

# Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Bases de Datos II

Proyecto I: TurismoCR

Prof. Miguel Corea Toruño

Integrantes:

Óscar Chavarría Campos

Elberth Adrián Garro Sánchez

Darío Monestel Corella

Daniel Solís Sánchez

II Periodo

2017

# Introducción

A continuación en este documento se explicará cómo se va realizar el proyecto a implementar, además se expondrá de manera gráfica el modelo del sistema para ayudar a su mejor comprensión y la aplicación que consume esos datos durante la ejecución. También se comentará acerca de la arquitectura para la solución de este proyecto para un mejor conocimiento del mismo así como las referencias tomadas para poder aplicar en torno a los requerimientos solicitados por el sistema TurismoCR.

## Descripción del Proyecto

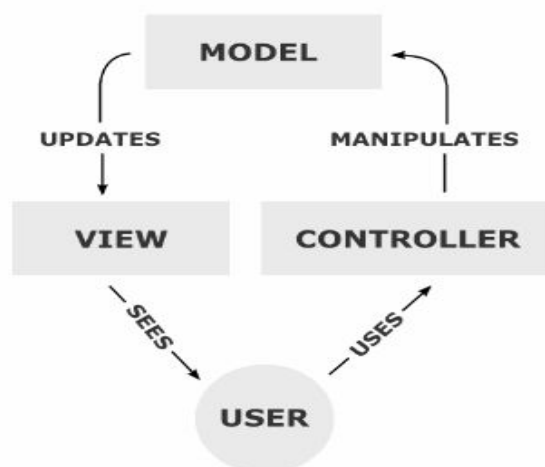
Este proyecto consiste en un sitio web para ordenar paquetes turísticos en Costa Rica, además de publicar servicios ofrecidos por pequeños empresarios del área del turismo en Costa Rica, los dueños de los establecimientos se deben registrar en el sistema para hacer uso de esas funcionalidades. Luego los clientes podrán consultar esos servicios habilitados por los propietarios donde los podrán agregar a un carrito para reservarlos para luego confirmar la compra, también estos podrán seguir a otros clientes para ver sus respectivos pedidos y recibir recomendaciones. Finalmente los administradores tienen todos los privilegios sobre sistema para manejarlo en caso de alguna irregularidad entre los usuarios del sistema.

# Arquitectura de la solución

La arquitectura solución implementada en este proyecto es el MVC donde nos basamos en 3 capas principales de esa manera se orientó al patrón modelo vista controlador. Solución de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y comunicaciones, para ello el MVC propone la construcción de tres componentes distintos donde mencionamos el primero.

El modelo lo podemos relacionar con la capa de acceso de datos, aquí se representa la información requerida por el sistema, para efectos de este proyecto está capa se tomó como la base de datos con los procedimientos almacenados para poder interactuar de manera más sencilla con la capa de lógica de negocio.

Esta capa implementada como el controlador responde o capta la información de la vista(interfaz de usuario) para enviarla mediante el módulo de acceso de datos a la base de datos, en otras palabras esta capa funciona de intermediario entre la interfaz de usuario y las bases de datos, esta capa es donde reside la lógica del sistema, para este proyecto se implementó una interfaz web para el manejo de usuarios y servicios. Finalmente se tiene la capa de vista como se mencionó de manera breve, anteriormente se representó como la interfaz de usuario donde el usuario interactúa con la misma para consumir la información de la base de datos mediante consultas de inserción, modificación y eliminación de diferentes registros.



# Contenido de la arquitectura del sistema

## Models:

En esta capa podemos encontrar los modelos tales como:

User, Service, Reseña, Owner, Imagen, ErrorViewModel y Carrito donde se definen los atributos y constructores para ser implementados en los Controladores y las vista respectivos mediante métodos.

## Views:

En esta capa se encuentra las vistas del sistema tales como:

**-Car:** vista para agregar paquetes turísticos previo a la compra.

**-Home:** vista de inicio de la página que ofrece diversas opciones para los usuarios del sistema para interactuar con el mismo.}

**-Orden:** vista que permite consultar los registros de las compras realizadas por los clientes del sistema.

**-Reseña:** vista para realizar reseñas así como consultarlas y borrarlas.

**-Service:** vista para administrar los servicios creados por los propietarios respectivos de la página.

**-Shared:** vista para manejar excepciones durante la ejecución del sistema.

**-User:** vista que maneja el inicio de sesión, registro y cierre de sesión del sistema.

## Controllers:

**-Car:** contiene los métodos para administrar el carrito para obtener y procesar los paquetes turísticos previo a la compra.

