

UNIVERSIDAD DA VINCI DE GUATEMALA
FACULTAD DE INGENIERÍA, INDUSTRIA Y TECNOLOGÍA
CARRERA: LICENCIATURA EN INGENIERÍA EN SISTEMAS



Caso Apple Music

Bases de Datos II

Ashley Dayane Alfaro Aguilar

Carné 202301982

Guatemala, 2025.

Link de github: <https://github.com/AshleyAlfaroAguilar/ashley-seeding.git>

Hacemos fork en el repositorio

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

 AshleyAlfaroAguilar / `ashley/seeding`

Your new repository will be created as `ashley-seeding`.
The repository name can only contain ASCII letters, digits, and the characters ., -, and -.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description

Proyecto para generar registros en una base de datos mongodb a partir de un schema

82 / 350 characters

Copy the `master` branch only
Contribute back to Branthonyc/seeding by adding your own branch. [Learn more.](#)

(i) You are creating a fork in your personal account.

Create fork

AshleyAlfaroAguilar / `ashley-seeding`

Code Pull requests Actions Projects Wiki Security Insights Settings

 **ashley-seeding** Public
forked from [Branthonyc/seeding](#)

master 1 Branch 0 Tags Go to file Add file > Code

This branch is up to date with [Branthonyc/seeding:master](#).

Branthonyc Create README.md 1 day ago · 4 Commits

LICENSE Initial commit last week

README.md Create README.md last week

package.json Create package.json last week

seed.js Create seed.js last week

README License

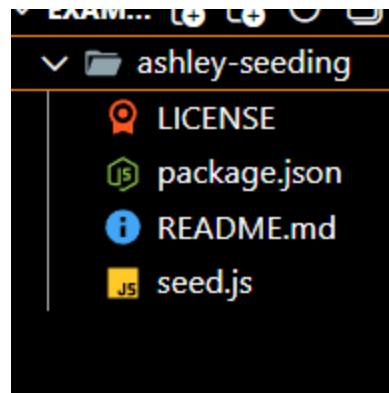
About
Proyecto para generar registros en una base de datos mongodb a partir de un schema

Readme MIT license Activity 0 stars 0 watching 0 forks

Releases
No releases published [Create a new release](#)

Packages

```
PS C:\Users\Lenovo\Documents\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982> git clone https://github.com/AshleyAlfaroAguilar/ashley-seeding.git
Cloning into 'ashley-seeding'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 2), reused 1 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 8.84 KiB | 2.21 MiB/s, done.
Resolving deltas: 100% (2/2), done.
PS C:\Users\Lenovo\Documents\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982> cd /ashley-seeding
Set-Location: Cannot find path 'C:\ashley-seeding' because it does not exist.
PS C:\Users\Lenovo\Documents\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982> cd ashley-seeding
PS C:\Users\Lenovo\Documents\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding>
```



Instalamos dependencias

```
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding> npm install
added 13 packages, and audited 14 packages in 27s
  1 package is looking for funding
    run `npm fund` for details
  found 0 vulnerabilities
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding>
```

Creamos Docker – compose dentro de database

```
docker-compose.yml U X
ashley-seeding > database > docker-compose.yml
1   version: "3.8"
2
3   services:
4     mongo:
5       image: mongo:6
6       container_name: apple-music-mongo
7       restart: always
8       ports:
9         - "27017:27017"
10      volumes:
11        - mongo_data:/data/db
12
13      volumes:
14        mongo_data:
15
```

