

Proyecto de Sistemas Informáticos,
Facultad de Informática,
Universidad Complutense de Madrid



Facultad
de
Informática



Generación de menús adaptativos

Curso 08/09

Prof. Director:

D. Miguel A. Blanco Rodríguez

Proyecto realizado por:

Santiago Novoa Bedito

M^a Dolores Perea Clemente

David Molina Rodríguez

Autorización a la UCM

Autorizo a la Universidad Complutense a difundir y a utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Fdo. Santiago Novoa Bendito

Fdo. M^a Dolores Perea Clemente

Fdo. David Molina Martínez

Índice

1.	Resumen	4
1.1.	Summary	4
2.	Palabras clave	5
3.	Introducción	6
4.	Desarrollo del proyecto	8
5.	Motivación	10
6.	Objetivo del proyecto	15
7.	Elementos que intervienen en la aplicación	16
7.1.	Usuarios	16
7.2.	Aplicaciones	17
7.3.	Nivel de acceso	18
7.4.	Filtrado	19
7.5.	Presentación de la información	19
8.	Ejemplo de uso	21
8.1.	Análisis de la información utilizada	24
9.	La base de datos	26
9.1.	Comprobación de las entidades	¡Error! Marcador no definido.
9.1.1.	Explicación de la figura	288
9.1.2.	Comprobación de las relaciones	10
10.	Motor de Inferencia	3515
10.1.	Ejemplo de funcionamiento del motor de inferencia	35
11.	Prototipo	10
11.1.	Interfaz gráfica con la base de datos	40
11.2.	Componentes del prototipo	42
12.	Conclusiones y extensiones	46
13.	Anexos	47
13.1.	Script SQL: generación de la base de datos	47
13.2.	Script SQL: datos para el prototipo	51
14.	Bibliografía	53

1. Resumen

El proyecto consiste en desarrollar una aplicación de carácter genérico que estudie el uso que de una determinada aplicación hacen los distintos usuarios. El objetivo principal es agrupar en niveles o perfiles a los usuarios para facilitar el uso y reducir el tiempo que un usuario emplea en hacer las consultas que normalmente realiza. Además se podrá personalizar la secuencia de formularios que se muestran permitiendo saltarse aquellos intermedios que no se usen.

Para ello la aplicación mantendrá un estudio estadístico sobre las opciones, botones o menús incluso el tiempo empleado en cada formulario. Por otro lado se guardará información sobre las secuencias de acciones que realiza un usuario y la frecuencia de éstas.

De esta forma se consigue que una vez conectado un usuario el programa sepa con cierta certeza aquella información que se va visualizar, el orden y las acciones a realizar de tal forma que pueda ir adaptando el aspecto de la aplicación al usuario.

1.1. Summary

This project consists of developing a generic application to study the way different users manage a certain application. The main goal is to classify those users in specific profiles to reduce their time spent when dealing with the system. Furthermore, it will be possible to customize the form sequence shown and skip those forms which are not used.

To achieve that, this program will keep a statistic study about the options, buttons, menus and even the time spent on each form. Moreover, it will register information about the action sequences executed by an user as well as its frequency.

Therefore, once logged, the system knows with certain accuracy the information which will be shown to a user as well as its order and the actions to perform so that the interface layout can be adapted to them.

2. Palabras clave

Base de datos, motor de inferencia, modelo de usuario, entidad, relación, menú adaptable, aplicación, acceso, información, SHA.

3. Introducción

Se trata de desarrollar un sistema que se ejecute en background bajo un programa genérico foreground.

La aplicación consistirá básicamente en una base de datos que registrará información acerca del uso de gestión que hace un usuario de su programa sin que sea necesario estar registrado, porque existirán determinadas aplicaciones que no requieran autenticación. Para ello se guardará la traza de las iteraciones que realiza un usuario en la aplicación informática.

Esta traza de seguimiento de todos los formularios e informes que recorre y el tiempo que dedica en cada uno, servirá para que un sistema en función de estos recorridos sea capaz de hacer propuestas de nuevos formularios y menús al usuario porque es reincidente o tiene un perfil de comportamiento que ya está estudiado.

La información contenida incluye datos del usuario, del tipo de aplicación foreground manejada, el nivel de acceso permitido según su rango, el filtrado de datos a mostrar dependiendo de su puesto concreto y el formato de la información.

La forma en la que la base de datos toma las decisiones será usando datos estadísticos proporcionados por un motor de inferencia que recibirá información de la base de datos y clasificará a los usuarios en grupos en función de la información recibida.

El objetivo de crear estos perfiles es que el usuario, con el tiempo, encuentre su programa guiado porque la interfaz se habrá personalizado de acuerdo con las secuencias de opciones más frecuentes que utiliza. También podrá sugerir las siguientes opciones a realizar. Con esto se

Generación de menús adaptativos

puede conseguir que cada usuario o trabajador ahorre tiempo en sus actividades cotidianas aumentando así su productividad.

4. Desarrollo del proyecto

El proyecto se ha desarrollado siguiendo las instrucciones y explicaciones que el profesor director nos ha ido proporcionando semanalmente. Durante el desarrollo se pueden distinguir tres fases claramente diferenciadas:

- Fase 1: en la primera fase del proyecto nos centramos en definir el objetivo del proyecto y todos los elementos que intervienen en éste, de forma que quedara claro todo lo necesario para realizar el diseño de la base de datos que es el núcleo del proyecto. Esta fase se corresponde con el punto 5 de esta memoria.
- Fase 2: en esta fase se realizó el diseño de la base de datos. Con la ayuda del desarrollo de ejemplos de uso, pudimos definir con claridad las distintas entidades que se necesitan junto con sus atributos. Una vez definidas las entidades y revisando los ejemplos se pudieron definir las relaciones necesarias para que la base de datos proporcionara la información suficiente y necesaria para lograr los objetivos perseguidos. Esta fase corresponde con los puntos 6 y 7 de la memoria.
- Fase 3: en la última fase realizamos dos programas básicos para comprobar el correcto funcionamiento de la base de datos. El primero consiste en una simple interfaz gráfica para insertar todos los datos necesarios a la utilización en un programa real. El segundo consiste en un prototipo de aplicación que utiliza la información almacenada previamente para modificar el aspecto de los distintos formularios que posee según se hayan usado previamente. Por otro lado, durante esta fase también se ha realizado esta memoria, reuniendo toda la información generada a lo largo del curso y organizándola convenientemente para su

correcta presentación.

Como se ha dicho anteriormente, durante todo el curso se han realizado reuniones semanales con el profesor. En estas reuniones presentábamos el trabajo realizado durante toda la semana y las propuestas de mejora o ideas que se nos podían ocurrir para mejorar tanto el desarrollo del proyecto como el proyecto en sí mismo. Una vez exponíamos lo realizado, el profesor nos indicaba el siguiente paso que debíamos realizar y los errores que habíamos cometido para corregirlos. De cada reunión se generaba un acta en la cual constaba la fecha, los asistentes, los temas tratados, los acuerdos alcanzados y el trabajo a realizar para la siguiente reunión.

Para llevar a cabo el trabajo, los miembros del grupo realizábamos dos reuniones a la semana. En la primera, que solía ser después de la reunión con el profesor o al día siguiente, nos repartíamos el trabajo a realizar, de tal manera que las tareas se ajustaran a los conocimientos que cada uno tenemos sobre los distintos temas que se han tratado. En la segunda reunión, previa a la asamblea con el profesor, poníamos en común el trabajo realizado, discutíamos las propuestas que cada uno realizaba y redactábamos nuestras conclusiones para exponérselas al profesor.

5. Motivación

Actualmente existen numerosos proyectos relacionados con sistemas adaptativos. Por ejemplo, es muy conocido el caso del spam, donde se envía anuncios y ofertas dependiendo de la información que se ha conseguido a partir del usuario, cuando éste ha cumplimentado alguna petición o búsqueda relacionada con un tema. A estos sistemas se los conoce como Sistemas Hipermedia Adaptativos (SHA).

El objetivo final consiste en que al usuario se le presente información ajustada a su necesidad, elaborando para cada uno un perfil de preferencias, objetivos y conocimientos con el fin de adaptarse dinámicamente a las necesidades de cada usuario.

Lo frecuente es que estas aplicaciones aparezcan integradas en páginas Web, aunque la idea es igual de válida que en las aplicaciones informáticas habituales.

Son muy numerosos los sectores interesados por los SHA: las plataformas de venta y distribución, la banca on-line, los buscadores y los portales generalistas, los sistemas de e-learning, etc. Hay quien afirma que, en muy pocos años, todos los sitios Web serán dinámicos y adaptativos, la edad dorada de los editores de HTML plano parece estar concluyendo. De hecho, existe una amplia oferta de portales dinámicos preconfigurados cuya implementación en entornos educativos ofrece considerables ventajas sobre otras soluciones Web.

Podríamos definir un SHA con las siguientes alternativas:

- Sistema que en función de una serie de variables que responden a diferentes perfiles de usuario adaptan dinámica o estáticamente los contenidos, los elementos multimedia, el mapeado del Website e

incluso el aspecto de la propia interfaz.

- Sistemas que se adaptan al usuario basándose en las suposiciones (heurísticos, inferenciales) o en las preferencias (paramétricos) implícitas o explícitas del mismo.
- Portales dinámicos en los que los elementos estructurales son independientes de los contenidos y que facilitan la adaptación manual o automática a una tipología de perfiles, preferencias o niveles de accesibilidad.
- Sistema que elabora para cada usuario un perfil de preferencias, objetivos y conocimientos con el fin de adaptarse dinámicamente a las necesidades de cada usuario.

