



Devenir Formación

PROYECTO FINAL DE CICLO

Gestión de Faltas de Asistencia

Autor: Alejandro Córdoba
Muñoz

Tutor: María Desirée Pérez
García

Ciclo: DAM

Curso: 2014

Índice

1. Introducción.
2. Justificación del Proyecto.
3. ¿A Quién Va Dirigida la Aplicación?
4. Planificación del Proyecto.
5. Descripción del Funcionamiento y Diseño de Pantallas.
6. Diseño de Base de Datos.
7. Propuestas de Mejora para la Aplicación.
8. Conclusión y Valoración Personal del Proyecto.
9. Bibliografía y Fuentes de Consulta.
10. Contraportada.

1. Introducción.

En esta documentación sobre la aplicación desarrollada, esta orientada para el usuario o persona que vaya a utilizarla en el uso de sus funciones. Con esta documentación queremos ofrecer una serie de pautas sobre el diseño de la base de datos, más adelante se explica sobre el desarrollo del diagrama E-R o el diagrama de clases de la aplicación, descripción del funcionamiento y diseño de pantallas atendiendo a las pautas de usabilidad.

Esta documentación se ha desarrollado para:

- Facilitar trabajo al usuario o persona que utilice la aplicación.
- Comprensión correcta y coherente de la aplicación.
- Comprensión de los componentes que forman nuestra aplicación.
- Reflejar el trabajo llevado a cabo a lo largo del proyecto.
- Otras.

2. Justificación del Proyecto.

Esta aplicación ha sido desarrollada como proyecto final del curso de Desarrollo de Aplicaciones Multiplataforma (DAM), concretamente nuestra aplicación tratará sobre la gestión de faltas de asistencia, asignada como tutora del alumno a María Desirée Pérez García.

Con esta aplicación se tratará la gestión de unos alumnos, matriculados en un curso, y los módulos profesionales a los que pertenecen dichos alumnos. También es posible registrar las faltas de asistencia de dichos módulos profesionales e imprimir un informe con los resultados obtenidos.

3. ¿A Quién Va Dirigida la Aplicación?

Esta aplicación desarrollada en el proyecto final va dirigida principalmente a los tutores del curso Desarrollo de Aplicaciones Multiplataforma (DAM).

Previa aprobación del proyecto desarrollado sobre gestión de faltas de asistencia, la aplicación va dirigida a todas aquellas empresas, instituciones y organizaciones que buscan nuevas maneras de acceder a la innovación.

Y esa innovación puede venir de su propia gestión de faltas de asistencia sobre sus alumnos, la gestión de sus alumnos o la gestión de los módulos profesionales que componen un determinado curso.

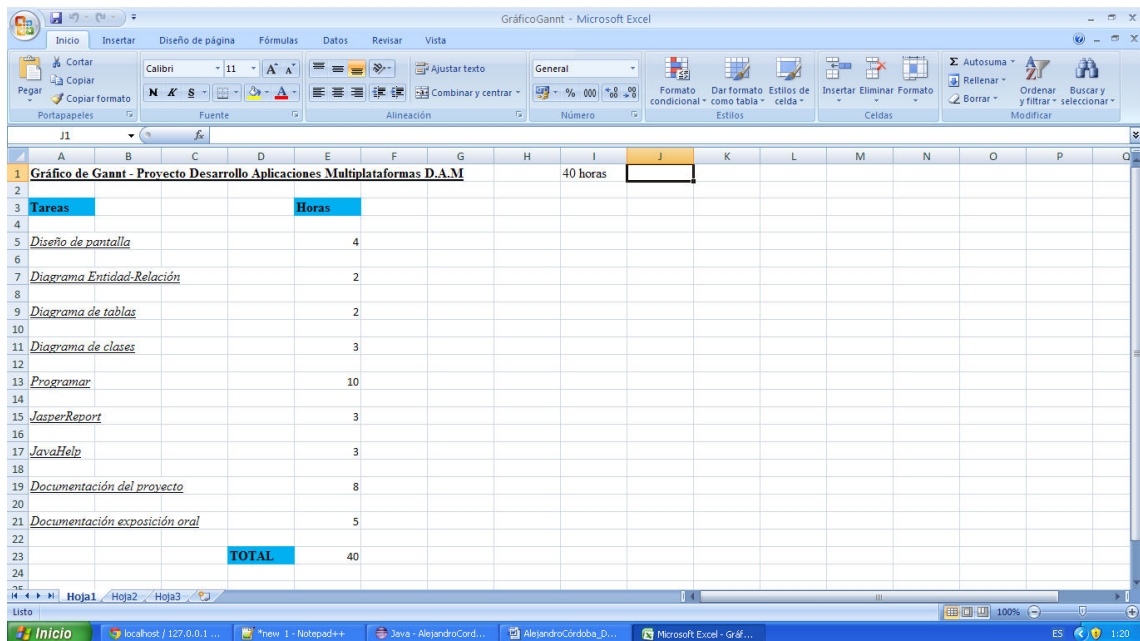
4. Planificación del Proyecto.

