



## Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

# Implementación de un Sistema de Recomendación basado en Deep Learning para las plataformas del proyecto z17.

Autor:  
Alejandro Figueroa Rodríguez

Tutores: Msc. Yadier Perdomo Cuevas  
Msc. Aneyty Martín García  
Ing. Yosbel Falero Vento

# INTRODUCCIÓN

Crecimiento de Información



Plataformas digitales saturadas de datos



Usuarios ante un abrumador volumen de contenido



Emergen los SR como solución

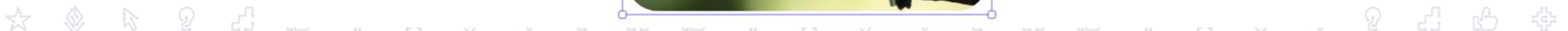


Dificultad de localización  
De contenido relevante y de calidad



# INTRODUCCIÓN

En Cuba ya existen plataformas que enfrentan este crecimiento de información.

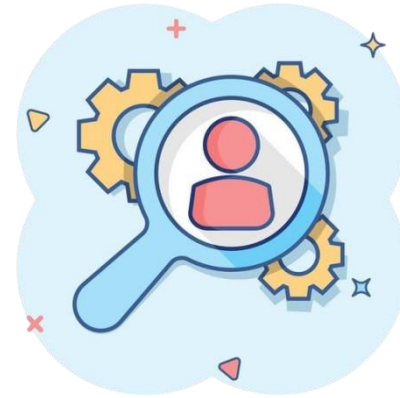


# PROBLEMÁTICA

Visibilidad limitada de  
contenido menos conocido



Preferencias de  
los usuarios



Personalización



Rendimiento



## PROBLEMA A RESOLVER

¿Como desarrollar un sistema de recomendación basado en Deep Learning que tenga en cuenta las preferencias de los usuarios y genere recomendaciones personalizadas para las plataformas del proyecto Z17?



# ● DISEÑO TEÓRICO DE LA INVESTIGACIÓN

## 🔍 Objeto de estudio

Los Sistemas de Recomendaciones

## 🖱️ Campo de acción

El Sistema de recomendación para las plataformas Picta, ToDus y Apklis.







# PREGUNTAS CIENTÍFICAS

1. ¿Cuáles son los principios teóricos y desarrollos recientes en la recomendación de información que sustenta la implementación de un sistema de recomendación basado en Deep Learning que tenga en cuenta las preferencias de los usuarios y genere recomendaciones personalizadas?
2. ¿Cómo implementar un sistema de recomendación basado en Deep Learning que tenga en cuenta las preferencias de los usuarios y genere recomendaciones personalizadas para las plataformas del proyecto Z17?
3. ¿Cómo evaluar a través de pruebas la calidad del sistema y de las recomendaciones para las plataformas del proyecto Z17?





# Deep Learning

- ❖ Campo de la inteligencia artificial (IA).
- ❖ Redes neuronales de múltiples capas.
- ❖ Se inspira en el funcionamiento del cerebro humano.
- ❖ Aprendizaje de patrones abstractos.



# Sistemas de Recomendaciones

- ❖ Basado en Contenido
- ❖ Basado en Filtrado colaborativo
- ❖ Basado en Popularidad
- ❖ Basado en Filtrado Demográfico
- ❖ Conversacionales
- ❖ Híbridos



# ESTUDIO DE SOLUCIONES SIMILARES



Sistema de recomendación  
de Youtube.



Sistema de recomendación  
de Netflix.

**Internacionales**



Sistema de recomendación  
de Google Play Store.



# ESTUDIO DE SOLUCIONES SIMILARES



Nacionales



Subsistema de recomendación  
de información para el  
buscador cubano Orión.

Módulo Recomendaciones  
del sistema para repositorios  
digitales REPXOS 3.0.



Desarrollo del Sistema de  
Recomendación de equipos de  
investigación para tesis de grado.



