

Proyecto II

Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Redes (IC 4302)
Primer Semestre 2023



1. Objetivo General

- Desarrollar un prototipo de aplicación con una base de datos NoSQL que implementa un motor de búsqueda full-text mediante Apache Lucene.

2. Objetivos Específicos

- Desarrollar una aplicación para realizar búsquedas de letras de canciones.
- Implementar una base de datos NoSQL Cloud Native mediante [Mongo Atlas](#).
- Implementar una solución de busque Full-text basada en Apache Lucene.
- Desarrollar una herramienta de carga de archivos CSV desde Azure Blob hacia Mongo Atlas.
- Automatizar una solución mediante el uso de [Docker](#).
- Implementar una aplicación en [JavaScript](#).
- Desarrollar un API en el lenguaje de programación Python.

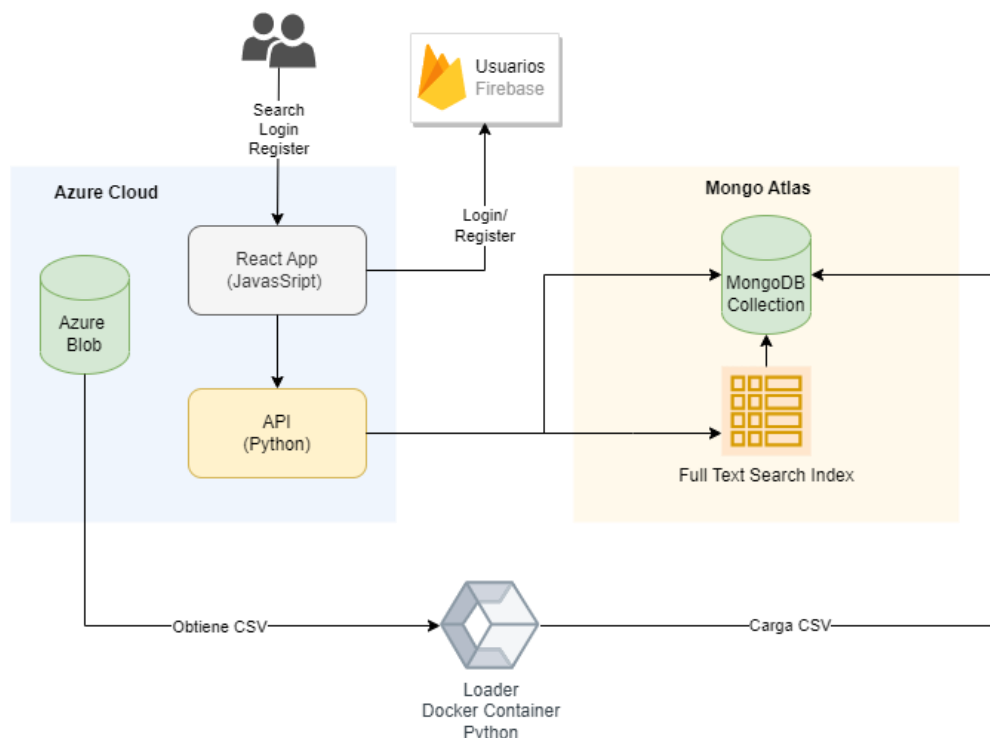
3. Datos Generales

- El valor del proyecto: 25%
- Nombre del proyecto: OpenLyrics Search.
- La tarea debe ser implementada en grupos de máximo 5 personas.
- La **fecha de entrega** es 19/05/2023 antes de las 10:00 pm.
- Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo con el reglamento. La copia incluye código/configuraciones que se puede encontrar en Internet y que sea utilizado parcial o totalmente sin el debido reconocimiento al autor.
- La revisión es realizada de forma virtual mediante citas en las cuáles todos los y las integrantes del grupo deben estar presentes, el proyecto debe estar completamente automatizado, el profesor decide en que computadora probar.
- Se debe incluir documentación con instrucciones claras para ejecutar el proyecto.
- Se deben incluir pruebas unitarias para verificar los componentes individuales de su proyecto, si estas no están presentes se obtendrá una nota de 0.
- Se espera que todos y todas las integrantes del grupo entiendan la implementación suministrada.
- Se deben seguir buenas prácticas de programación en el caso de que apliquen, entre ellas:
 - ◆ Documentación interna y externa.
 - ◆ Estándares de código.
 - ◆ Diagramas de arquitectura.
 - ◆ Diagramas de flujo
 - ◆ Pruebas unitarias.
- Toda documentación debe ser implementada en Markdown.