En la sección de este apartado se mostrara a continuación la planificación del desarrollo de las tareas propuestas y llevadas a cabo a lo largo del desarrollo del proyecto,

El desarrollo de la planificación del proyecto se ha realizado mediante unas hojas de cálculos Microsoft Office Excel. Se han desarrollado un diagrama de Gannt, realizado sobre las horas estimadas en clase para el desarrollo del proyecto.

A continuación se explicaran detalladamente cada una de las tareas llevadas a cabo:

Planificación en Horas Estimadas.



-Diseño de pantalla. Diseño de las pantallas de la aplicación desarrollada para el proyecto.

-Diagrama Entidad-Relación. Diseño del diagrama entidad-relación de la base de datos diseñada.

-Diagrama de tablas. Documentación apartada junto con el proyecto justificando el cumplimiento de la normalización de la base de datos.

-Diagrama de clases. Desarrollo de los diagramas UML sobre las clases implementadas en java en el desarrollo del proyecto.

-Programar. Implementación de clases, Hibernate, mapeos y clasesBean, tratamiento con java en la aplicación.

-JasperReport. Realización y implementación de informe, iReport,

-JavaHelp. Implementación de la ayuda de la aplicación, con la librerías JavaHelp.

-Documentación del proyecto. Desarrollo del presente documento.

-Documentación exposición oral. Desarrollo de la documentación sobre la exposición del proyecto.

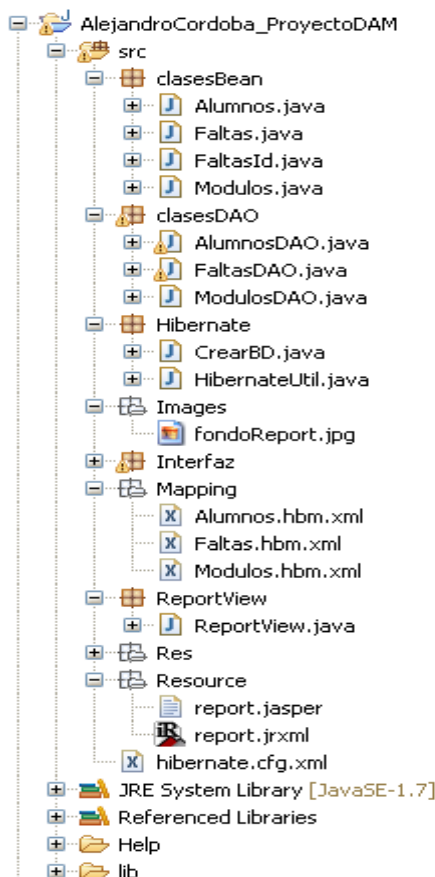
5. Descripción del Funcionamiento y Diseño de Pantallas.

En este apartado del documento podríamos dividirlo en dos partes, en una de ellas explicaremos el funcionamiento de la aplicación desarrollada para el proyecto de Desarrollo de Aplicaciones Multiplataforma de gestión de faltas de asistencia a través de los diagramas de clases realizados a través de diagramas UML.

En la otra parte del documento se describe el diseño de las pantallas implementadas en la aplicación atendiendo a las pautas de usabilidad.

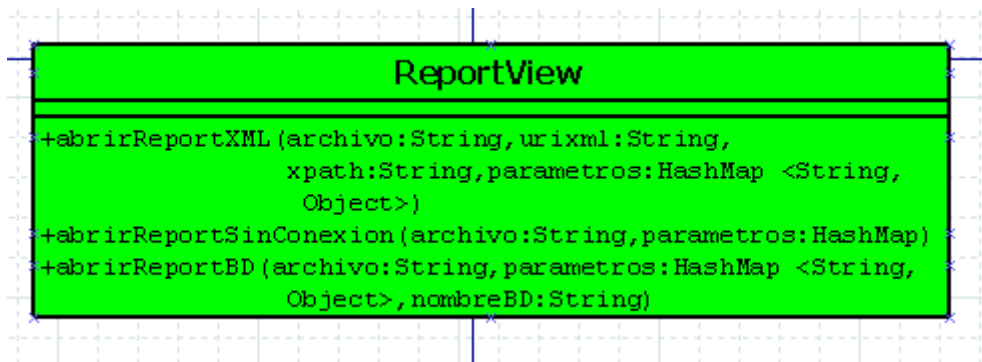
Descripción del Funcionamiento

En primer lugar expondremos la estructura de nuestro proyecto para cuando hagamos referencia a una clase o a varias de ellas saber en qué parte del proyecto nos encontramos para entender mejor el funcionamiento llevado a cabo en la aplicación.