Existen diversos tipos de SHA que podríamos clasificar en:

- **Hipermedias adaptables:** Aquellos que requieren la participación activa y voluntaria del usuario (vg. mediante cuestionarios de preferencias, de conocimientos...) para elaborar un perfil explícito y por lo general estático.
- **Hipermedias adaptativos:** Los que incorporan algoritmos que monitorizan de manera automática el comportamiento de los usuarios en el portal, analizando las acciones de navegación, los tiempos de permanencia en determinadas secciones, las palabras clave introducidas en los formularios de búsqueda, las descargas realizadas, etc. reconfigurando la información e incluso la estructura de la interfaz en sucesivas visitas.
- **Hipermedias dinámicos:** Comparten las características de los anteriores, pero la información sobre los contenidos, los elementos estructurales y los perfiles de usuario está atomizada en bases de

datos que permiten reconstruir dinámicamente una página *Web* (o documento hipermedia) diferente para cada usuario.

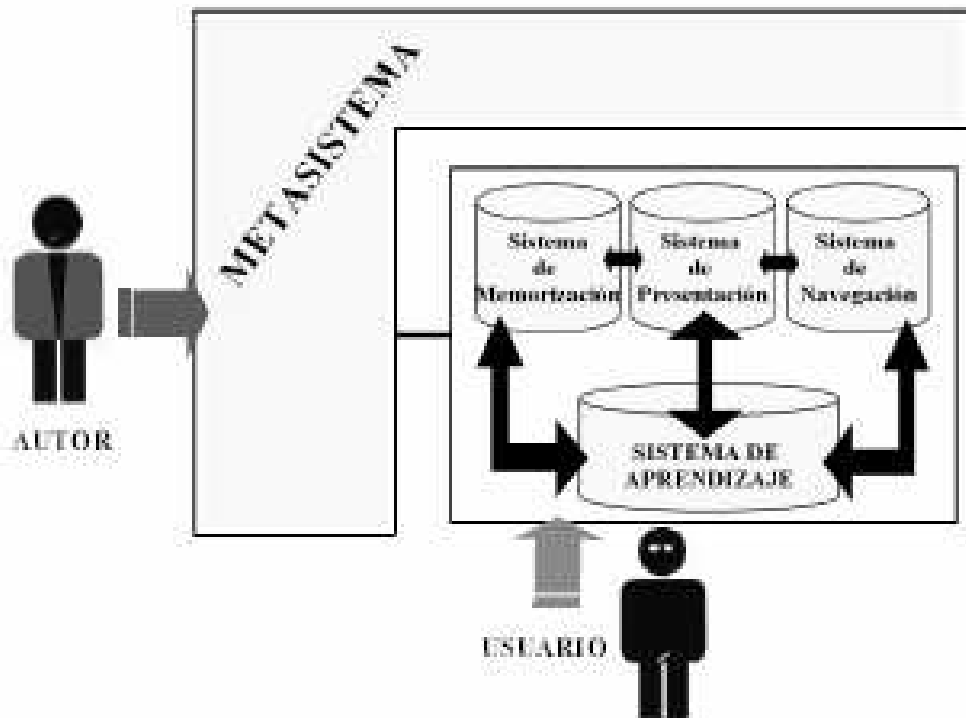
Desde un punto de vista estructural, los SHA están formados por los siguientes cuatro elementos:

- **Dispositivos de interacción con el usuario para la recogida de información explícita e implícita.** En el primer caso, son frecuentes los formularios, menús de opciones, botones, *banners* y barras de selección. En el segundo, recurre a *scripts* que monitorizan el seguimiento de la actividad en el portal: Ruteadores, cronogramas, detectores de “clickeo”, seguimiento de vínculos internos y externos y detectores de actividad en *warm zones*.
- **Procesos de filtrado y análisis de variables.** Se trata de complejos algoritmos que tienen como finalidad el análisis de las características y preferencias del usuario. El dispositivo puede construir una tabla de perfiles para la ulterior asignación de variables a procesos, objetos y contenidos. Algunas de las variables harán referencia al dispositivo empleado (Ordenador, teléfono *wap*, PDA...), otras al tipo de acceso (modalidad y velocidad de conexión) y un tercer grupo a las características específicas del usuario.
- **Motor de decisión.** Constituye el elemento clave del sistema. Analiza las variables de entorno y los perfiles de usuario para decidir qué tipos de contenido y en qué formato se mostrarán ante una determinada petición de acceso. Existen múltiples arquitecturas, desde un sistema experto construido en torno a una base de conocimientos más o menos flexible, hasta un motor inferencial basado en lógica difusa, algoritmos genéticos o redes

neuronales.

- **Gestor de contenidos.** Su función es construir y entregar los contenidos al usuario final. Por ejemplo, en el caso de páginas *Web* dinámicas, estos serán elaborados a partir de multitud de elementos atómicos almacenados en bases de datos. Si se trata de un sitio estático -sistema ya en declive- el gestor procederá a elegir una entre las diferentes versiones de páginas almacenadas.

Gráficamente podríamos entender un sistema adaptativo genérico como sigue:



Aquí se ilustra cómo un usuario, al interactuar con la aplicación, en realidad actúa sobre varias partes de la misma. Por un lado, el sistema memoriza las acciones realizadas por el usuario, el tiempo que emplea con cada una, etc. Éste sería el Sistema de Memorización. Por otro lado, esto lo consigue porque se le ha mostrado la información a través de una interfaz, cuya configuración contiene el Sistema de Presentación.

Generación de menús adaptativos

Finalmente, el orden de ejecución de cada elemento de la interfaz queda almacenado en el Sistema de Navegación.

Al ser adaptativo, estos sistemas tienen que comunicarse entre ellos facilitándose información a través de un motor de inferencia.

6. Objetivo del proyecto

El objetivo de este proyecto es realizar un SHA genérico que sea capaz de modificar la apariencia (formularios, menús, etc.) de varios tipos de aplicaciones, tanto aplicaciones Web como aplicaciones de oficina.

Entre los distintos tipos de SHA descritos anteriormente hemos optado por centrarnos en los **Hipermedias dinámicos** que son los que presentan las características más adecuadas para la realización de este proyecto.

Como este tipo de SHA necesita de una base de datos, esta es la prioridad del proyecto, es decir, realizar un diseño completo y eficaz de una base de datos para conseguir que esta pueda ser aplicable a cualquier tipo de aplicación informática. Los componente del SHA que vamos a investigar e intentar desarrollar son **los procesos de filtrado y análisis de la información y el motor de decisión** que estan íntimamente relacionados con la base de datos y por tanto dependen de esta.

Fuera del alcance del proyecto quedan los dos elementos de los SHA que no gestionan datos de usuarios, estos son el **Gestor de contenido y los dispositivos de interacción con el usuario**. Para realización de este proyecto suponemos que estos dos componentes existen y son funcionales.

7. Elementos que intervienen en la aplicación

7.1. Usuarios

La información necesaria sobre los usuarios dependerá en gran medida de la aplicación, ya que puede haber aplicaciones que requieran autenticación y otras no.

En el caso de que no requieran una autenticación bastará con recoger la información del usuario que tenga el sistema operativo sobre las distintas sesiones que se puedan abrir en él. Si fuera necesario se podrían pedir algunos datos básicos a la hora de iniciar nuestra aplicación.

Si fuera una aplicación que requiera autenticación se utilizará el perfil que esta aplicación tenga sobre el usuario y todos los datos asociados a éste. Dependiendo de cómo y dónde estén almacenados estos datos se podrían recoger y clasificar automáticamente sin necesidad de que se introdujeran manualmente. Esto puede resultar extremadamente complicado debido a la diversidad en los diseños de las bases de datos utilizadas por las distintas empresas o aplicaciones, por lo que lo más razonable es que se cree una interfaz gráfica para introducir los datos a mano.

Es indispensable disponer de información sobre el usuario que está usando el sistema ya que sin esta información resulta totalmente imposible realizar cualquier tipo de cambio en las aplicaciones.

De cuantos más datos se disponga de los usuarios mejor se podrán clasificar en perfiles y se podrá diferenciar con menos tiempo de estudio de un usuario concreto las actividades que éste podría llevar a cabo. Los datos que se pueden almacenar son de tipos muy variados, desde datos personales (DNI, dirección, número de hijos, estado civil, edad...) hasta

datos académicos (titulación, media...) o médicos (minuvalías...), ya que cualquiera que sea la diferencia que pudiera haber puede ser suficiente para que el motor de inferencia clasifique antes al usuario, y por tanto el trabajo de esta sección del programa sea menos costosa. Además se podrían tener en cuenta notas, comentarios o evaluaciones que se realicen dentro de la empresa para saber la evolución o rendimiento de los empleados; incluso se podría tener en cuenta el nivel en el manejo de las distintas aplicaciones (puede que se hayan realizado cursos de formación para el correcto manejo de una determinada aplicación).

7.2. Aplicaciones

Las aplicaciones a las que va dirigidas este proyecto son de diverso tipo, desde aplicaciones de ofimática hasta aplicaciones específicas de gestión o de ingeniería, tanto para consultar datos como para insertarlos o modificarlos. Suelen ser aplicaciones que su explotación se realiza mediante menús (formularios y reports) y acceso con múltiples pestañas.

Para realizar correctamente la modificación de la interfaz gráfica de una determinada aplicación se necesita información muy precisa acerca de dicha interfaz. Por un lado se necesita saber todas las opciones (botones, menús...) que incluye, y en el caso de que una dé acceso a otra opción no disponible anteriormente se debe guardar esa relación entre opciones. Por otro lado, se deben conocer todos los formularios que se muestran en la aplicación y el tipo de información que muestran, ya que no es lo mismo presentar un fichero de texto que presentar una serie de gráficas.

Toda esta información combinada junto con el perfil del usuario puede permitir crear una interfaz adecuada al uso que se realiza de la aplicación en un momento determinado.

Hay que tener en cuenta que no todas las aplicaciones informáticas pueden hacer uso de este sistema, como por ejemplo los videojuegos o los reproductores de vídeo o música.

