



NaturallsTIC Web App

04/09/2019

Isaac Mena López - 2016130651

Bases de Datos II

Tecnológico de Costa Rica

SIUA, Alajuela, Costa Rica

Descripción general del proyecto

La aplicación web NaturallsTIC es un sistema de proporción pequeña que permite a un usuario poder registrar *observaciones* previamente realizadas para contribuir a la base de datos del sistema. Los usuarios pueden además de ingresar *observaciones* actualizar, eliminar y visualizar las ya existentes.

Se incluye un módulo para poder realizar las mismas operaciones pero sobre *imágenes* de taxones en una base de datos (MongoDB), la galería se puede visualizar accediendo desde la página principal, al igual que las observaciones.

El sistema cuenta con un *login* el cual se presenta al iniciar la página, si no se posee cuenta asociada se puede crear una inmediatamente.

Adicionalmente, para efectos académicos, se pueden realizar operaciones CRUD de igual manera sobre los *usuarios*.

Descripción de la arquitectura.

Como patrón de arquitectura se siguió el modelo Model View Controller (MVC).


La lógica o *backend* del producto se implementó utilizando Java Servlets, JSP y JSP tags. Los servlets tomaron el rol de *Controladores* ya que se encargan de interactuar con el *Modelo*, el cual provee una interfaz para acceder y escribir sobre los datos en la base.

Se implementó un Modelo por cada relación o tabla de datos sobre las que el usuario puede operar. Se realizaron entonces 3 Modelos distintos (ModelImage, ModelObservation y ModelUser).

Las interfaz de los modelos permitían a quien los usase (en este caso a los controladores) realizar las operaciones básicas sobre las tablas en su respectiva base de datos.

Para la *Vista* o interfaz gráfica con la que el usuario puede interactuar, se utilizó HTML5, CSS3, Bootstrap y JavaScript. Se hizo uso de una plantilla base¹ para la implementación de esta parte. Dicha plantilla fue modificada por mi persona para lograr suplir los requerimientos del proyecto.

¹ <https://html5up.net/astral>



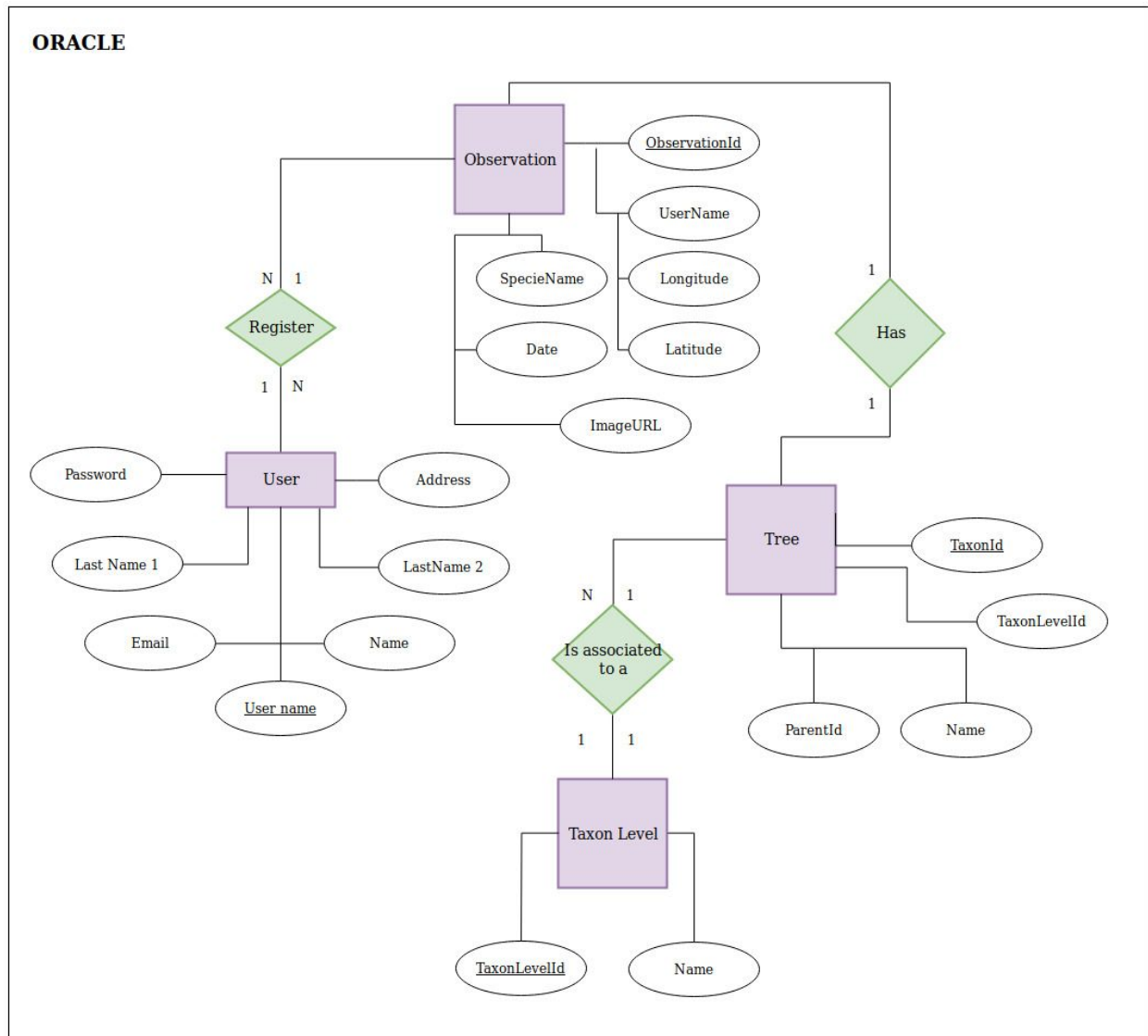
Se implementó el uso de las clases y estilos proveídos por Bootstrap para mejorar la parte visual, especialmente en la sección de Observaciones, para poder visualizar las imágenes asociadas.

