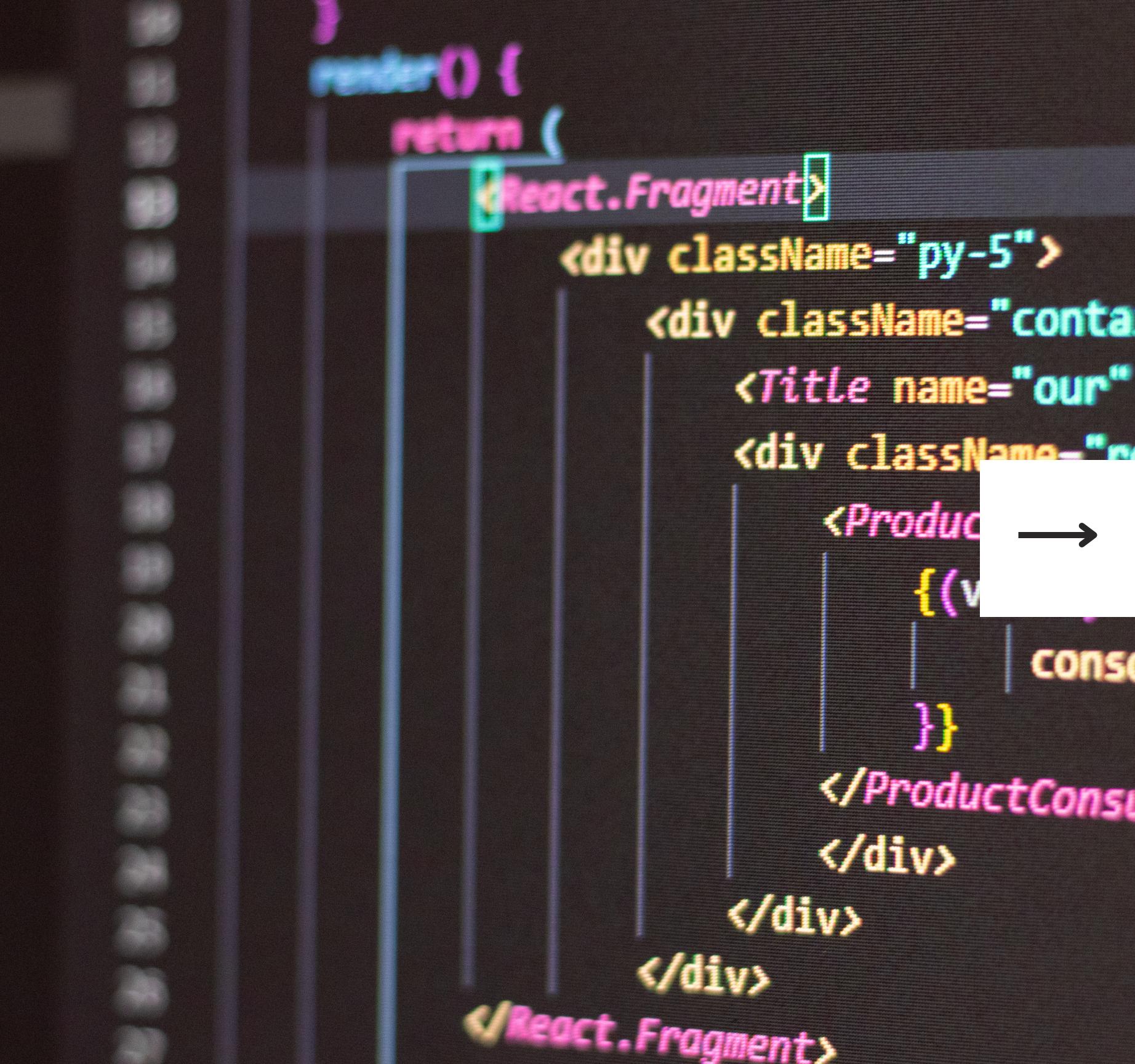
TRABAJO DE FINAL DE GRADO

DESAFIANDO EL MODELO DE BERTRAND:

CÓMO LOS ALGORITMOS PUEDEN GENERAR COLUSIÓN TÁCITA EN MERCADOS OLIGOPÓLICOS

Autor: Genís Sánchez García

Tutor: Pablo Lopez Argüello



```
render() {
  return (
    <React.Fragment>
      <div className="py-5">
        <div className="conta
          <Title name="our"
            <div className="prod
              <ProductConso
                <div>
                  <div>
                    <React.Fragment>
```

Índice

1. Introducción
2. Objetivos
3. Metodología
4. Conceptos generales
5. Q-learning y duopolio de Bertrand
6. Resultados
7. Conclusiones



Introducción

La colusión es el acuerdo conjunto que realizan firmas o empresas independientes para determinar precios y niveles productivos concretos, con el fin de maximizar ganancias (Stigler, 1964) y de “coordinar y/o monitorear las acciones para alcanzar beneficios por encima de los niveles competitivos” (Garrod & Olczak, 2018) cercanos a los umbrales monopolísticos (Motta, 2004).

Sin embargo, la colusión tácita se da cuando:

- Se produce una situación típicamente coordinada o colusoria, pero sin mediar **comunicación o acuerdo directo**.
- La coordinación se basa en la **confianza mutua**
- Las ganancias derivadas de la colusión son **mayores** que la **libre competencia** o las represalias ante la **desviación colusoria**



La colusión tácita **NO** es una práctica o conducta inherentemente humana. Ciertos algoritmos informáticos son capaces de propiciarla o de ser su origen.

Objetivos



Objetivo general

Analizar y comprender cómo los algoritmos fijadores de precios son capaces de aprender a coludir tácitamente de forma completamente autónoma (sin intervención humana) y sin mediar ningún tipo de comunicación.

1

Revisión de la literatura

Sobre la colusión, las condiciones óptimas para su aparición, modalidades algorítmicas y mejores algoritmos y sistemas para su consecución

2

Simulación computacional de un mercado oligopólico de Bertrand

Desarrollar un algoritmo que simule un oligopolio de Bertrand en el que dinámicamente se fijen precios y se monitoricen las decisiones tomadas por cada agente económico con los que medir su nivel colusorio

3

Análisis de resultados

Comparar los resultados obtenidos con los hallados en otras investigaciones previas, identificar similitudes y diferencias, y determinar el éxito del algoritmo desarrollado

Metodología

MÉTODOS EMPLEADOS

El marco teórico se ha basado en el análisis exhaustivo de numerosas fuentes académicas, y el marco empírico en el análisis de los resultados obtenidos en la simulación o experimento computacional

ENFOQUE DE INVESTIGACIÓN

La totalidad de la investigación se ha desarrollado alrededor de la consecución del objetivo general: analizar y comprender como los algoritmos informáticos pueden propiciar u originar colusión tácita



FUENTES UTILIZADAS

Se han utilizado fuentes académicas de prestigio, priorizando aquellas con un alto número de citaciones y las avaladas por expertos por el Servicio de Bibliotecas de la Universidad Autónoma de Barcelona

NATURALEZA DE LOS DATOS

Los datos utilizados para el desarrollo del algoritmo son aleatorios y completamente diferentes para cada ejecución

Conceptos generales

Condiciones de mercado que propician prácticas colusorias

- 1 Bajo número de competidores
- 2 Simetrías de mercado
- 3 Fuertes barreras de entrada y baja rotación competitiva
- 4 Alta transparencia de mercado
- 5 Alta interacción de mercado

La **simetría de mercado** incluye la simetría de las funciones de coste, las funciones de capacidad productiva y la simetría u homogeneidad de productos en calidad y diferenciación

Tipos de colusión algorítmica y sus diferencias

- 1 Tipo messenger
- 2 Tipo Hub and Spoke
- 3 Tipo Predictable Agent
- 4 Tipo Digital Eye

Tipos de aprendizaje automático

- 1 Aprendizaje supervisado
- 2 Aprendizaje no supervisado
- 3 Aprendizaje semi supervisado
- 4 Aprendizaje por refuerzo