Clase ReportView

Es la clase que se ha desarrollado en java para crear unos métodos que sean accesibles a las librerías de **JasperReport**. Concretamente en esta clase implementamos la funcionalidad de abrir informes mediante un archivo XML, un informe sin conexión de datos ninguna y un informe mediante una base de datos.

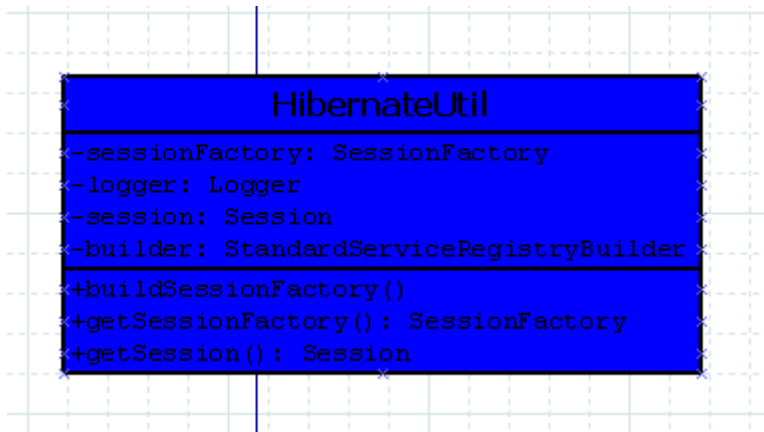


Clase HibernateUtil

Es la clase que se ha desarrollado en java para poder obtener la conexión a la base de datos que soporta la aplicación mediante las librerías que implementa **Hibernate**, en ella se declaran los métodos **buildSessionFactory**, para construir un objeto tipo **SessionFactory** para a partir del poder obtener la conexión a la base de datos.

Luego se implementa el método **getSessionFactory**, para que ese objeto que hemos creado en nuestra clase anterior nos devuelva un valor, dependiendo del valor que nos devuelve ese objeto podemos obtener o no la conexión.

Y por ultimo un método para obtener un objeto **Session** que este nos devolverá la conexión a la base de datos denominado **getSession**.



Clases Bean

Estas clases no son más que los ficheros java desarrollados en la aplicación correspondiente al mapeo realizado de las tablas de las bases de datos, establecida también las relaciones, para luego poder acceder a cada uno de sus campos para poder gestionar el dato obtenido en la aplicación.

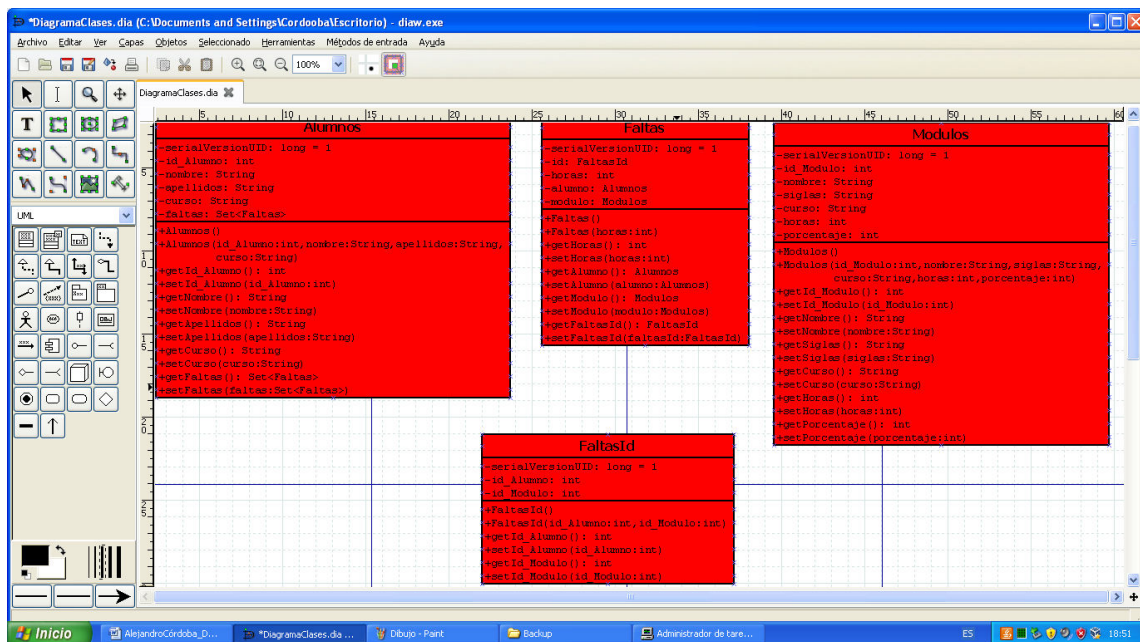
A continuación se define cuales los atributos de cada clase.

