Instituto Tecnológico de Costa Rica

Escuela de computación

Ingeniería en computación

IC-4302 Bases de Datos ll

Proyecto Programado

Sistema de Restaurantes

ESTUDIANTES:

Josué Arrieta Salas 2014008153

Adrián López Quesada

PROFESORA:

Alicia Salazar

Forma De trabajo:

Parejas

VALOR: 30%

17/11/2015

**Índice**

# Manual de Usuario

# Manual Técnico

## División de trabajo

Debido al ser un grupo de 2 personas (Josué y Adrián), y SQL Server al ser un motor de bases de datos con arquitectura cliente servidor; la división de responsabilidades fue la siguiente:

* Josué Arrieta: se encargó de la realización de toda la base de datos: modelo relacional o de implementación, todos los procedimientos almacenados, manejo de roles de usuario y seguridad, todos los procedimientos almacenados (ya sea espaciales), Jobs, triggers, vistas, etc. Realizó esta misma documentación también.
* Adrián López: se encargó de la aplicación: web service de tipo REST, archivos formato JSON, creador de la interfaz utilizando: HTML 5, AngularJS, Javascript, JQuery, CSS, nodeJS. Se encargó de poder almacenar y mostrar imágenes para el usuario (base de datos multimedial).

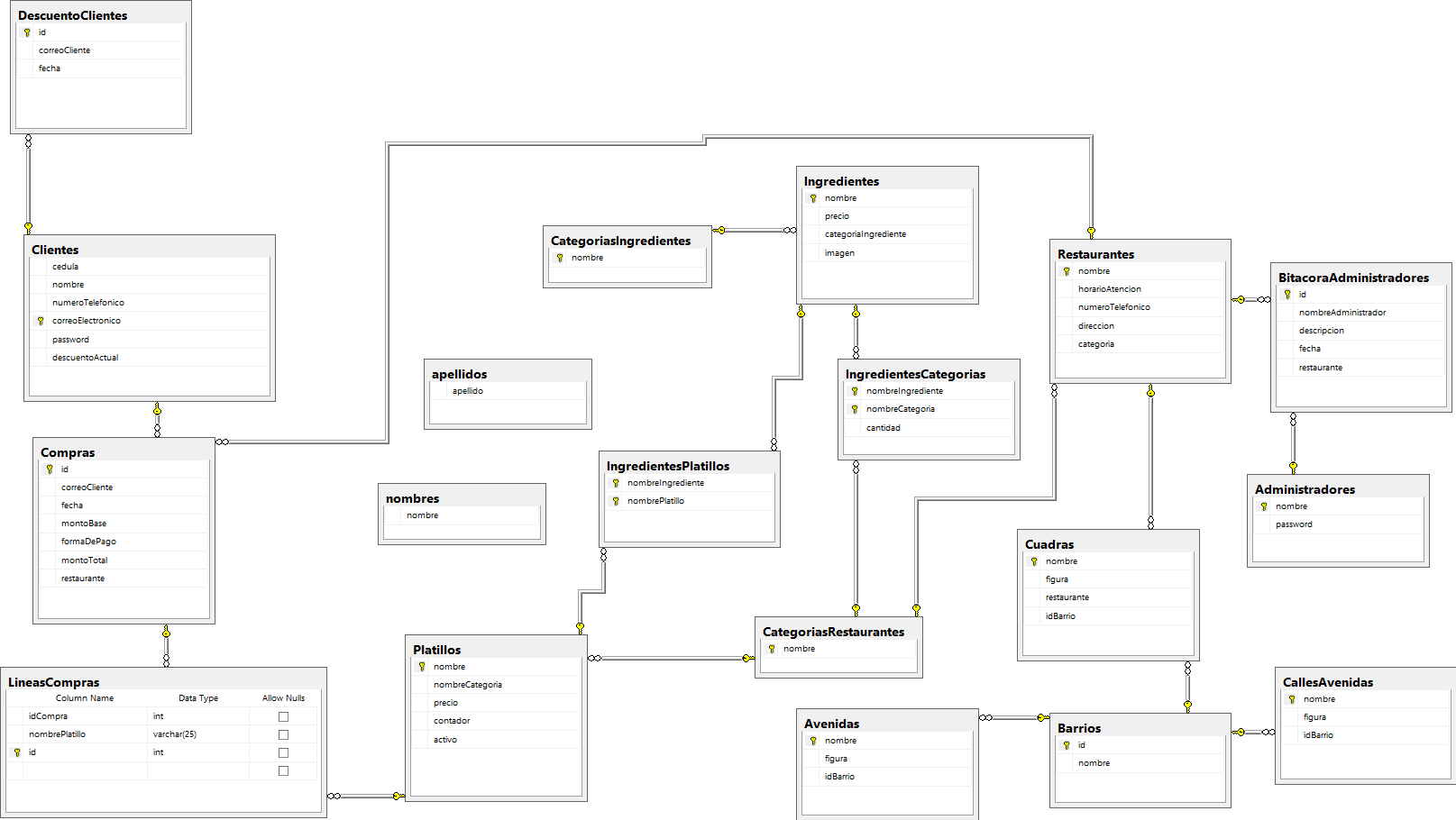
Se utiliza github para el manejo de versiones.