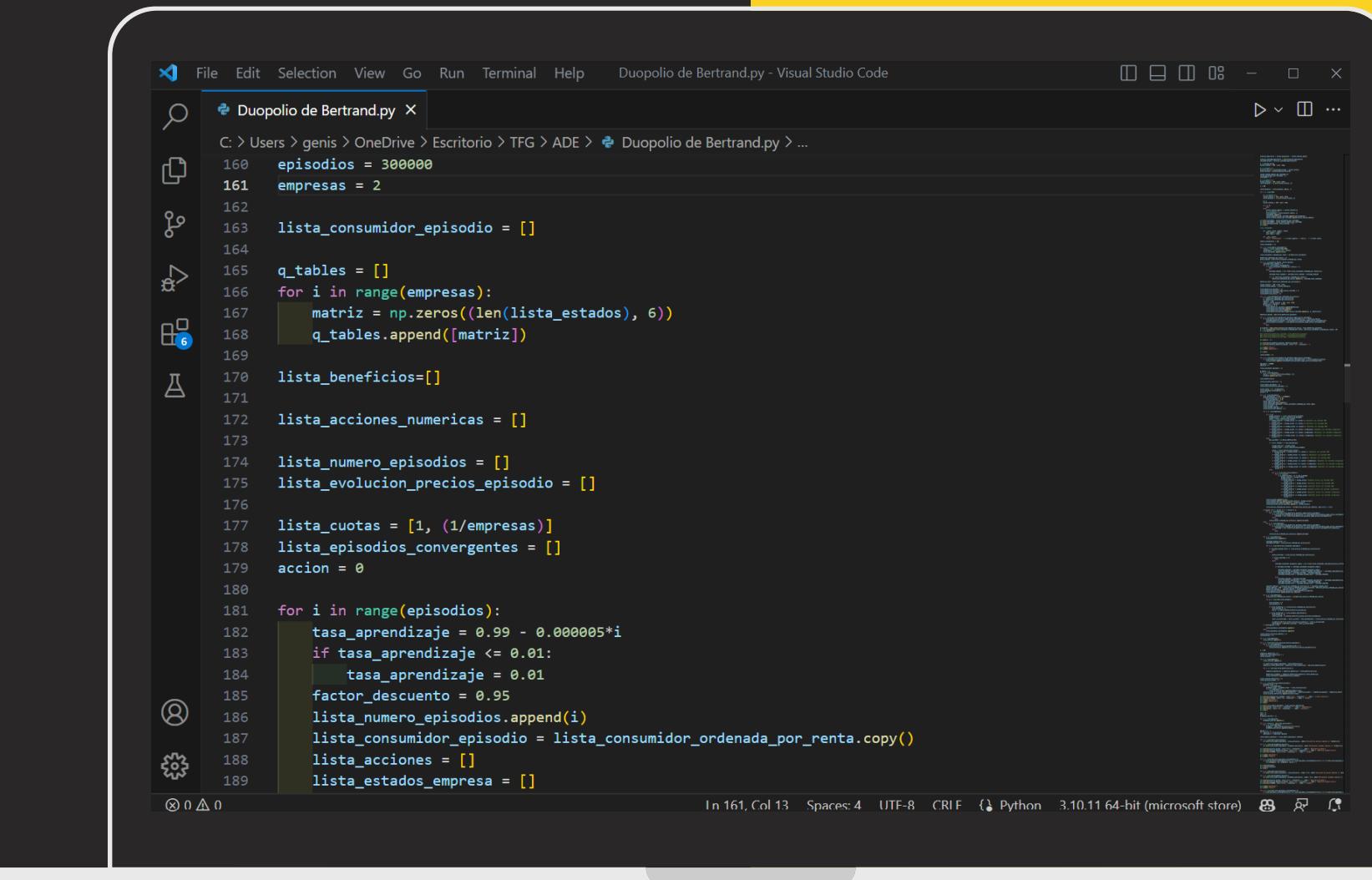
Q-learning y duopolio de Bertrand

Mejor algoritmo fijador de precios

- 1 Mejor toma de decisiones como resultado de la experiencia
- 2 Información extraída del entorno mediante la interacción: decisiones dinámicas

Limitaciones

- 1 Determinada función de coste pero indeterminada función de demanda
- 2 Solo caben 6 tipos de acciones determinadas y concretas
- 3 Solo caben precios y unidades enteras limitados
- 4 Rentas inmutables y consumidores sin preferencias ni valoraciones de utilidad



A screenshot of the Visual Studio Code interface. The title bar says "Duopolio de Bertrand.py - Visual Studio Code". The code editor shows a Python script named "Duopolio de Bertrand.py". The script contains several lists and loops related to a Bertrand oligopoly simulation. The code is partially visible, starting from line 160 and ending at line 189. The status bar at the bottom indicates the file has 161 lines, 4 spaces, and is in UTF-8 encoding, using Python 3.10.11 64-bit (microsoft store).

```
160 episodios = 300000
161 empresas = 2
162 lista_consumidor_episodio = []
163 q_tables = []
164 for i in range(empresas):
165     matriz = np.zeros((len(lista_estados), 6))
166     q_tables.append([matriz])
167
168 lista_beneficios=[]
169 lista_acciones_numericas = []
170 lista_numero_episodios = []
171 lista_evolucion_precios_episodio = []
172
173 lista_cuotas = [1, (1/empresas)]
174 lista_episodios_convergentes = []
175
176 accion = 0
177
178 for i in range(episodios):
179     tasa_aprendizaje = 0.99 - 0.000005*i
180     if tasa_aprendizaje <= 0.01:
181         tasa_aprendizaje = 0.01
182     factor_descuento = 0.95
183     lista_numero_episodios.append(i)
184     lista_consumidor_episodio = lista_consumidor_ordenada_por_renta.copy()
185     lista_acciones = []
186     lista_estados_empresa = []
```

Captura de pantalla parcial del algoritmo desarrollado

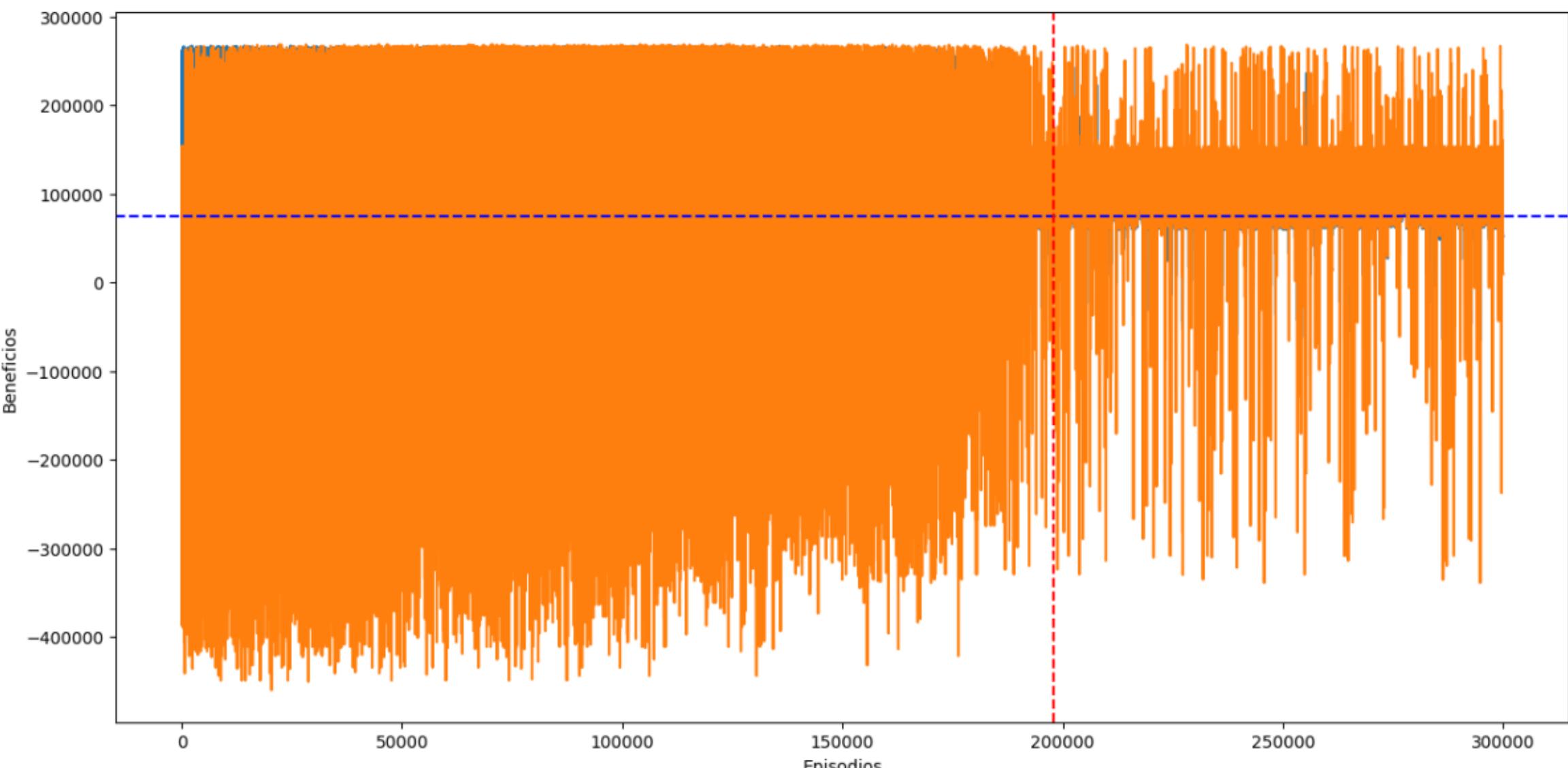
Resultados

Leyenda

- Beneficios empresa 1
- Beneficios empresa 2
- Tasa de aprendizaje mínima
- Beneficio promedio