# ESTUDIO DE SOLUCIONES SIMILARES

Basado en Deep Learning



Personalización



Preferencias de usuarios



Enfoque híbrido



# ESTUDIO DE SOLUCIONES SIMILARES

Sistemas	Personalización	Preferencias de usuarios	Algoritmo híbrido	Basdo en Deep Learning
Sistema de Recomendación Youtube	x	x	x	x
Sistema de Recomendación de Netflix	x	x	x	x
Sistema de Recomendación de Google Play Store	x	x	x	x

# ESTUDIO DE SOLUCIONES SIMILARES

Sistemas	Personalización	Preferencias de usuarios	Algoritmo híbrido	Basdo en Deep Learning
Subsistema de recomendación de información para el buscador cubano Orión	x	x	x	-
Módulo Recomendaciones del sistema para repositorios digitales REPXOS 3.0	x	x	x	-
Desarrollo del Sistema de Recomendación de equipos de investigación para tesis de grado	x	x	x	-



# TECNOLOGIAS Y HERRAMIENTAS



# PROPUESTA DE SOLUCIÓN

## Retroalimentación

### Explicit Feedback



- Thumbs up / down
- Star ratings from 1 to 5
- Written reviews



#### Customer Reviews

★★★★☆ 4.0

5 star	<div><div></div></div>	62%
4 star	<div><div></div></div>	15%
3 star	<div><div></div></div>	8%
2 star	<div><div></div></div>	1%
1 star	<div><div></div></div>	14%

