

# Interfaz gráfica para interactuar con resultados de evaluación orientada a sistemas de recomendación

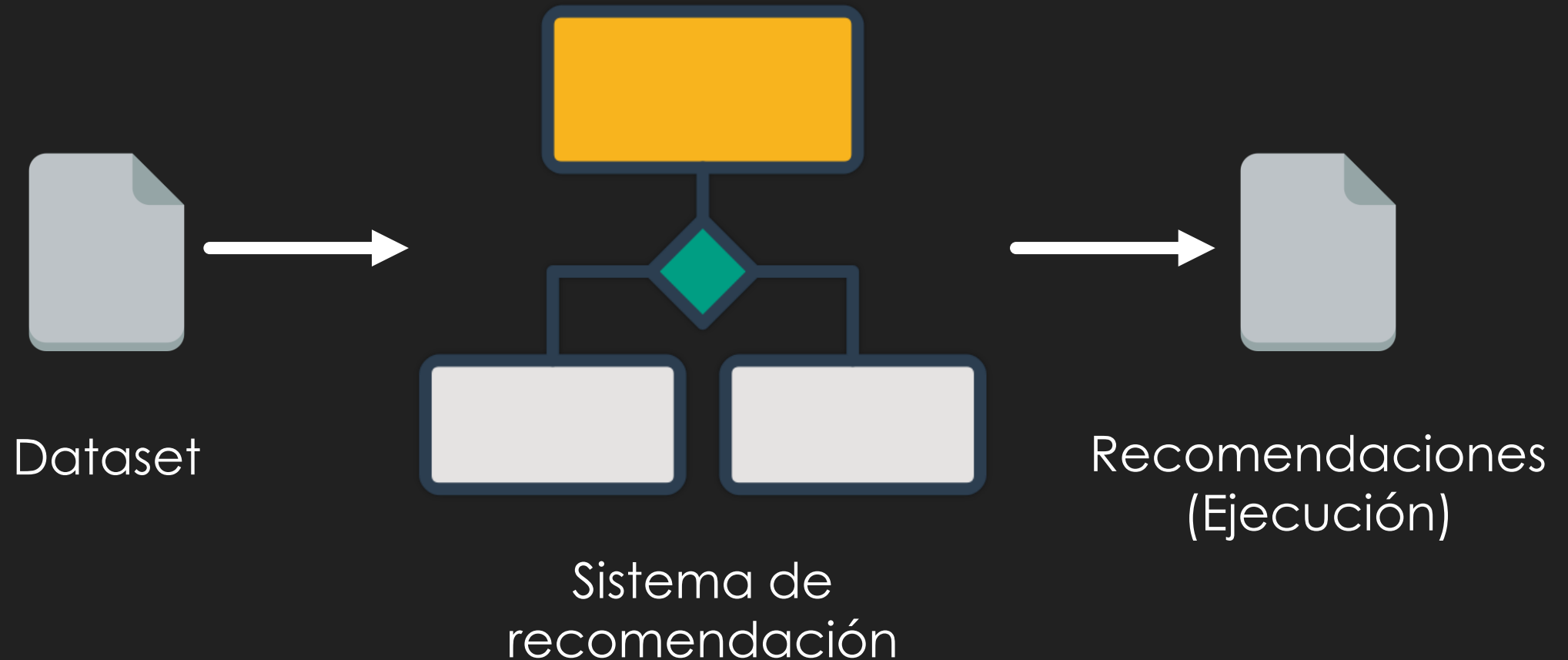
Ricardo Morato Mateos

# OBJETIVO

- Desarrollar interfaz gráfica
- Entender sistemas de recomendación y evaluación

# ¿QUE ES UN SISTEMA DE RECOMENDACIÓN?

# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?

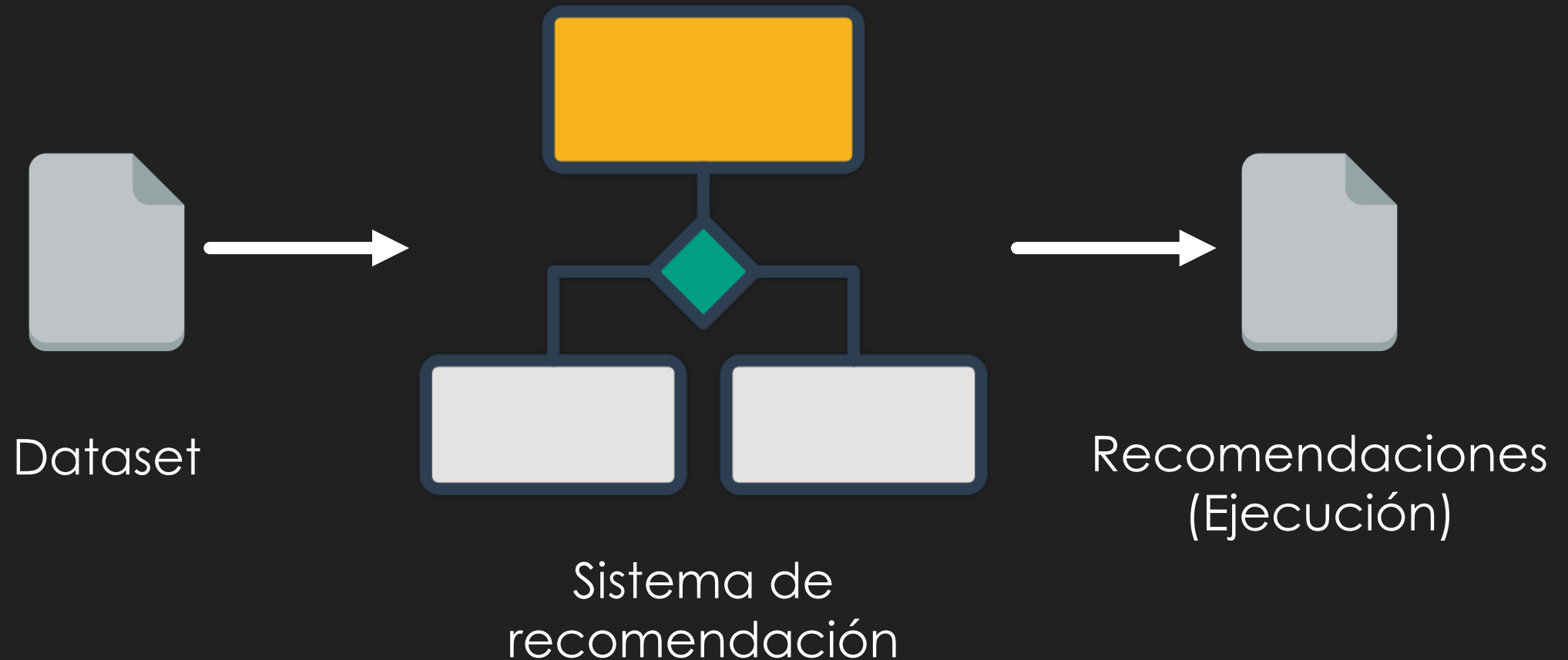


# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?