-Clase Alumnos, **int id_Alumno**, **String nombre**, **String apellidos**, **String curso**, **Set <Faltas> faltas**.

-Clase Faltas, **int FaltasId**, **int horas**, **Alumnos alumno**, **Modulos modulo**.

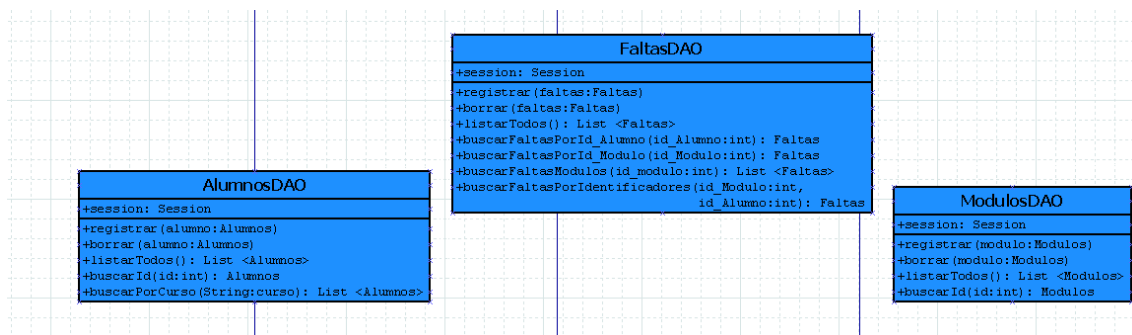
-Clase Modulos, **int id_Modulo**, **String nombre**, **String siglas**, **String curso**, **int horas**, **int porcentaje**.

-Clase FaltasId, **int id_Alumno**, **int id_Modulo**.

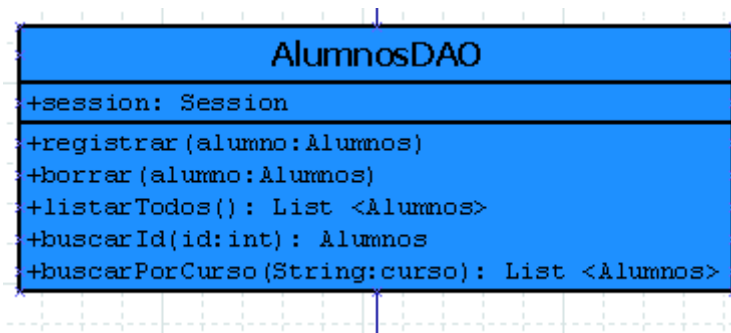


Clases DAO

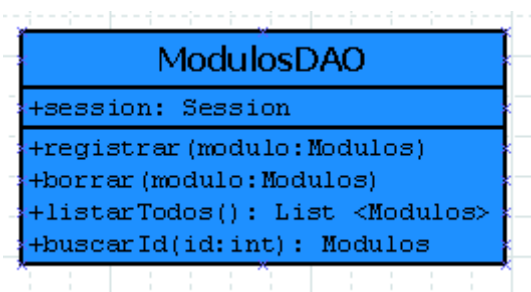
Las clases denominadas como DAO son las clases utilizadas para el tratamiento de datos con Hibernate a través de java, como por ejemplo poder buscar a Módulo Profesional en concreto a través de algún valor de sus campos como por ejemplo el identificador, nombre, entre otros.