## Convenciones Básicas

* Si es un nombre se usa: varchar(25), también para forma de pago (efectivo o crédito).
* Si es una descripción se usa: varchar(100), igual que un horario de atención y dirección, correos electrónicos , etc.
* Cualquier número será un int (incluyendo números telefónicos). No se usa el tipo money, ya que este tipo solo pertenece a SQL Server y C# no será usado; entonces para evitar
* Todo error manejado será de tipo RAISERROR para creación de mensajes de error definidos por el usuario (además del estándar try catch de SQL server).

## Modelo de Implementación

Se obtuvo el siguiente modelo relacional para la resolución del problema (si no se aprecia se adjuntó imagen Modelo.png):



Se produce un modelo altamente escalable. Se producen las siguientes observaciones:

* Las tablas “apellidos” y “nombres”, no tienen relación con ninguna otra tabla debido a que solo contienen nombres y apellidos para crear clientes aleatorios.
* Clientes y administradores no se utiliza herencia, debido a que los datos de solo poseen un valor en común (password) de modo que no se aprovecharía la herencia y más bien habría que hacer un join de más al buscar alguno de estos (habría que pegar la tabla padre con su respectiva hija); de modo que se perdería eficiencia. Se obtienen 2 tablas por aparte con las características de cada uno: “Clientes” y “Administradores”.
* Para el manejo de que se puedan tener varios platillos en una compra se añade la tabla “LineasCompras”; de modo que una factura posee su información general y a su vez una serie de muchas líneas de compra. Estas líneas de compra solo pertenecen a una factura claramente.
* “BitacoraAdministradores” y “DescuentoClientes”; funcionan como historiales de acciones realizadas por administradores o descuentos aplicados a clientes.
* “Avenidas”, “Barrios”, “Cuadras” y “Barrios”: para el manejo de consultas espaciales. Cada restaurante pertenece a una cuadra en específico.
* “Ingredientes”: se podría decir que es la tabla más importante. Estos pertenecen a una categoría (“CategoriasIngredientes”), además pertenecen a una categoría de restaurantes (“IngredientesCategorias”), además pertenecen a una serie de platillos (“IngredientesPlatillos”). Tiene un campo para la imagen, donde esta será almacenada de forma binaria para ser mostrada en la aplicación.
* “Platillos”: hay un job que se corre todos los días en la madrugada, que los genera. Tienen una casilla de “activo” que es un bit, de modo que si es un ‘1’ es un platillo que se vende en el día actual. Es una aglomeración de ingredientes, donde su precio es la suma de estos. Además cada platillo pertenece a una categoría de restaurante.