### Implicit Feedback



- Played songs / videos
- Purchased items
- Browsing history

#### Your Browsing History



**Sterlizia**  
Potted Plant



**SWATCH #SB02**  
Man Watch



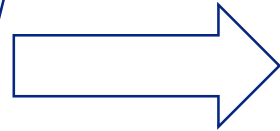
**The Tempest**  
Book

# PROPUESTA DE SOLUCIÓN

## Etapas



Generación de  
Candidatos



Clasificación



Re-  
Clasificación

# PROPUESTA DE SOLUCIÓN

## Etapa

- ❖ Recuperación / Búsqueda
- ❖ Realizado por 1 Modelo



Generación de  
Candidatos

- ❖ Datos Implícitos
- ❖ X cantidad de candidatos

# PROPUESTA DE SOLUCIÓN

## Etapa

- ❖ Recibe las candidatos de la etapa anterior
- ❖ Ordenamiento de candidatos



## Clasificación

- ❖ Datos Explícitos
- ❖ X cantidad de candidatos

# PROPUESTA DE SOLUCIÓN

## Etapa

- ❖ Recibe las candidatos de la etapa anterior



## Re- Clasificación

- ❖ Equidad
- ❖ Diversidad
- ❖ Actualidad

- ❖ Da como salida las recomendaciones del sistema

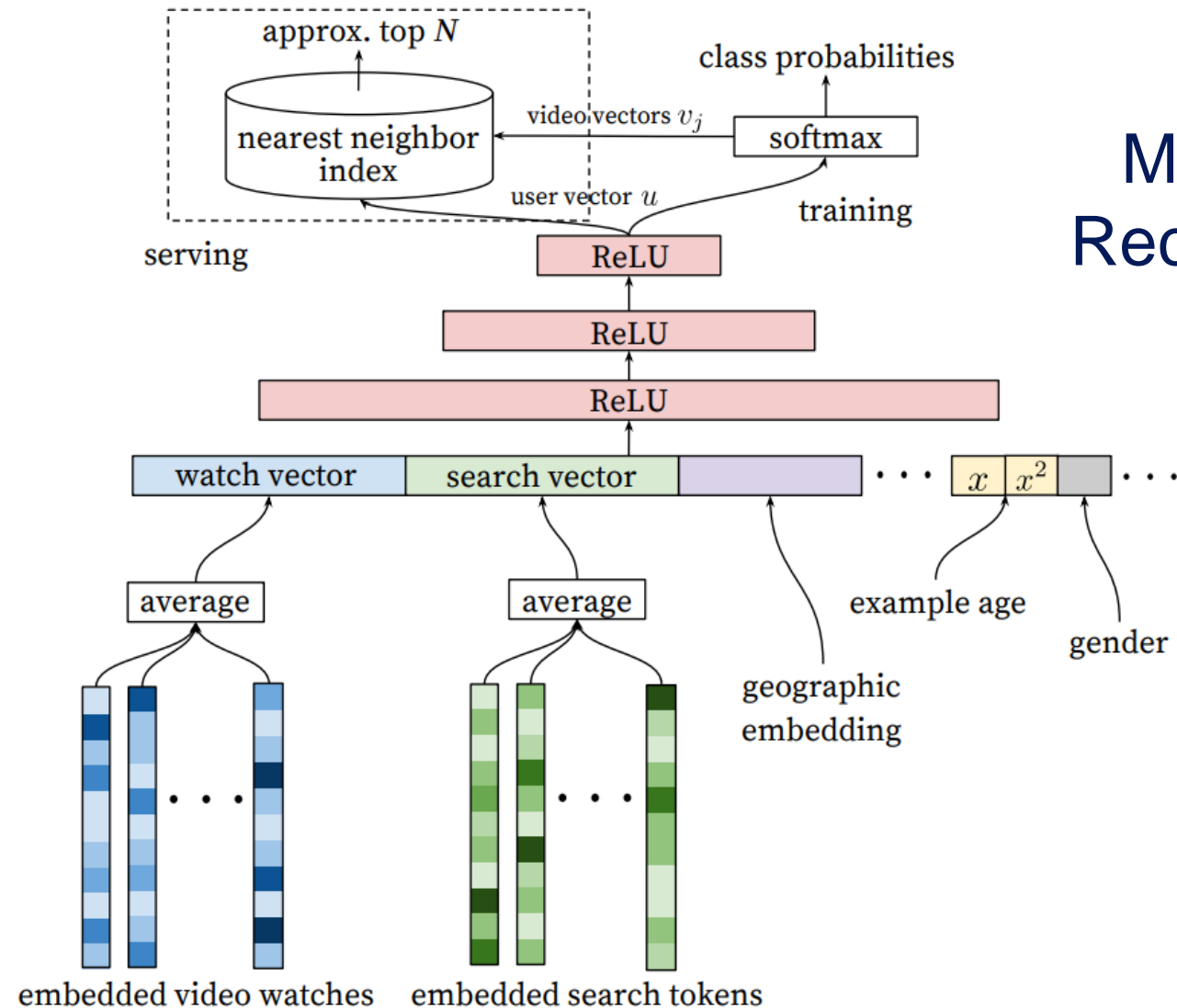
# PROPUESTA DE SOLUCIÓN



Modelos de  
Deep Learning

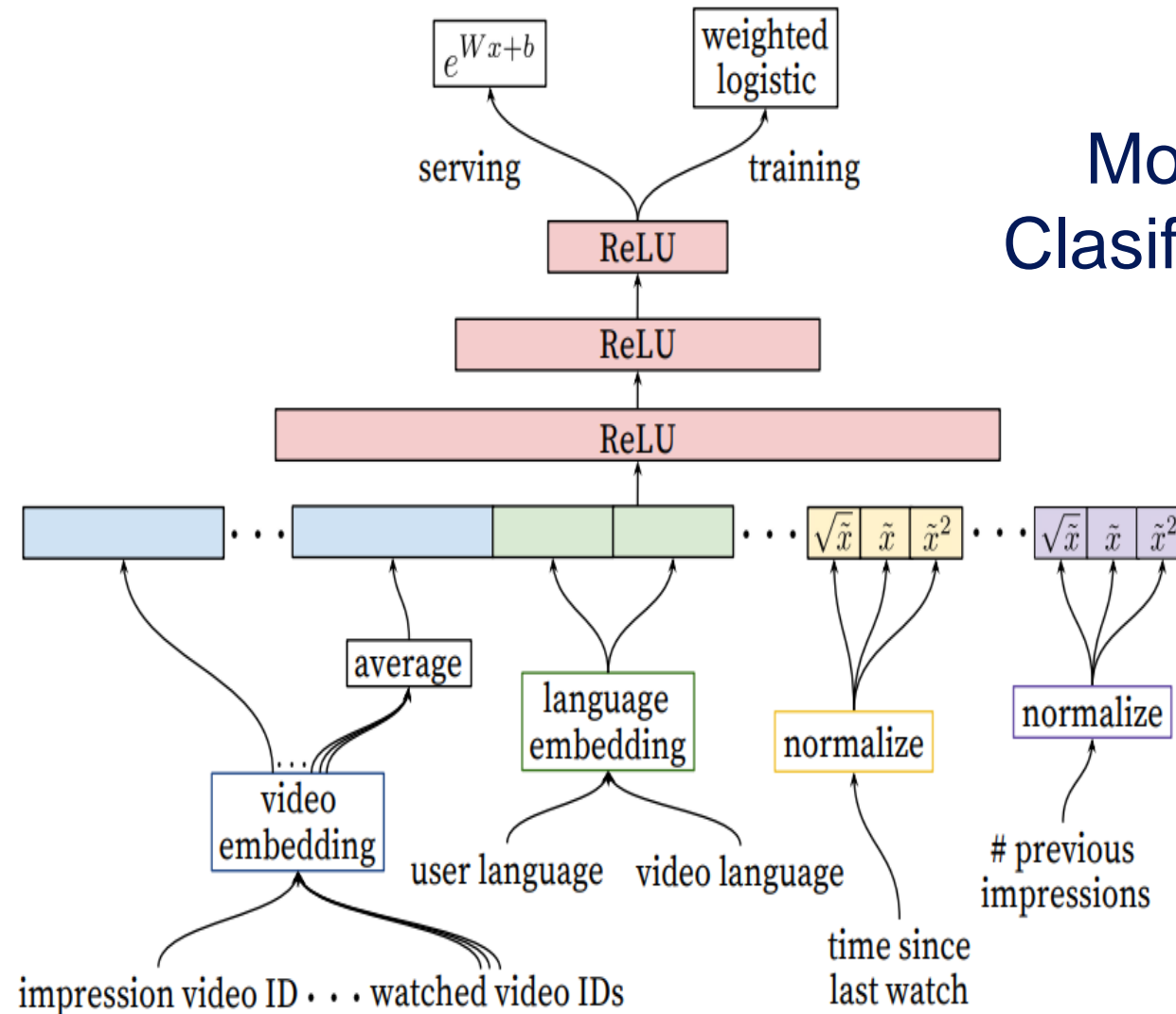


# PROPUESTA DE SOLUCIÓN



Modelo de  
Recuperación

# PROPUESTA DE SOLUCIÓN



Modelo  
Clasificación

# REQUISITOS

## Requisitos Funcionales (13)

- ❖ Entrenar Modelo(s)
- ❖ Actualizar Modelo(s)
- ❖ Recomendar

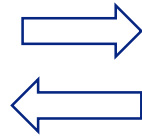


## Requisitos No Funcionales (12)

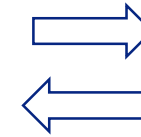
- ❖ Usabilidad (2)
- ❖ Seguridad (2)
- ❖ Rendimiento (2)
- ❖ Restricciones de Implementación y Diseño (4)
- ❖ Software (1)
- ❖ Hardware (1)