Beneficios en colusión cooperativa



Elaboración propia mediante python

Resultados

Leyenda

- Precios empresa 1
- Precios empresa 2
- Promedio móvil empresa 1
- Promedio móvil empresa 2
- Área de convergencia
- Precio monopolístico
- Precio de Nash
- Tasa de aprendizaje mínima



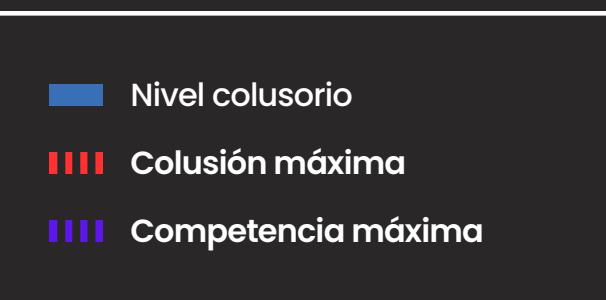
Precios en colusión cooperativa



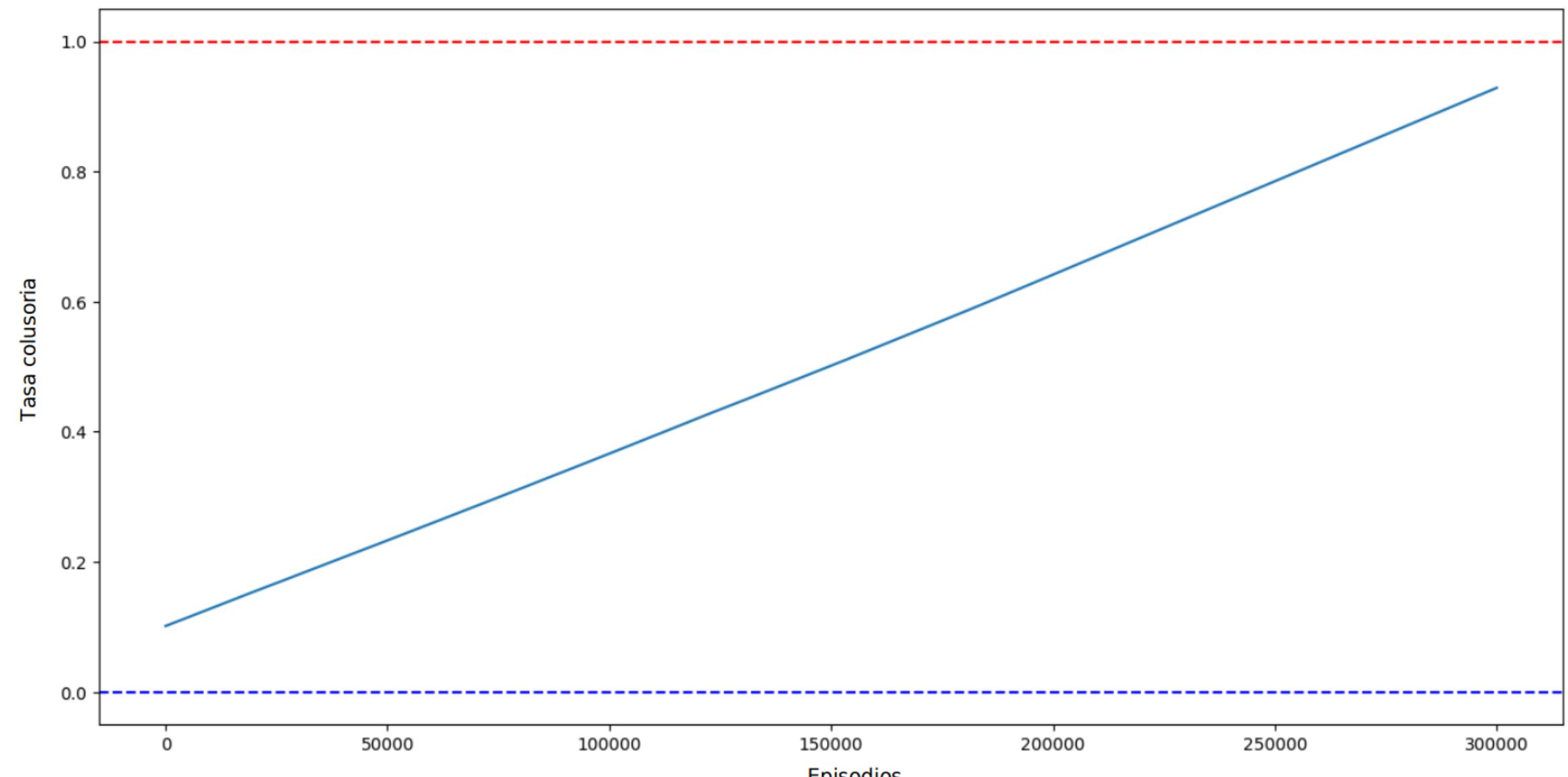
Elaboración propia mediante python

Resultados

Leyenda



Colusión cooperativa