- Si el proyecto no se encuentra automatizado obtendrá una nota de 0. Si el proyecto no se entrega completo, la porción que sea entregada debe estar completamente automatizada, de lo contrario se obtendrá una nota de 0.
- En caso de no entregar documentación se obtendrá una nota de 0.
- El email de entrega debe contener una copia del proyecto en formato tar.gz y un enlace al repositorio donde se encuentra almacenado, debe seguir los lineamientos en el programa de curso.
- Los grupos de trabajo se deben autogestionar, la persona facilitadora del curso no es responsable si un miembro del equipo no realiza su trabajo.
- En consulta o en mensajes de correo electrónico o Telegram, como mínimo se debe haber leído del tema y haber tratado de entender sobre el mismo, las consultas deben ser concretas y se debe mostrar que se intentó resolver el problema en cuestión.

4. Descripción

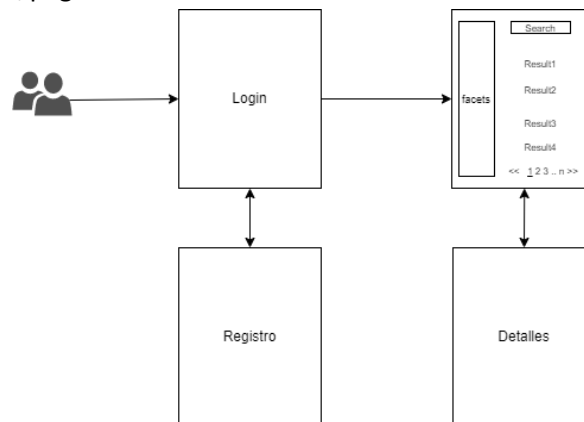
Para este proyecto implementaremos un sistema que permite realizar búsquedas Full-text search sobre letras de canciones mediante Mongo Atlas Search, el siguiente diagrama muestra los componentes que se deben configurar e implementar:



- **Loader:** Consiste en una aplicación Python la cual se ejecuta como un Docker container en una maquina local, esta se conecta a Azure Blob y descargar archivos CSV que contienen información que se debe cargar en una o más colecciones de Mongo Atlas, para este ejercicio se cargara la

información del dataset [Song lyrics from 79 musical genres](#), este contiene dos archivos: artists-data.csv y lyrics-data.csv, esta aplicación debe ser capaz de cargar estos dos archivos y cualquier otro con igual formato, se recomienda leer la descripción del dataset y explorar los datos detalladamente para poder hacer la carga de forma satisfactoria y encontrar como relacionar los datos. Es importante tener cuidado en la configuración de la lectura del CSV ya que los cambios de línea, filas multilínea entre otros pueden causar problemas al leer los archivos.

