

## Tarea 1

### Bases de Datos

*La pizza de Don Cangrejo es la mejor, para ti para mi ♪*

Profesora Cecilia Reyes C.  
reyes@inf.utfsm.cl

Andrea Figueroa R.  
abfiguer@alumnos.inf.utfsm.cl

Camilo Rivas.  
crivas@alumnos.inf.utfsm.cl

Anibal Catalán.  
acatalan@alumnos.inf.utfsm.cl

Alejandro Díaz O.  
ajdiaz@alumnos.inf.utfsm.cl

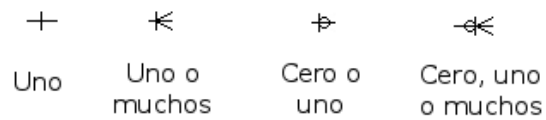
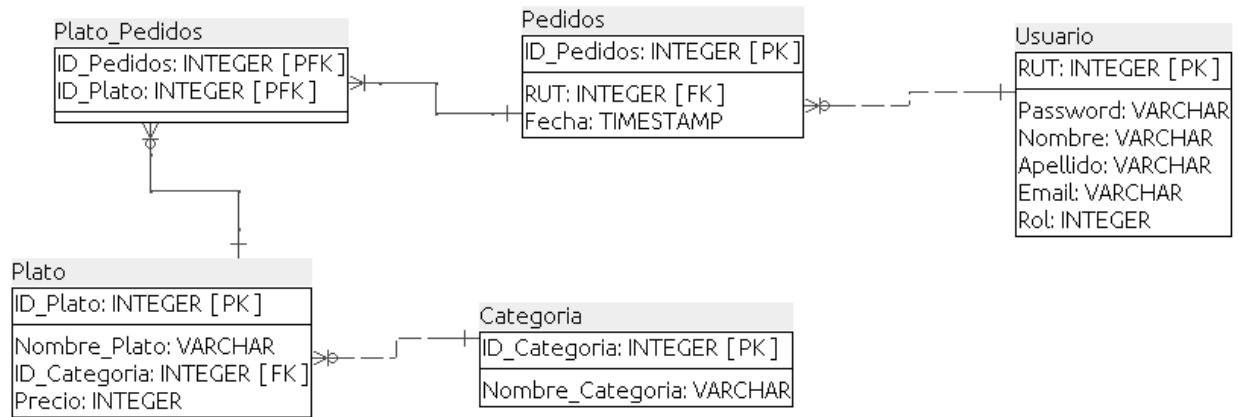
24 de marzo de 2014

## Introducción

El gerente general del prestigioso restaurante "El Crustáceo Cascarudo" está cansado de contratar a un contador cada mes para saber el detalle de sus ventas. En busca de abaratar al máximo los costos, le ha pedido a los entusiastas y responsables estudiantes de la asignatura Base de Datos que desarrollen un sistema web que permita llevar la administración de los pedidos del restaurante. Al darse cuenta de la buena voluntad de los estudiantes, ha decidido agregar el requerimiento de permitir a los clientes realizar pedidos desde el sistema y retirar en el local. Afortunadamente, el modelo de la base de datos usada en el restaurante para los pedidos ya existe y será usada como punto de partida.

Se requiere desarrollar una plataforma que permita el registro de clientes los cuales pueden realizar un pedido y retirar directamente en la tienda, por otra parte los cajeros también tendrán acceso a la plataforma, siendo su función la de administrador del sistema.

## Modelo de Datos



## Requerimientos Técnicos

El principal requerimiento técnico existente es una arquitectura multi-capa que permita el aislamiento de la lógica de la aplicación, la presentación y la persistencia de los datos en componentes separados reutilizables en otras aplicaciones.

Las responsabilidades de cada capa son:

- Capa de Presentación: presentar de forma intuitiva la información, mantener la comunicación con los usuarios a través de elementos idóneos como formularios y listas de selección, no sobrecargar el entorno de trabajo, sólo realizar operaciones inherentes a la presentación de la información.
- Capa Lógica: aplicar las lógicas del negocio a los datos almacenados en la BD, para luego presentarlos como información útil al usuario; establecer un nivel de seguridad sobre el sistema a través de autenticación; interpretar las necesidades de información del usuario y traducirlas, de acuerdo a las reglas del negocio, como mensajes a la capa de datos.
- Capa de Datos: cuidar la integridad de los datos, establecer protocolos de comunicación con la capa lógica para responder las necesidades de datos, manejar el almacenamiento de los datos, proteger los datos ante fallas en el sistema, manejar roles y perfiles que permitan asegurar que los datos serán asequibles sólo por usuarios identificados por el sistema, etc.

