

Proyecto de Software - Entrega 3 -

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

Objetivo del Trabajo

Se desea desarrollar un sistema que permita llevar a cabo la administración de los productos y servicios relacionados con la gestión del Buffet de la Facultad de Informática.

Destinatarios

Estará destinado al personal del buffet, los cuales tendrán diferentes roles (administración y gestión) y a los usuarios online, los cuales trabajan en la facultad y podrán registrarse para realizar pedidos.

Especificación

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

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

- ❖ Módulo de Menú del día
- ❖ Módulo de Pedidos Online
- ❖ Módulo de Balance (Listados/Estadísticas)

Módulos del sistema:

A continuación se describen las características que deberá poseer el sistema. Debajo del nombre de cada módulo se indican qué roles estarán autorizados a acceder al mismo.

Módulo de Menú del día

Roles: Gestión, Administración

Cada día, el empleado confeccionará la lista de productos que podrán ser comprados en forma online, dichos productos deben tener stock suficiente para poder incluirlos. El detalle de los productos del menú, deberá estar asociado a un día del calendario, pudiéndose crear diferentes menús para distintos días. Por ejemplo, para el Lunes 17/10/2016 podría haber alfajores y sándwiches de jamón y queso, y para el martes 18/10/2016, sándwiches de milanesa y turrónes. El menú del día debe ser visto en la página principal del sitio.

Se deberá crear una bot de Telegram (<https://core.telegram.org/bots>) para consultar el menú del día. Deberá tener (al menos) 2 comandos:

- **/hoy**: Devolverá el menú del día de hoy, en el caso que exista. Si no está creado, deberá informar esta situación.
- **/mañana**: Devolverá el menú del día para el día siguiente, en el caso que exista. Si no está creado, deberá informar esta situación.

Además, desde la aplicación se debe permitir enviar el menú del día, a todos los usuarios suscritos al bot.

Nota: *considerar los cambios al modelo necesarios para registrar las suscripciones.*

Módulo de Pedidos Online

Roles: Usuario online

Esta funcionalidad permite que los denominados “usuarios online” puedan realizar pedidos de ciertos productos seleccionados en el día (los seleccionados para el menú). Se mostrará la siguiente información de los productos ofrecidos:

1. Producto (con nombre, marca, tipo, etc.)
2. Precio por producto
3. Cantidad disponible
4. Observación: para registrar cualquier cambio o comentarios sobre el pedido.

El pedido tendrá tres estados posibles:

- **Pendiente**: aún no entregado.
- **Entregado**: se registra el pago y se descuenta del stock.
- **Cancelado**: en caso que el usuario online desee hacerlo en la primer media hora de haber hecho el pedido o que el empleado del buffet lo haga por alguna razón particular.

Solamente podrán utilizar esta funcionalidad aquellas personas que tengan un lugar de trabajo/ubicación conocida. Por ejemplo, docentes de los laboratorios, o personal de administración. Esto es importante para que el pedido pueda ser entregado a la oficina/espacio de donde proviene el usuario.

El usuario puede elegir uno o más productos de ese listado y ese pedido es creado con estado **Pendiente**.

El usuario online podrá visualizar todos los pedidos realizados por él en un rango de fechas y con el estado correspondiente.

Administración de estados del Pedido online

Roles: Gestión, Administración

El usuario de Gestión y Administración, puede **aceptar o cancelar un pedido**:

- En el caso de que **acepte** el pedido, el mismo pasa de estado **Pendiente** a **Entregado**, sumar el total del precio del pedido a “ingresos” (dependiendo de cómo hayan armado el modelo) y debe controlar el stock y luego descontar los productos seleccionados.
- En el caso de que **cancele** el pedido, el mismo pasa de estado **Pendiente** a estado **Cancelado**, y puede ingresar algún comentario sobre el motivo de la cancelación (no deberá descontarse stock, ni sumar a “ingresos”).

Roles: Usuario online

El Usuario online, puede cancelar un pedido, pero sólo en la primer media hora de haberlo realizado, en ese caso el estado del pedido pasa de **Pendiente** a **Cancelado** y puede ingresar algún comentario sobre el motivo de la cancelación.

Módulo de Balance

Roles: Administración, Gestión

Este módulo permite llevar un control de los ingresos/egresos en función del capital. El balance se realizará, por día o mes

El objetivo del módulo **Balance** es visualizar las **ganancias**. Se define la ganancia como

$$\text{GANANCIA}(\text{fecha_fin} - \text{fecha_inicio}) = \text{INGRESO}(\text{fecha_fin} - \text{fecha_inicio}) - \text{EGRESO}(\text{fecha_fin} - \text{fecha_inicio})$$

Es decir, se calcula la ganancia en el intervalo de tiempo que el usuario define.

Se consideran como:

3.1 Ingresos :

- **Venta buffet:** ingresos recaudados a partir de la venta del buffet, incluidas las ventas por pedidos online

3.2 Egresos

- **Proveedores:** pago a proveedores. Esto está relacionado con la compra de productos. Al dar de alta una compra de cualquier producto, se registrará el pago al proveedor correspondiente como un gasto.

Deben realizarse las siguientes estadísticas:

- Listado y Gráfico de barras con las ganancias (ingreso-egreso) del buffet por día, entre dos fechas. Por ejemplo, si se eligen las fechas 01/11/2016 y 30/11/2016, se mostrará por cada día de Noviembre, una barra que representará la ganancia de ese día.
- Listado y Gráfico de torta con las cantidades de venta de cada producto, entre dos fechas.

Ambos gráficos y listados deben exportarse a PDF.

Para las estadísticas se puede utilizar cualquier librería siempre y cuando sea de uso libre, como por ejemplo Highcharts (<http://www.highcharts.com/>), Charts.js (<http://www.chartjs.org/>) o CanvasJS (<http://canvasjs.com/>)

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, teniendo la posibilidad de utilizar algún ORM. 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.
- La aplicación no debe ser susceptible a **SQL Injection, XSS ni CSRF**
- 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/> y <https://jigsaw.w3.org/css-validator/> respectivamente).
- Tener en cuenta que el servidor de la cátedra utiliza los siguientes servicios y versiones:
 - Servidor Web: Apache 2.4.10
 - Servidor BD: Mariadb 10.0.26
 - Intérprete Php: Php 5.6.24
- **Importante:**
 - Les recordamos que la entrega está pensada para ser realizada por un grupo de 3 integrantes. Quedando en ustedes la responsabilidad de su conformación. Pudiendo ser menor la cantidad de integrantes, pero debiendo realizar el mismo trabajo que todos, sin excepciones.
 - La repetición de código podría implicar la desaprobación de la entrega.

Fecha límite de Entrega: 7 de noviembre a las 8:00 (cierre del servidor)

Fecha de Reentrega: 12 de diciembre a las 08:00 (cierre del servidor)