**-Home:** se define una clase para redireccionar a otras páginas.

**-Imagen:** maneja las acciones para el manejo de las imágenes de los lugares y servicios ofrecidos por los mismos.

**-Orden:** implementa los métodos para manejar las órdenes de compra realizadas por los clientes.

**-Reseña:** contiene los métodos para crear, ver y borrar reseñas realizadas por los clientes sobre servicios de los propietarios.

**-Service:** controlador encargado de manejar todos los métodos relacionados con el manejo de los servicios del sistema por parte de los propietarios y la posterior consulta por parte de clientes y administradores del sistema.

**-User:** contiene las acciones referentes al manejo de usuarios del sistema.

## Arquitectura de la bases de datos a implementar

**Neo4j:** Es un software libre de base de datos orientado a grafos, implementado en Java. Los desarrolladores describen este motor como de persistencia embebido, completamente transaccional, que almacena datos estructurados en grafos en lugar de tablas, además de las relaciones entre ellos.

La base descrita anteriormente será implementada para el manejo de los usuarios del sistema los cuales se dividen en tres roles (cliente, propietario, administrador), luego los clientes se pueden seguir entre sí para ello se usa las relaciones entre los nodos que ofrece esta base de datos orientada a grafos.



**MongoDB:** Sistema de base de datos NoSQL orientado a documentos. MongoDB guarda estructuras de datos en documentos similares a JSON con un esquema dinámico (MongoDB utiliza una especificación llamada BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. Estos documentos se almacenan en estructuras llamadas colecciones donde se puede acceder mediante diversas consultas. De acuerdo a lo anterior esta base de datos se usará para almacenar los servicios turísticos y manejo de los archivos de tipo imagen del sistema.



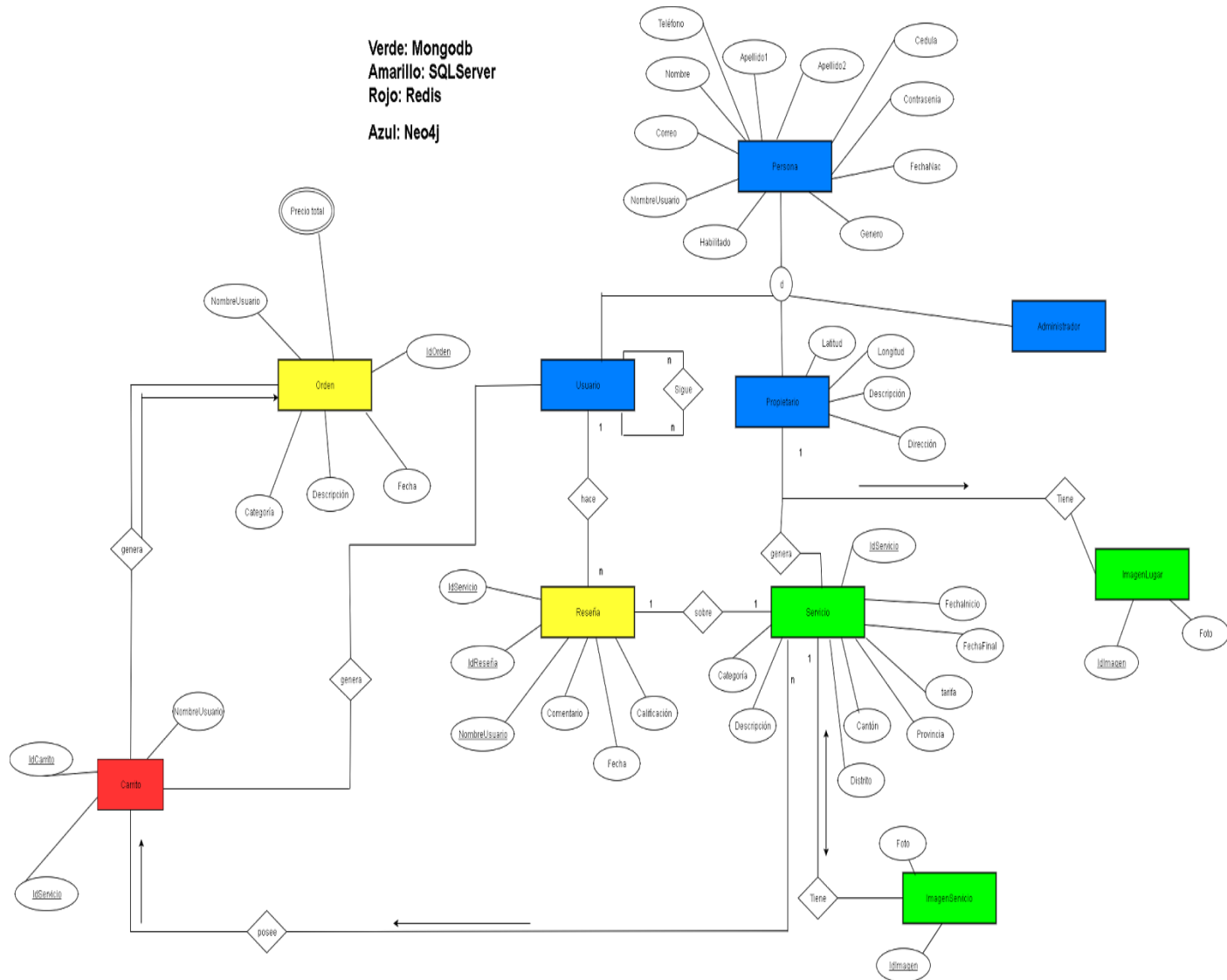
**Redis:** Es un motor de base de datos en memoria basado en el almacenamiento en tablas de hashes (clave-valor), donde presenta un comportamiento persistente. El modelo de datos de Redis se basa en la estructura de datos de tipo diccionario que relaciona una llave a un contenido almacenado en un índice. Esta base cumplirá la función en el sistema de implementar el carrito de compras donde recibe los diferentes servicios de un cliente determinado para procesar la compra cuando esta sea confirmada.



