

Orientación de Proyecto

Bases de Datos
Licenciatura en Ciencia de Datos
Curso: 2025-2026

Este documento proporciona una orientación para el proyecto final de la asignatura. **La realización de este proyecto constituye un ejercicio por equipo de dos estudiantes.**

Escenario

GreenScape es una plataforma digital de red social dedicada a conectar a entusiastas de las plantas y la jardinería de todo el mundo. Su misión es crear una comunidad global donde los usuarios puedan compartir conocimientos sobre el cultivo y cuidado de plantas, descubrir nuevas especies, contribuir con sus experiencias mediante publicaciones y reacciones, y acceder a una tienda en línea para adquirir plantas y productos relacionados. GreenScape busca democratizar el conocimiento botánico y fomentar prácticas de jardinería sostenible a través de la colaboración y el intercambio de experiencias entre usuarios.

El equipo ha sido contratado como analistas de datos en la empresa GreenSolutions, creadores de la plataforma GreenScape. Debido a despidos masivos, todo el personal relacionado con el desarrollo de GreenScape ya no se encuentra disponible. Sin embargo, con vistas a mejorar la plataforma en el futuro, la empresa ha encargado obtener información acerca de su utilización por los usuarios e implementar una propuesta de mejora al diseño de la base de datos.

Como únicos profesionales disponibles con conocimientos técnicos, el equipo desempeñará un papel crucial en la continuidad operativa de GreenScape. El trabajo consistirá en realizar ingeniería inversa sobre la base de datos existente para comprender su estructura y funcionamiento, extraer información valiosa mediante consultas analíticas complejas, identificar patrones de comportamiento de usuarios y vendedores, y proponer mejoras al diseño de la base de datos que amplíen las funcionalidades de la plataforma. Además, será necesario desarrollar herramientas interactivas que permitan a los directivos de la empresa visualizar métricas clave y tomar decisiones informadas sobre el futuro de GreenScape. La capacidad para trabajar en equipo y el dominio de bases de datos será fundamental para el éxito de esta misión.

En equipo, y sin contar con ayuda de nadie en la empresa que conozca la base de datos de GreenScape, se deberá completar las tareas asignadas por la empresa.

Materiales provistos

Para el desarrollo del proyecto se le provee de los siguientes recursos disponibles:

- **Configuración Docker:** Archivo `docker-compose.yml` para desplegar un contenedor MySQL con la base de datos GreenScape.
- **Script de inicialización:** Archivo `init.sql` con datos iniciales.
- **Documento de orientación:** El presente documento (`guidelines.tex`) con los requerimientos y las tareas a desarrollar.
- **Instrucciones de uso:** Archivo `README.md` con instrucciones detalladas para la instalación, configuración y conexión a la base de datos desde diferentes entornos.

Tareas

1. Realizar un **fork** del repositorio del proyecto disponible en https://github.com/Project-Orientations/2526-cd-bd-project-setup-green_scape. Este repositorio será la base del trabajo y deberá contener todo el código y documentación desarrollada durante el proyecto. Asegurar mantener el repositorio actualizado con los avances.
2. A partir de la base de datos proporcionada, elaborar una propuesta de **modelo entidad-relación extendido (MERX)** que permita comprender conceptualmente el escenario planteado, respetando la

concepción vista en clases. Se puede utilizar cualquier herramienta para su construcción; sin embargo, se recomienda emplear diagrams.net o Excalidraw.