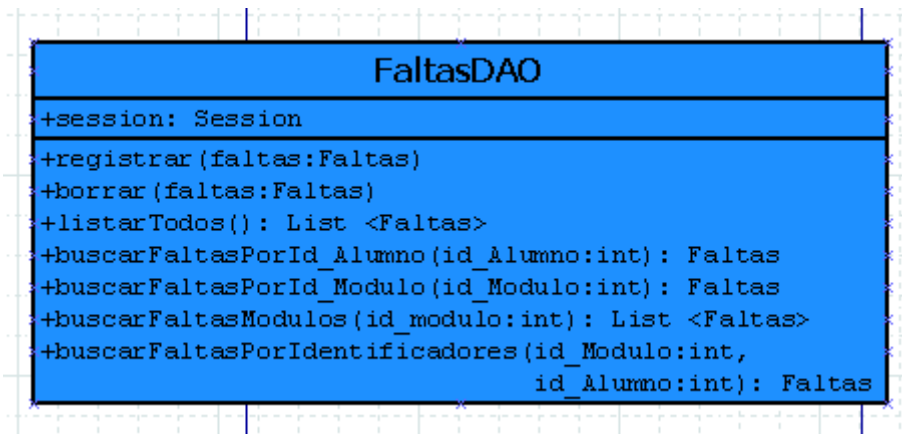
-AlumnosDAO, se compone de los métodos **registrar** (introducir un nuevo registro en la base de datos a través de un Alumno pasado por parámetros), **borrar** (borrar un registro existente en la base de datos a través de un objeto pasa por parámetros de tipo Alumno), **listarTodos** (obtener todos los registros existentes en la base de datos a través de un objeto tipo List <Alumno>, así podremos acceder a todos sus campos de la base de datos), **buscarId** (buscar un registro en la base de datos a través del campo ID_ALUMNO pasado como parámetros al método citado), **buscarPorCurso** (buscar un alumno en la base de datos a través del campo CURSO, esta consulta nos devolverá un conjunto de valores para ello este método devuelve un objeto de tipo List <Alumnos>).



-ModulosDAO, se compone de los métodos **registrar** (introducir un nuevo registro en la base de datos a través de un Modulo pasado por parámetros), **borrar** (borrar un módulo profesional existente en la base de datos a través de un objeto pasado por parámetros de tipo Modulo), **listarTodos** (obtener todos los módulos profesionales en la base de datos a través de un objeto tipo List <Modulo>, así será más fácil acceder a todos los módulos profesionales registrados en la base de dato), **buscarId** (buscar un registro en la base de datos a través del campo ID_MODULO pasado como parámetros al método citado).



-FaltasDAO. se compone de los métodos **registrar**(introducir un nuevo registro en la base de datos a través de una Faltas pasada por parámetros), **borrar**(borrar un registro en la base de datos a través de una Faltas pasada por parámetros), **listarTodos**(obtener todos las faltas de asistencia en la base de datos a través de un objeto tipo List <Faltas>, así será más fácil acceder a todos los módulos profesionales registrados en la base de dato), **buscarFaltasPorId_Alumno**(buscar las fatlas de asistencia a través del identificador del alumno), **buscarFaltasPorId_Modulo**(buscar las faltas de asistencia a través del identificador del modulo), **buscarFaltasModulos**(buscar todas las faltas de asistencia a través del identificador del alumno, devolviendo un objeto tipo List <Faltas> así será más fácil el tratamiento de datos), **buscarFaltasPorIdentificadores**(buscar una falta de asistencia en concreto a través de los parámetros que se identifican con los campos ID_ALUMNO y ID_MODULO).