Elaboración propia mediante python

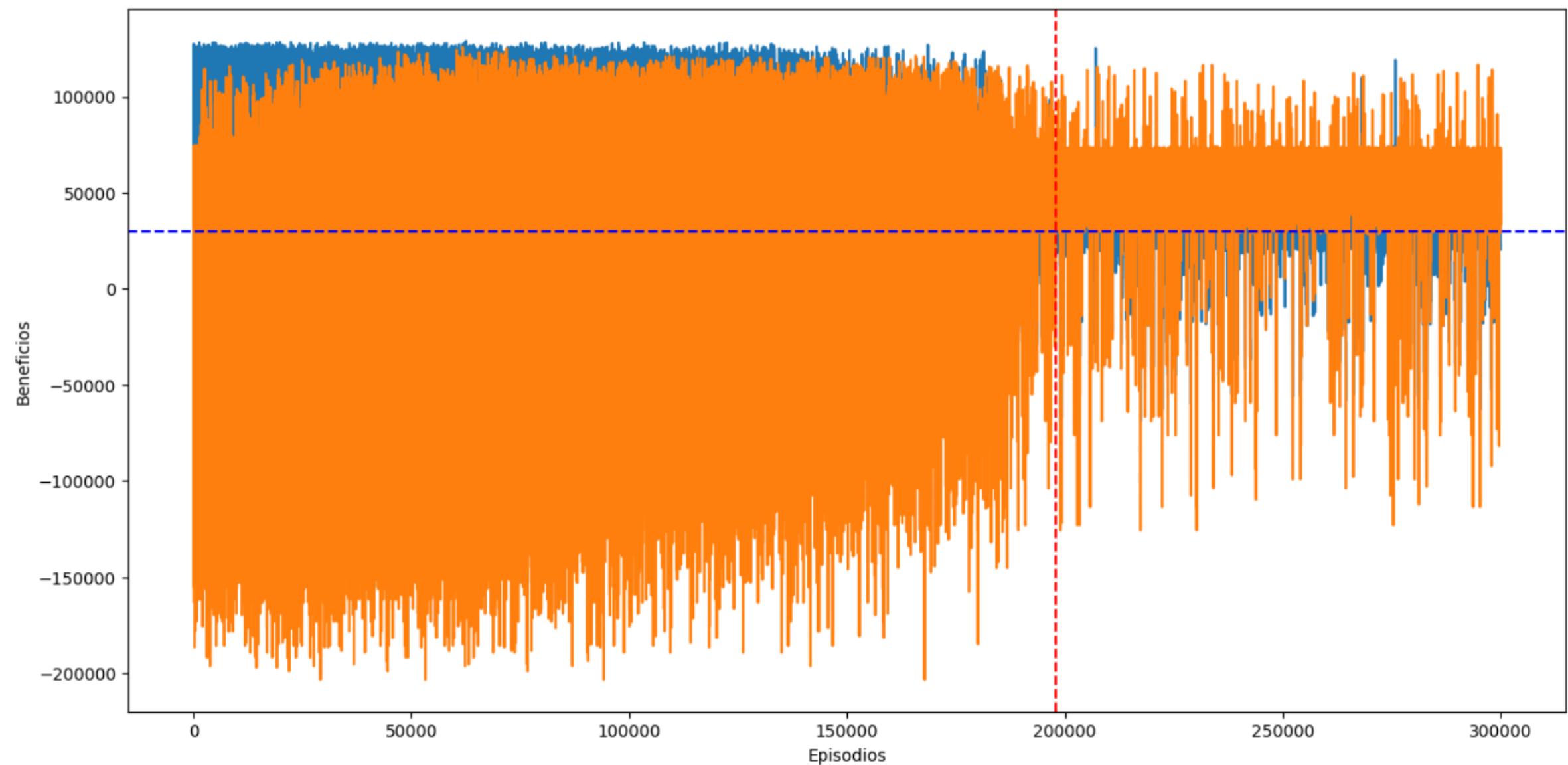
Resultados

Leyenda

- Beneficios empresa 1
- Beneficios empresa 2
- Tasa de aprendizaje mínima
- Beneficio promedio



Beneficios en colusión dominante



Elaboración propia mediante python

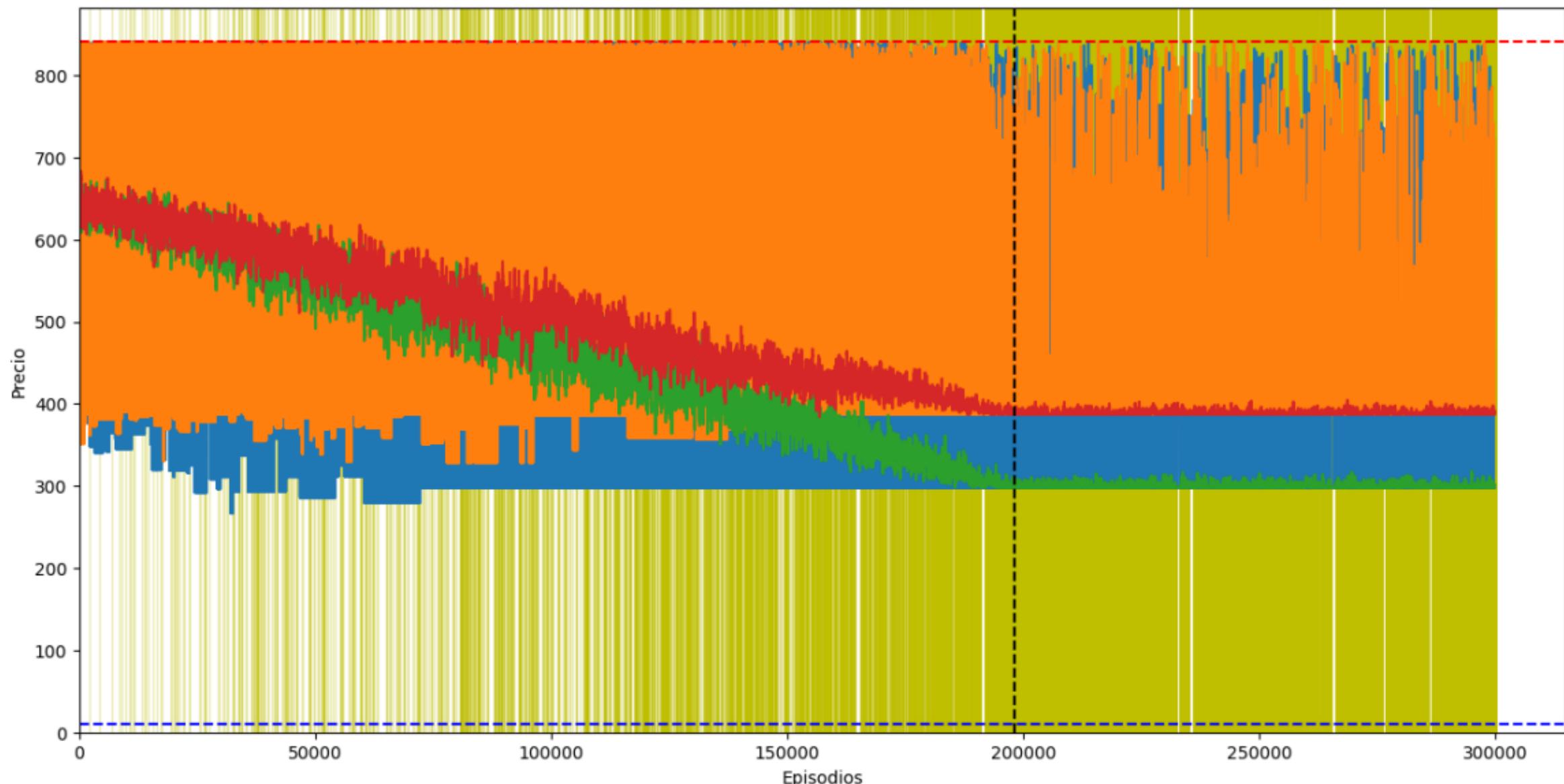
Resultados

Leyenda

- Precios empresa 1
- Precios empresa 2
- Promedio móvil empresa 1
- Promedio móvil empresa 2
- Área de convergencia
- Precio monopolístico
- Precio de Nash
- Tasa de aprendizaje mínima