7.3. Nivel de acceso

Muchas de las aplicaciones necesitan autenticación del usuario para poder ser utilizadas, pues usan este conocimiento en su filtro de opciones, para que, por ejemplo, dependiendo del usuario que se haya conectado, se tenga acceso a unas opciones mientras que otras estén restringidas.

Esto puede deberse a diversos motivos, uno de los más frecuentes es el rango en la empresa o puesto de responsabilidad. Un usuario puede disponer de una mayor o menor responsabilidad dentro de una organización jerárquica. Esto implica que podrá tener acceso a información más restringida que un usuario que, por ejemplo, esté a su cargo.

Esta información es muy crítica ya que sin ella se podría poner a disposición de cualquier usuario una información restringida, lo que podría causar grandes perjuicios para la empresa u organización. Por esta razón será necesario tener la información sobre niveles de acceso, organización jerárquica (interna a cada departamento y entre departamentos) o cualquier método que se pueda utilizar para realizar esta diferenciación.

En nuestro caso nos vamos a centrar en la estructura jerárquica dentro de las empresas ya que es una aplicación orientada al mundo

empresarial, en el cual lo más frecuente es tener una estructura bien definida.

7.4. Filtrado

Otra forma de filtrar la información a la que un usuario se le otorga permiso, totalmente compatible con el filtrado por nivel de acceso, puede ser por la organización estructural (departamentos, sectores, áreas...) que pueda tener una empresa.

Se puede dar el caso de que una misma aplicación sea usada por distintos departamentos dentro de una empresa, y las opciones que se habiliten en cada caso sean distintas. Por esta razón es necesario conocer qué aplicaciones son usadas por los distintos departamentos y qué opciones se habilitan de cada aplicación para cada departamento. Aunque esta información ya esté recogida en la aplicación usada por los usuarios, nosotros también necesitamos conocerla (mediante nuestra base de datos o mediante comunicación con la primera aplicación) para evitar mostrar o habilitar opciones a quien no se debe.

Para el desarrollo del proyecto consideraremos que la mejor opción es tener esta información en nuestra base de datos para así poder hacer la aplicación lo más genérica posible. Además supondremos una estructura en departamentos, que es la más común en el mundo empresarial.

7.5. Presentación de la información

Este punto quizás sea el más complicado de todos, ya que la información tiene muchos formatos, desde texto plano hasta formularios con tablas, gráficas, etc.

Generación de menús adaptativos

Es necesario conocer qué tipo de información y qué tipo de formato se presenta en cada formulario de la aplicación para poder modificarlo correctamente sin producir una pérdida de información que pueda afectar al usuario de ésta o al desarrollo de su trabajo. También resultará necesario saber si el formulario permite introducción de datos o modificación de información, ya que éstos pueden disponer de ciertas opciones que no podrán ser ocultadas (guardar, restaurar...).

Otra opción que puede ser útil es saber que información muestra una aplicación (teniendo en cuenta todos los formularios que forman parte de ésta). La elección entre las dos formas de almacenar esta información se realizará cuando se diseñe la base de datos y después de haber realizado un estudio de posibles casos de usos y aplicaciones hipotéticas.

8. Ejemplo de uso

Con el objetivo de identificar los elementos más relevantes que debemos tener en cuenta a la hora de almacenar información y, por tanto, de realizar el diseño de la base de datos, se ha realizado un ejemplo en el que se ilustra el uso y el resultado de aplicar la generación dinámica de menús o lo que nos gustaría conseguir con ello:

Una empresa suele estar dividida en diferentes departamentos, como Recursos Humanos, Comercial, Contabilidad, Producción y otros. A su vez cada departamento suele estar organizado de forma jerárquica encabezada por un jefe, los jefes de subdepartamentos y los empleados.

Si la empresa dispusiera de una aplicación informática que gestionara todo el funcionamiento interno, que tuviera acceso a toda la información y que todos los empleados la usaran habitualmente, facilitaría mucho la gestión de la misma. No obstante, es necesario algún tipo de precaución, porque si todos tuvieran acceso a toda la información y pudieran consultarla y en el peor de los casos modificarla sería un desastre.

Esta aplicación, gracias a nuestro sistema, filtrará la información dependiendo de qué usuario esté conectado. Si se conectara el jefe del departamento comercial puede que le muestre toda la información sobre el departamento ya que es el máximo responsable. Pero como en realidad él no usa toda la información sino que se dedica a comprobar que sus empleados cumplen con las tareas que se le han encomendado y a planificar las nuevas tareas según sea necesario, la aplicación le presentará los formularios correspondientes en el orden más frecuente en el que el jefe lo haya visto. Por otro lado el jefe debe pasar un informe a sus superiores por lo que la aplicación le mostrará un botón para ir al

Generación de menús adaptativos

formulario correspondiente, ya que esta opción no la usa tan a menudo como para redirigirle directamente.

Uno de los empleados de este departamento es el comercial, el cual tiene una agenda que, al conectarse a la aplicación, es lo primero que le aparece porque siempre es lo primero que visualiza. Cada vez que vuelve de tratar con un cliente rellena un parte de trabajo por lo que, después de ver la agenda, siempre le presenta un botón para ir al formulario correspondiente. Estas dos acciones las realiza cada vez que inicia la aplicación. Después puede o no presentar algún informe extra, por lo que tendrá los botones correspondientes para realizar estas acciones (presentar informes de gastos, informes de oportunidad de negocio, nuevos contratos...). Al igual que su jefe tiene otro botón para presentar su informe de trabajo por lo que la opción también la tendrá aunque le redirigirá a su correspondiente formulario.

Los administrativos del área comercial se tienen que ocupar de registrar nuevas empresas con las que han llegado a algún tipo de acuerdo y se encargan de hacer un informe a los responsables del departamento de ventas para que tengan los productos necesarios para el negocio. También realizan los informes de los gastos de los comerciales y aceptan o rechazan las facturas que éstos les presentan, por si intentan timar a la empresa con algún gasto personal. Por lo tanto, cuando se conecta lo primero que va a hacer es comprobar si hay informes nuevos de los comerciales. Después de esto, realiza sus informes correspondientes. La aplicación a estos empleados les mostrará un formulario con todos los informes que no haya procesado, al final le pedirá al empleado que elija uno y la opción que desea realizar sobre ese informe.

El jefe de la empresa se encarga de supervisar los informes de los jefes de departamento por lo que la aplicación le permitirá tener acceso a cada uno de ellos. Además estos informes se le muestran en el orden en el que suele visitarlos, pues ello le permite organizar mejor la visión general del estado de la empresa.

El caso del departamento de recursos humanos es algo más complejo ya que es un departamento bastante más grande que el departamento descrito anteriormente. Dos personas dirigen el departamento, uno de estos jefes tiene mucha experiencia y lleva años en la empresa pero aun así es muy preciso y lee muy detenidamente cada informe que le llega, ya que sabe de la importancia de realizar bien su trabajo y, si es posible, que sea a la primera pasada. Su compañero es más joven y dinámico y le gusta poco estar sentado delante del ordenador, además hace relativamente poco que ha salido de la carrera, por lo que sólo echa un vistazo rápido a los informes mirando simplemente los gráficos o estadísticas que aparecen. La aplicación en el caso del primer jefe mostrará todo el contenido de los informes mientras que para el segundo mostrará las gráficas con las leyendas.

En el departamento también trabajan dos personas que se encargan de hacer el recuento de las horas que los trabajadores han contabilizado en el informe y hacer los cheques correspondientes o informar al jefe de que hay alguna anomalía. Una de estas personas es un poco patosa y se suele equivocar al elegir las opciones y se suele meter en sitios que no le interesan, la aplicación en este caso lo primero que le mostrará será una lista de las opciones en las que se ha detenido más tiempo para evitar que tenga que buscar en los menús y se vuelva a equivocar y perder tiempo. Su compañero sí sabe donde están las cosas que necesita y es muy eficiente, en vez de usar solo una ventana de la aplicación suele usar varias simultáneamente para mirar información en una y rellenar en otra,

por lo que la aplicación acabará por mostrarle una única ventana dividida en dos, una para realizar las búsquedas o consultas que necesite y la otra para introducir información en el formulario que corresponda.

Estos dos últimos empleados tienen a su cargo a un becario al que le tienen que enseñar el funcionamiento del departamento y en concreto su trabajo, él también tiene acceso a la aplicación pero solo le muestra las opciones a las que le permiten acceder sus responsables. Dentro de estas opciones solo puede consultar información y no modificar, por lo que la aplicación le mostrará solo las opciones a las que tiene acceso y en el orden en que suele acceder.

8.1. Análisis de la información utilizada

Del ejemplo se pueden sacar una lista de elementos relevantes que influyen en el funcionamiento de la aplicación:

- Departamentos de la empresa.
- Nivel jerárquico en los departamentos o en la empresa.
- Características laborales, sociales, personales y académicas del usuario.
- Características de la empresa: sector de actividad, número de empleados, facturación, tiempo en el sector.
- Formato de la presentación de la información (gráfica, formulario, reportes, tipo, formato...).
- Qué información accede cada usuario.
- Qué modo y nivel de acceso tiene a la información cada usuario (lectura, escritura,..).
- Aplicaciones que estudia nuestro programa: científicas, administrativas, legales, ingeniería,..
- Opciones elegidas por los usuarios.

- Secuencia de presentación de la información.

Todos los elementos observados aparecen ya en los puntos anteriores excepto el último. Con este último elemento, conseguiremos saber los pasos que ha seguido un usuario durante el manejo de la aplicación, facilitando la modificación de los distintos elementos a mostrar con algo de antelación. Por otro lado conseguimos “guiar” al usuario presentándole sólo la información y los formularios que más usa en el orden en el que los utiliza.

9. La base de datos

Evidentemente, para poder gestionar toda esta información es preciso que se utilice una **base de datos**. Para construirla debemos identificar las entidades, sus atributos y diseñar el diagrama E-R a partir de nuestro ejemplo, tras analizar el problema en particular.