# Arquitectura

Modelo



Controlador



Vista



# PATRONES DE DISEÑO

## Patrones GRASP

Experto

Creador

Controlador

## Patrones GoF

Singleton

Template Method

# PATRONES DE DISEÑO

## Experto

```
class RetrievalModel(tfrs.models.Model):
    def __init__(self, ...

    def call(self, inputs): ...

    def compute_loss(self, features, training=False): ...

    def fit_model(self, ...

    def evaluate_model(self, cached_test, cached_train) -> None: ...

    def index_model(self) -> tfrs.layers.factorized_top_k.BruteForce: ...

    def predict_model(self, ...

    def save_model(self, path: str, dataset: tf.data.Dataset) -> None: ...

    def load_model(self, path: str, cached_train, cached_test) -> None: ...
```

# PATRONES DE DISEÑO

## Template Method

```
class AbstractDataPipeline(ABC):
    @abstractmethod
    def __init__(self) -> None: ...

    @abstractmethod
    def __str__(self) -> Text: ...

    @abstractmethod
    def get_path(self, path: str) -> str: ...

    @abstractmethod
    def read_csv_data(self, paths: List[str]) -> Tuple[pd.DataFrame]: ...

    @abstractmethod
    def load_dataset(self, path: str): ...

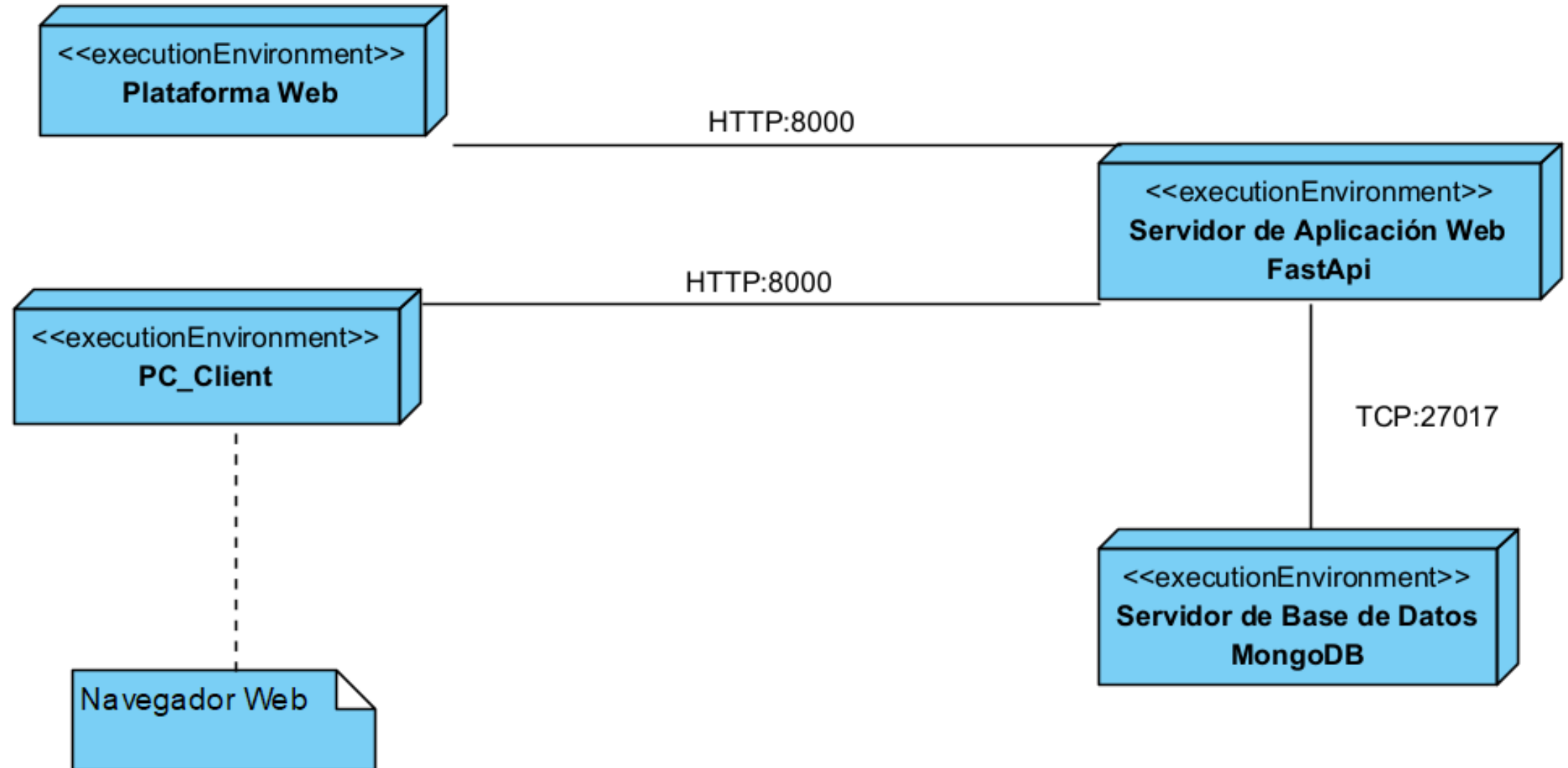
    @abstractmethod
    def merge_data(self, ...

    @abstractmethod
    def convert_to_tf_dataset(self, data: pd.DataFrame) -> tf.data.Dataset: ...

    @abstractmethod
    def load_vocabularies(self, path: str): ...
```



