

## Proyecto agenda V3

Ahora han sido los amigos de Carlos los que han enseñado a sus propios amigos la Agenda. El resultado es que, han sido tantos los que se han registrado deseando poder gestionar sus contactos a través de la aplicación de Carlos que la aplicación se ha quedado pequeña. Por tanto, Carlos se ha propuesto escalar un nivel o dos más la aplicación:

1. Por un lado, desea instalar un Sistema Gestor de Bases de Datos y crear en él una base de datos en la que persistir todos los datos de la aplicación. Esto, además, le permitirá incorporar seguridad adicional a la aplicación
2. Por otro, también quiere incorporar el patrón MVC a la misma

Así que, tomada la decisión en firme, Carlos ha decidido:

- Instalar SQL Server Express Edition, versión gratuita
- Instalar el driver SQLSRV para comunicar la aplicación PHP con la BD SQL Server

Tras haber llevado a cabo las resoluciones anteriores, Carlos se plantea que, aprovechando la incorporación del SGBD SQL Server a la gestión de los datos de la aplicación, la Agenda pueda gestionar nuevos datos y proporcionar nuevas funcionalidades. Así que, a los datos que ya administra:

- Usuarios
- Datos de contactos
- Permisos de los usuarios

Se plantea agregar:

- Familiares de los contactos, y tal vez, relaciones entre los contactos, los familiares, etc. Entre esta información, se plantea incorporar datos sobre lugar de trabajo de cada uno de ellos, etc.
- Eventos de tipo calendario: agenda de actividades a realizar en el futuro, cumpleaños, etc.
- Imágenes de personas, mascotas, momentos, etc., relacionados con los usuarios de la agenda, sus contactos, familiares y relaciones

Teniendo en cuenta todo lo anterior Carlos se plantea:

- Necesitaré crear un modelo E/R a partir del cual implementar la base de datos
- Necesitaré crear una o más rutinas que me permitan migrar los datos actuales de usuarios, roles y agendas a las tablas de la BD

Una vez realizado lo anterior, y por el lado operacional, Carlos se propone:

- Adaptar el código actual de la aplicación al patrón MVC, separando el código según su funcionalidad, situando:
  - la lógica de la aplicación, en uno o varios controladores,
  - la lógica de negocio, en el modelo, y
  - las ventanas con las que los usuarios interactúan con la aplicación, en diferentes vistas
- Dado el gran número de nuevos objetos de código que necesitaré, y la posible incorporación de rutinas ajenas en las que se definen otros objetos de código, dividir en espacios de nombres los objetos de código de la aplicación para evitar colisión de nombres
- Permitir a los usuarios que puedan modificar su perfil
- Agregar las funcionalidades necesarias para permitir a los usuarios de la aplicación gestionar los nuevos datos que ahora pueden incorporar a sus agendas