Para facilitar el manejo de los datos de cada tabla se crearon objetos *User*, *Observation* y *Image*, cada uno con sus respectivos atributos vinculados a los de la base de datos.

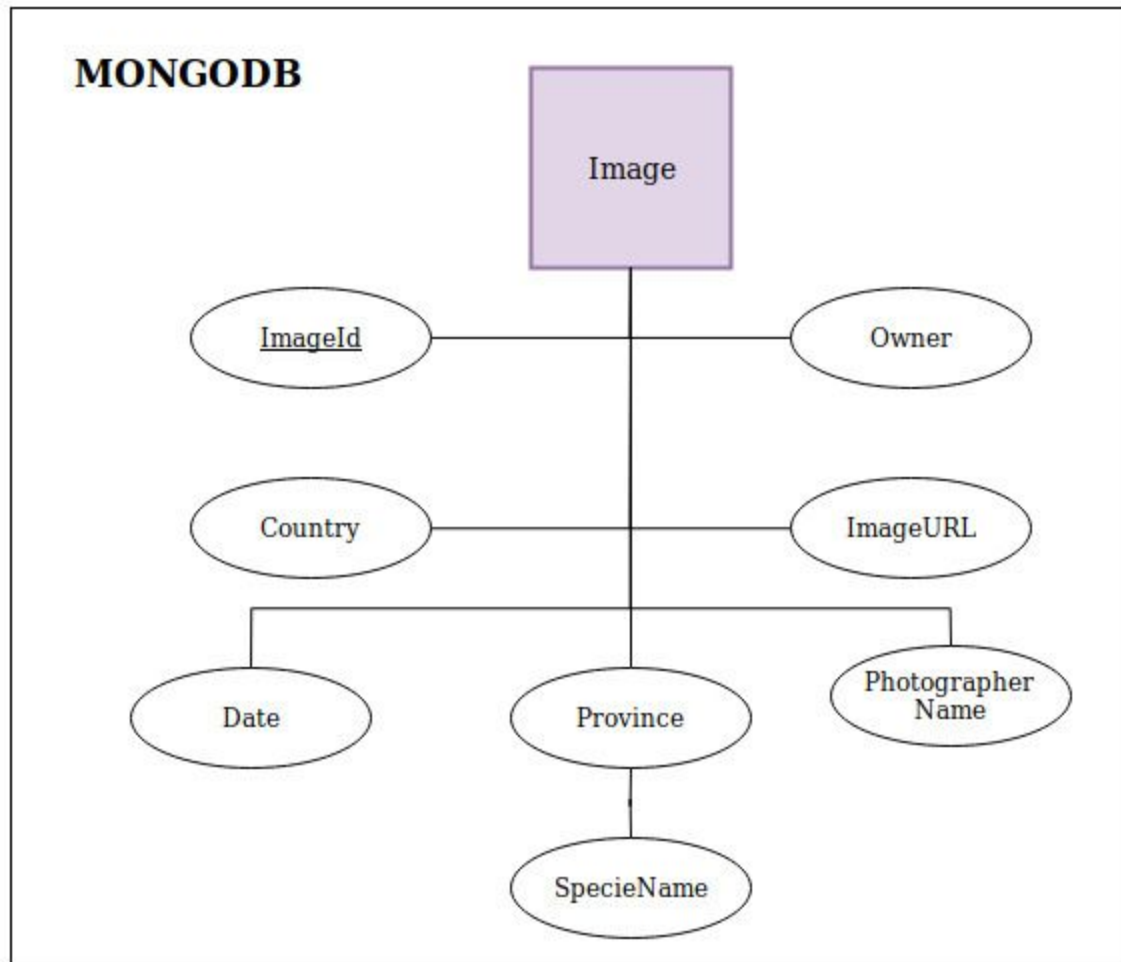
Finalmente para la parte de la conexión a la base de datos en oracle se realizó un *Connection Pool*, sin embargo este no fue sometido a prueba con múltiples usuarios. Para efectos del proyecto funciona correctamente con un usuario.