# Diagrama de Despliegue



# Estrategia de Pruebas

## Pruebas Funcionales

Prueba de Caja  
Negra

Prueba de Caja  
Blanca

Técnica Ladov

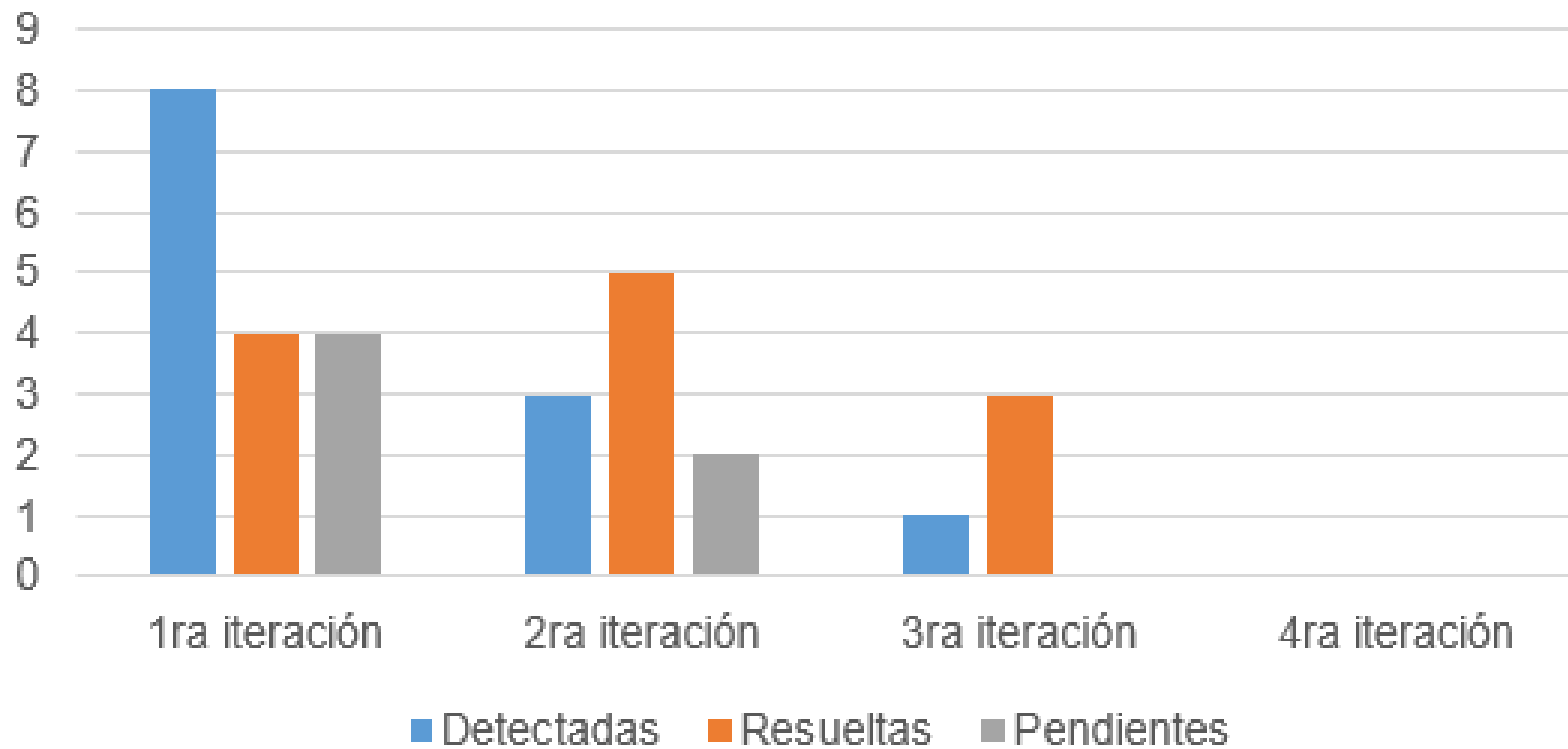
## Pruebas No Funcionales

Prueba de  
Rendimiento

Prueba de  
Seguridad

# Prueba de Caja Negra

## No conformidades

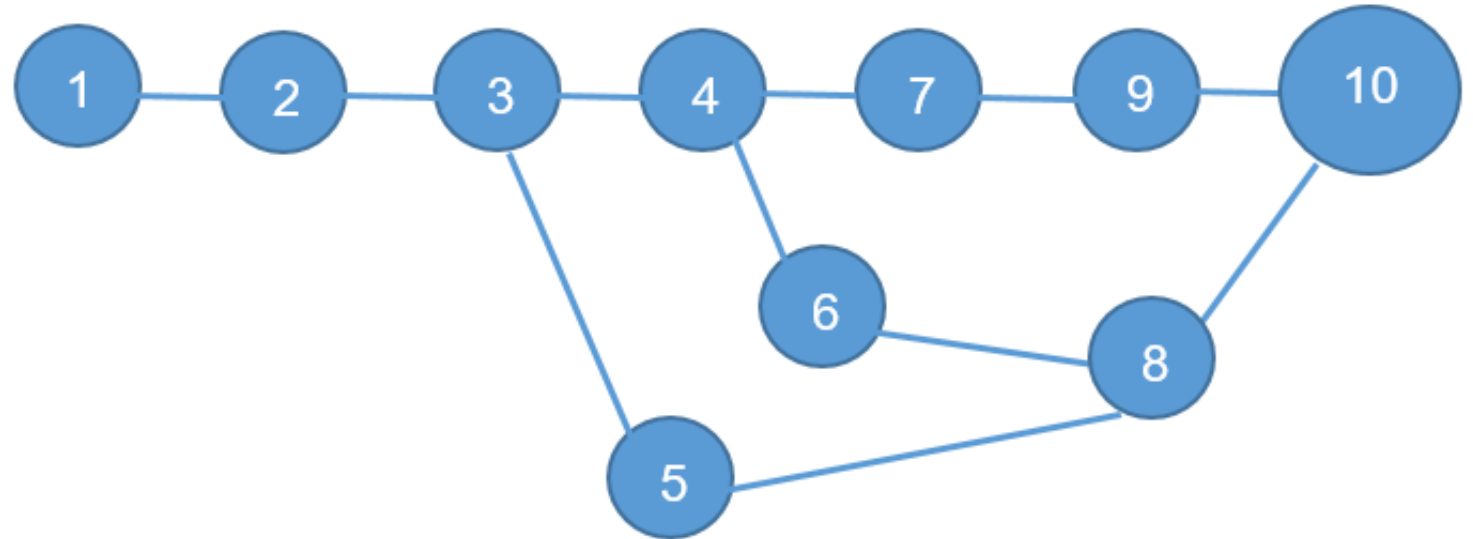


- Ortografía
- Interfaz
- Funcionalidades

Total de no conformidades encontradas: 12

Total Resueltas: 12

# ● Prueba de Caja Blanca



**Cantidad de caminos mínimos = 3**

**Complejidad Ciclomática = 3**

# Prueba de Seguridad

Tipo	Cantidad	Descripción	Recomendaciones
<b>Falsificación de petición (CSRF) en API de recomendaciones.</b>	1	El sistema no evalúa que solo las plataformas autorizadas puedan realizar las peticiones para generar recomendaciones.	Crear un sistema de autenticación y autorización de plataformas o sistemas externos para evitar intrusos no permitidos.
<b>Ataques de inyección.</b>	1	El sistema permite la inyección de código en el requisito Recomendar.	Crear validaciones para los parámetros recibidos en las funciones y procesos que intervienen en el requisito Recomendar.