Precios en colusión dominante



Elaboración propia mediante python

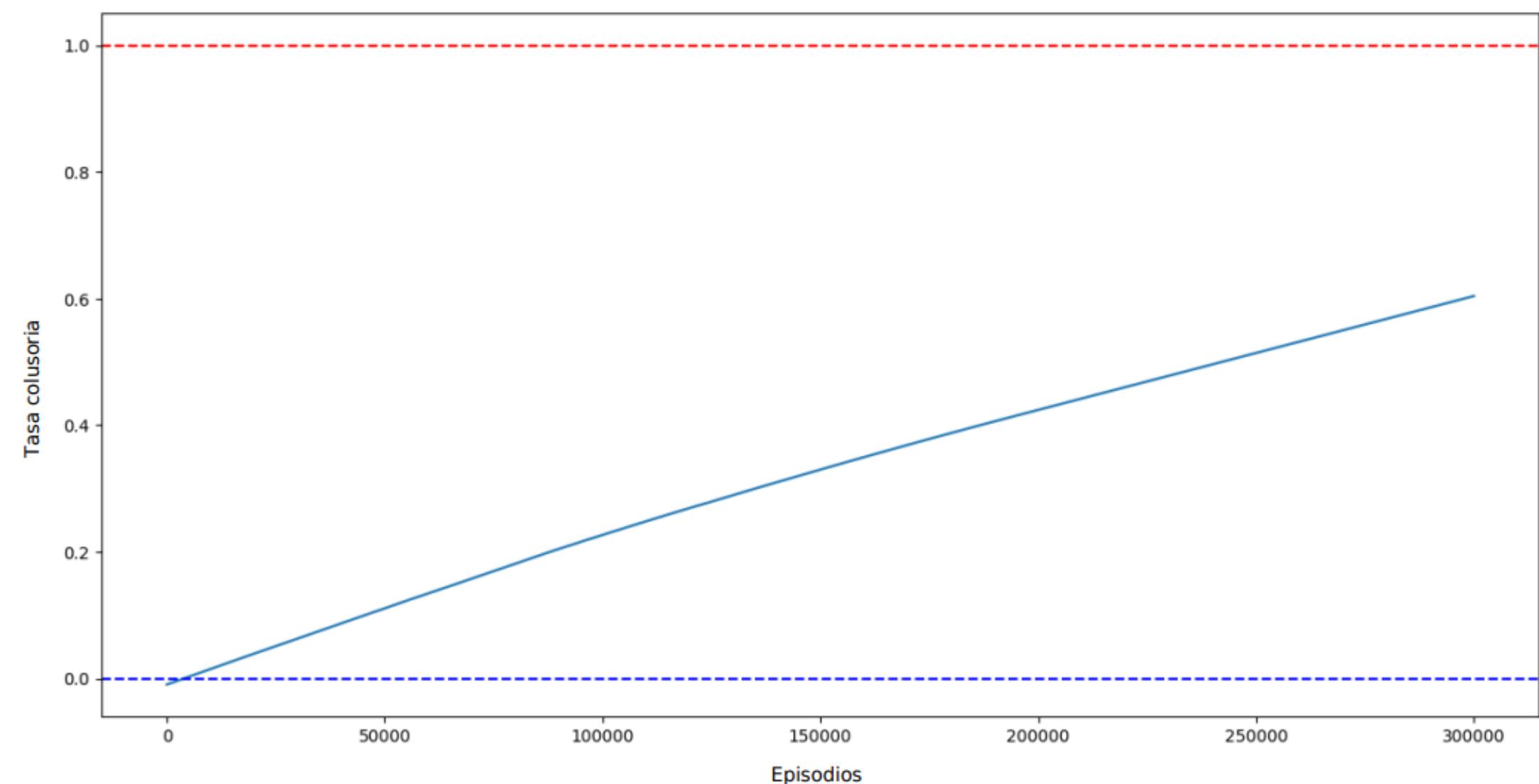
Resultados

Leyenda

- Nivel colusorio
- Colusión máxima
- Competencia máxima



Colusión dominante



Elaboración propia mediante python

Resultados

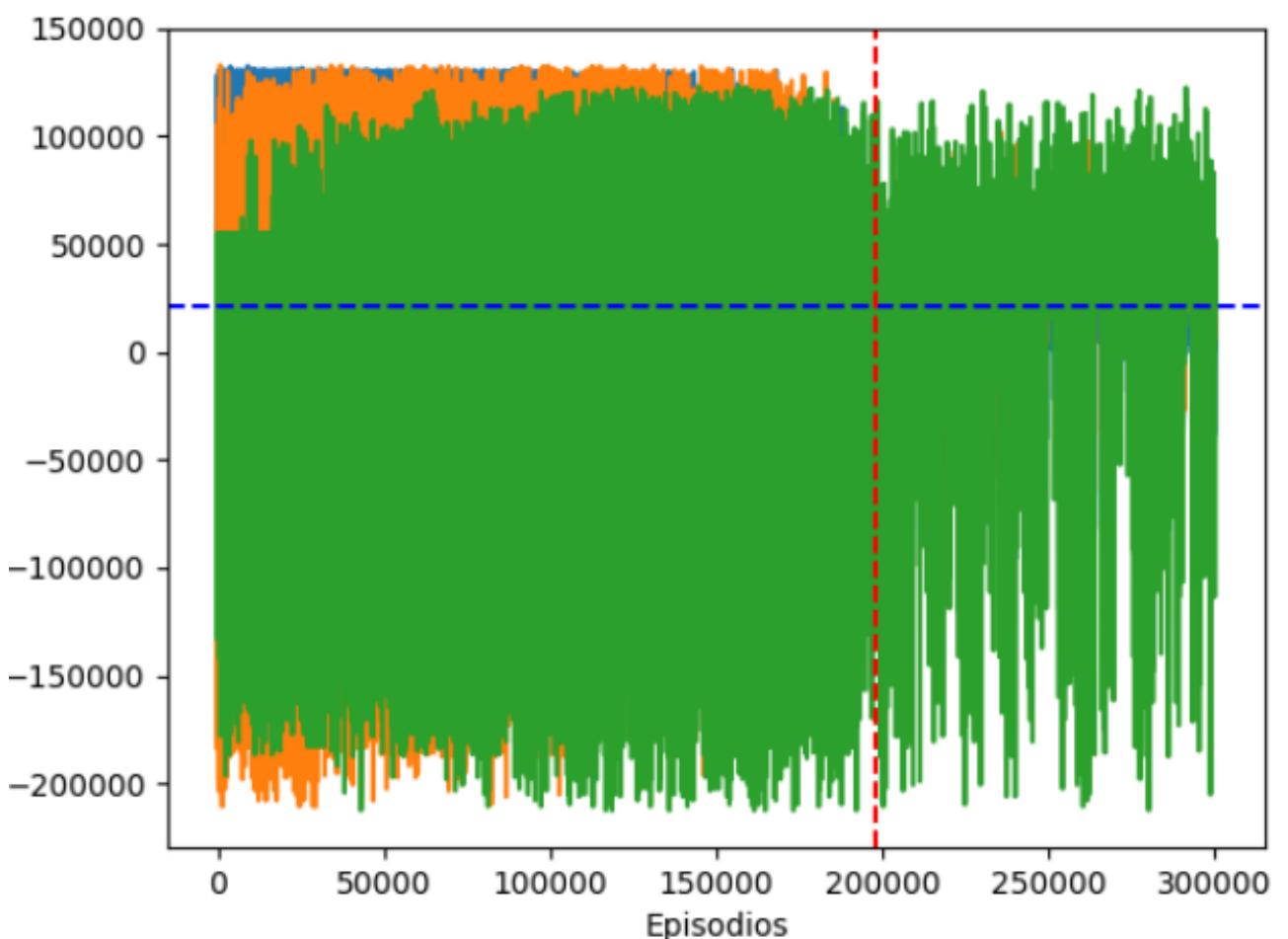
Beneficios en n=3 y n=4

Leyenda

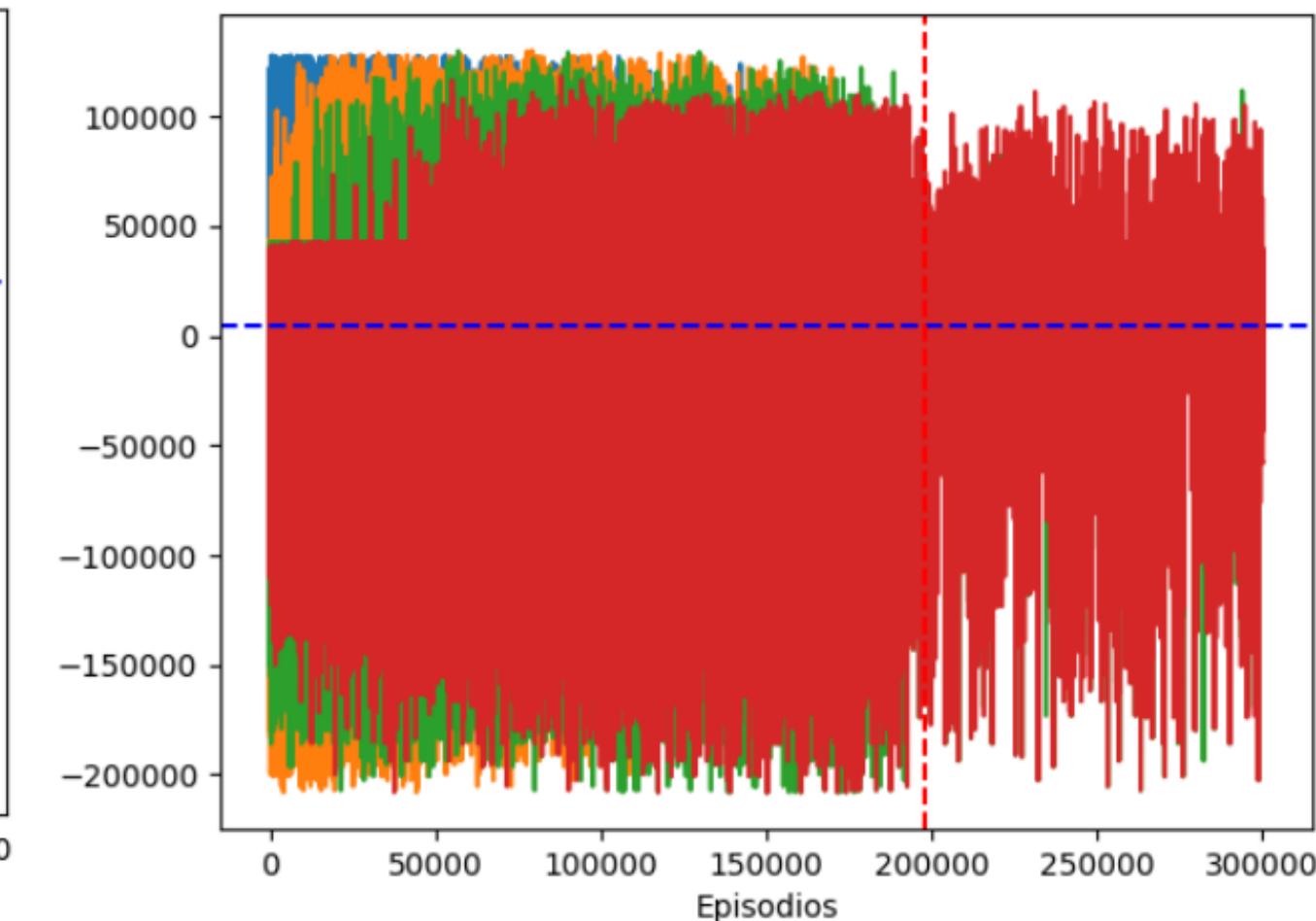
- Beneficios empresa 1
- Beneficios empresa 2
- Beneficio empresa 3
- Beneficios empresa 4
- Área de convergencia
- Tasa de aprendizaje mínima
- Beneficio promedio



