

## **Enunciado proyecto PHP “Almacén de cajas”**

### **Fase I**

Diseñar una aplicación basada en el diseño de clases de la jerarquía “Caja” con acceso a tablas en un servidor en el que debemos implementar un determinado almacén de cajas que se encuentran dispuestas en las lejas por estanterías.

La primera parte del diseño de la aplicación consistirá en el diseño de las tablas que configuran el entorno de trabajo.

Aspectos a tener en cuenta:

- El almacén está organizado en pasillos que contienen huecos en donde se colocan las estanterías. Cada pasillo está identificado por una letra y dentro de cada pasillo, los huecos están identificados por un número, de forma parecida a la distribución de plazas de garaje en un centro comercial. Es decir, cada estantería tiene asociado un lugar del almacén mediante coordenadas pasillo-número (Ej. C-2).
- Las distintas estanterías se irán colocando en esos huecos de manera discrecional, a elección del administrador del almacén.
- El número de huecos por pasillo podrá ser diferente.
- Las cajas que formarán parte del almacén son del tipo genérico Caja, pudiendo contener cualquier objeto.
- Cada estantería podrá tener un número diferente de lejas.
- Cuando se creen cajas, estas se depositarán en una leja libre de una determinada estantería. Se hará también de manera discrecional, a criterio del administrador del almacén.

Las operaciones a realizar sobre la web del almacén son:

#### **1.- Alta de Estanterías. Ubicación de las mismas en el almacén.**

De la Estantería reflejaremos:

- Código.
- Material del que está hecha.
- Número de lejas.
- Ubicación (pasillo y hueco).
- Fecha de alta.

Elegiremos un pasillo y hueco donde colocarlas. Solo se habilitarán en el formulario de alta los huecos libres en el almacén, por tanto, esta operación de elección de pasillo y número se debe implementar con generación dinámica de componentes web.

Al verse afectadas tres tablas, en esta operación es obligatorio realizar un control de transacciones.

#### **2.- Creación de cajas y asignación de leja dentro de una estantería.**

De la Caja registraremos:

- Código.
- Material del que está hecha.
- Color.
- Dimensiones (altura, anchura y profundidad) en centímetros.

- Contenido.
- Ubicación (estantería y leja destino).
- Fecha de alta.

La estantería destino se puede elegir dentro de las disponibles, es decir, que tengan lejas libres y dentro de esta se podrá elegir la leja no ocupada que se estime oportuna. Esta operación de elección de estantería y leja se debe implementar con generación dinámica de componentes web.

Al verse afectadas tres tablas, en esta operación también es obligatorio realizar un control de transacciones.

**3.- Listado de ambas opciones.** Con todos sus datos y además:

- Estanterías. Indicando letra de pasillo y número de hueco.
- Cajas. Indicando código estantería y leja donde se ubica.

**4.- Baja de Estanterías.** Opción para dar de baja una estantería por deterioro u otros motivos. Si tuviese cajas no dejar dar de baja hasta que esté vacía. El procedimiento para dar de baja lo debe adoptar el programador, teniendo en cuenta que se debe anotar la fecha de baja y que las cajas en backup que hubiesen estado en dicha estantería podrán ser referenciadas.

**5.- Control de errores.** En todo proceso que lo requiera se implementará un control de errores con captura de excepciones. Crear clases de excepciones propias. Derivar su visualización a una página de errores común.

**6.- Uso de sentencias preparadas.** En todo campo de formulario susceptible de captar código malicioso en forma de SQL Injection, se utilizarán sentencias preparadas para las órdenes SQL oportunas.

**7.- Inventario del almacén.** Consiste en obtener una relación detallada de las distintas estanterías del almacén ordenadas por pasillo y número, y la descripción completa en cada una de ellas de las cajas que contuvieran ordenadas a su vez por número de leja.

*[Comienzo 25-01-2022] [Fecha prevista finalización: 08-02-2022]*

## Fase II

### 8.- Implementar la salida de cajas.

Se debería hacer pidiendo el código de la misma y mostrando su descripción completa antes de eliminarla del almacén.

Cualquier caja que abandone el almacén tendrá su registro en una tabla de backup junto con la fecha de salida.

Esta operación es obligatorio realizarla con disparadores de base de datos.

Se guardará la ubicación (estantería y leja) en estructuras de tabla de backup creadas a tal efecto. Estas tablas deberán reflejar siempre la ubicación que tenía la caja antes de ser sacada del almacén. Se debe tener en cuenta que la estantería puede eventualmente darse de baja.

Ya que ahora existe otra tabla que registra cajas, para dar de alta nuevas debemos comprobar que el código no se repite en ninguna de las dos.

*[Fecha prevista finalización: 18-02-2022]*

### **Fase III**

**9.- Implementar la devolución de una caja al almacén.** Se tomará como dato el código que tenía en el almacén. Se procederá a rescatarla mostrando su ubicación anterior y borrarla de las tablas de backup correspondientes e insertarla en su tabla respetando el código que tenía antes. La ubicación puede ser distinta.  
Esta operación es obligatorio realizarla con disparadores PHP.

*[Fecha prevista finalización: 25-02-2022]*

### **Fase IV**

#### **10.- Login de administrador.**

Solamente habrá una persona con credenciales para operar con la gestión de almacén. Para respetar este requisito, se creará una tabla con usuario y password de ese administrador.

Dicha tabla la creará vacía el administrador de la base de datos separadamente de la aplicación. La primera vez que ejecute la aplicación se pedirá el registro de administrador. La contraseña se guardará encriptada y con un nivel de seguridad suficiente.

A la entrada de la aplicación se comprobará si ya existe uno, de ser así se pedirán sus credenciales.

*[Fecha prevista finalización: 02-03-2022]*

### **Fase V**

#### **11.- Web híbrida.**

Inclusión de algún script de otras web que realcen los servicios aportados por nuestra aplicación. (Tiempo, RRSS, Mapas, etc.)

*[Fecha prevista finalización: 04-03-2022]*

### **Fase VI**

#### **12.- Documentación de la aplicación.**

Como mínimo se deberá aportar:

- 1.- Diagrama de clases.
- 2.- Esquema de tablas y sus relaciones.
- 3.- Casos de uso.
- 4.- Flujo de datos. En cada caso de uso o general.
- 5.- Variables de sesión empleadas.

*[Fecha prevista finalización proyecto: 08-03-2022]*

**Notas:**

- *La codificación en PHP se hará siempre con el uso de objetos y respetando el paradigma Modelo-Vista-Controlador. Además, para la interacción entre los objetos y las tablas se implementará una capa DAO (Data Access Object).*
- *Los códigos de Caja y Estantería se formarán con dos primeras letras indicando el tipo de objeto (CA, ES) y posteriormente 3 caracteres numéricos generados manualmente (Ejemplo CA211).*
- *Para que el proyecto pueda ser considerado evaluable, todas las fases deben estar implementadas.*