9.1. Comprobación de las entidades

Como en el ejemplo que se ha desarrollado hay dos departamentos, éstos han de figurar en la entidad. Por otro lado, como la empresa sólo tiene una aplicación para gestionar todo, la entidad Aplicaciones contendrá información acerca de esta aplicación.

En la entidad Nivel jerárquico tiene que estar reflejada la estructura de cada departamento. En el caso del departamento comercial tiene que haber un jefe, un comercial y un administrativo. Para el caso del de recursos humanos se tiene que tener constancia de que hay dos jefes, dos empleados y además tiene que estar reflejado que estos dos empleados son responsables del becario. Por otro lado tiene que estar presente la información necesaria que permita saber quién es el responsable de los distintos empleados.

En la entidad Empresa se guardará toda la información sobre la empresa que se considere necesaria. En este caso se podría guardar que es una empresa consolidada en el mercado que se encarga de distribuir vinos de marcas conocidas. Tiene bastantes clientes, un volumen de venta importante y varias sedes (delegaciones).

El jefe más experimentado en la entidad Características personales tendremos sus datos (nombre, DNI, número de cuenta...), su estado civil (casado, número de hijos...), sus años en la empresa, su rendimiento profesional y su formación académica (supongamos que tiene una

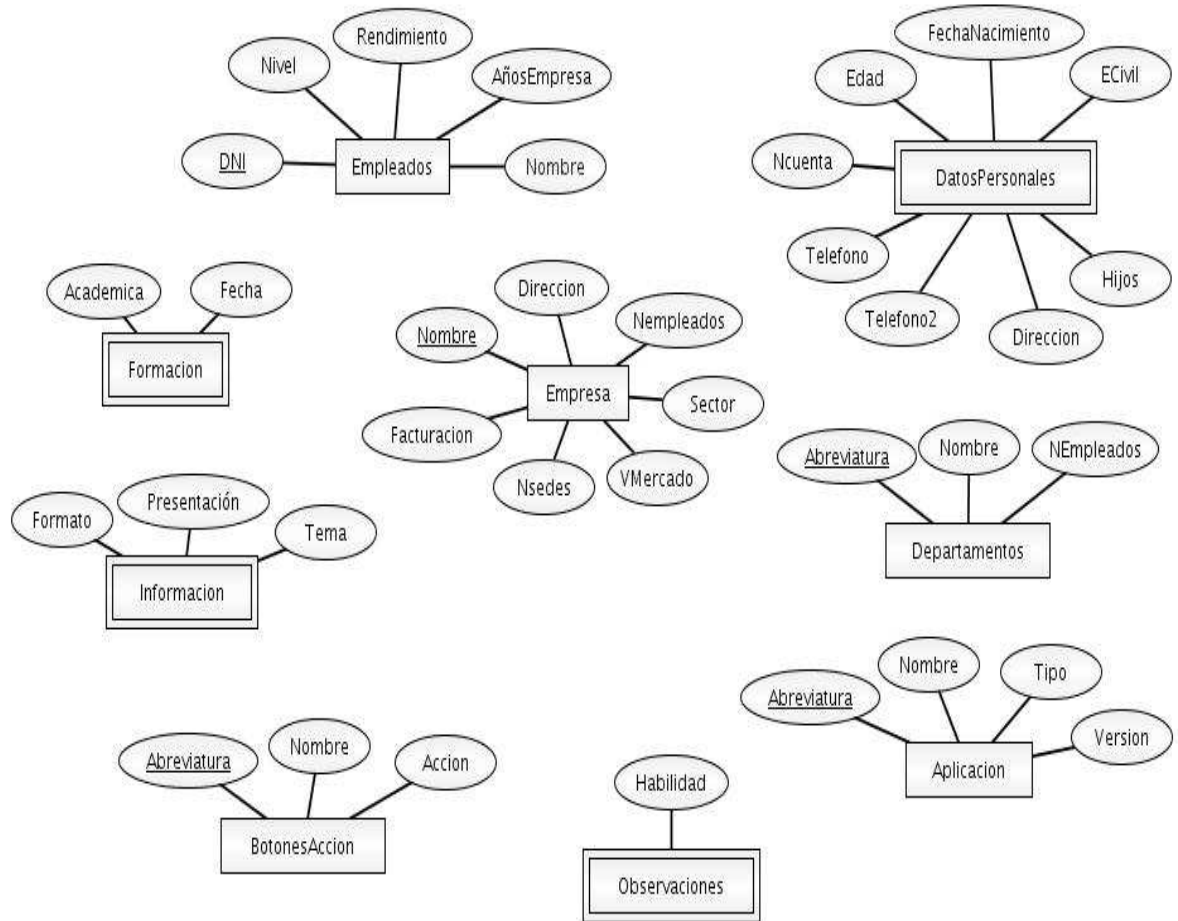
carrera y un par de masteres). Este jefe accede a los informes que sus empleados le dirigen de forma minuciosa; en una entidad tenemos que tener presente la información sobre los informes exactos a los que accede (que son los de sus empleados) y en otra que visualiza toda la información de dicho informes. En cuanto a las opciones que elige y el orden en que lo hace en este caso no está definido ya que no suele seguir ningún patrón concreto.

El jefe más joven, aparte de la información personal y social al igual que su compañero, añade que es una persona dinámica y rápida. En este caso el tipo de información que visualiza suele ser gráficos resumen por lo que esto se guardará en la correspondiente entidad y al igual que antes sólo accede a los informes de los empleados bajo su jerarquía mando. En este caso sí sigue un orden de visualización concreto y es que mira los informes por orden alfabético. Los dos jefes tienen acceso a la información de todo el departamento y en ambos modos (escritura y lectura).

Para el empleado patoso las opciones que se guardan sólo son aquellas en las que permanece más tiempo, y el orden en el que las elige se intenta guardar según el mismo criterio. En cambio, para su compañero, que se sabe que es eficaz, se tiene una gran cantidad de opciones y el orden en que realiza las tareas también está muy definido. Estos empleados tienen acceso a la información que generan y además pueden modificar los permisos de su becario.

El becario solo tiene acceso a lo que le permiten sus jefes por lo que su acceso está muy restringido, por otro lado su condición de becario inexperto figurará en sus datos laborales. Las opciones que elige son escasas ya que trabaja poco, y el orden en que examina los documentos es preciso ya que lo único que hace es pulsar a siguiente.

Un posible esquema de las entidades con sus atributos podría ser el que sigue:



9.1.1. Explicación de la figura

Al realizar el análisis las entidades que se dedujeron del ejemplo han resultado las siguientes:

- **Empleados**: esta entidad guarda toda la información sobre los distintos empleados de la empresa. Está relacionada con una entidad *Formación* para reflejar el nivel de formación del empleado que contendrá varios atributos derivados. Lo importante es almacenar datos básicos como el *Nombre y Apellidos*, *DNI*, *Nivel de formación*, *Rendimiento*, *Años en la empresa*. Estos dos

últimos hacen referencia a la experiencia del empleado y al rendimiento personal en la realización de su trabajo que demuestra en la utilización de las aplicaciones.

- Formación: agrupa la información académica y la evaluación de su expediente.
- Datos personales: recoge datos del usuario personales. Está formado por el *Edad, Fecha nacimiento, E. Civil, N° cuenta, Dirección, Hijos, Teléfono 1 y Teléfono 2.*
- Observaciones: aquí se incluye cualquier información relativa al empleado que pueda ser importante registrar, como sus destrezas o su facilidad para desarrollar ciertas tareas.
- Empresa: información básica sobre la empresa, como el Nombre, Sector, Dirección, Número de Empleados, Número de Sedes, Facturación y Volumen de Mercado.
- Información: en esta se guardan datos sobre la información que se muestra en la aplicación correspondiente, nos permitirá saber ciertos aspectos para indicar cómo presentar la información en pantalla del modo más adecuado posible. Esta entidad no tiene clave primaria por lo que será el número de fila por ejemplo el que desempeñe esta función. En esta entidad guardaremos *Formato, Presentación y Tipo*. El formato distingue si es un gráfico, texto plano, etc. La presentación indica si se mostrará como un formulario, una imagen, un documento... El tipo indica lo que va a ser representado, por ejemplo, si es un balance, un nombre...
- Aplicación: información sobre las distintas aplicaciones de las que dispone la empresa, de esta forma se podría tener una aplicación distinta por departamento y el estudio de las distintas aplicaciones

no implicaría tener distintas bases de datos. Aquí interesa tener el *Nombre, Tipo y Versión*.

- Botones de Acción: es ésta se guardan las distintas opciones de las distintas aplicaciones que los usuarios van utilizando, junto con la acción que realiza. Registra el *Nombre* y la *Acción*.
- Departamentos: aquí se guardará la información de los distintos departamentos existentes en la empresa. Guardaremos el *Nombre* y *Número de empleados*.

Las claves primarias de las entidades *Aplicación, Departamentos* y *Opciones* son abreviaturas de sus nombres siguiendo algún convenio, como por ejemplo usar las letras de inicio de cada palabra del nombre completo.

9.1.2. Comprobación de las relaciones

Para asociar estas entidades surgen un conjunto de relaciones que las interconectan. A continuación se detalla cada una de ellas:

- Para representar el nivel jerárquico de la empresa se ha usado la relación *Supervisado por*, en esta se presupone que un empleado solo tiene un supervisor o un superior y que varios empleados puede ser dirigidos por un único jefe.
- La relación *Accede a* representa la información a la que cada empleado accede. Como se ve en el esquema esta relación tiene unos atributos que indican qué tipo de acceso tienen los empleados a la información que acceden mostrada por la aplicación ya que estos pueden no tener permisos de escritura sobre cierta información.

