



# Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

## **PECL 3 – ARTEFACTO 9**

### **MODELO DE DATOS**

### **Ing. Software**

Laboratorio Martes 12:00 – 14:00

Grado en Ingeniería Informática – Curso 2018/2019

Eduardo Graván Serrano – 03212337L

Marcos Barranquero Fernández – 51129104N

Sonia Rodríguez-Peral Bustos - 54302528B

Adrián Montesinos González – 51139629A

Alejandro Caballero Platas – 50891258D

Tabla de contenido

Decisiones de diseño del modelo relacional ..... 3

Modelo relacional ..... 5

Otros elementos de almacenamiento..... 6

## Decisiones de diseño del modelo relacional

Se ha decidido usar el sistema gestor de base de datos PostgreSQL en su versión 10 para la creación de la base de datos, ya que se ajusta a las especificaciones descritas y además los ingenieros tienen un nivel de conocimiento superior con este sistema, pudiendo dar un mejor servicio usando PostgreSQL que el que se pudiese dar usando otros gestores similares.

Durante la creación del modelo relacional se ha intentado ser lo más fiel posible al modelo entidad relación que teníamos anteriormente, siempre que la estructuración de la base de datos fuese viable y no afectase al rendimiento.

Esto se ve reflejado sobre todo en la transformación de las relaciones de herencia que se ha hecho. No se han eliminado ninguna de las entidades hijas que teníamos en el modelo ER, si no que se ha hecho una transformación 1 a 1, lo cual ha resultado en que algunas de las relaciones hijas, sobre todo de la tabla empleado, tengan solo un atributo que es clave primaria y foránea.

Sobre la relación de herencia de Empleados, se puede remarcar que se ha decidido dejar todos los subtipos de empleados para tener un control más directo sobre las diferentes funciones de los empleados que haya en la base de datos y que sea más directa su clasificación. Se mantiene la relación Superusuario para almacenar también aquellos usuarios que tengan estos privilegios totales y tenerlos bien identificados, aunque no sea estrictamente necesario ya que realmente depende del rol de acceso a la base de datos.

En cuanto a la forma en que interactúan los pedidos con las piezas que van en dichos pedidos, se ha determinado que la mejor forma de llevar a cabo esta relación es hacer una relación identificativa en la cual se pasa la clave primaria de pieza como clave foránea y primaria de los pedidos, tanto especiales como automáticos. De esta forma conseguimos que pueda haber más de un tipo de pieza por pedido. La tabla Pedido de la cual heredan las otras 2 tiene un atributo llamado cantidad para contabilizar los números de pieza que hay por cada pedido.

Se han generado dos relaciones nuevas que son las tablas intermedias entre Parte\_Trabajo y Presupuesto con Piezas. Ambas tablas intermedias tienen las dos claves primarias de las relaciones correspondientes como clave primaria y foránea. A parte de esto, el atributo cantidad que antes correspondía a la propia relación, se ha añadido a estas nuevas tablas intermedias para poder registrar el número de piezas que se usan/estiman con cada Parte/Presupuesto.

Toad crea índices sobre las claves foráneas de manera automática, estos índices no se han modificado, ya que se está usando PostgreSQL, estos índices son los creados por defecto, que son de tipo árbol.

En cuanto a integridad referencial, la gran mayoría de relaciones entre tablas se han configurado para que la integridad referencial tanto en update como en delete se encuentren en modo restrict, de esta forma la base de datos mantiene un nivel de sobrecarga bajo cada vez que se intente llevar a cabo una de estas operaciones y se mantiene un alto nivel de coherencia entre los datos que se almacenan.

Hay un par de excepciones, que es entre las tablas Factura y Petición\_Trabajo, y la relación de herencia de la tabla Empleado con las tablas Técnico, Coordinador, Superusuario y Responsable\_Almacén. La integridad referencial entre estas relaciones está configurada para ser en cascada, tanto para update como para delete. Esto se ha configurado así ya que las tablas "hijas" de estas relaciones solo se relacionan con las tablas de las que heredan la clave foránea, por lo que el borrado/actualización en cascada no supone una sobrecarga alta en la base de datos, además de que las claves foráneas forman parte de la clave principal de las tablas, por lo que se considera que la integridad referencial en este modo es válida para estos casos en concreto.

Se han introducido constraints en las tablas que tienen un dominio específico para de esta forma evitar que se introduzcan datos fuera de rango en la base de datos. Ejemplos de estos constraints pueden ser los posibles valores de prioridad de los pedidos, o el estado de las peticiones de trabajo.

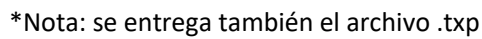
Como cambios de cara a la entrega final, se puede comentar que se ha creado una nueva entidad “Informe”, en la cual se almacenarán los informes generados por la aplicación, relacionándolo con la oficina desde la cual se ha creado dicho informe.

Para que la interacción entre las piezas y los pedidos fuese más limpia, se ha decidido introducir dos tablas intermedias entre estas relaciones. La única función de estas tablas es relacionar los pedidos con las piezas que van en estos pedidos. Esto permite la existencia de un atributo cantidad que se encargará de contabilizar el número de piezas de un mismo tipo que va en cada pedido.

Se han añadido dos atributos a la relación “Proveedor”. Ambos campos están relacionados con la forma en que se hacen los pedidos a dichos proveedores. El primero de ellos almacena la URL del Google Drive al cual se subirá el archivo XML con el que se realizan los pedidos. El segundo campo almacena la plantilla a rellenar para hacer el pedido en formato XML.

No se ve conveniente modificar la relación entre las tablas “Proveedor” y “Pieza”, ya que entendemos que el identificador es único para una pieza en una oficina. Esto quiere decir que, si se tiene otra pieza igual en otra oficina, creemos más conveniente que tenga un identificador totalmente nuevo.

Este es el modelo relacional obtenido:



\*Nota: se entrega también el archivo .txp

## Otros elementos de almacenamiento

Los pedidos a proveedores se hacen a través de la subida de archivos XML a un Google Drive asociado a cada proveedor que se genera, se rellena y se sube automáticamente a esta plataforma. Los proveedores descargan estos archivos XML y atienden a la petición de compra de suministros.

El formato de estos archivos XML se acuerda con el proveedor antes de empezar la actividad comercial. Cuando se ha llegado a un acuerdo, se almacena una plantilla en la base de datos, que se rellenará de cara a hacer los pedidos.

A parte de esto no se guarda ningún tipo de información en archivos de otro tipo.