Levantamos mongoDB con Docker

```
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding\database> docker-compose up -d
time="2025-12-06T00:39:01-06:00" level=warning msg="C:\\\\Users\\\\Lenovo\\\\Documents\\\\U\\\\SEMESTRE6\\\\BASES_DE_DATOS_II\\\\ExamenFinal202301982\\\\ashley-seeding\\\\database\\\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 9/9
  ✓ mongo Pulled
    ✓ a2be9b1fc57a Pull complete
    ✓ af6eca94c810 Pull complete
    ✓ dc9abaf2c8ff Pull complete
    ✓ cde22982277a Pull complete
    ✓ 1ad79419bb76 Pull complete
    ✓ afdef96660ff Pull complete
    ✓ 9d6e74542339 Pull complete
    ✓ 9d81a4e85c6a Pull complete
[+] Running 3/3
  ✓ Network database_default Created
  ✓ Volume "database_mongo_data" Created
  ✓ Container apple-music-mongo Started
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding\database> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
613b7a36dc72      mongo:6           "docker-entrypoint.s..."   27 seconds ago   Up 25 seconds   0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp
NAMES
apple-music-mongo

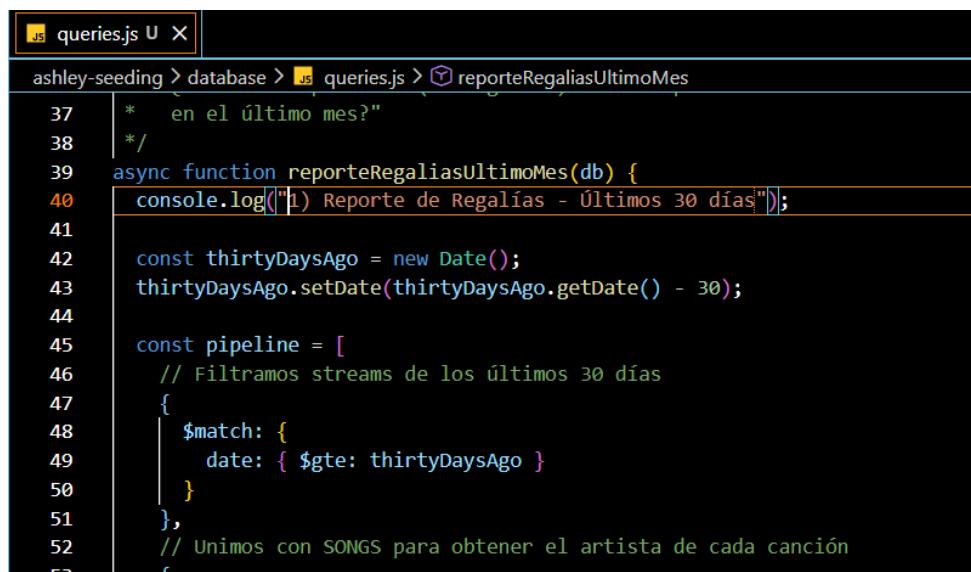

```

Ejecutamos el seeder

```
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding> npm start
> parcial-db2-apple-music-seeder@1.0.0 start
> node seed.js

🚀 Conectado a MongoDB. Limpiando colecciones viejas...
📝 Generando Catálogo Musical...
👤 Generando Usuarios...
🎥 Simulando Reproducciones (Streams)...
✅ EXITO: Base de datos poblada.
- 100 Usuarios (20 son Zombies sin streams)
- 50 Canciones
- 4700 Streams generados
-----
TIPS PARA EL ALUMNO:
1. Revisa los usuarios que NO están en la colección 'streams'.
2. Observa que 'songs' tiene el nombre del artista embebido.
```

Creamos queries.js con la primera agregación



```
ashley-seeding > database > queries.js > reporteRegaliasUltimoMes
37     * en el último mes?"*
38     */
39     async function reporteRegaliasUltimoMes(db) {
40         console.log("1) Reporte de Regalías - Últimos 30 días");
41
42         const thirtyDaysAgo = new Date();
43         thirtyDaysAgo.setDate(thirtyDaysAgo.getDate() - 30);
44
45         const pipeline = [
46             // Filtramos streams de los últimos 30 días
47             {
48                 $match: {
49                     date: { $gte: thirtyDaysAgo }
50                 }
51             },
52             // Unimos con SONGS para obtener el artista de cada canción
53         ]
```

Ejecutamos la primera agregación

```
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding\database> node queries.js
Conectado a MongoDB para ejecutar agregaciones.
```

1) Reporte de Regalías - Últimos 30 días

(index)	Artista	Segundos Totales	Cantidad de Streams
0	'Bad Bunny'	125232	768
1	'Metallica'	91265	385
2	'Feid'	85705	374
3	'Taylor Swift'	76200	389
4	'Ricardo Arjona'	75908	382

Top 10 canciones más escuchadas en Guatemala durante los últimos 7 días

```
queries.js U X
ashley-seeding > database > queries.js > reporteRegaliasUltimoMes > top10Guatemala > pipeline > $match >
39  async function reporteRegaliasUltimoMes(db) {
80  * 2) Top 10 Regional - Guatemala
81  * Pregunta:
82  * "¿Cuáles son las 10 canciones más escuchadas en 'Guatemala'
83  * en los últimos 7 días?"
84  */
85  async function top10Guatemala(db) {
86  console.log("2) Top 10 canciones en Guatemala - Últimos 7 días");
87
88  const sevenDaysAgo = new Date();
89  sevenDaysAgo.setDate(sevenDaysAgo.getDate() - 7);
90
91  const pipeline = [
92    // Filtramos streams de los últimos 7 días
93  {
94    $match: {
95      date: { $gte: sevenDaysAgo }
96    }
97  }
98
99  ]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
PS C:\Users\Lenovo\Documents\U\SEMESTRE6\BASES_DE_DATOS_II\ExamenFinal202301982\ashley-seeding\database> node queries.js
Conectado a MongoDB para ejecutar agregaciones.
```