**Microsoft SQL Server:** Es un sistema de manejo de base de datos del modelo relacional. Este motor se encargará de almacenar las órdenes de compra de los clientes del sistema, además de registrar las reseñas realizadas por los mismos clientes sobre los diversos servicios del sistema con el objetivo de mejorar la experiencia de usuario cuando está buscando paquetes turísticos para su posterior compra al tener a su alcance diversas recomendaciones dadas por otros usuarios y la plataforma del sistema.



# Modelo del sistema TurismoCR





## Conclusión

Este proyecto nos permitió como equipo aprender a comunicar diversas bases de datos en un mismo sistema mediante la plataforma de .net incorporado a una aplicación web. Además se aprendió a usar diversos tipos de base de datos entre las cuales podemos mencionar las NoSQL, orientada a grafos, clave-valor y un repaso de bases de datos relacionales vistas en el curso anterior de base de datos. Finalmente se implementó de manera profunda el MVC en el proyecto TurismoCR como una buena práctica a la hora de estructurar las diversas funcionalidades de una aplicación de software de forma modular de tal manera que al realizar mantenimiento sobre la misma facilita los cambios entre las diversas capas. También se aplicó la programación orientada a objetos como recurso ofrecido del lenguaje de programación c#

## Referencias

MongoDB. (2008). Documentación de MongoDB. Sitio web:  
<https://docs.mongodb.com/manual/reference/command/>

Youtube. (2016). Curso de .net MVC 5. Septiembre 13, 2017, de Youtube Sitio web:  
<https://www.youtube.com/watch?v=YzC-FYg66xA>

Wikipedia. (2014). Modelo-Vista-Controlador. Mayo 20, 2017, de Wikipedia Sitio web:  
<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

Wikipedia. (2014). Programación por capas. Junio 15, 2017, de Wikipedia Sitio web:  
[https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_por\\_capas](https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas)

Youtube. (2014). Introducción Neo4j. Septiembre 14, 2017, de Youtube Sitio web:  
<https://www.youtube.com/watch?v=8NMfJmTPmeM&list=LLOgDVR9TH7UPRuP90pXiBTA&index=26>

Microsoft (2008). Controladores y métodos apps ASP.NET MVC. Sitio web:  
[https://msdn.microsoft.com/es-co/library/dd410269\(v=vs.100\).aspx](https://msdn.microsoft.com/es-co/library/dd410269(v=vs.100).aspx)

Github (2016). Cypher CS ejemplos. Septiembre 10 2017. Sitio web:  
<https://github.com/Readify/Neo4jClient/wiki/cypher-examples>