- Full Text Search Index: Una vez que la colección ha sido creada en Mongo Atlas, se debe crear un índice de búsqueda, instrucciones de como hacerlo se pueden encontrar en [Create an Atlas Search Index](#), para familiarizarse con las configuraciones pueden seguir el [tutorial](#) y usar los datos de prueba que son proporcionados por Mongo Atlas.
- API: Este componente es un contenedor escrito en Python, que se ejecuta dentro de Azure Container App, el mismo se encarga de interactuar con Mongo Atlas y realizar todas las consultas requeridas para implementar la búsqueda. Información de como interactuar con Mongo Atlas mediante Python puede ser encontrada en [Atlas Search Tutorials](#), [pymongo Driver](#), [Run Atlas Search Queries](#), [\\$search Stage](#) y [\\$facet Stage](#). Es necesario implementar pruebas unitarias para todos los endpoints del API.
- Firebase Usuarios: Esta es una simple base de datos en tiempo real para realizar el registro e inicio de sesión desde la aplicación React.
- React App: Este componente es un container que se encuentra alojado en Azure App Service, el mismo ejecuta una aplicación escrita en JavaScript utilizando un framework seleccionado por cada grupo, esta aplicación permitirá registrar usuarios, iniciar sesión y realizar búsquedas con facets, se esperan facets por Artista, Lenguaje, Género Musical del Artista, Popularidad del Artista (rango) y Numero de Canciones del Artista (rango). La aplicación se espera que tenga los siguientes componentes/paginas:



- Login: Permite ingresar a la aplicación mediante email y password.
- Registro: Permite registrar un nuevo usuario en el sistema.
- Search: Permitirá realizar búsquedas (estilo Google) y tendrá facets para refinar la búsqueda, los resultados de búsqueda deben mostrar el título de la canción y una porción del texto donde fue encontrado el match (texto de búsqueda), se debe paginar los resultados, el usuario puede dar clic en una canción para ver los detalles de esta.

- Detalles: Muestra la letra completa de la canción, junto con toda la información del artista e información de la canción.

Azure Cloud y Mongo Atlas

Si por alguna razón alguno de los grupos se ve imposibilitado para usar Azure Cloud, la aplicación se puede revisar de manera local corriendo en Kubernetes, de igual forma si por alguna razón un grupo no puede crear una cuenta de Mongo Atlas pueden usar Mongo corriendo en Kubernetes y Elasticsearch para crear el índice de búsqueda.

Documentación

Se deberá entregar una documentación que al menos debe incluir:

- Instrucciones claras de como ejecutar su proyecto.
- Pruebas realizadas, con pasos para reproducirlas.
- Resultados de las pruebas unitarias.
- Recomendaciones y conclusiones (al menos 10 de cada una).
- La documentación debe cubrir todos los componentes implementados o instalados/configurados, en caso de que algún componente no se encuentre implementado, no se podrá documentar y tendrá un impacto en la completitud de la documentación.
- Referencias bibliográficas donde aplique.

Imaginen que la documentación está siendo creada para una persona con conocimientos básicos en el área de computación y ustedes esperan que esta persona pueda ejecutar su proyecto y probarlo sin ningún problema, el uso de imágenes, diagramas, code snippets, videos, etc. son recursos que pueden ser útiles.

La documentación debe estar implementada en Markdown y compilada en un PDF.

5. Recomendaciones

- Utilizar telnet/curl/[postman](#) para interactuar con los componentes y verificar que los mismos están funcionando correctamente.
- Utilizar [Docker Desktop](#) para instalar Docker en Windows.
- Realicen peer-review/checkpoints semanales y no esperen hasta el último momento para integrar su aplicación.
- Crear un repositorio de GitHub e invitar al profesor.
- Realizar commits y push regulares.
- Para la aplicación de React se recomienda usar algún tipo de JS Framework como [VueJS](#). Existen cantidades de tutoriales de cómo hacer autenticación con Firebase.
- Para la revisión, se recomienda tener clientes de administración de las bases de datos para poder

realizar consultas y observar su comportamiento.

6. Entregables

- Documentación.
- Documentación con las pruebas unitarias del API.
- Docker files y código para generar el API, Loader y la aplicación React.
- Archivos de infraestructura como código con los cambios requeridos por cada grupo.

7. Evaluación

Funcionalidad / Requerimiento	Porcentaje
Documentación (*)	10%
Pruebas unitarias (*)	10%
Configuración de Mongo Atlas	10%
API	40%
Loader	10%
React App	20%
	100

(*) La completitud de la documentación, así como pruebas unitarias se encuentra acotado por la completitud de otros requerimientos.