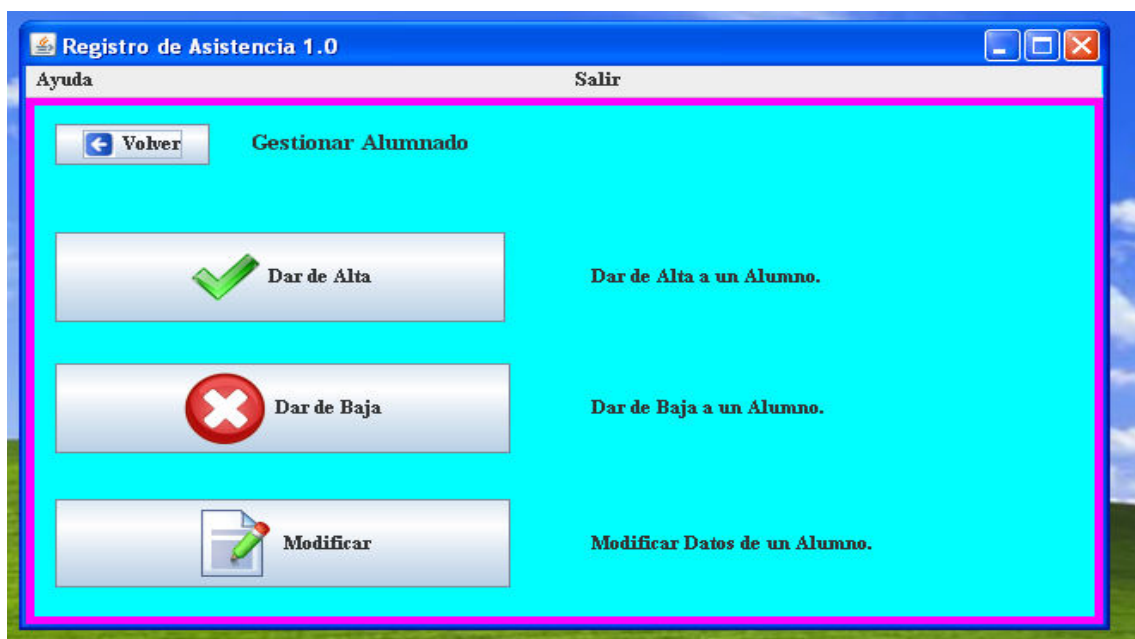


Una vez explicada ya las clases que componen nuestra aplicación pasaremos a saber cómo funciona nuestra aplicación desarrollada de faltas de asistencia.

La primera vista que obtenemos de la aplicación es la siguiente, y está compuesta de cuatros botones principales.



Cada botón representa un conjunto de acciones que se relacionan y están agrupadas por funcionalidades, es decir, dentro de gestionar alumnado podremos realizar todas las funcionalidades específicas que podemos llevar a cabo en nuestra aplicación relacionada con la gestión de alumnos como por ejemplo dar de alta a un alumno, dar de baja a un alumno o modificar los datos de un alumno seleccionado específicamente.



Dar de Alta

Para dar de alta un alumno solo deberemos introducir los campos que se muestran en la pantalla. Todos los campos son obligatorios para poder insertar un nuevo alumno.

Una vez introducidos todos los campos solo deberemos seleccionar el botón confirmar para insertar el registro.

Si el registro se ha insertado correctamente la aplicación lanzara un mensaje de información al usuario, informando de que se ha insertado un nuevo registro en la base de datos.

En esta pantalla se han implementados dos botones más, uno denominado volver para volver a la pantalla anterior y otro denominado limpiar campos que nos facilitara la operación de borrar el texto introducido en los campos.

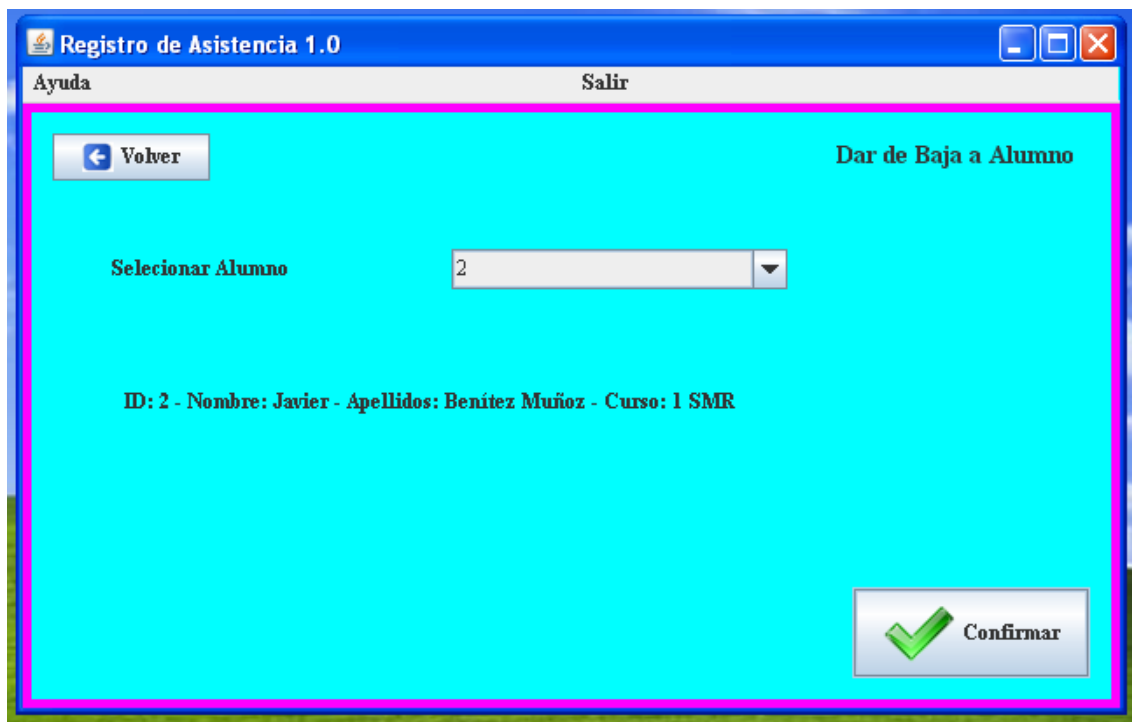
The screenshot shows a Windows-style application window titled "Registro de Asistencia 1.0". It has a menu bar with "Ayuda" and "Salir". The main content area has a cyan background and is titled "Dar de Alta a Alumno". In the top left corner of this area is a button with a left arrow and the text "Volver". The form contains four input fields: "Identificador" (text), "Nombre" (text), "Curso" (a dropdown menu currently showing "1 DAM"), and "Apellidos" (text). At the bottom right of the form area are two buttons: "Limpiar Campos" (with a circular arrow icon) and "Confirmar" (with a green checkmark icon).