○ Datos de entrada

UserId	MoviedId	Rating
1	192	4.0
1	195	5.0
1	9	3.5

# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?



# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?

○ Datos de salida

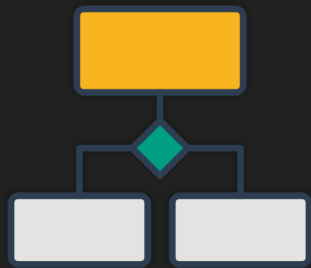
UserId	MovielId	Rating
1	318	5.0
1	902	5.0
1	475	5.0
1	405	4.0
1	306	3.94
1	872	3.19
1	597	2.96
2	...	...

# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?

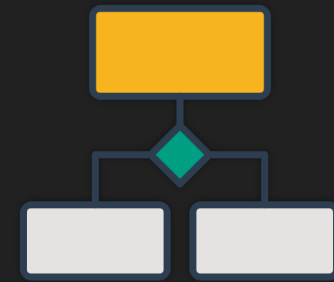




# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?



Sistema 1



Sistema N

# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?

- Basadas en error
  - MAE
  - RMSE

- Basadas en precisión
  - Precisión
  - Recall
  - F
  - MRR
  - MAP
  - NDCG

# DESARROLLO DEL PROYECTO

# PATRÓN DE DISEÑO

## ○ MVC

- Permite abstraer cada parte de la aplicación y que, al realizar un cambio en alguna de ellas, no afecte a las demás.

- Tres partes:

- Modelo: Se encarga del acceso y guardado de los datos en la base de datos.

- Vista: Se encarga de la representación de los datos y de las interacciones del usuario.

- Controlador: Se encarga de realizar la lógica de negocio.

# VISTA

- Consultar, subir y borrar datasets y ejecuciones, visualizar métricas y gráficas.
- HTML
- CSS
- JavaScript

# VISTA - CSS

- Librería Bulma
  - Ligera.
  - Diseño moderno.
  - No utiliza jQuery.

# VISTA - JAVASCRIPT

- React
  - Utilización de un DOM virtual, aumentando la velocidad de renderizado.
  - Desarrollada por Facebook.
  - Componentes con estado. Al cambiarlo, se renderiza ese componente.
  - Ligero: sólo 132 Kb. Angular más de 500 Kb.
  - Añadir funcionalidad añadiendo librerías externas.
  - Transpilar (Compilar a una versión más antigua de JavaScript).

# CONTROLADOR

- Cálculo de métricas y comunicación de la vista con el modelo.
- JavaScript
  - Razones:
    - No mantenimiento de dos servidores.
    - Interés personal.
- Node.js



# CONTROLADOR - NODE.JS

- Basado en motor V8 de Chrome.
- Asíncrono.
- Monohilo.
  
- Escasas librerías sobre sistemas de recomendación.

# MODELO

- Acceso y guardado de datasets, ejecuciones y métricas.
- MongoDB
  - Rapidez.
  - Búsqueda por atributos.
  - Facilidad de integración con Node.js gracias a la librería Mongoose.

# MODELO

- Descripción de campos
  - Tipo.
  - Obligatorio.
  - Valores por defecto.

```
const DatasetSchema = new Schema({
  name: {
    type: String,
    required: [true, 'Empty name']
  },
  description: {
    type: String,
    required: [true, 'Empty description']
  },
  train_file: {
    type: String,
    required: [true, 'Empty file']
  },
  test_file: {
    type: String,
    required: [true, 'Empty file']
  },
  timestamp: {
    type: Date,
    default: Date.now
  }
})
```

# PRUEBAS

# PRUEBAS

- Pruebas unitarias
  - Caja negra.
  - Cobertura de código: 94%.
- Pruebas de integración
  - Pruebas para el API.
- Pruebas de sistema
  - Verificar que todos los requisitos funcionales se cumplen.

# APLICACIÓN

**MEJORAS**

# MEJORAS

- Mejoras en rendimiento
  - Guardar las métricas en base de datos.
  - Instanciar el objeto que calcula las métricas en la primera petición de una ejecución.
- Mejoras futuras
  - Sistema completo de recomendaciones.
  - Evaluaciones desde un fichero externo.
  - Agregación de usuarios.



# CONCLUSIONES

# CONCLUSIONES

- Este trabajo se desarrolló para dar una herramienta funcional a desarrolladores de sistemas de recomendación.
- Se ha aprendido sobre los tipos de sistemas, los algoritmos que usan estos sistemas para generar recomendaciones y, por supuesto, sobre su evaluación.
- Se ha desarrollado todo el proyecto en JavaScript, un lenguaje que el autor no había utilizado nunca de forma exhaustiva.

# PREGUNTAS

# ÍNDICE

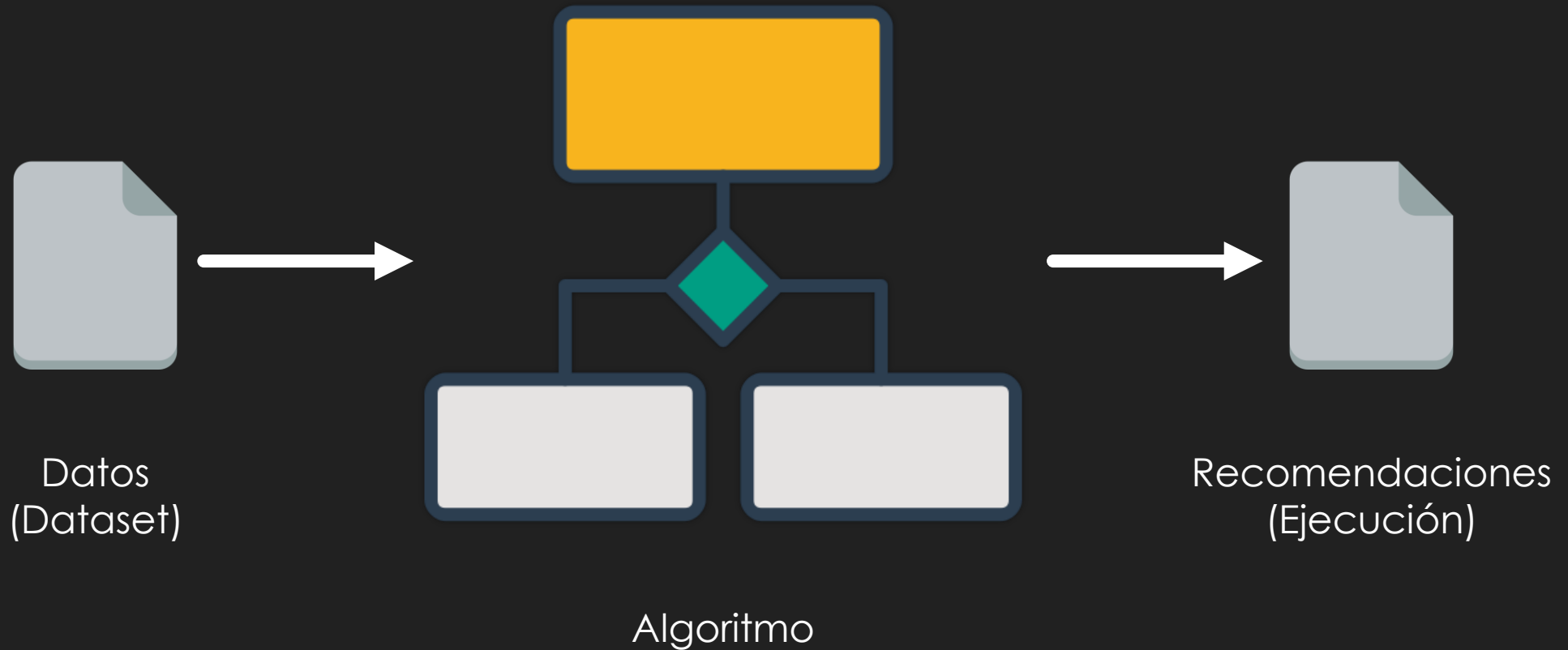
- Estado del arte
- Estudio de las tecnologías utilizadas
- Diseño
- Desarrollo
- Pruebas
- Conclusiones
- Aplicación