1) Reporte de Regalías - Últimos 30 días

(index)	Artista	Segundos Totales	Cantidad de Streams
0	'Bad Bunny'	125232	768
1	'Metallica'	91265	385
2	'Feid'	85705	374
3	'Taylor Swift'	76200	389
4	'Ricardo Arjona'	75908	382

2) Top 10 canciones en Guatemala - Últimos 7 días

(index)	Posición	Canción	Artista	Streams
0	1	"Can't Get Enough of Your Love"	'Bad Bunny'	68
1	2	'The Streak'	'Metallica'	13
2	3	'Up Up & Away'	'Feid'	12
3	4	'Do Wah Diddy Diddy'	'Ricardo Arjona'	11
4	5	'Band On the Run'	'Taylor Swift'	11
5	6	'Tik-Toc'	'Bad Bunny'	11
6	7	"Ain't No Sunshine"	'Metallica'	11
7	8	'Another Night'	'Metallica'	10
8	9	'Go Your Own Way'	'Bad Bunny'	10
9	10	'When Doves Cry'	'Bad Bunny'	9

Usuarios Zombis (Churn Risk)

```

queries.js U X
ashley-seeding > database > queries.js > main
157  }
158
159 /**
160 * 3) Usuarios Zombis (Churn Risk)
161 * Pregunta:
162 * "Listar usuarios que tienen suscripción 'Premium',
163 * pero que NO han reproducido ninguna canción en los últimos 30 días."
164 */
165 async function usuariosZombis(db) {
166   console.log("3) Usuarios Zombis (Premium sin actividad en 30 días)");
167
168   const thirtyDaysAgo = new Date();
169   thirtyDaysAgo.setDate(thirtyDaysAgo.getDate() - 30);
170
171   const pipeline = [
172     // Primero filtramos solo usuarios Premium
173     {
174       $match: { subscription: "Premium" }
175     },
176
177     // Hacemos LEFT JOIN con Streams (lookup) para ver actividad
178     {
179       $lookup: {
3) Usuarios Zombis (Premium sin actividad en 30 días)

```

(index)	Usuario	Email	País	Subscripción
0	'Ryder97'	'Chris4@yahoo.com'	'GT'	'Premium'
1	'Kayden_Miller1'	'Nels_Sanford-Rohan34@yahoo.com'	'GT'	'Premium'
2	'Virgil_Willms0'	'Tanner_Conroy@hotmail.com'	'GT'	'Premium'
3	'Kailyn_Pagac'	'Sheldon_Mante18@gmail.com'	'RO'	'Premium'
4	'Jeffry_Upton'	'Jerry1@gmail.com'	'IM'	'Premium'
5	'Alberta_Dach'	'Alvina.Lindgren@gmail.com'	'GT'	'Premium'
6	'Annetta_Morissette51'	'Garth17@yahoo.com'	'BS'	'Premium'
7	'Walker_Heller'	'Libbie_Bahringer@yahoo.com'	'GT'	'Premium'
8	'Maritza_OHara98'	'Garrick84@hotmail.com'	'CD'	'Premium'
9	'Priscilla_Emmerich'	'Lon13@hotmail.com'	'GT'	'Premium'
10	'Zack.Wunsch'	'Ashly_Smith19@hotmail.com'	'SX'	'Premium'
11	'Icie.McLaughlin'	'Edds2@hotmail.com'	'GT'	'Premium'

Análisis demográfico de usuarios que escuchan reggaeton (edad)

```

queries.js U ●
ashley-seeding > database > queries.js > ...
233 /**
234 * 4) Análisis de Demografía por Género (Reggaeton)
235 * Pregunta:
236 * "De todos los usuarios que escuchan 'Reggaeton',
237 * ¿cuál es la distribución por edades?"
238 */
239 async function demografiaReggaeton(db) {
240   console.log("4) Demografía de usuarios que escuchan Reggaeton (rangos de edad)");
241
242   const pipeline = [
243     // Partimos de streams
244     {
245       $lookup: {
246         from: "songs",
247         localField: "song_id",
248         foreignField: "_id",
249         as: "song"
250       }
251     },

```

4) Demografía de usuarios que escuchan Reggaeton (rangos de edad)

(index)	Rango	Usuarios	Porcentaje
0	'15-20'	16	20
1	'21-30'	20	25
2	'31-40'	15	18.75
3	'41-50'	29	36.25