3. Escribir código MySQL que responda a los siguientes ejercicios:

- a) **Listar todos los productos disponibles:** Mostrar todos los productos registrados, incluyendo sus detalles.
- b) **Contar las reacciones por publicación:** Calcular la cantidad total de reacciones recibidas por cada publicación, incluyendo información sobre la publicación y su autor.
- c) **Tipos de plantas preferidos:** Listar los tres tipos de plantas que han recibido la mayor cantidad de reacciones positivas, junto al total de reacciones.
- d) **Usuarios activos en contribuciones y reacciones:** Determinar la actividad de los usuarios mostrando, para cada uno, su información personal y la última fecha (dentro de los últimos seis meses) en que realizó una contribución o reaccionó a una publicación. Si el usuario no presenta actividad reciente, la fecha debe mostrarse como NULL.
- e) **Publicaciones más populares considerando las reacciones:** Identificar las publicaciones que presentan mayor cantidad de reacciones positivas que negativas, incluyendo sus detalles y la cantidad total de reacciones.
- f) **Contribuciones constantes:** Mostrar todas las plantas que han recibido contribuciones en dos meses consecutivos.
- g) **Promedio de actividad:** Determinar el promedio de actividad mensual de los diez usuarios más activos en agregar contenido multimedia a sus publicaciones durante el último año (promedio de videos/fotos agregados por mes).
- h) **Distribución de edades:** Analizar la distribución de los usuarios por rangos de edad de diez años (por ejemplo: 11 – 20, 21 – 30, 31 – 40, 41 – 50, ...). Devolver la cantidad de usuarios y el por ciento de edad por categoría.
- i) **Productos sin incremento en ventas mensuales:** Identificar los productos que no han mostrado un incremento en sus ventas mes a mes durante el último año.
- j) **Tendencias de contribución según el clima:** Examinar cómo varían las contribuciones de acuerdo con el tipo de clima, identificando la planta más popular en cada categoría climática.
- k) **Cambio de preferencia en categorías de plantas:** Identificar a los usuarios que han cambiado su categoría de planta más contribuida al comparar la actividad entre dos años consecutivos.
- l) **Compras contradictorias:** Destacar a los usuarios que han comprado más plantas no marcadas como "Me gusta" que plantas marcadas como tales.
- m) **Usuarios de solo texto:** Listar los usuarios que nunca han agregado contenido multimedia a sus publicaciones.
- n) **Vendedores mejor calificados:** Mostrar los cinco vendedores con mejor calificación promedio, ordenados de forma descendente. De cada uno se debe incluir el total de productos vendidos, calificación promedio, nombre, correo electrónico y dirección particular.
- ñ) **Trigger de auditoría de precios:** Implementar un disparador que se active automáticamente cuando se modifique el precio de un producto en la tabla **Producto**. Este recurso debe:
 - Registrar en una tabla de auditoría (**HistorialPrecios**) la información del cambio: producto afectado, precio anterior, precio nuevo, fecha y hora del cambio.
 - Calcular el porcentaje de cambio en el precio.
 - La tabla de auditoría debe ser creada con la estructura apropiada para almacenar estos datos.
- o) **Procedimiento almacenado para análisis de actividad de usuario:** Crear un procedimiento almacenado llamado **sp_analisis_usuario** que reciba como parámetros:
 - Identificador del usuario a analizar
 - Fecha inicial del período de análisis
 - Fecha final del período de análisisEl procedimiento debe retornar un conjunto de resultados que incluya:

- Total de publicaciones realizadas en el período
- Total de reacciones dadas y recibidas
- Total de comentarios realizados
- Total de compras y monto gastado
- Total de contribuciones realizadas
- Planta más comprada y planta más contribuida

Este procedimiento debe ser invocable desde la aplicación Streamlit, permitiendo al usuario introducir los parámetros de forma interactiva y visualizar los resultados en un formato amigable.