# ESTADO DEL ARTE

# ¿QUÉ ES UN SISTEMA DE RECOMENDACIÓN?

- Un sistema de recomendación es un conjunto de herramientas y software que proporcionan sugerencias que sean de utilidad para un usuario.
- Se basan en algoritmos de predicción.

# FLUJO



# TIPOS DE SISTEMAS DE RECOMENDACIÓN

- Sistemas basados en contenido: Recomiendan ítems asociados a los ítems que gustaron a un usuario en el pasado.
- Sistemas de filtrado colaborativo: Recomiendan ítems en base a los gustos de otros usuarios similares.



# EVALUACIÓN DE LOS SISTEMAS DE RECOMEDACIÓN

- Online: Es necesario la implicación de los usuarios.
- Offline:
  - Comparan ejecuciones (recomendaciones) contra los datos originales (datasets)
  - Métricas basadas en error
  - Métricas basadas en precisión

# MÉTRICAS BASADAS EN ERROR

- Métodos estadísticos que se basan en el error.
- Principal inconveniente: No hacen distinciones entre las posiciones de los ítems.
- Métricas:
  - MAE
  - RMSE

# MÉTRICAS BASADAS EN PRECISIÓN

- Miden la cantidad de elementos recuperados relevantes y no relevantes.
- Métricas:
  - Precisión
  - Recall
  - F
  - MRR
  - MAP
  - NDCG
- Se suelen calcular hasta un límite (cutoff).

# ESTUDIO DE LAS TECNOLOGÍAS UTILIZADAS

# VISTA

- HTML
- CSS
- JavaScript

# REACT

- Desarrollada por Facebook.
- Utilización de un DOM virtual, aumentando la velocidad de renderizado.
- Componentes con estado. Al cambiarlo, se renderiza ese componente.
- Ligero: sólo 132 Kb. Angular más de 500 Kb.
- Añade funcionalidad añadiendo librerías externas.
- Transpilar (Compilar a una versión más antigua de JavaScript).

# CONTROLADOR

- JavaScript
  - Razones:
    - No mantenimiento de dos servidores.
    - Interés personal.
- Node.js

# NODE.JS

- Basado en motor V8 de Chrome.
- Asíncrono.
- Monohilo.
  
- Escasas librerías sobre sistemas de recomendación.



# MODELO

- Base de datos NoSQL
  - Rapidez.
- MongoDB
  - Facilidad de integración con Node.js gracias a la librería Mongoose.
  - Búsqueda por atributos.

**DISEÑO**

# REQUISITOS FUNCIONALES

- Almacenar, borrar, consultar y listar datasets.
- Almacenar, borrar, consultar, comparar y listar ejecuciones.
- Cálculo métricas sobre las ejecuciones.

# REQUISITOS NO FUNCIONALES

- Diseño moderno y sencillo.
- Diseño intuitivo: El usuario debe poder navegar por la aplicación sin necesidad de la creación de un manual de usuario.
- Rapidez.

# DESARROLLO

# PATRÓN DE DISEÑO

- MVC
  - Permite abstraer cada parte de la aplicación y que, al realizar un cambio en alguna de ellas, no afecte a las demás.
  - Tres partes:
    - Modelo: Se encarga del acceso, recuperación y guardado de los datos en la base de datos.
    - Vista: Se encarga de la representación de los datos y de las interacciones del usuario.
    - Controlador: Se encarga de realizar la lógica de negocio.

# MODELO

- Descripción de campos
  - Tipo
  - Obligatorio
  - Valores por defecto

```
const DatasetSchema = new Schema({
  name: {
    type: String,
    required: [true, 'Empty name']
  },
  description: {
    type: String,
    required: [true, 'Empty description']
  },
  train_file: {
    type: String,
    required: [true, 'Empty file']
  },
  test_file: {
    type: String,
    required: [true, 'Empty file']
  },
  timestamp: {
    type: Date,
    default: Date.now
  }
})
```

# VISTA - CSS

- Librería Bulma.
  - Ligeras.
  - No utiliza jQuery.



# VISTA - JAVASCRIPT

- Cada pantalla es un componente distinto con toda la lógica de esa pantalla.
- Google Charts.
  - Gran variedad de gráficos.
  - Sencilla de usar.
  - Personalizable.
  - Actualización en tiempo real.

# CONTROLADOR

- Consideraciones generales:
  - Devolución en formato JSON.
  - Tratamiento de errores genérico.
- Endpoints para:
  - Subida de datasets y ejecuciones.
  - Consulta y listado de datasets y ejecuciones.
  - Borrado de datasets y ejecuciones.
  - Consulta de métricas.

# MEJORAS DE RENDIMIENTO

- Guardar las métricas en base de datos.
- Instanciar el objeto que calcula las métricas una única vez por cada ejecución.

# PRUEBAS

# PRUEBAS UNITARIAS

- Comprobar que un fragmento de código tiene el comportamiento esperado.
- Caja negra.
- Cobertura de código: 94%.

# PRUEBAS DE INTEGRACIÓN

- Verifican el correcto funcionamiento de varios componentes que actúan en conjunto.
- Pruebas para el API.

# PRUEBAS DE SISTEMA

- Pruebas que demuestran el funcionamiento de la aplicación de forma general.
- Verificar que todos los requisitos funcionales se cumplen.