n = 3



n = 4



Elaboración propia mediante python

Resultados

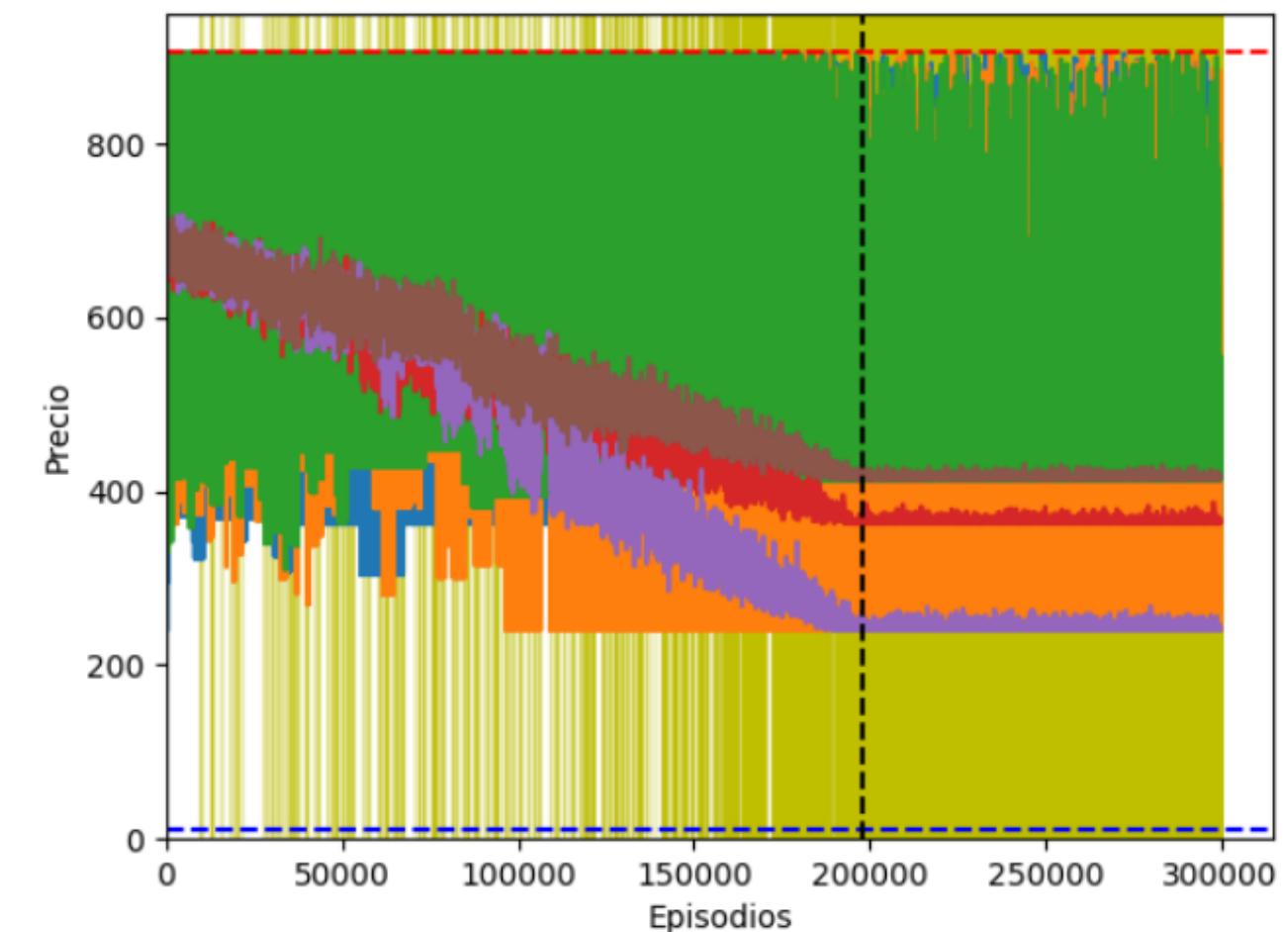
Precios en n=3 y n=4

Leyenda

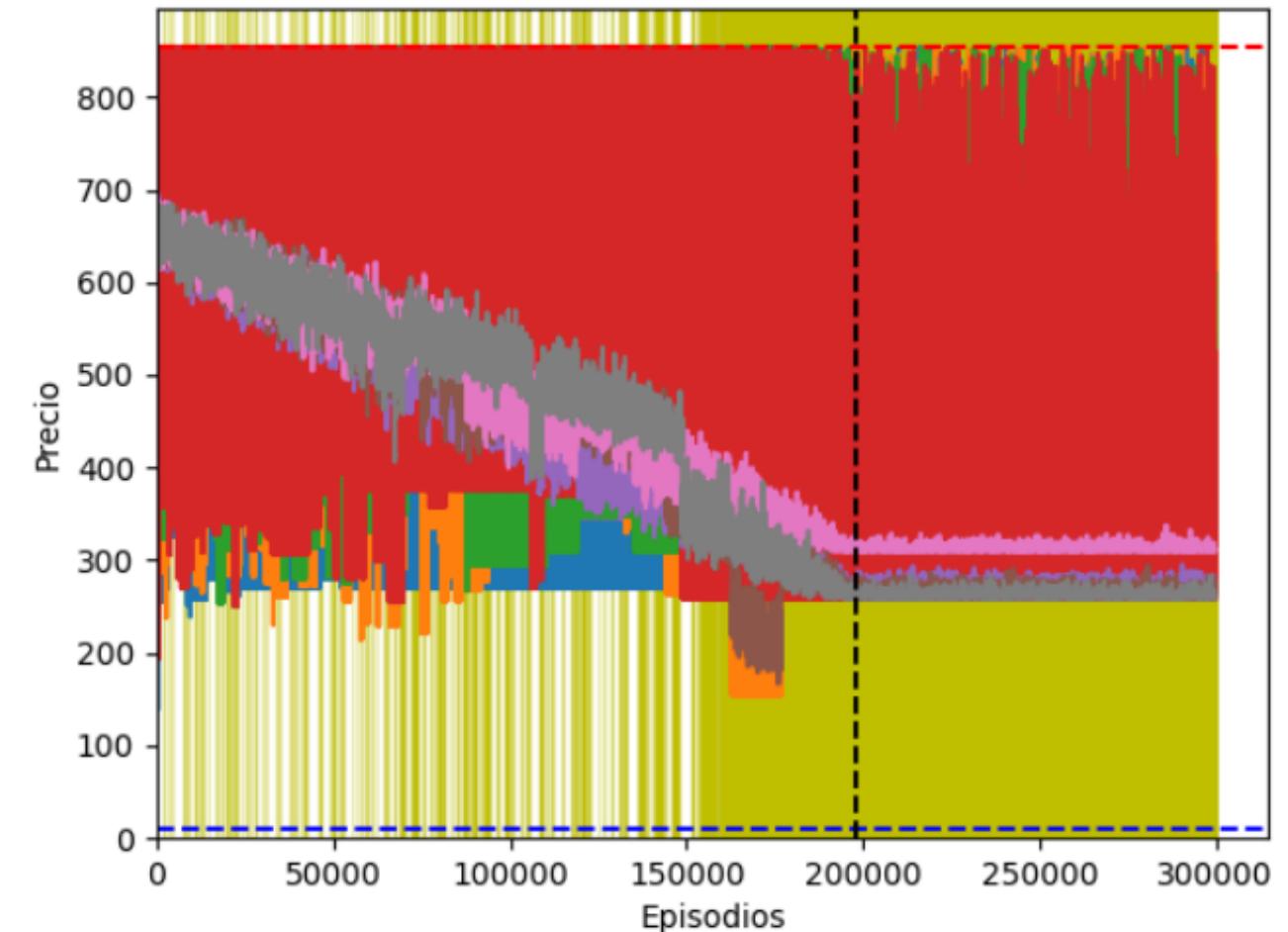
- Precios empresa 1
- Precios empresa 2
- Precios empresa 3
- Precios empresa 4
- Promedio móvil empresa 1
- Promedio móvil empresa 2
- Promedio móvil empresa 3
- Promedio móvil empresa 4
- Área de convergencia
- Tasa de aprendizaje mínima
- Beneficio promedio
- Tasa de aprendizaje mínima