# ● Prueba de Rendimiento

## Servidor Web

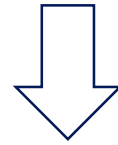
Usuarios	Muestra	Media	Min	Max	% Error	Rendimiento	KB/s Recibidos
200	200	233	12	450	0.00%	150.0/sec	699.34
500	500	1275	88	2151	0.00%	154.0/sec	718.87
500	2000	1536	68	3331	0.00%	220.8/sec	1029.06

## Sistema de Recomendación

- Tiempo aproximado de entrenamiento: 12 horas.
- Tiempo aproximado de actualización: 15 Minutos.
- Tiempo aproximado de inferencia o recomendación del sistema: 1 – 2.50 Segundos.

# Validación de la Solución

+1	Máximo de satisfacción
0.5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0.5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción



Resultado	Cantidad	%
Total de usuarios de la muestra	25	100
Máximo de satisfacción	20	80
Más satisfecho que insatisfecho	3	12
No definida	0	0
Más insatisfecho que satisfecho	2	8
Clara insatisfacción	0	0
Contradictoria	-	-

$$ISG = \frac{A(+1) + B(0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

$$ISG = \frac{20(+1) + 3(0.5) + 0(0) + 2(-0.5) + 0(-1)}{25}$$



$$ISG = 0.82$$



# CONCLUSIONES

- El análisis de sistemas de recomendaciones existentes evidenció la ausencia de una solución que cumpliera con los requisitos del cliente.
- Se implemento un sistema de recomendación basado en Deep Learning que tiene en cuenta las preferencias de los usuarios y genera recomendaciones personalizadas para las plataformas del proyecto Z17

## CONCLUSIONES

- La validación del problema de investigación mediante la técnica LadoV demuestra que el sistema de recomendación desarrollado tiene en cuenta las preferencias de los usuarios y genera recomendaciones personalizadas para las plataformas del proyecto Z17 (Apklis, Picta y ToDus).

# RECOMENDACIONES

- ✓ Aumentar la cantidad de datos a utilizar para entrenar los modelos y mejorar el hardware utilizado en el proceso de entrenamiento para mejorar tanto el rendimiento como la precisión.
- ✓ Probar el Sistema con las 3 plataformas.



Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

# Implementación de un Sistema de Recomendación basado en Deep Learning para las plataformas del proyecto z17.

Autor:

Alejandro Figueroa Rodríguez

Tutores: Msc. Yadier Perdomo Cuevas

Msc. Aneyty Martin García

Ing. Yosbel Falero Vento