# CONCLUSIONES

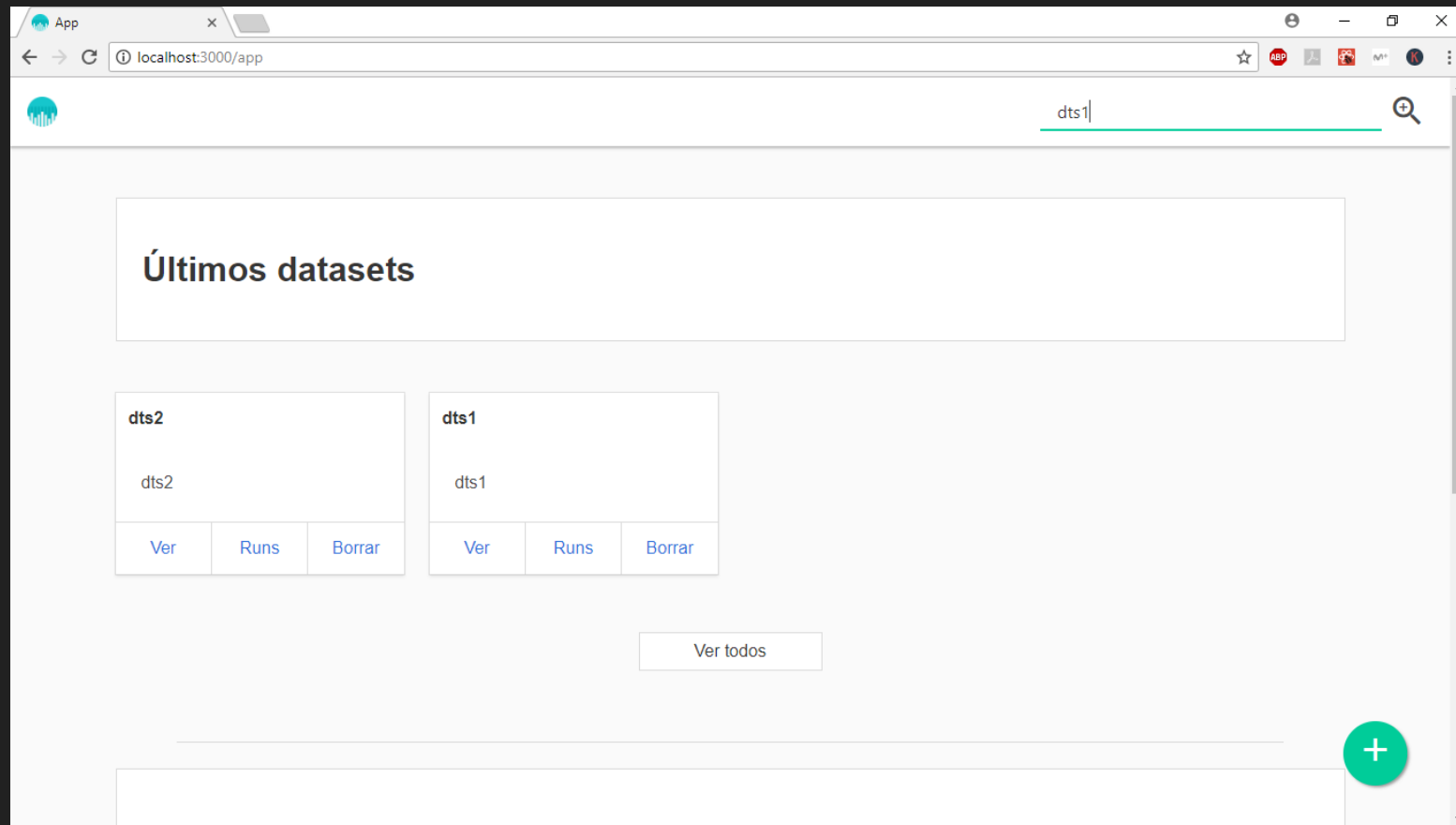


# CONCLUSIONES

- Este trabajo se desarrolló para dar una herramienta funcional a desarrolladores de sistemas de recomendación.
- Se ha aprendido sobre los tipos de sistemas, los algoritmos que usan estos sistemas para generar recomendaciones y, por supuesto, sobre su evaluación.
- Se ha desarrollado todo el proyecto en JavaScript, un lenguaje que el autor no había utilizado nunca de forma exhaustiva.

# APLICACIÓN

# PANTALLA PRINCIPAL



# PANTALLA PRINCIPAL

The screenshot shows a web browser window with the URL localhost:3000/app. The page content is as follows:

