

Proyecto de Desarrollo / Proyecto de Software - Entrega 3 -

La finalidad de esta entrega es extender el trabajo de la entrega anterior con la totalidad de la funcionalidad del sistema.

Objetivo del Trabajo

Se desea realizar un sistema de gestión para un banco de alimentos. Se van a manejar pedidos de mercadería, entregas, pedidos modelo, alertas, turnos de entregas, usuarios y configuraciones generales.

Destinatarios

Estará destinado al personal de la asociación, los cuales tendrán diferentes roles (administración, gestión, consultas).

Especificación

El sistema debe contar con ciertos módulos que permitan dicha gestión.

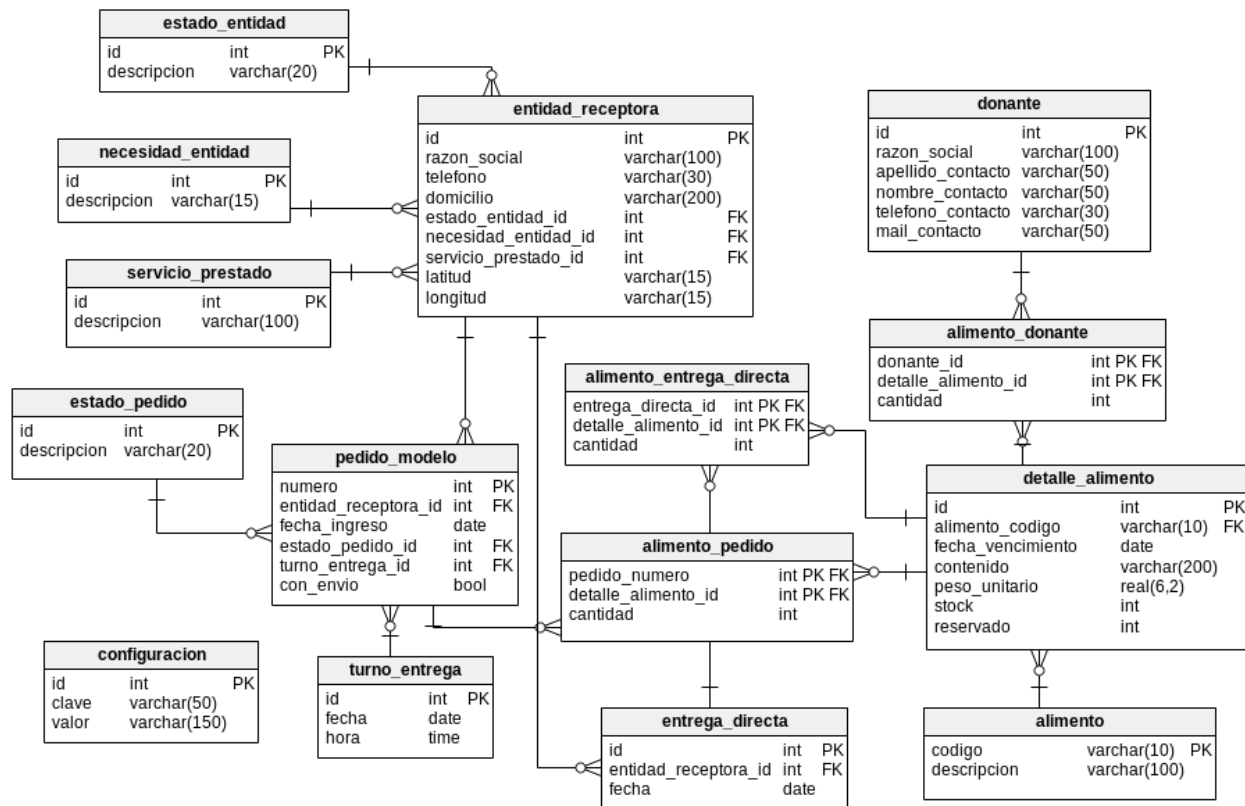
En esta **tercer etapa**, será necesaria la implementación de los siguientes módulos:

1. **Alertas** (se muestran en el home del backend):
 - a. Pedidos a entregar en el día actual mostrando su estado y marcar aquellos que ha pasado su hora de entrega y no han sido entregados.
 - b. La cantidad de unidades con vencimiento cercano en el stock del banco (de N días configurable por el administrador).
2. **Pedidos Modelo**: Es el listado con todos los alimentos que van a ser ofrecidos a una E.R. Se establece un turno de entrega.
3. **Turno de entregas**: Refiere a una agenda interna para tener listos los pedidos en un día y horario.