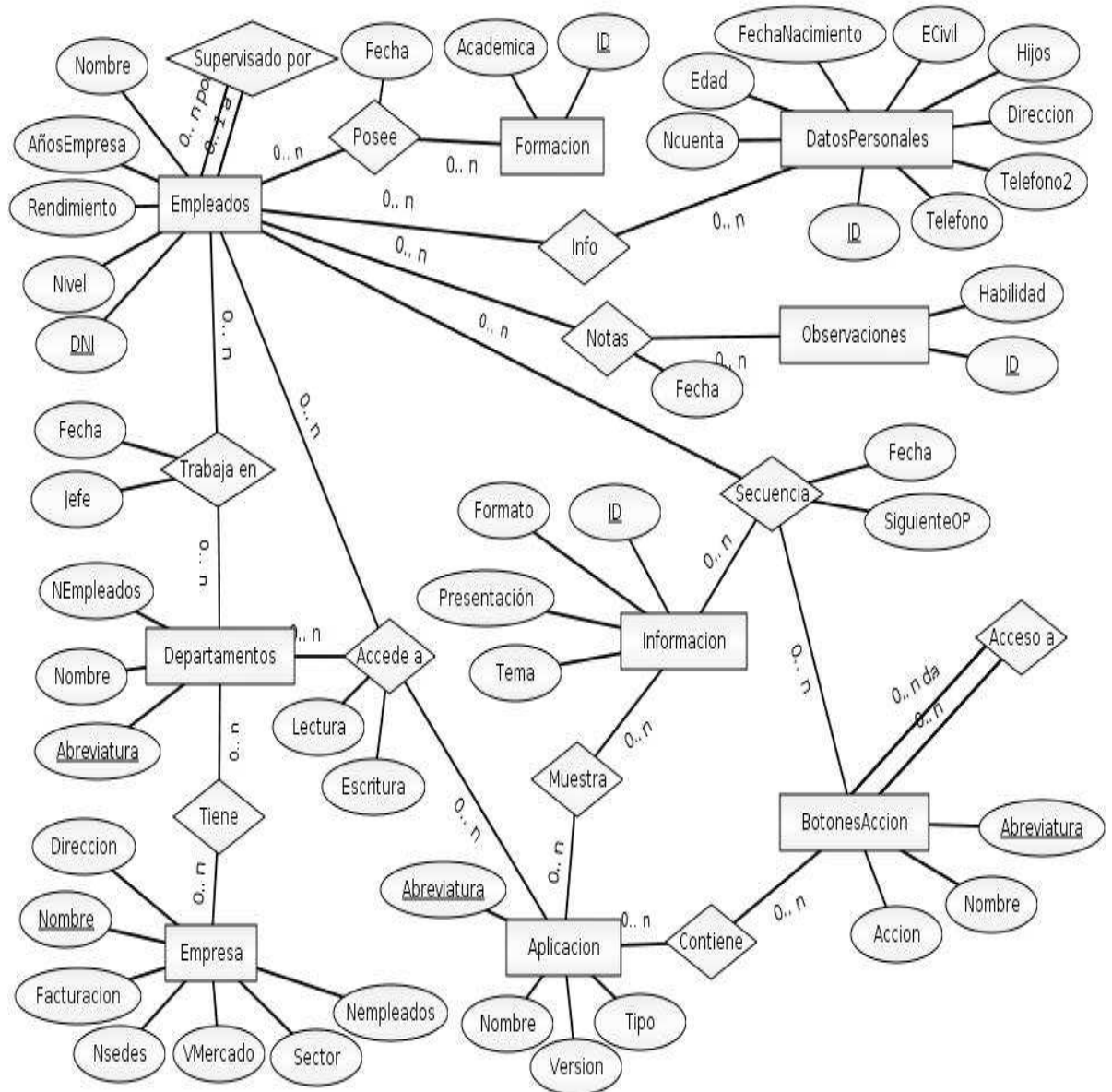
Generación de menús adaptativos

- Para saber los empleados de cada departamento se dispone de la relación *Trabaja en*. En esta relación todos los componentes de la entidad *Empleados* tienen que estar presentes en la relación y sólo pueden ser parte de un departamento. Además registra la *fecha* pues se necesita saber el periodo en el que un empleado ha trabajado con algún jefe y/o departamento, por si cambiase de puesto.
- En la relación *Tiene* participan las entidades *Empresa* y *Departamentos*. Con esta relación se completa la información sobre la empresa y se sabe si los distintos departamentos tienen aplicaciones distintas para realizar sus actividades a través de la relación *Accede a*.
- Con la idea de saber las opciones de las que disponen las aplicaciones se tiene la relación *Contiene*.
- *Acceso a* es una relación binaria que nos permite saber qué opciones dan acceso a otras opciones.
- Para guardar la secuencia de acciones que lleva a cabo un usuario hemos incluido la relación *Secuencia* en la que intervienen las entidades *Empleados*, *Información* y *Opciones*. Por otro lado tiene un atributo *Siguiente Opción* que nos indica la opción elegida a continuación. Además contiene un campo *fecha* para saber cuándo ha realizado cada acción para mantener el historial de acciones ordenado cronológicamente y para poder calcular el tiempo que el usuario emplea con cada una.
- *Notas*: relaciona al empleado con la entidad *Observaciones*. Tiene en cuenta la fecha de la observación.
- *Info*: permite acceder a los datos personales del empleado.

Generación de menús adaptativos

- *Posee*: guarda la formación académica del empleado y la fecha de cada evento.

El diagrama E-R resultante queda como sigue:



Una vez mostrado el modelo E-R se puede generar un listado de las tablas que compondrán nuestra base de datos:

- **DatosPersonales** (ID, Edad, FechaNacimiento, ECivil, Ncuenta, Telefono, Telefono2, Direccion, Hijos)

Generación de menús adaptativos

- Empresa (Nombre, Direccion, Nempleados, Nsedes, VMercado, Facturacion, Sector)
- Aplicacion (Abreviatura, Nombre, Tipo, Version)
- Departamentos (Abreviatura, Nombre, NEmpleados)
- Observaciones (ID, Habilidad)
- BotonesAccion (Abreviatura, Accion, Nombre)
- Informacion (ID, Formato, Presentación, Tema)
- Formacion (ID, Academica)
- Empleados (DNI, Nombre, Nivel, Rendimiento, AñosEmpresa)
- Acceso_a (Abreviatura, _da_Abreviatura)
- Secuencia (DNI, ID, Abreviatura, SiguienteOP, Fecha)
- Contiene(Abreviatura_Aplicacion, abreviatura_BotonesAccion)
- Muestra (Abreviatura, ID)
- Accede_a (DNI, Abreviatura Departamentos, Abreviatura Aplicacion, Lectura, Escritura)
- Notas (DNI, ID, Fecha)
- Info (DNI, ID)
- Posee (DNI, ID, Fecha)
- Tiene (Abreviatura, Nombre)
- Trabaja_en (DNI, Abreviatura, Fecha, Jefe)
- Supervisado_por (_por DNI, _a DNI)

Generación de menús adaptativos

Con la ayuda de la aplicación DBCASE, proyecto de la UCM que permite generar scripts SQL a partir del esquema E-R, hemos generado el anexo 1, que representa la creación de la base de datos diseñada en este apartado.

10. Motor de Inferencia

Para averiguar qué acciones va a realizar un usuario conectado al sistema es necesario un motor de inferencia que recoja información de la base de datos, la procese y pregunte al usuario si desea realizar la acción que se haya considerado más probable.

Para conseguir esto necesitará información de casi toda la base de datos por lo que su análisis de la situación se realizará en varios pasos. En primer lugar deberá recabar toda la información sobre el usuario que está interactuando con el sistema. Esta información la obtendrá de la entidad *Empleados* y de las relaciones *Tiene*, *Info* y *Notas*. Una vez evaluado el empleado y conociendo sus características se podrá recabar la información sobre la aplicación que está usando y el departamento donde trabaja. Con esta información podemos acceder exactamente a la información más relevante que es la contenida en la relación *Secuencia*. Esta última consulta a la base de datos le proporcionará todas las trazas anteriores por lo que podrá averiguar el paso actual y ver cuál de las posibles trazas es más probable que se repita. Con todo esto debe ser capaz de predecir con un cierto grado de certeza la siguiente acción que deseará realizar el usuario.

10.1. Ejemplo de funcionamiento del motor de inferencia

Para describir el funcionamiento del motor de inferencia de una forma más detallada vamos a suponer que en la base de datos tenemos la información de una empresa que consta de dos departamentos.

Con la información sobre los empleados (sus características personales, departamento, aplicación que usa...) el motor de inferencia realiza un estudio sobre qué datos tienen relevancia a la hora de realizar acciones sobre los programas y, por tanto, influir en el desarrollo del

trabajo de un empleado. Este estudio lo realiza con la ayuda de la información de las secuencias que realizan los distintos empleados. El resultado de este estudio es un árbol de decisión por el cual se guiará el acceso a la información estrictamente necesaria para modificar de manera correcta la apariencia de la aplicación usada por los empleados.

Imaginemos que los departamentos usan una misma aplicación pero usan partes de ella totalmente disjuntas. De esta situación se podría sacar la raíz del árbol de decisión ya que es un factor muy influyente en el resto de la información. A continuación, el motor de inferencia pasará a estudiar primero un departamento con el objetivo de realizar el subárbol correspondiente con toda la información discriminatoria de forma independiente.

En el estudio del primer departamento se ve con claridad que el estado civil o el número de hijos no es un factor que influya en el desarrollo del trabajo del personal. En cambio, hay un salto generacional entre los empleados, es decir, el departamento está formado por un grupo de gente joven y dinámica, y por otro grupo con experiencia y que no tiene tanta destreza con las nuevas tecnologías. Estos datos sí influyen en la forma de trabajar de los empleados ya que los jóvenes prefieren la información de forma gráfica y con un breve resumen, y por el contrario la gente mayor prefiere la información de la forma en la que la ha visto durante la mayor parte de su vida profesional, texto plano con algún que otro gráfico sencillo. Por esto podemos deducir que el siguiente nodo del árbol será la edad.