Para el diseño físico de las capas se requiere:

- Utilización del Sistema de Bases de Datos PostgreSQL para la capa de los datos.
- Construcción de la capa lógica del negocio en PHP.
- La capa de presentación debe ser en código HTML con uso de plantillas CSS para el diseño de las interfaces. Es recomendable manejar este punto a través de plantillas prediseñadas. Se prohíbe el uso de Flash.

## Requerimientos Funcionales

Se identifican 2 actores relevantes que cumplen los siguientes roles:

1. Cajeros: Cumple rol Administrador
2. Clientes: Cumple rol Usuario

El sistema debe permitir:

1. **Registro de clientes**

Debe existir un proceso de registro de clientes que contemple un RUT **válido**, una contraseña, un correo electrónico y un nombre real.

2. **Login de clientes**

Debe contar con un sistema de login y logout para los clientes. El sistema en todo momento debe verificar que los cajeros no puedan acceder como clientes.

3. **Login de cajeros**

Debe contar con un sistema de login y logout para los cajeros. El sistema en todo momento debe verificar que los clientes no puedan acceder como cajeros. Ojo, no se pide un sistema de registro para los cajeros.

4. **Menú de cliente**

Luego de autenticarse en el sistema como cliente, se debe redirigir a este menú que:

- a) Debe poseer un link que permita realizar un pedido, el cual consistirá en los elementos que se encuentren en el 'carrito de compra'

- b) Cada pedido debe tener un link que lleve al requerimiento 'Búsqueda'.
- c) Debe poseer un link que lleve al requerimiento 'Historial'

#### 5. Perfil de plato

Debe mostrar los detalles del plato (Nombre, categoría, precio) y dar la opción de agregarlo a un 'carrito de compra' (hint: use variables de sesión)

#### 6. Historial

- a) Si está autenticado como cliente debe mostrar un historial de pedidos del usuario, siendo posible filtrar por mes y/o año de forma tal que muestre el total de dinero gastado en los pedidos que cumplan con los criterios de filtrado. Si está autenticado como cajero se aplican las mismas condiciones pero debe mostrar todos los pedidos sin importar a qué cliente pertenecen.
- b) Cada pedido debe tener un link que lleve al requerimiento 'detalle pedido'.

#### 7. Detalle pedido

Debe mostrar el detalle de platos que posee el pedido, tales como nombre del plato y precio. Un cliente no debe poder tener acceso a ver detalles de pedidos que no le pertenecen.

#### 8. Búsqueda

El sistema debe ser capaz de buscar platos por categoría y/o rango de precio y mostrar un link que lleve al requerimiento 'Perfil de plato'.

#### 9. Menú de cajero

Luego de autenticarse en el sistema como cajero, se debe redirigir a este menú.

- a) Debe poseer un link que redirija al requerimiento 'Historial'.
- b) Debe poseer un link que redirija al requerimiento 'Agregar plato'.
- c) Debe poseer un link que redirija al requerimiento 'Agregar categoría'.

#### 10. Agregar plato

Solo debe ser accesible para el cajero, debe permitir agregar nuevos platos.

#### 11. Agregar categoría

Solo debe ser accesible para el cajero, debe permitir agregar nuevas categorías.