n = 3



n = 4

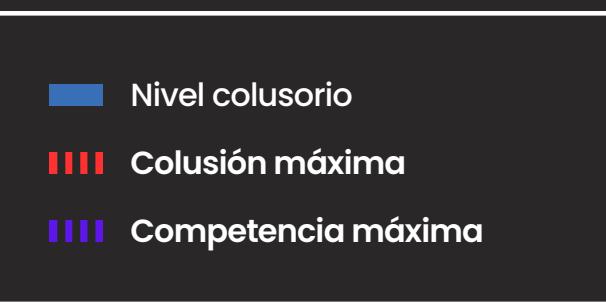


Elaboración propia mediante python

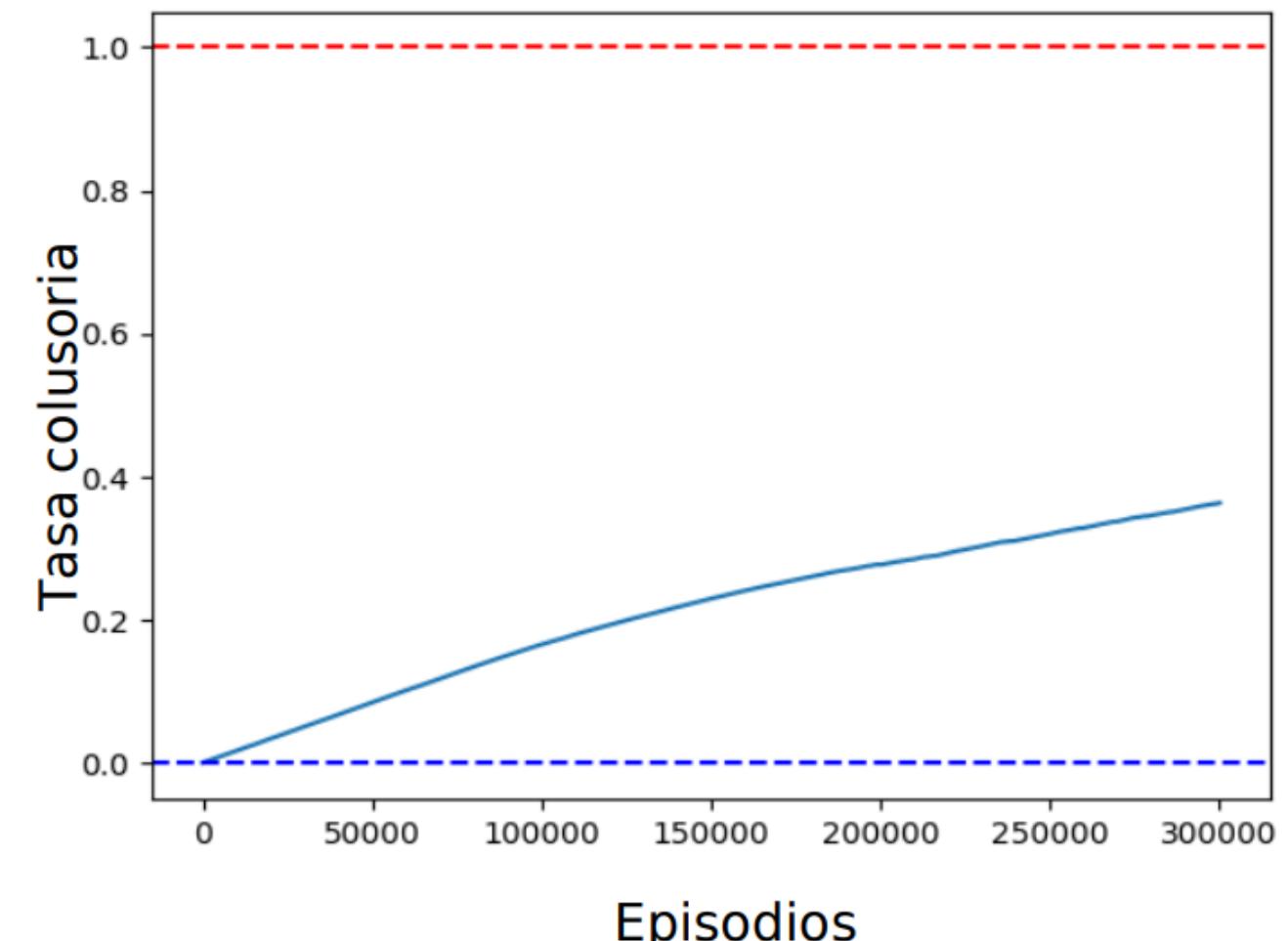
Resultados

Colusión en n=3 y n=4

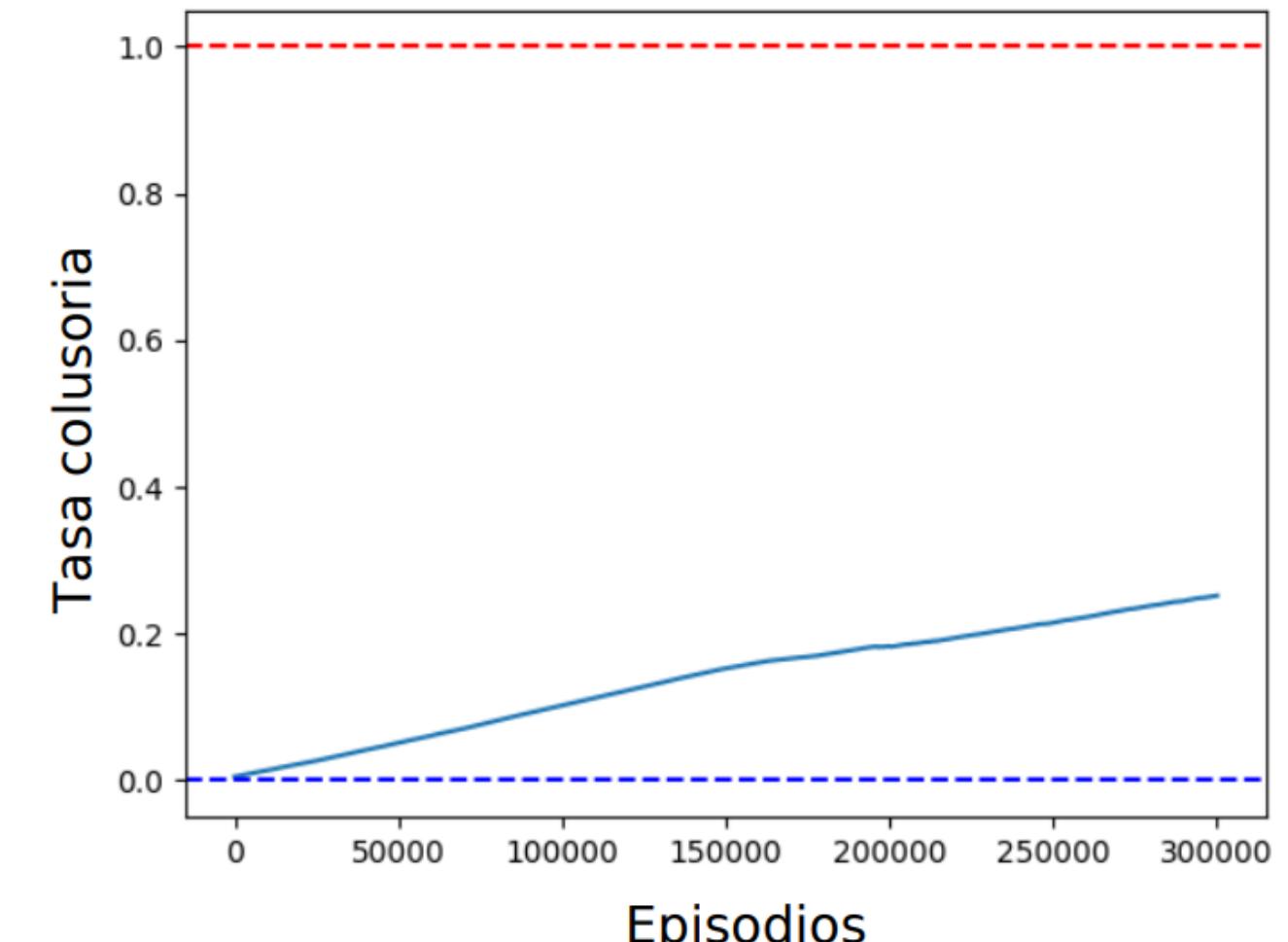
Leyenda



n = 3



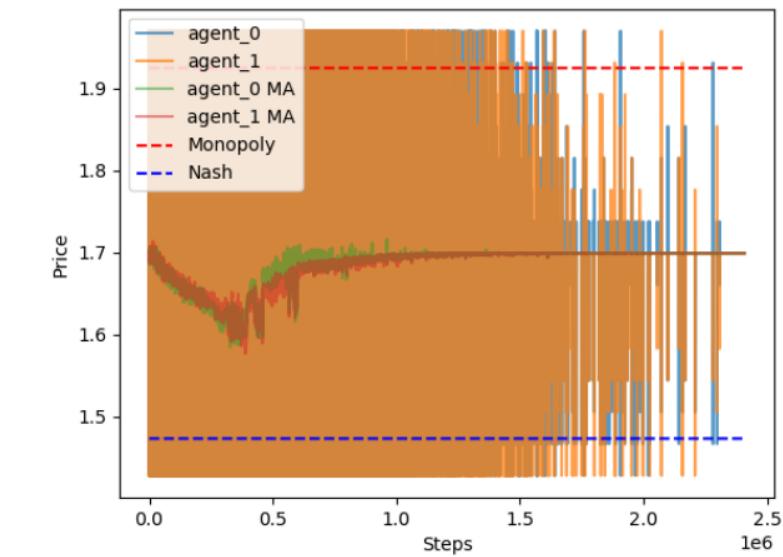
n = 4



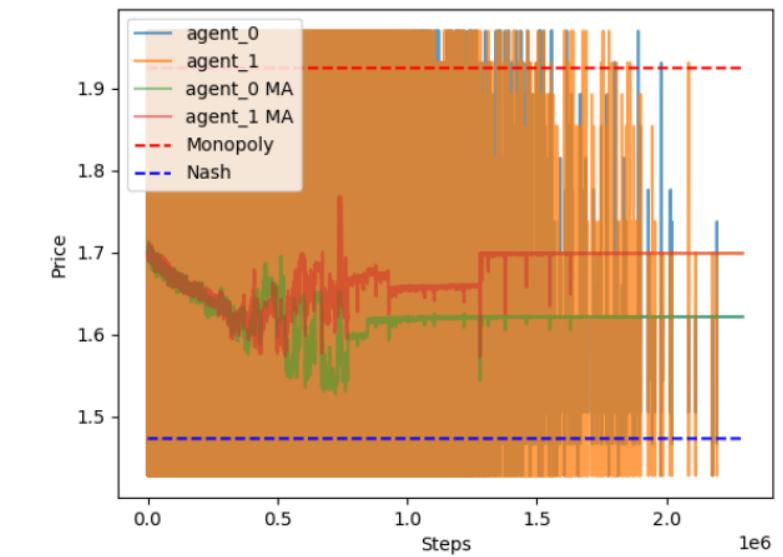
Elaboración propia mediante python

Conclusiones

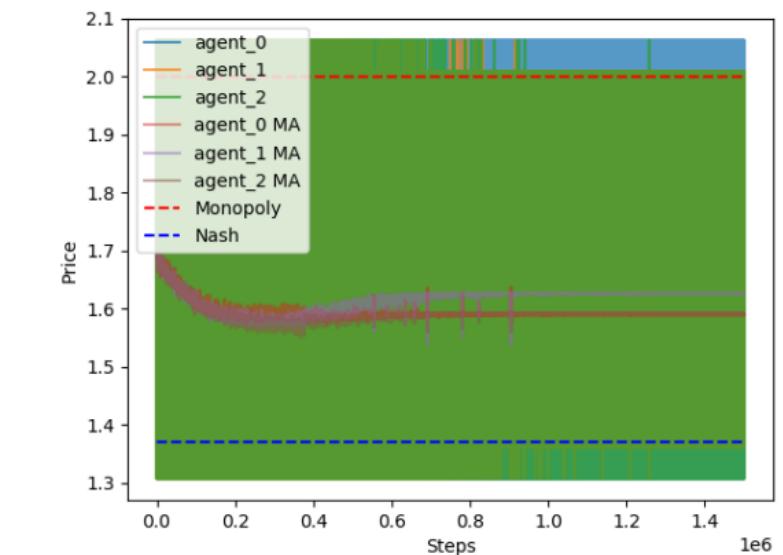
- 1 La tasa colusoria se incrementa conforme aumenta el número de interacciones
- 2 El algoritmo se centra en explotar la experiencia adquirida y se alcanzan áreas de convergencia significativas cuando se alcanza la tasa de aprendizaje mínima
- 3 El algoritmo no mantiene estable un precio monopolístico debido a los incentivos para desviarse