En la rama de los jóvenes se descarta la información acerca de la jerarquía, ya que todos tienen casi el mismo rango, pero sí se tiene en cuenta el rendimiento demostrado, ya que unos son capaces de ver más información que otros. En el caso de los experimentados sí que se tiene

en cuenta el rango en la empresa ya que no todos pueden ser jefes y esto últimos acceden a información más general que los que tienen un trabajo más concreto.

Por último, para este departamento, vamos a suponer que cada empleado independientemente de su puesto o edad elige, al iniciar la aplicación, una opción concreta. Puede ser que algunos empleados coincidan en alguna acción pero se distinguirían por otra característica anterior. Así, todas las ramas que llevamos hasta ahora en este departamento se terminarán con la opción elegida, de esta forma podemos saber qué patrón de consulta de información será el más probable que realicen.

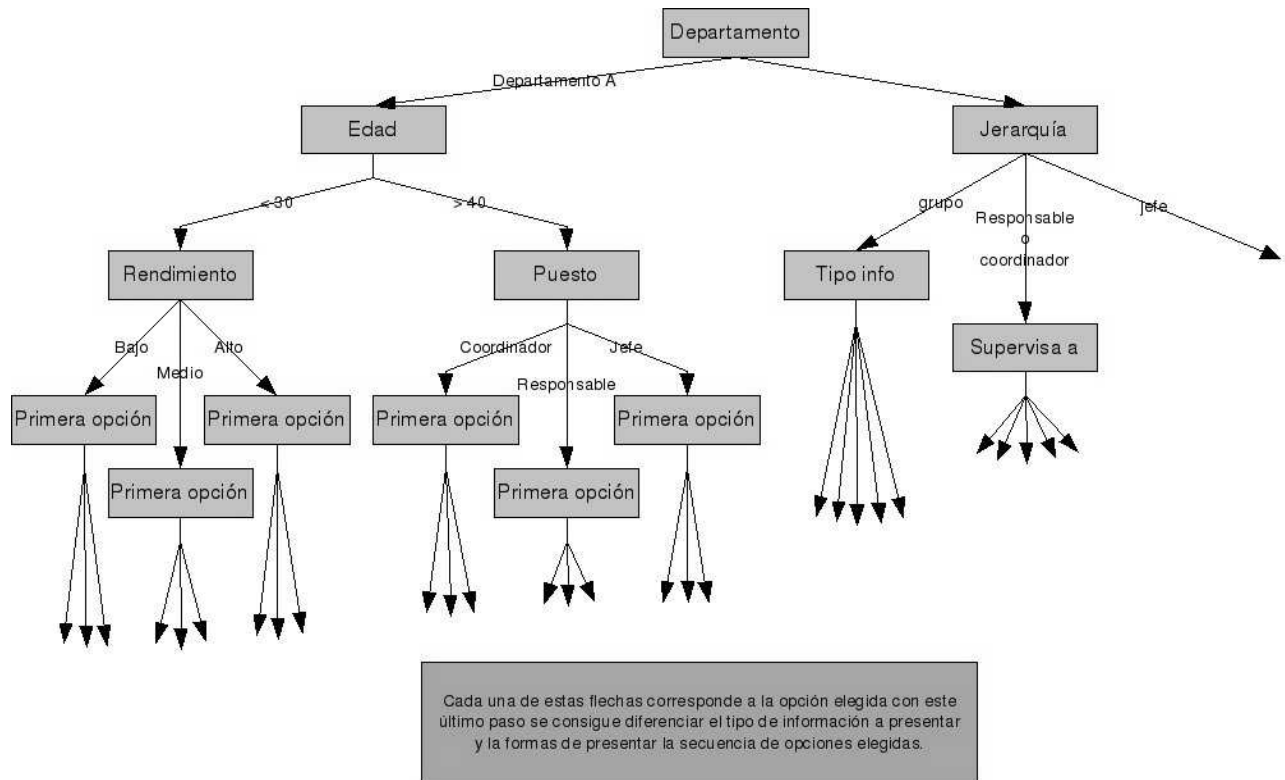
El otro departamento está distribuido en pequeños grupos de trabajo, cada uno con un responsable y, por encima de estos responsables, hay un coordinador que se encarga de organizar a varios grupos y así hasta el jefe del departamento. Esto significa que hay más rangos jerárquicos y esto inevitablemente influye de una manera muy significativa en el trabajo de cada uno, por lo que esta información será el primer nodo de este subárbol.

Con esta información el motor ya tiene suficiente información para distinguir casi con exactitud qué recorrido de opciones seguirá el usuario, lo único que necesita es saber a qué tipo de información accede al entrar.

En la rama de responsables y coordinadores, todos empiezan por la misma opción ya que ven los informes que han realizado sus empleados y una vez leídos crean un nuevo informe de conclusiones para presentar a su superior. Por lo que debajo de estas ramas se tiene que saber a quién supervisa, con el objetivo de presentarle los informes generados por sus subordinados.

Generación de menús adaptativos

Con todo esto el árbol generado sería algo parecido a:



Cuando un usuario se registra y comienza a usar la aplicación, se recorre este árbol en función de los datos del usuario registrado, y se obtiene la información necesaria para poder modificar correctamente el comportamiento de la aplicación para que se ajuste a lo que suele hacer. Además podrá acceder, de forma más eficiente, a la secuencia de opciones que dicho usuario realiza; incluso puede cambiar el recorrido si el usuario pasa a hacer otra de sus secuencias.

La inferencia consta de tres módulos: uno que guarda la secuencia de traza que ha sido utilizada por los diferentes perfiles de usuarios; otro que crea las reglas discriminatorias con los valores introducidos, analizando su significancia y que también son guardados (consistirá en un programa estadístico de árboles de decisión o un Sistema experto) y el tercero que se utiliza para guiar a los usuarios en

Generación de menús adaptativos

la explotación de la aplicación cuando el segundo módulo ha procesado los datos del primero, y construido las reglas de comportamiento.

11. Prototipo.

11.1. Interfaz gráfica con la base de datos

Con la idea de probar el correcto funcionamiento de la base de datos y de facilitar una herramienta para la inicialización de ésta, se ha desarrollado una aplicación sencilla en Java que consiste en una serie de formularios en los que se le va pidiendo al usuario la información necesaria.

Lo primero que pide la aplicación son las credenciales del administrador de la base de datos. Una vez introducidos los datos correctos aparece el menú principal.

Esta dividida en varias secciones, donde cada una pide toda la información necesaria para rellenar las tablas relacionadas con dicha sección. Por ejemplo, al insertar un empleado no sólo pedirá la información necesaria de la tabla Empleados sino que se pedirá toda la información de las tablas relacionadas con ésta. Además, esta aplicación permite visualizar y modificar el contenido de la base de datos, pero sólo de las secciones anteriores, ya que no tiene sentido modificar los datos introducidos por el manejo de la aplicación.

Las opciones de consultar y modificar no están implementadas, ya que esta herramienta la hemos considerado secundaria y nos hemos centrado en la segunda, en la que se prueba un prototipo funcional.

Con esta estructuración del programa, se puede decir que la base de datos se divide en las siguientes secciones:

- Empleados: esta sección esta compuesta por toda la información que se puede tener de un empleado. En la base de datos diseñada corresponde con las entidades *Empleados*, *Formación*, *Datos*

Personales y Observaciones y con las relaciones *Posee, Info* y *Fecha*.

Para insertar todos los datos necesarios se presentan 3 formularios, en el primero se piden los datos del empleado, a continuación se piden los datos personales. Si en algún momento se decide cancelar antes de pasar al tercer y último formulario, no se produce ninguna modificación en la base de datos ya que estos datos los hemos considerado básicos para el correcto funcionamiento de la aplicación. En el tercero se piden los datos de formación y observaciones, que son opcionales.

- Empresas: en este caso se piden todos los datos sobre la empresa en la que se está usando la aplicación con el objetivo de conocer su estructura interna. De esta sección forman parte las entidades *Departamentos* y *Empresa* y las relaciones *Supervisado por*, *Trabaja en* y *Tiene*.

En esta sección se tienen 3 formularios, en el primero se piden los datos de la empresa y en el segundo se piden los datos sobre los distintos departamentos. Como mínimo la empresa ha de tener un departamento para el correcto funcionamiento de la aplicación. En el caso de que se cancele sin insertar uno, se avisará de que se borrarán todos los datos insertados hasta el momento. Cada vez que se inserta un departamento se preguntará si se desea insertar más, en el caso de que se haya terminado y elijamos 'No', se pasará al último formulario en el que se indica qué trabajadores pertenecen a los distintos departamentos y el jefe de cada departamento.

- Aplicaciones: aquí se piden todos los datos sobre todas las aplicaciones que vaya a ser objeto de estudio. De esta sección forman parte las entidades *Aplicación*, *Botones de acción* e *Información* y las relaciones *Accede a*, *Muestra* y *Acceso a*.

Esta parte tampoco está implementada, pues con las dos anteriores nos ha resultado suficiente para comprobar que no hay defectos en la base de datos (ni en el script que hemos usado para generarla), que puedan producir errores internos o inesperados fuera de su uso normal.

- Inferencia: aquí se piden todos los datos sobre todas las aplicaciones que vaya a ser objeto de estudio.

11.2. Componentes del Prototipo

Para comprobar que la base de datos contiene la información necesaria y suficiente para conseguir los objetivos planteados, se ha desarrollado una aplicación en Java que consta de dos partes:

- Una **interfaz gráfica**, en la que se intenta representar una aplicación de gestión, que consta de un menú y una serie de formularios, los cuales contienen diversos tipos de información y sus propios menús.

Además, la interfaz hará las funciones de las partes de los SHA que han quedado fuera del alcance de este proyecto. Por un lado se encarga de recoger la información necesaria de las actividades realizadas por el usuario en los distintos formularios, y se encarga de pasar esta información a la interfaz con la base de datos. Por otro lado, también se encarga de modificar su propio aspecto, gracias a la información que recibe de las distintas consultas que realiza a la interfaz con la base de datos.