Diagramas de Entidad - Relación

DER - Oracle

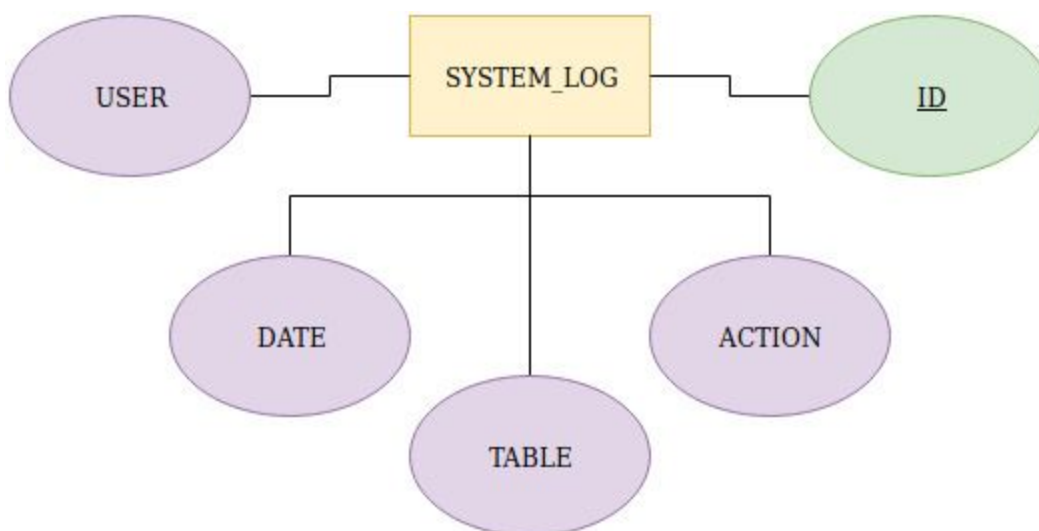


DER - MongoDB



Bitácora de operaciones sobre relaciones

Adicionalmente en el esquema de Oracle se encuentra una relación llamada *SYSTEM_LOG* la cual mantiene una especie de bitácora de cambios u operaciones que se han realizado sobre cualquier tabla dentro de la base.



Diccionario de datos

Observation: representa las grabaciones de especies hechas por un usuario.

Observation Id: PK number

Identificador único de cada observación.

Latitude: number

Latitud geográfica para indicar el lugar donde se realizó la observación.

Longitude: number

Latitud geográfica para indicar el lugar donde se realizó la observación.

Especie Name: String

Sólo se contempla el nombre de la especie para las observaciones.

Image URL: String



Dirección web donde se puede localizar la imagen.

User Name: String

Nombre de usuario al cual está ligada la observación (quién la subió)

Tree: representa toda la jerarquía de taxones

Taxon level ID: FK number

Identificador único para el nivel al cual está asociado cada taxón (Reino, Filo, Clase, ...).

Taxon ID: PK number

Identificador único de cada taxón.

Parent ID: number

Identificador del taxón padre. Para formar una jerarquía.

Taxon Level: representa los 7 distintos niveles en una jerarquía taxonómica

Level ID: PK number

Número de nivel.

Level Name: String

Representa el nombre del nivel.

Image: son fotografías de especies distintas

ID: PK number

Identificador único de las imágenes (por defecto lo incluye Mongo)

URL: String

Dirección para localizar la imagen.

Photographer Name: String

Nombre del fotógrafo que tomó la imagen.

Specie Name: String

Nombre de la especie que representa.

Date: Date



Fecha en la que se tomó la foto.

Country: String

País donde se tomó la fotografía.

Province: String

Provincia del país donde se tomó la fotografía.

Owner: String

Nombre del dueño de la fotografía (no es necesariamente)

User: persona que puede operar sobre observaciones y añadir nuevas

User Name: String

Nombre del usuario con el cual se debe registrar en la página.

Name: String

Nombre real del usuario

Last Name 1: String

Primer apellido del usuario

Last Name 2: String

Segundo apellido del usuario

Address: String

Dirección de vivienda del usuario (puede constar de provincia y país)

Email: String

Correo electrónico del usuario.

User password: String

Contraseña asociada a la cuenta.

System Log: funciona como un *log* del sistema para el control de operaciones sobre tablas

User Name: String



Nombre del usuario que realizó la acción sobre la tabla.

Log Date: Date

Fecha y hora en la que ocurrió la acción

Action: String

Operación CRUD realizada

Table Name: String

Nombre de la tabla que se vio afectada.