#### 12. Bonificación

Se dará un bono de 5 puntos a los grupos que agreguen una imagen asociada a cada plato (ésto incluye que al agregar un nuevo plato desde el panel de cajero, se debe poder adjuntar una imagen)

## Condiciones Generales

- Estimen cuál será la mejor forma de resolver los problemas que pueda notar. Su decisión adjúntela en un documento llamado “supuestos” (en el formato que como equipo estimen conveniente), en donde además podrá explicar cosas que su ayudante deberá tener en cuenta al momento de revisar su tarea. Dado el formato de esta tarea es EXTREMADAMENTE IMPORTANTE el desarrollo de este documento. No se permitirán supuestos que contradigan requerimientos expuestos en este pdf
- Se deja abierta la posibilidad de ampliar y realizar cambios en las tablas (agregar atributos, tablas etc.) o alguna mejora en el modelo. Todo arreglo y funcionalidad extra será bienvenida. Pero debe ser justificado en la página de supuestos.
- Se aplicarán descuentos en el caso de existir links rotos (5 puntos menos por cada link roto).
- No deben aparecer mensajes de error por pantalla, provenientes de la lógica de negocio o del DBMS. Trate los mensajes de manera que un usuario entienda.
- Debe garantizarse el acceso de todas las tablas a sus ayudantes, para eso debe correr el siguiente comando en su consola de trabajo de postgresQL:  
GRANT ALL ON nombre\_de\_tablas\_separadas\_por\_coma TO ayubd;  
(El no cumplimiento de esto implicará nota 0 en la revisión final)

## Consideraciones de Entrega

- **Sobre el trabajo:** La entrega debe ser realizadas en grupos de dos personas previamente inscritos en Moodle, es libre de elegir a quien desee de compañero.
- **Almacenamiento:** Prefiera usar borrado lógico en el caso de necesitar esos datos para algún requerimiento de la tarea o si considera importante dicho dato. En caso contrario, utilice borrado físico.
- **Presentación:** Se prohíbe usar Flash, tendrá nota 0 si incluye animaciones de este estilo.
- **Sobre el modelo de datos:** La implementación del modelo de datos será revisado en forma especial, por lo tanto revise muy bien e investigue o pregunte cómo hacer la base de datos profesionalmente. Su nota será fruto del modelo que usted implemente.
- **Explicaciones:** Luego de la publicación de la tarea, la primera sesión de laboratorio será el momento donde se explicará la tarea, la asistencia es obligatoria. Además, en las siguientes sesiones de laboratorio podrá encontrar a los ayudantes del curso dispuestos a resolver todas las dudas que posean de manera que realicen un excelente desarrollo de la tarea. Los horarios se publicarán vía Moodle.
- **Sobre el trabajo:** Se recomienda avanzar constantemente en el desarrollo del sitio y finalizar la tarea con anticipación a la entrega por la congestión que se produce en los servidores los días finales. No se responderá por problemas de último momento en este sentido.
- **Sobre actitudes indebidas:** Se sancionará con la nota mínima, y sin derecho a reclamo, cualquier situación que no corresponda durante el desarrollo de la tarea. Esto incluye:
  - Copia de trabajos
  - Utilización indebida o no autorizada de cuentas que no correspondan a la asignada por su ayudante.

Que su tarea sea atacada a través de XSS o SQL Injection no se considera actitud indebida, es por ésto que debe cuidar en todo momento la sanitización de los datos que son ingresados a su aplicación.

## Modo de Trabajo

- Se aconseja encarecidamente que aproveche TODOS los horarios del laboratorio para realizar las consultas que posea (la disposición está, de usted depende aprovecharla).
- La tarea puede ser realizada en el Laboratorio de Computación (LabComp) (subterráneo Edif.F) o bien desde su casa a través de ssh que se explicará en ayudantía.
- Cada equipo tiene asignada una cuenta de usuario para creación de las Bases de Datos, las cuales DEBEN tener por nombre bdXX, donde XX es su número de grupo. Todas las páginas PHP creadas deben quedar dentro de una carpeta que debe llamarse html dentro de su cuenta bdXX.
- La página de inicio se debe llamar “index.php”.
- Cuando se cumpla el plazo de entrega, las cuentas serán deshabilitas. No existen casos especiales.
- Cada semana los ayudantes estarán en horario de laboratorio en el labcomp para responder sus dudas y así no sufra contratiempos en su desarrollo.
- Información adicional se publicará en la plataforma Moodle, es de vital importancia que revise constantemente las dudas y anuncios importantes relacionados con su tarea.

## Fechas de entrega

- Modelo de datos implementado en el servidor tareas.inf.utfsm.cl : 06/04/2014 a las 23:55.
- Entrega final: 05/05/2014