- Una **interfaz con la base de datos**, que será la encargada de realizar las actualizaciones y consultas necesarias para conseguir el objetivo final de modificar la interfaz gráfica. Contiene la información necesaria para manejar las distintas secuencias de acciones que se hayan podido encontrar, en función del usuario

que se haya registrado en la aplicación.

Para realizar las pruebas se ha generado un script SQL que aparece en el punto 2 del anexo que contiene la información básica necesaria para realizar las pruebas.

En este script se inserta lo necesario para realizar las pruebas. En muchos casos la información que sería necesaria para el motor de inferencia se ha omitido ya que esta parte de la aplicación queda fuera del alcance del proyecto.

Se insertan dos usuarios, uno con rol de empleado y otro como jefe dentro de un mismo departamento con el objetivo de poder mostrar diferencias en la interfaz gráfica según el usuario utilizado. Para poder completar correctamente la relación *Secuencia* se ha insertado en *Información* un registro que valdrá para todos los formularios. Por último, se han insertado en la entidad *BotonesAccion* todos los botones u opciones de menú que se pueden escoger. Con esta información se consigue modificar la interfaz gráfica ligeramente en cada formulario y seguir la secuencia correspondiente.

Al iniciar la aplicación se pide autenticación. En el caso de que ésta sea correcta, se inicializa la interfaz con la base de datos, que guardará la información necesaria sobre el usuario que está haciendo uso de la herramienta.

El primer efecto que se puede observar se da en el formulario principal de la aplicación, que presenta una serie de botones de los cuales algunos serán deshabilitados en el caso de que no se use al usuario Jefe. Esta información se deduce de la relación *Trabaja_en*, ya que el DNI del usuario coincide con el DNI del Jefe lo que indica que es el jefe del departamento.

Generación de menús adaptativos

Al iniciar la interfaz con la base de datos también se guardan por fechas todas las secuencias que se han realizado anteriormente. Si no se han encontrado, todos los formularios se presentarán en su formato original, con todas sus opciones habilitadas, y cada vez que se elija una opción ésta se registrará.

En la siguiente ejecución del programa con el mismo usuario se encontrará esa secuencia y, en el caso de que el usuario elija la misma primera opción con la que empieza, los formularios variarán su presentación, tendrán sus opciones modificadas (si procede) y aparecerá un botón llamado “siguiente”. Este botón permite pasar a la siguiente opción de la secuencia que está siendo utilizada.

Las secuencias se filtran por día, lo que permite registrar todas las acciones realizadas en un mismo día como una única secuencia. Con esto se consigue que, en el caso de que se esté usando una secuencia anterior y en un momento dado el usuario la rompe usando otra opción, se registre toda la secuencia que se ha usado junto con el cambio como una nueva; de esta forma se pueden registrar todas las variantes realizadas. Por otro lado, si se llega al final de una secuencia y el usuario sigue trabajando, ocurrirá lo mismo, se agregará una nueva secuencia con las ampliaciones correspondientes. Así podremos tener información más que suficiente para que el motor de inferencia (no implementado) trabaje y produzca las conclusiones más realistas posibles. En el caso de que existan varias secuencias que empiecen por la misma opción siempre se elegirá la más reciente ya que se supone que es la más completa.

Si se da el caso de que al entrar en un formulario la siguiente opción que se realizó fue una interna a éste, directamente se presentará la información a la que da acceso dicha opción, ahorrando la presentación

Generación de menús adaptativos

(en este caso vacía) del formulario. Esto muestra que es posible ahorrar algo de tiempo usando este sistema.

Aunque es un prototipo extremadamente sencillo, nos ha servido para ver que, con lo realizado, se puede perfectamente conseguir una interfaz que maneje adecuadamente la información almacenada en la base de datos, y que permita modificar correctamente la interfaz gráfica.

12. Conclusiones y extensiones

Durante el desarrollo de este proyecto el equipo de diseño ha conseguido cumplir los objetivos que se había propuesto inicialmente.

Hemos conseguido diseñar un prototipo de aplicación que se adapta a las necesidades y características del usuario, permitiéndole disponer de una herramienta agradable y fácil de usar.

Con respecto a los conocimientos que hemos requerido para el diseño e implementación de nuestra herramienta (tanto a nivel teórico como práctico) nos hemos basado en los conocimientos adquiridos a lo largo de nuestra formación académica.

Con los ejemplos utilizados, se ha podido mostrar que el prototipo construido permite ser utilizado en aplicaciones de gestión mostrando resultados fiables y razonables. Su aplicación a otras áreas enriquecerá la extensión del prototipo y permitirá ser validado para usos posteriores.

Por ejemplo, en el futuro empresas de todo tipo podrán usar la aplicación para su gestión tanto interna como externa, para realizar operaciones y trámites de forma más rápida y eficaz.

Cierto es que nuestro prototipo acepta diversas ampliaciones y mejoras de muchos tipos. Por este motivo el diseño empleado ha sido elegido para facilitar cualquier intento de mejora o actualización para el caso en el que otros desarrolladores estuvieran interesados en aplicar esta tecnología para sus propósitos.

13. Anexos

13.1. Script SQL: generación de la base de datos

Para crear la base de datos es necesario un script SQL que implemente el diagrama mostrado anteriormente. Como se ha dicho anteriormente para generarlo se ha usado el proyecto de la UCM llamado DBCASE, sirviendo así como prueba de dicho proyecto.

```
-- SCRIPT GENERATED BY DATABASE DESIGN TOOL

-- Script generated using MySQL syntax

-- SECCION DE CREACION DE TABLAS

DROP TABLE IF EXISTS Acceso_a;

CREATE TABLE Acceso_a(Abreviatura VARCHAR(5), _da_Abreviatura
VARCHAR(5)) ENGINE=InnoDB;

DROP TABLE IF EXISTS Secuencia;

CREATE TABLE Secuencia(SiguienteOP VARCHAR(5), Fecha DATETIME NOT
NULL, DNI VARCHAR(9), ID INTEGER, Abreviatura VARCHAR(5))
ENGINE=InnoDB;

DROP TABLE IF EXISTS Contiene;

CREATE TABLE Contiene(Abreviatura_Aplicacion VARCHAR(5),
Abreviatura_BotonesAccion VARCHAR(5)) ENGINE=InnoDB;

DROP TABLE IF EXISTS Muestra;

CREATE TABLE Muestra(Abreviatura VARCHAR(5), ID INTEGER)
ENGINE=InnoDB;

DROP TABLE IF EXISTS Accede_a;

CREATE TABLE Accede_a(Lectura VARCHAR(1) NOT NULL, Escritura
VARCHAR(1) NOT NULL, DNI VARCHAR(9), Abreviatura_Departamentos
VARCHAR(10), Abreviatura_Aplicacion VARCHAR(5)) ENGINE=InnoDB;

DROP TABLE IF EXISTS Notas;

CREATE TABLE Notas(Fecha DATE NOT NULL, DNI VARCHAR(9), ID INTEGER)
ENGINE=InnoDB;

DROP TABLE IF EXISTS Info;

CREATE TABLE Info(DNI VARCHAR(9), ID INTEGER) ENGINE=InnoDB;

DROP TABLE IF EXISTS Posee;

CREATE TABLE Posee(Fecha DATE NOT NULL, DNI VARCHAR(9), ID INTEGER)
ENGINE=InnoDB;
```

Generación de menús adaptativos

```
DROP TABLE IF EXISTS Tiene;
```

```
CREATE TABLE Tiene(Abreviatura VARCHAR(10), Nombre VARCHAR(20))  
ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Trabaja_en;
```

```
CREATE TABLE Trabaja_en(Fecha DATE NOT NULL, Jefe VARCHAR(9) NOT  
NULL, DNI VARCHAR(9), Abreviatura VARCHAR(10)) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Supervisado_por;
```

```
CREATE TABLE Supervisado_por(_a_DNI VARCHAR(9), _por_DNI VARCHAR(9))  
ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS DatosPersonales;
```

```
CREATE TABLE DatosPersonales(Edad INTEGER NOT NULL, FechaNacimiento  
DATE NOT NULL, ECivil VARCHAR(10) NOT NULL, Ncuenta VARCHAR(20) NOT  
NULL, Telefono INTEGER NOT NULL, Telefono2 INTEGER, Direccion  
VARCHAR(30) NOT NULL, Hijos INTEGER NOT NULL, ID INTEGER NOT NULL  
AUTO_INCREMENT, PRIMARY KEY(ID)) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Empresa;
```

```
CREATE TABLE Empresa(Nombre VARCHAR(20) NOT NULL, Direccion  
VARCHAR(30) NOT NULL, Nempleados INTEGER NOT NULL, Nsedes INTEGER NOT  
NULL, VMercado REAL NOT NULL, Facturacion REAL NOT NULL, Sector  
VARCHAR(15) NOT NULL) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Aplicacion;
```

```
CREATE TABLE Aplicacion(Abreviatura VARCHAR(5) NOT NULL, Nombre  
VARCHAR(15) NOT NULL, Tipo VARCHAR(15) NOT NULL, Version REAL NOT  
NULL) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Departamentos;
```

```
CREATE TABLE Departamentos(Abreviatura VARCHAR(10) NOT NULL, Nombre  
VARCHAR(20) NOT NULL, NEmpleados INTEGER NOT NULL) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Observaciones;
```

```
CREATE TABLE Observaciones(Habilidad VARCHAR(20) NOT NULL, ID INTEGER  
NOT NULL AUTO_INCREMENT, PRIMARY KEY(ID)) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS BotonesAccion;
```

```
CREATE TABLE BotonesAccion(Abreviatura VARCHAR(5) NOT NULL, Accion  
VARCHAR(10), Nombre VARCHAR(10) NOT NULL) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Informacion;
```

```
CREATE TABLE Informacion(Formato VARCHAR(20) NOT NULL, Presentacion  
VARCHAR(20) NOT NULL, Tema VARCHAR(20) NOT NULL, ID INTEGER NOT NULL  
AUTO_INCREMENT, PRIMARY KEY(ID)) ENGINE=InnoDB;
```

```
DROP TABLE IF EXISTS Formacion;
```

```
CREATE TABLE Formacion(Academica VARCHAR(20) NOT NULL, ID INTEGER NOT  
NULL AUTO_INCREMENT, PRIMARY KEY(ID)) ENGINE=InnoDB;
```