- p)* **Análisis de influencers y su impacto en ventas:** Identificar a los 5 usuarios “*influencers*” (aquellos cuyas publicaciones generan más interacciones) y determinar si existe correlación entre su actividad y las ventas de las plantas con las que interactúan. Para cada *influencer*, calcular:
- 1) Su puntaje de interacciones ponderado (Me gusta=1, Me encanta=2, Me asombra=1,5, comentarios=2);
 - 2) Las plantas con las que más ha interactuado (publicaciones, contribuciones, compras);
 - 3) El incremento porcentual en ventas de esas plantas en las 2 semanas posteriores a sus publicaciones/contribuciones versus las 2 semanas anteriores;
 - 4) La tasa de conversión: porcentaje de usuarios que compraron plantas después de reaccionar a sus publicaciones.
- q)* **Detección de patrones de comportamiento anómalo en vendedores:** Identificar vendedores con patrones de venta sospechosos mediante el análisis de:
- 1) Vendedores que venden el mismo producto a precios significativamente diferentes ($> 30\%$ de variación) sin justificación temporal;
 - 2) Vendedores con productos que tienen calificaciones extremadamente polarizadas (muchos 5 y muchos 1, pero pocos valores intermedios);
 - 3) Vendedores cuyos compradores nunca o raramente han comprado otros productos en la plataforma (posible manipulación);
 - 4) Productos vendidos por múltiples vendedores donde uno tiene un patrón de ventas muy diferente al resto. Para cada vendedor sospechoso, generar un “índice de sospecha” ponderado y listar las evidencias específicas detectadas.
4. Implementar y comparar soluciones para conversaciones en comentarios utilizando diferentes modelos de datos:
- a)* Modificar el diseño de la base de datos relacional para permitir conversaciones en los comentarios, es decir, permitir que los usuarios puedan responder comentarios de otros usuarios creando hilos (conversaciones internas).
 - Escribir el código para modificar la base de datos de acuerdo al diseño y generar conversaciones de prueba de longitud no menor que 20.
 - Escribir código para dado un comentario inicial obtener la conversación entera surgida a partir de dicho comentario.
 - b)* Modelar y resolver el mismo problema de conversaciones en comentarios utilizando otro modelo de datos visto en conferencias. Los identificadores de los comentarios deben ser los mismos en ambas bases de datos para facilitar la comparación.
 - c)* Comparar ambos diseños (relacional y no relacional) de acuerdo a las ventajas y desventajas que presentan para resolver este escenario.
5. Implementar un sistema de documentación jerárquica para las plantas. Cada planta debe tener asociado un **documento principal** (Ficha Técnica), que contiene información esencial sobre el tipo de planta y sus cuidados necesarios. Además, cada documento principal puede tener asociados múltiples **documentos secundarios** que complementan la información, tales como:
- **Certificado Fitosanitario:** Información sobre el estado de salud y ausencia de plagas de la planta.
 - **Guía de Riego Estacional:** Instrucciones específicas de riego según la estación del año.

- **Manual de Tratamiento de Plagas:** Procedimientos para prevenir y tratar plagas comunes.
- **Historial de Crecimiento:** Registro del desarrollo esperado de la planta en diferentes etapas.
- **Análisis de Suelo:** Especificaciones detalladas sobre el tipo de suelo y nutrientes requeridos.

Para esta tarea:

- Diseñar una solución de almacenamiento apropiada considerando que los documentos tienen estructuras variables y se organizan de forma jerárquica. Los identificadores de las plantas deben coincidir con los de la base de datos provista. Justificar la elección del modelo de datos.
 - Implementar la solución diseñada para almacenar documentos principales y secundarios asociados a plantas.
 - Insertar datos de prueba: al menos 5 fichas técnicas, cada una con un mínimo de 3 documentos secundarios de tipos diferentes.
 - Implementar una operación de consulta que, dado el identificador de una planta, retorne todos sus documentos (principal y secundarios) organizados jerárquicamente.
6. Desarrollar una aplicación web interactiva utilizando **Streamlit** que proporcione una interfaz gráfica para interactuar con la base de datos GreenScape. La aplicación debe incluir las siguientes funcionalidades:

- **Selector de consultas:** Un desplegable que permita seleccionar cualquiera de las consultas SQL implementadas en las tareas anteriores, mostrando los resultados en una tabla interactiva.
- **Análisis de usuario con procedimiento almacenado:** Interfaz interactiva que permita al usuario introducir los parámetros del procedimiento almacenado `sp_analisis_usuario` (ID de usuario, fecha inicio, fecha fin) y visualizar los resultados del análisis de actividad en formato amigable (tablas, gráficos, métricas) según convenga.
- **Gestión de conversaciones:** Interfaz para crear nuevos comentarios y asociarlos a hilos de conversación existentes, permitiendo la navegación jerárquica de las conversaciones.
- **Explorador de documentos:** Funcionalidad para seleccionar una planta y visualizar todos sus documentos asociados (principales y secundarios) de forma organizada y jerárquica.
- **Gestor de precios de productos:** Interfaz que permita seleccionar un producto, visualizar su precio actual y modificarlo. Esta funcionalidad debe:
 - Mostrar una lista de productos con sus precios actuales
 - Permitir al usuario seleccionar un producto y cambiar su precio
 - Después de actualizar el precio, consultar y mostrar el historial de cambios de precio registrado por el *trigger* de auditoría, demostrando que el *trigger* se activó correctamente
 - Visualizar la tabla de auditoría con los cambios históricos de precios para el producto seleccionado

La aplicación debe ser completamente funcional, con manejo de errores robusto y una interfaz intuitiva que facilite la interacción con los datos de GreenScape.

Restricciones Técnicas

Para el desarrollo de la aplicación Streamlit y la interacción con la base de datos, se deben cumplir las siguientes restricciones técnicas:

- **No se admite el uso de ORM:** No se permite utilizar bibliotecas ORM (como SQLAlchemy, Django ORM, Peewee, etc.) para el trabajo con la base de datos. El objetivo es que los estudiantes trabajen directamente con SQL y comprendan las operaciones de base de datos a bajo nivel.
- **Uso obligatorio de bibliotecas SQL nativas:** Debe utilizarse bibliotecas de Python que permitan definir la sintaxis SQL directamente, tales como:

- `mysql-connector-python`
 - `pymysql`
 - `psycopg2` (para PostgreSQL)
 - `sqlite3` (módulo estándar de Python)
- **Consultas SQL explícitas:** Todas las operaciones de base de datos deben realizarse mediante consultas SQL escritas explícitamente en el código, sin abstracciones de alto nivel. Esto incluye operaciones de `SELECT`, `INSERT`, `UPDATE`, `DELETE`, y llamadas a procedimientos almacenados.

Entrega y Defensa

La entrega y discusión del proyecto se realizará en la **semana 16** del curso académico, o en fechas posteriores si así se decide y comunica oportunamente.

Formato de entrega

El equipo deberá preparar un **repositorio** (público o privado con acceso compartido al profesor) que contenga:

- **Imagen del MERX:** Documento en formato `.pdf` o imagen (`.png`, `.jpg`) que muestre el modelo entidad-relación extendido de la base de datos original.
- **Carpeta del proyecto Streamlit:** Carpeta contenedora con el código fuente completo de la aplicación web y todos los scripts SQL utilizados en el proyecto, organizados de manera clara y estructurada.
- **Informe técnico:** Documento en formato `.pdf` que contenga:
 - Comparación entre los diseños relacional y no relacional para la implementación de conversaciones en comentarios (Tarea 4.c), incluyendo análisis de ventajas, desventajas y conclusiones sobre cuál modelo resulta más apropiado para este escenario.
 - Justificación de la elección del modelo de datos utilizado para el sistema de documentación jerárquica de plantas (Tarea 5), explicando por qué es la solución más apropiada considerando la estructura variable y jerárquica de los documentos.
 - Cualquier otra consideración técnica relevante sobre las decisiones de diseño tomadas durante el proyecto.
- **README:** Archivo `README.md` con instrucciones claras para la configuración y ejecución del proyecto, incluyendo la estructura del repositorio, requisitos previos, pasos de instalación, y descripción breve de las funcionalidades implementadas.
- **Archivo docker-compose.yml:** Fichero `docker-compose.yml` actualizado con todas las configuraciones necesarias para desplegar los contenedores de base de datos utilizados en el proyecto (MySQL y cualquier otra base de datos adicional requerida).

Nota importante: Asegurar que el repositorio incluya un archivo `.gitignore` apropiado que excluya credenciales, archivos temporales y dependencias que puedan ser reinstaladas.