## Estadísticas

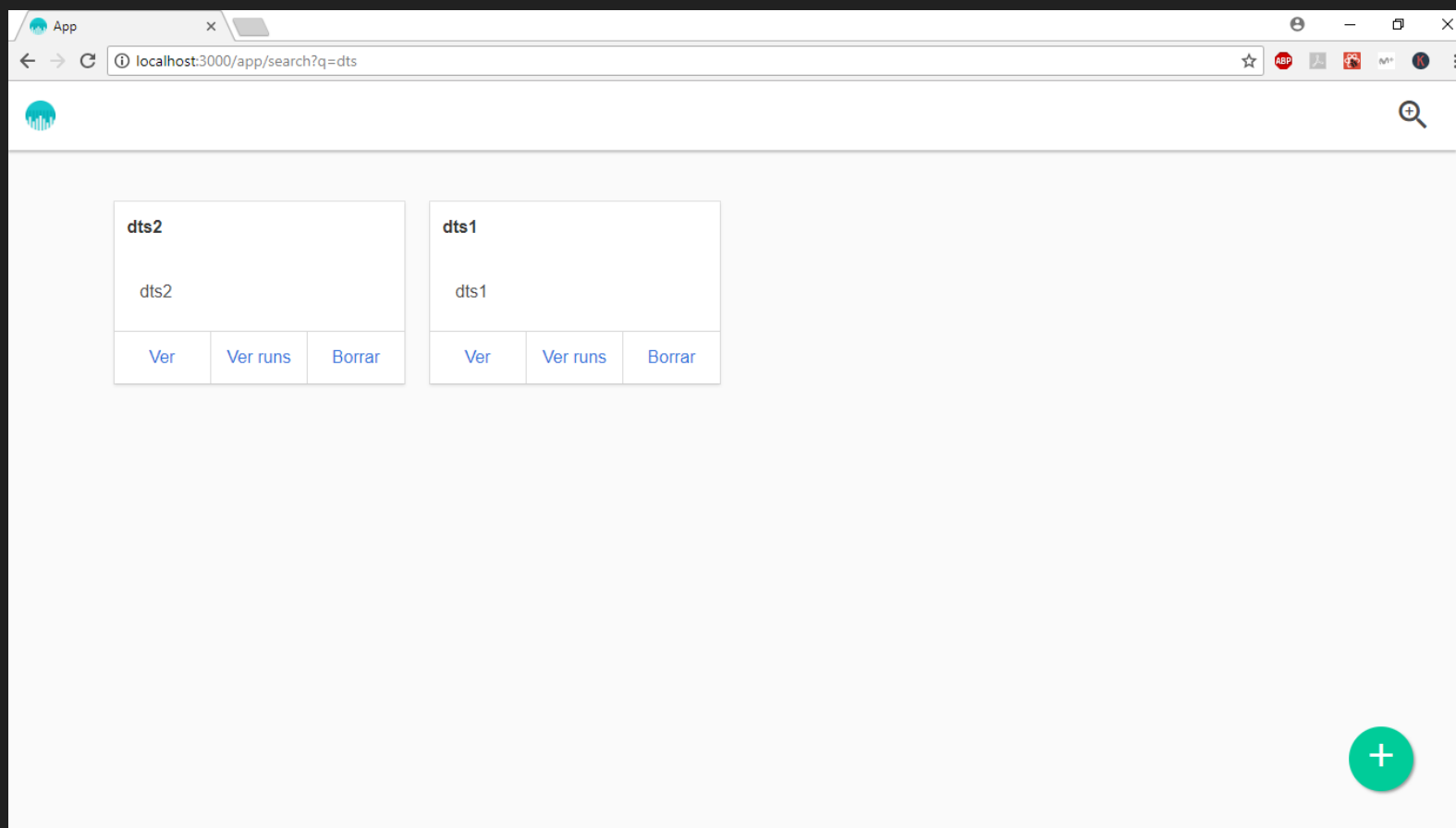
NÚMERO DE DATASETS	NÚMERO DE RUNS
2	4

### Has consultado las siguientes métricas estas veces

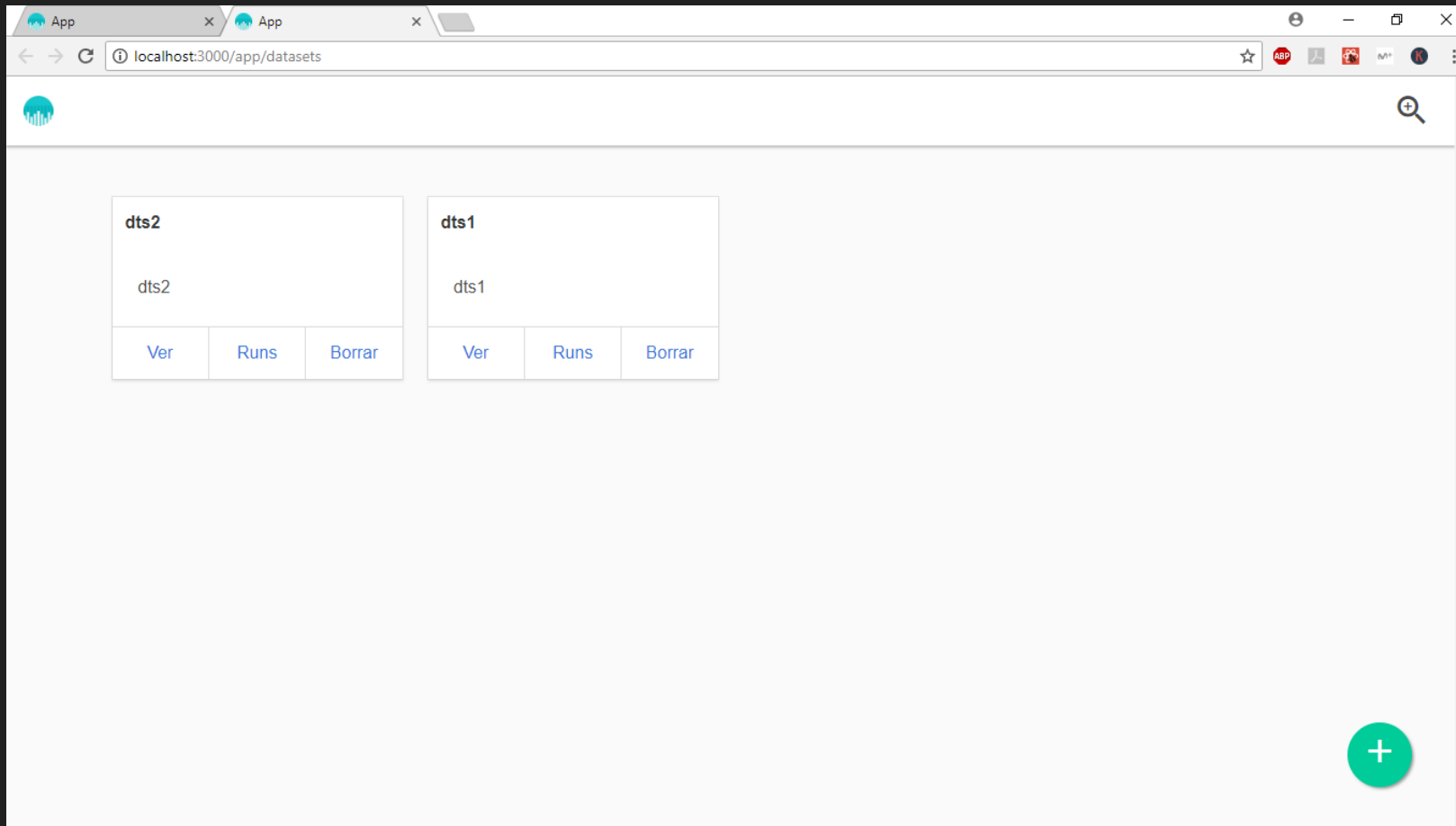
MAE	MSE	RMSE	P@K	R@K	F@K	MRR	MAP	NDCG
10	10	0	3	3	1	5	8	10

A green circular button with a white plus sign is located in the bottom right corner of the application area.