Generación de menús adaptativos

```
DROP TABLE IF EXISTS Empleados;

CREATE TABLE Empleados(DNI VARCHAR(9) NOT NULL, Nombre VARCHAR(30)
NOT NULL, Nivel INTEGER NOT NULL, Rendimiento INTEGER NOT NULL,
AnosEmpresa INTEGER NOT NULL) ENGINE=InnoDB;

-- SECCION DE CREACION DE TIPOS ENUMERADOS

DROP TABLE IF EXISTS BOOL;

CREATE TABLE BOOL(value_list VARCHAR(1)) ENGINE=InnoDB;

ALTER TABLE BOOL ADD PRIMARY KEY (value_list);

INSERT INTO BOOL values ('1');

INSERT INTO BOOL values ('0');

-- SECCION DE ESTABLECIMIENTO DE CLAVES

ALTER TABLE Empresa ADD PRIMARY KEY (Nombre);

ALTER TABLE Aplicacion ADD PRIMARY KEY (Abreviatura);

ALTER TABLE Departamentos ADD PRIMARY KEY (Abreviatura);

ALTER TABLE BotonesAccion ADD PRIMARY KEY (Abreviatura);

ALTER TABLE Empleados ADD PRIMARY KEY (DNI);

ALTER TABLE Acceso_a ADD PRIMARY KEY (Abreviatura, _da_Abreviatura);

ALTER TABLE Acceso_a ADD FOREIGN KEY (Abreviatura) REFERENCES
BotonesAccion(Abreviatura);

ALTER TABLE Acceso_a ADD FOREIGN KEY (_da_Abreviatura) REFERENCES
BotonesAccion(Abreviatura);

ALTER TABLE Secuencia ADD PRIMARY KEY (Fecha, Abreviatura, DNI, ID);

ALTER TABLE Secuencia ADD FOREIGN KEY (DNI) REFERENCES
Empleados(DNI);

ALTER TABLE Secuencia ADD FOREIGN KEY (ID) REFERENCES
Informacion(ID);

ALTER TABLE Secuencia ADD FOREIGN KEY (Abreviatura) REFERENCES
BotonesAccion(Abreviatura);

ALTER TABLE Contiene ADD PRIMARY KEY (Abreviatura_Aplicacion,
Abreviatura_BotonesAccion);

ALTER TABLE Contiene ADD FOREIGN KEY (Abreviatura_Aplicacion)
REFERENCES Aplicacion(Abreviatura);

ALTER TABLE Contiene ADD FOREIGN KEY (Abreviatura_BotonesAccion)
REFERENCES BotonesAccion(Abreviatura);

ALTER TABLE Muestra ADD PRIMARY KEY (Abreviatura, ID);

ALTER TABLE Muestra ADD FOREIGN KEY (Abreviatura) REFERENCES
Aplicacion(Abreviatura);
```

Generación de menús adaptativos

```
ALTER TABLE Muestra ADD FOREIGN KEY (ID) REFERENCES Informacion(ID);

ALTER TABLE Accede_a ADD PRIMARY KEY (DNI, Abreviatura_Departamentos,
Abreviatura_Aplicacion);

ALTER TABLE Accede_a ADD FOREIGN KEY (Lectura) REFERENCES
BOOL(value_list);

ALTER TABLE Accede_a ADD FOREIGN KEY (Escritura) REFERENCES
BOOL(value_list);

ALTER TABLE Accede_a ADD FOREIGN KEY (DNI) REFERENCES Empleados(DNI);

ALTER TABLE Accede_a ADD FOREIGN KEY (Abreviatura_Departamentos)
REFERENCES Departamentos(Abreviatura);

ALTER TABLE Accede_a ADD FOREIGN KEY (Abreviatura_Aplicacion)
REFERENCES Aplicacion(Abreviatura);

ALTER TABLE Notas ADD PRIMARY KEY (DNI, ID);

ALTER TABLE Notas ADD FOREIGN KEY (DNI) REFERENCES Empleados(DNI);

ALTER TABLE Notas ADD FOREIGN KEY (ID) REFERENCES Observaciones(ID);

ALTER TABLE Info ADD PRIMARY KEY (DNI, ID);

ALTER TABLE Info ADD FOREIGN KEY (DNI) REFERENCES Empleados(DNI);

ALTER TABLE Info ADD FOREIGN KEY (ID) REFERENCES DatosPersonales(ID);

ALTER TABLE Posee ADD PRIMARY KEY (DNI, ID);

ALTER TABLE Posee ADD FOREIGN KEY (DNI) REFERENCES Empleados(DNI);

ALTER TABLE Posee ADD FOREIGN KEY (ID) REFERENCES Formacion(ID);

ALTER TABLE Tiene ADD PRIMARY KEY (Abreviatura, Nombre);

ALTER TABLE Tiene ADD FOREIGN KEY (Abreviatura) REFERENCES
Departamentos(Abreviatura);

ALTER TABLE Tiene ADD FOREIGN KEY (Nombre) REFERENCES
Empresa(Nombre);

ALTER TABLE Trabaja_en ADD PRIMARY KEY (DNI, Abreviatura);

ALTER TABLE Trabaja_en ADD FOREIGN KEY (DNI) REFERENCES
Empleados(DNI);

ALTER TABLE Trabaja_en ADD FOREIGN KEY (Abreviatura) REFERENCES
Departamentos(Abreviatura);

ALTER TABLE Supervisado_por ADD PRIMARY KEY (_a_DNI);

ALTER TABLE Supervisado_por ADD FOREIGN KEY (_a_DNI) REFERENCES
Empleados(DNI);

ALTER TABLE Supervisado_por ADD FOREIGN KEY (_por_DNI) REFERENCES
Empleados(DNI);
```

El script consiste en la creación de las distintas tablas que componen las entidades y las relaciones de acuerdo con el diagrama E-R, la creación de una tabla auxiliar para almacenar valores booleanos y la asignación de claves primarias y externas.

13.2.Script SQL: datos para el prototipo

Este script ha sido generado a mano teniendo en cuenta la aplicación sobre la que se iban a realizar las pruebas, contiene las siguientes sentencias SQL:

```
insert into Empleados values (000000001, 'David', 1, 1, 0);
insert into Departamentos values ('CSS', 'David', 2);
insert into Empleados values (000000002, 'Jefe', 2, 2, 5);
insert into Trabaja_en values (08-05-2009, 000000002, 000000001, 'CSS');
insert into Trabaja_en values (08-05-2009, 000000002, 000000002, 'CSS');
insert into Informacion values ('Info', 'Normal', 'Tema', 0);
insert into BotonesAccion values ('op1', 'Opcion1', 'Opcion 1');
insert into BotonesAccion values ('op2', 'Opcion2', 'Opcion 2');
insert into BotonesAccion values ('op3', 'Opcion3', 'Opcion 3');
insert into BotonesAccion values ('op4', 'Opcion4', 'Opcion 4');
insert into BotonesAccion values ('op5', 'Opcion5', 'Opcion 5');
insert into BotonesAccion values ('op6', 'Opcion6', 'Opcion 6');
insert into BotonesAccion values ('op7', 'Opcion7', 'Opcion 7');
insert into BotonesAccion values ('op8', 'Opcion8', 'Opcion 8');
insert into BotonesAccion values ('op1A', 'Entradas', 'Entradas');
insert into BotonesAccion values ('op1B', 'Salidas', 'Salidas');
insert into BotonesAccion values ('op1C', 'Pendientes', 'Pendientes');
insert into BotonesAccion values ('op2A', 'Compras', 'Compras');
insert into BotonesAccion values ('op2B', 'Ventas', 'Ventas');
insert into BotonesAccion values ('op2C', 'Balance', 'Balance');
```

Generación de menús adaptativos

```
insert into BotonesAccion values ('op3A','Ventas','Ventas');
insert into BotonesAccion values ('op3B','Compras','Compras');
insert into BotonesAccion values ('op5A','Todos','Todos');
insert into BotonesAccion values ('op5B','MesAct','MesAct');
insert into BotonesAccion values ('op5C','Resumenes','Resumenes');
insert into BotonesAccion values ('op5D','Ventas','Ventas');
insert into BotonesAccion values ('op5E','Compras','Compras');
insert into BotonesAccion values ('op6A','Hoy','Hoy');
insert into BotonesAccion values ('op6B','Semana','Semana');
insert into BotonesAccion values ('op6C','Mes','Mes');
insert into BotonesAccion values ('op7A','Niveles','Niveles');
insert into BotonesAccion values
('op7B','Intercambios','Intercambios');
insert into BotonesAccion values
('op7C','INversiones','Inversiones');
```

Lo primero que insertamos son los usuarios y su departamento junto con la relación que hay entre ellos. Luego para realizar el ejemplo suponemos que solo hay un tipo de información. Para que todo funcione correctamente insertamos los botones de los cuales dispone nuestra aplicación de prueba, los ocho primeros corresponden con botones que dan paso a un formulario distinto al que nos encontramos, mientras que los que acaban en letra son los botones internos a los distintos formularios y que modican la información mostrada por este.

14. Bibliografía

- Fundamentos de bases de datos / Henry F. Korth, Abraham Silberschatz; Traducción M^a Ángeles Vaquero Martín, Antonio Vaquero García; revisión técnica M^a Amparo Vila Miranda.
- <http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=485>
- <http://lsi.ugr.es/~gedes/grupo/PubN/02%20Med%20TallerSHCyA%20JISBD.pdf>