4. **Entrega Directa:** Cuando la fecha de vencimiento es muy cercana, se entregan sin pasar por el pedido modelo. Solo se podrán seleccionar alimentos que estén dentro del rango configurado por el administrador tal cual indica el inciso 1-b.
5. **Envíos:** Se provee la opción de hacer el envío del pedido por parte del banco hacia alguna E.R. Se podrá seleccionar un día y mostrará los pedidos que hay que enviar (solo de los envíos que realiza el banco) dicho día en un mapa mostrando la trayectoria (logística) del flete. A su vez se debe mostrar información del clima del recorrido obteniendo la información de <http://openweathermap.org> (obteniendo un JSON como resultado de la consulta). Para ello se deben adicionar las coordenadas a las E.R (partiendo desde el banco) desarrolladas en la etapa anterior y en base a cada ubicación mostrar el clima del lugar en el mapa.
6. **Acerca del Banco:** Desde el frontend mediante un link se deben mostrar los datos cargados en LinkedIn por el Banco. La información debe ser obtenida a partir de la API que LinkedIn provee para tal fin (debe ser en formato JSON). Ver: <https://developer.linkedin.com/documents/quick-start-guide>. Para esto deberá crear una cuenta y configurarla apropiadamente y cargar datos del banco. Los datos necesarios para realizar la autenticación deben ser almacenadas en la configuración general del sistema (inciso 8)
7. **Impresiones:** Listados y Estadísticas (debe poder descargarse en PDF)
 - a. Listado (entre fechas) de los kilos de pedidos que fueron entregados (gráfico de barra)
 - b. Listado (entre fechas) de cada E.R y los kilos de alimento que le fueron entregados (gráfico de torta)
 - c. Alimentos vencidos sin entregar (agrupados por descripción)
8. **Configuraciones:** Se podrán modificar las configuraciones generales del sistema, como ser la cantidad de días límite de vencimiento del stock, las coordenadas del banco de alimentos para el mapa de los envíos, datos requeridos para la autenticación de LinkedIn, y todo lo que considere necesario para el sistema.
9. **Usuarios:** Se debe realizar el ABM de los usuarios del sistema y asociarlos a un rol:
 - Administrador: Puede hacer todas las operaciones (aquellas del rol de Gestión más manejo de usuarios y configuraciones del sistema)
 - Gestión: Puede agregar y modificar pedidos modelo, envíos, turnos de entrega y entregas directas, así como también se le visualizarán las alertas y podrá consultar los listados de alimentos en stock y aquellos del inciso 7.
 - Consulta: Puede listar los alimentos en stock y consultar los listados y estadísticas del inciso 7.

Sugerencia

Para simplificar el desarrollo del trabajo, la cátedra sugiere la utilización de la siguiente extensión del modelo de base de datos sugerido en la entrega anterior:



Notas:

- La solución utilizada para planificar (logística) los envíos quedan a criterio del alumno. Al igual que el mapa elegido. en el mapa se muestra solo un día de entrega.
- **La obtención de los datos sobre el clima, mediante su API Rest, debe realizarse desde Javascript, parseando el JSON y configurando el mapa para su visualización.**
- La aplicación no debe ser susceptible a **SQL Injection, XSS ni CSRF**

Consideraciones generales

- Debe ser desarrollado utilizando PHP, HTML5, CSS3 y MySQL, respetando estrictamente el modelo en capas MVC. Debe utilizar PDO como mecanismo de abstracción de bases de datos. Debe tener en cuenta los conceptos de Semántica Web proporcionada por HTML5 siempre y cuando sea posible con una correcta utilización de las etiquetas del lenguaje.
- Debe utilizar **Twig** como motor de plantillas.
- El trabajo será evaluado desde el servidor de la cátedra que cada grupo deberá gestionar mediante Git. **NO se aceptarán entregas que no estén realizadas en tiempo y forma en el servidor provisto por la cátedra.**
- El ayudante a cargo evaluará el progreso y la participación de cada integrante mediante las consultas presenciales y el seguimiento mediante GitLab.
- Toda vista (**HTML5** y **CSS3**) debe validar contra las especificaciones corrientes de la W3C (<http://validator.w3.org/>).
- Tener en cuenta que el servidor de la cátedra utiliza los siguientes servicios y versiones:
 - Servidor Web: nginx 1.6.1-3
 - Servidor BD: mariadb 10.0.13-1
 - Intérprete Php: php 5.5.16-2

Fecha límite de Entrega: Lunes 10 de noviembre a las 8:00 (cierre del servidor)