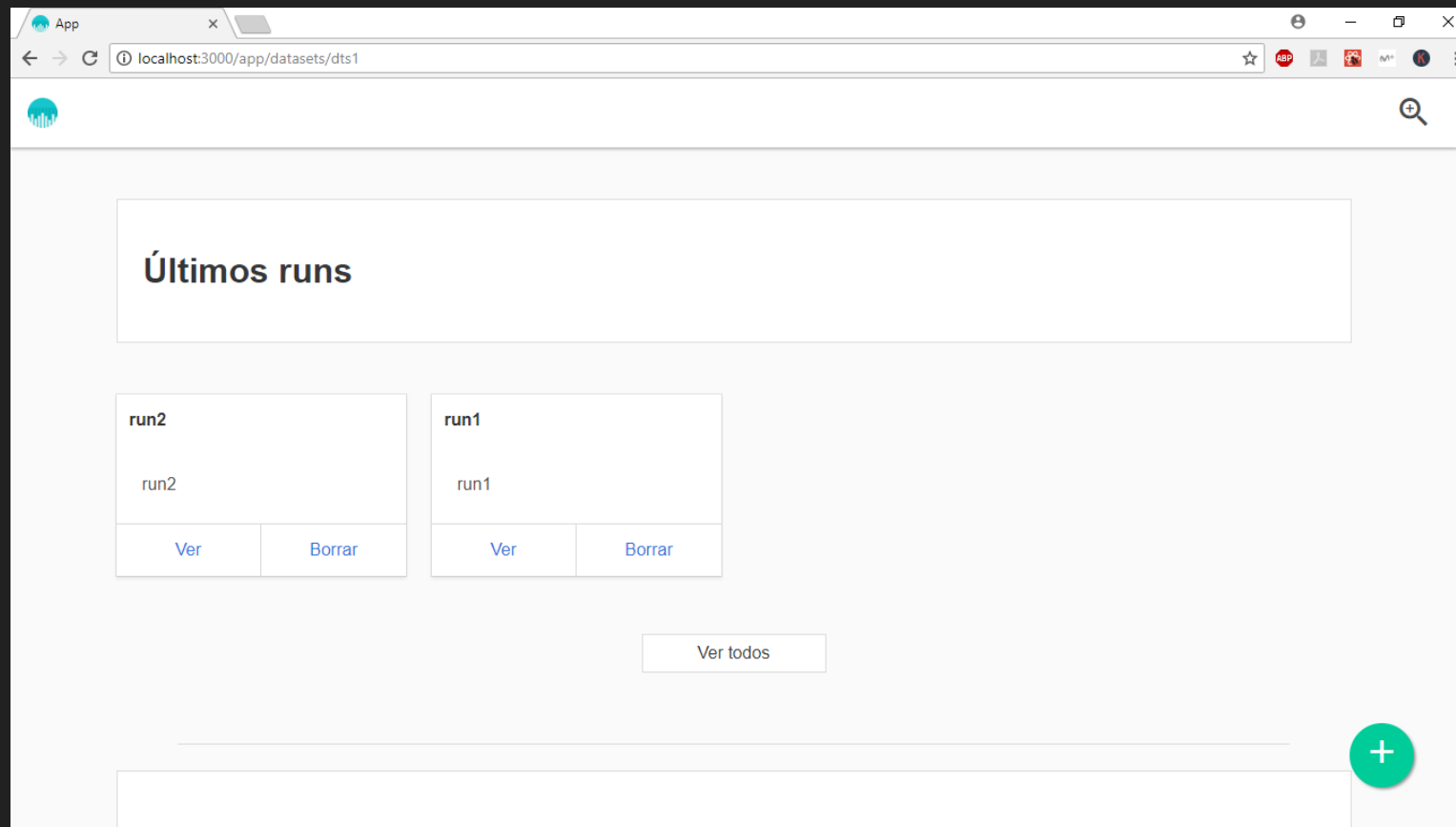
# BÚSQUEDA DE DATASETS



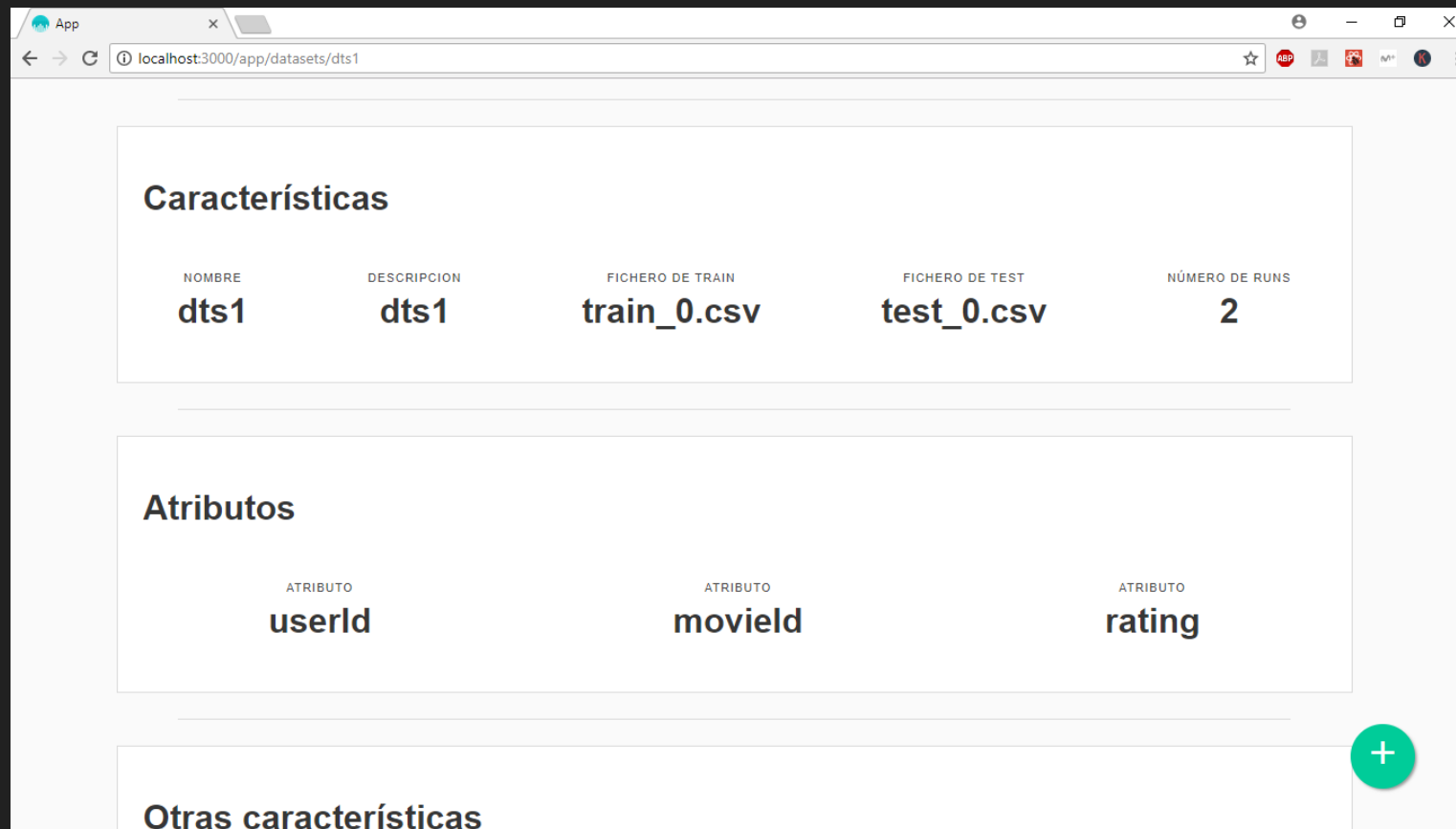
# LISTADO DE DATASETS



# VISTA DE UN DATASET



# VISTA DE UN DATASET



The screenshot shows a web browser window with the URL `localhost:3000/app/datasets/dts1`. The page is divided into three main sections:

- Características:** A table with 5 columns: NOMBRE, DESCRIPCION, FICHERO DE TRAIN, FICHERO DE TEST, and NÚMERO DE RUNS. The data row shows: `dts1`, `dts1`, `train_0.csv`, `test_0.csv`, and `2`.
- Atributos:** A table with 3 columns: ATRIBUTO, ATRIBUTO, and ATRIBUTO. The data row shows: `userId`, `movieId`, and `rating`.
- Otras características:** A section partially visible at the bottom of the page.

A green circular button with a white plus sign is located in the bottom right corner of the application area.