Dar de Baja

Para dar de baja a un alumno solo deberemos seleccionar el identificador del alumno a eliminar.

Una vez seleccionado, automáticamente se mostrara abajo toda la información correspondiente a ese alumno.

Si queremos eliminar el alumno seleccionado solo deberemos pulsar el botón confirmar. Si la aplicación ha eliminado el registro correspondiente a través del identificador del alumno seleccionado, la aplicación mostrara un mensaje de información al usuario.



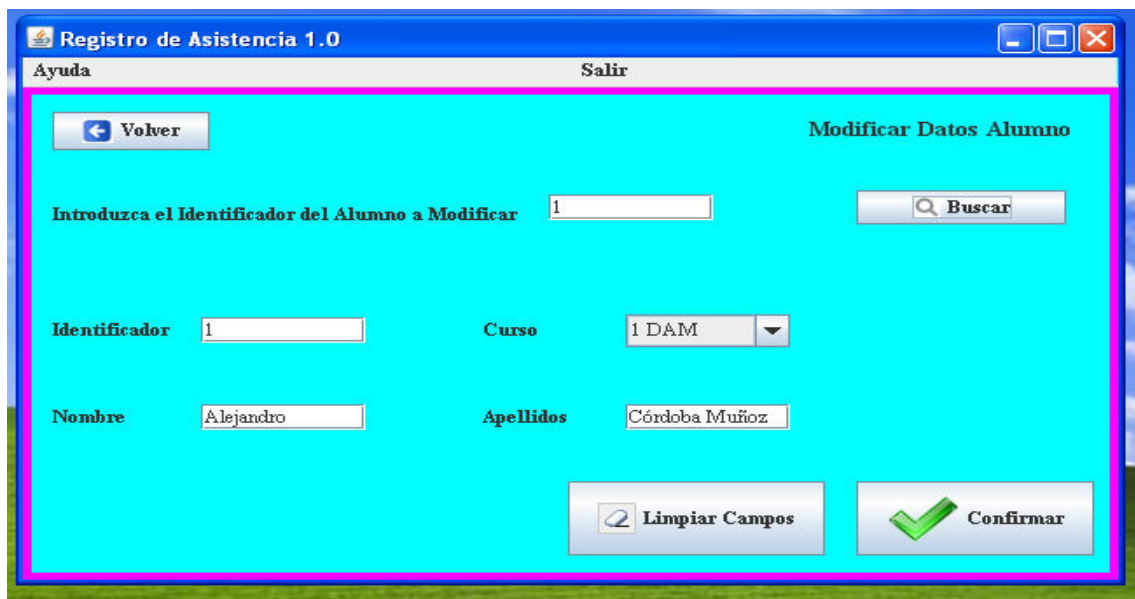
The screenshot shows a Windows-style application window titled "Registro de Asistencia 1.0". The window has a menu bar with "Ayuda" and "Salir". The main content area has a cyan background and is titled "Dar de Baja a Alumno" in the top right corner. In the top left, there is a "Volver" button with a left arrow. Below it, the label "Seleccionar Alumno" is followed by a text box containing the number "2" and a dropdown arrow. Further down, the text "ID: 2 - Nombre: Javier - Apellidos: Benítez Muñoz - Curso: 1 SMR" is displayed. In the bottom right corner, there is a "Confirmar" button featuring a green checkmark icon.

Modificar

Para poder modificar un dato del alumno, es necesario introducir el identificador del alumno.