- 4 El número de agentes económicos disminuye significativamente la colusión
- 5 Los resultados obtenidos son acordes a los hallados en otras investigaciones y textos académicos
- 6 Objetivos establecidos alcanzados con éxito



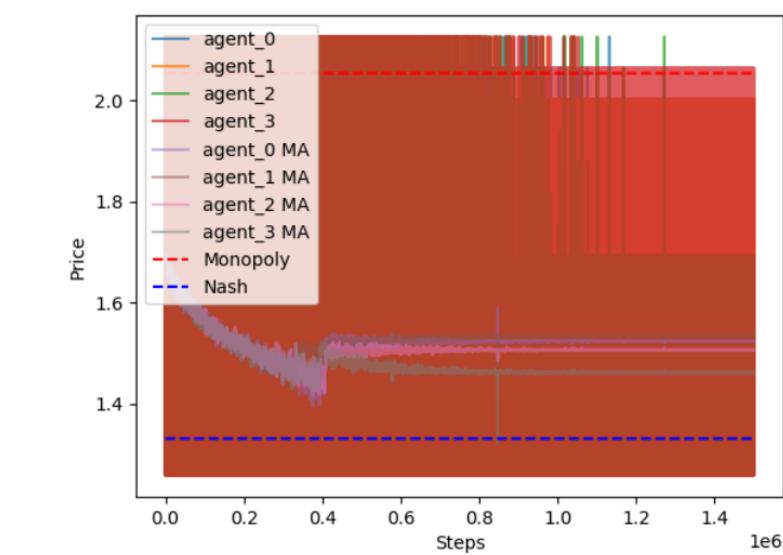
(a) Cooperating collusive equilibrium



(b) Dominating collusive equilibrium



(b) Three agents ($n = 3$)



(c) Four agents ($n = 4$)

Price	Profit	Δ (Profit Gain)
1.45	0.213	-0.085
1.5	0.234	0.098
1.55	0.254	0.271
1.6	0.272	0.433
1.65	0.289	0.580
1.7	0.304	0.709
1.75	0.317	0.819
1.8	0.327	0.905
1.85	0.333	0.965
1.9	0.337	0.996
1.95	0.337	0.996

Table 3.1: Profit gain Δ

Gráficos y tablas extraídos de Nicolas Lepore (2021), *AI Pricing Collusion: Multi-Agent Reinforcement Learning Algorithms in Bertrand Competition*. Bachelor's thesis, Harvard College