# VISTA DE UN DATASET

The screenshot shows a web browser window with the URL `localhost:3000/app/datasets/dts1`. The page content is as follows:

### Otras características

ATRIBUTO PRINCIPAL	ATRIBUTO DE RATING	ITEM RELEVANTE CUANDO
<b>userId</b>	<b>rating</b>	<b>&gt; 2.999999</b>

---

### Has consultado las siguientes métricas estas veces

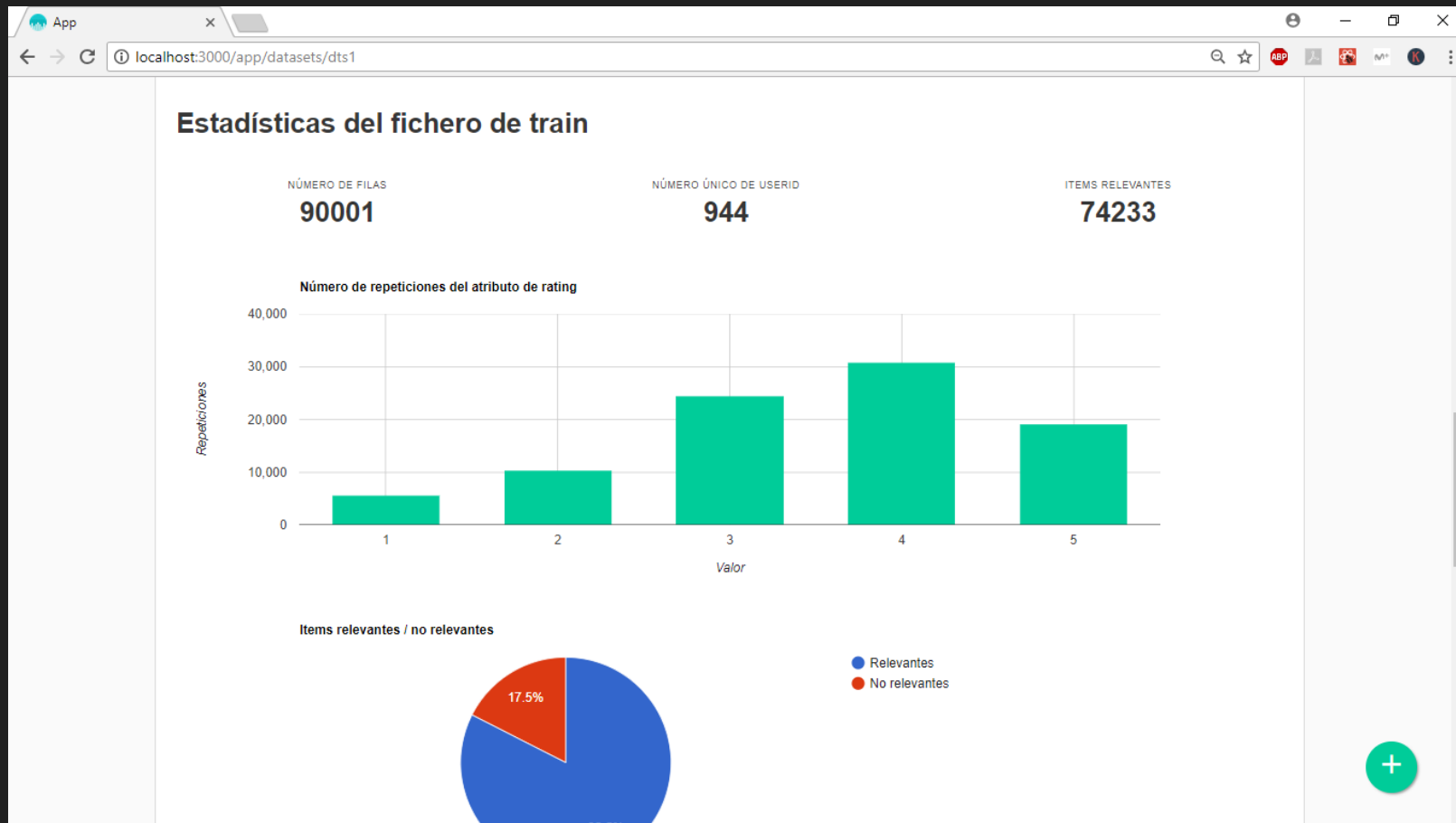
MAE	MSE	RMSE	P@K	R@K	F@K	MRR	MAP	NDCG
<b>11</b>	<b>11</b>	<b>0</b>	<b>3</b>	<b>5</b>	<b>2</b>	<b>5</b>	<b>8</b>	<b>10</b>

---

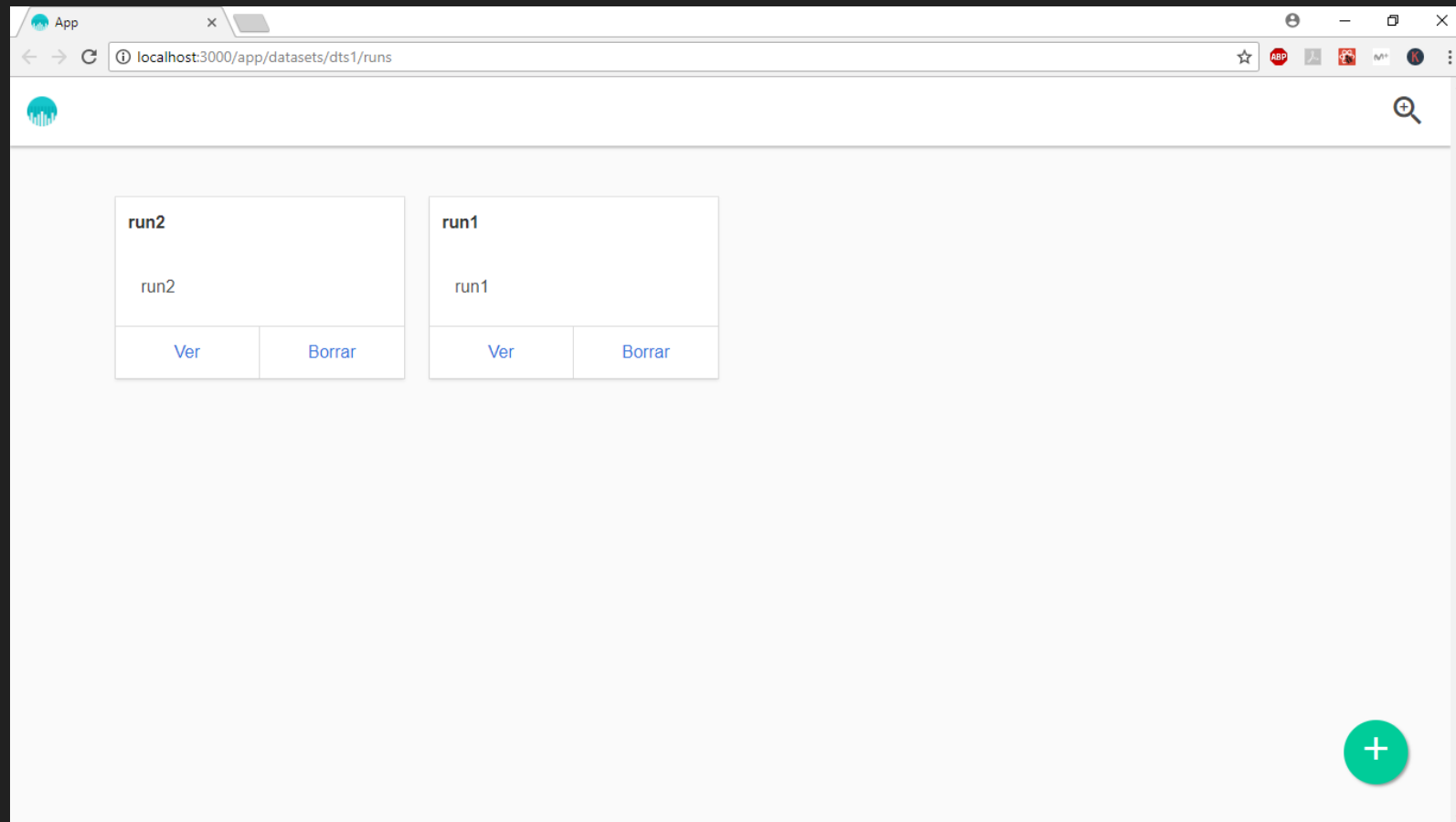
### Estadísticas del fichero de train

A green circular button with a white plus sign is located in the bottom right corner of the page content area.