5 usuarios que han escuchado más canciones distintas del artista Bad Bunny

```

 377 /**
 378  * 5) Heavy Users de Bad Bunny
 379  * Pregunta:
 380  * ¿Cuáles son los 5 usuarios que más canciones distintas han escuchado del artista 'Bad Bunny'?
 381 */
 382 async function heavyUsersBadBunny(db) {
 383   console.log("5) Heavy Users de Bad Bunny (Top 5)");
 384
 385   const pipeline = [
 386     // Unimos streams -> songs para obtener info de la canción
 387     {
 388       $lookup: {
 389         from: "songs",
 390         localField: "song_id",
 391         foreignField: "_id",
 392         as: "song"
 393       }
 394     },
 395   ];
 396   db.collection("users").aggregate(pipeline)
 397   .limit(5)
 398   .then(result =>
 399     console.log(result)
 400   )
 401   .catch(error =>
 402     console.error(error)
 403   );
 404 }
 405 
```

5) Heavy Users de Bad Bunny (Top 5)

(index)	Usuario	Email	CancionesDistintas
0	'Alena72'	'Betty.Langosh@gmail.com'	10
1	'Maryjane.Blick'	'Deion89@yahoo.com'	10
2	'Julia.Weber81'	'Laney88@yahoo.com'	10
3	'Carolyn_Welch94'	'Monique.Shields48@hotmail.com'	10
4	'Ford51'	'Colton_VonRueden-Brown@gmail.com'	10

Diseñar API

API Spec – Apple Music Analytics (MongoDB)

Base URL (ejemplo para el proyecto): <http://localhost:3000/api>

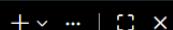
La API expone endpoints de solo lectura para alimentar el dashboard de analíticas.
Todos los datos provienen de la base `apple_music_db` en MongoDB.

1. Reporte de Regalías por Artista (Últimos 30 días)

Endpoint

- GET /api/royalties

Descripción



Creamos el dashboard en v0.dev

Royalties Report
Total streaming time per artist (seconds) - Last 30 Days

Artist	Total Streaming Seconds
The Weeknd	~100M
Bad Bunny	~95M
Taylor Swift	~90M
Drake	~85M
Ariana Grande	~80M
Billie Eilish	~75M
Post Malone	~70M
Dua Lipa	~65M

Total Seconds
29.2M
Average per Artist
3.65M
Top Artist
The Weeknd
Tracked Artists
8

Top 10 Songs in Guatemala
Most played songs in the last 7 days

Position	Song Title	Artist	Streams
1	Tú Me Preguntas	Bad Bunny	145,000
2	Un x100to	Grupo Frontera & Bad Bunny	128,000
3	Ella Baila Sola	Esteban Armado x Peso Pluma	115,000
4	MAMILL	Becky G & Karol G	102,000
5	La Jumpa	Arcángel & Bad Bunny	98,000
6	Dakdi	Bad Bunny & Jay Cortez	92,000
7	Tíki	Bad Bunny	87,000
8	Yo No Sé	Arcángel & Bad Bunny	84,000
9	La Cenecera	Bad Bunny & Arcángel	81,000
10	Ella y Yo	Aventura & Bad Bunny	76,000

Total Streams
1.01M
Avg. per Song
101K

Zombie Users
Premium users with no activity in the last 30 days

⚠️ These premium subscribers have been inactive. Consider targeting them with re-engagement campaigns.

Username	Email	Country	Subscription
alex_listen	alex@example.com	Guatemala	Premium
maria_music	maria@example.com	Guatemala	Premium
juan_beats	juan@example.com	Guatemala	Premium
sofia_premium	sofia@example.com	Guatemala	Premium
carlos_silent	carlos@example.com	Guatemala	Premium
isabel_silencio	isabel@example.com	Guatemala	Premium
luis_inactive	luis@example.com	Guatemala	Premium
diego_dormant	diego@example.com	Guatemala	Premium

Total Zombie Users
8
Lost Revenue (B/mo)
\$180/mo
Days Inactive
30+

Reggaeton Demographics
Demographic distribution of users who listen to Reggaeton

Age Group	Percentage
15-20 years	26%
21-30 years	42%
31-40 years	22%
41-50 years	8%

Largest Group
21-30 yrs
% of Audience
42%
Youngest
15+ yrs
Oldest
50 yrs

Heavy Users — Bad Bunny
Top 5 users who listened to the most distinct Bad Bunny songs

Ei Conejo - Bad Bunny's #1 Fans
These power users are deeply engaged with Bad Bunny's catalog. Ideal for early access campaigns and exclusive releases.

Rank	Username	Email	Distinct Songs
1	bunny_fan_1	fan1@example.com	127
2	bunny_lover_2	fan2@example.com	115
3	bad_bunny_stan	fan3@example.com	108
4	reggaeton_addict	fan4@example.com	102
5	el_conejo_fans	fan5@example.com	98

Top User
127 Songs
Avg. per User
110
Engagement
Very High
Total Songs
550+

Creamos el diagrama de esquema

Esquema de Datos - Apple Music Analytics (MongoDB)