# VISTA DE UN DATASET



# LISTADO DE EJECUCIONES DE UN DATASET



# VISTA DE UNA EJECUCIÓN

50 + Seleccione filtro Seleccione condición Parametro Aplicar filtro

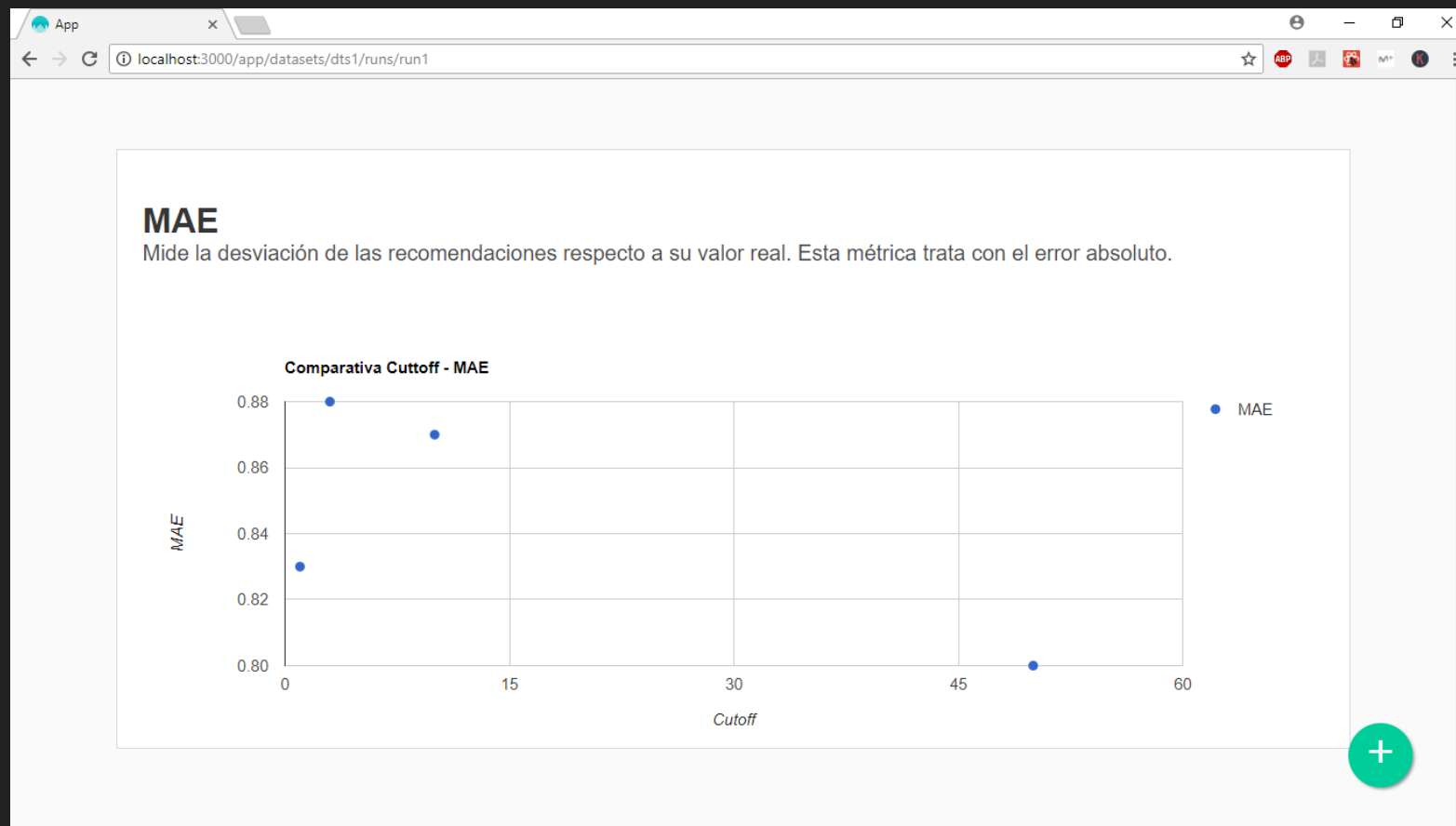
MÉTRICAS

- MAE
- MSE
- RMSE
- P@K
- R@K
- F@K
- MRR
- MAP
- NDCG

Cutoff	MAE	MSE	R@K	Borrar tabla
1	0.83	1.33	0.00	Borrar
3	0.88	1.65	0.01	Borrar
10	0.87	1.55	0.02	Borrar
50	0.80	1.19	0.02	Borrar

+

# VISTA DE UNA EJECUCIÓN



# COMPARACIÓN DE EJECUCIONES

3 + Borrar

MÉTRICAS

- MAE
- MSE
- RMSE
- P@K
- R@K
- F@K
- MRR
- MAP
- NDCG

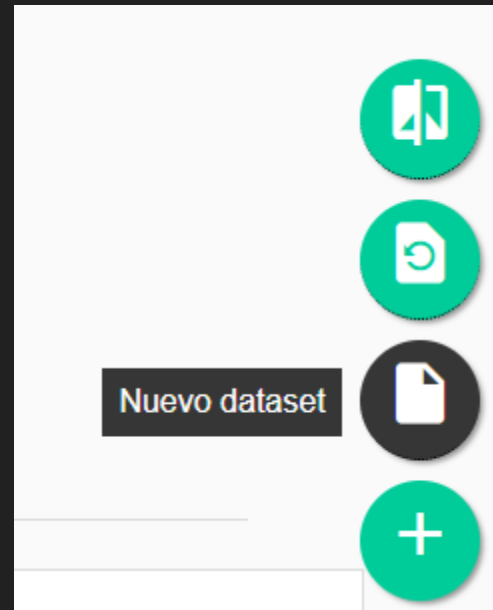
Cutoff	Run	MAE	MSE	Borrar tabla
1	run2	1.00	1.00	Borrar
1	run1	0.70	1.31	Borrar
2	run2	1.00	1.00	Borrar
2	run1	0.79	1.53	Borrar
3	run2	1.00	1.00	Borrar
3	run1	0.84	1.64	Borrar

+ 70

# COMPARACIÓN DE EJECUCIONES



# BOTÓN PRINCIPAL





# PREGUNTAS