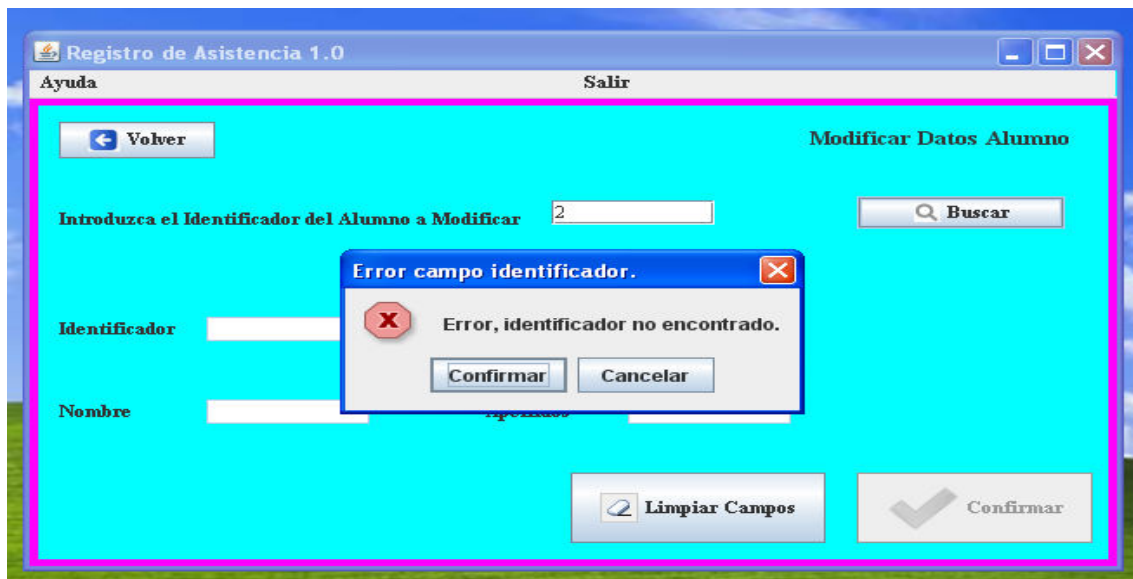
Una vez introducido un identificador para buscar, es necesario seleccionar el botón buscar para poder realizar la consulta a la base de datos.

Si obtiene resultado mostrara los datos del alumno en la cajas de texto, y modificaremos los valores. Una vez modificado seleccionamos el botón confirmar.



The screenshot shows a Windows-style application window titled "Registro de Asistencia 1.0". It has a menu bar with "Ayuda" and "Salir". The main area is titled "Modificar Datos Alumno" and has a cyan background. At the top left is a "Volver" button with a left arrow. Below it, a label "Introduzca el Identificador del Alumno a Modificar" is followed by a text box containing "1" and a "Buscar" button with a magnifying glass icon. Further down, there are four input fields: "Identificador" (containing "1"), "Curso" (a dropdown menu showing "1 DAM"), "Nombre" (containing "Alejandro"), and "Apellidos" (containing "Córdoba Muñoz"). At the bottom, there are two buttons: "Limpiar Campos" with an eraser icon and "Confirmar" with a green checkmark icon.

Si la consulta no devuelve ningún resultado en la búsqueda, la aplicación nos mostrara un mensaje de error.



Una vez explicado el botón de gestionar alumnos, explicaremos el botón de gestionar módulos profesionales, en el cual se ofrecen todas las funcionalidades que abarca el área de módulos profesionales como dar de alta, dar de baja o incluso gestionar alumnado para poder matricular o desmatricular un alumno en específico.



Dar de Alta

Para dar de alta un módulo profesional solo deberemos introducir los campos que se muestran en la pantalla.

Una vez introducidos todos los campos solo deberemos seleccionar el botón confirmar para insertar el registro.

Si el registro se ha insertado correctamente la aplicación lanzara un mensaje de información al usuario, informando de que se ha insertado un nuevo registro en la base de datos.

En esta pantalla se han implementados dos botones más, uno denominado volver para volver a la pantalla anterior y otro denominado limpiar campos que nos facilitara la operación de borrar el texto introducido en los campos.

The screenshot shows a Windows-style application window titled "Registro de Asistencia 1.0". The window has a menu bar with "Ayuda" and "Salir". The main content area has a cyan background and is titled "Dar de Alta un Módulo Profesional". In the top left corner of the content area is a button labeled "Volver" with a left-pointing arrow. The form contains six input fields arranged in two columns. The left column has labels "Identificador", "Nombre", and "Siglas". The right column has labels "Curso", "N° Horas", and "Porcentaje". The "Curso" field is a dropdown menu currently showing "1 DAM". The "Porcentaje" field has a "%" symbol to its right. At the bottom of the form are two buttons: "Limpiar Campo" with an eraser icon and "Confirmar" with a green checkmark icon.

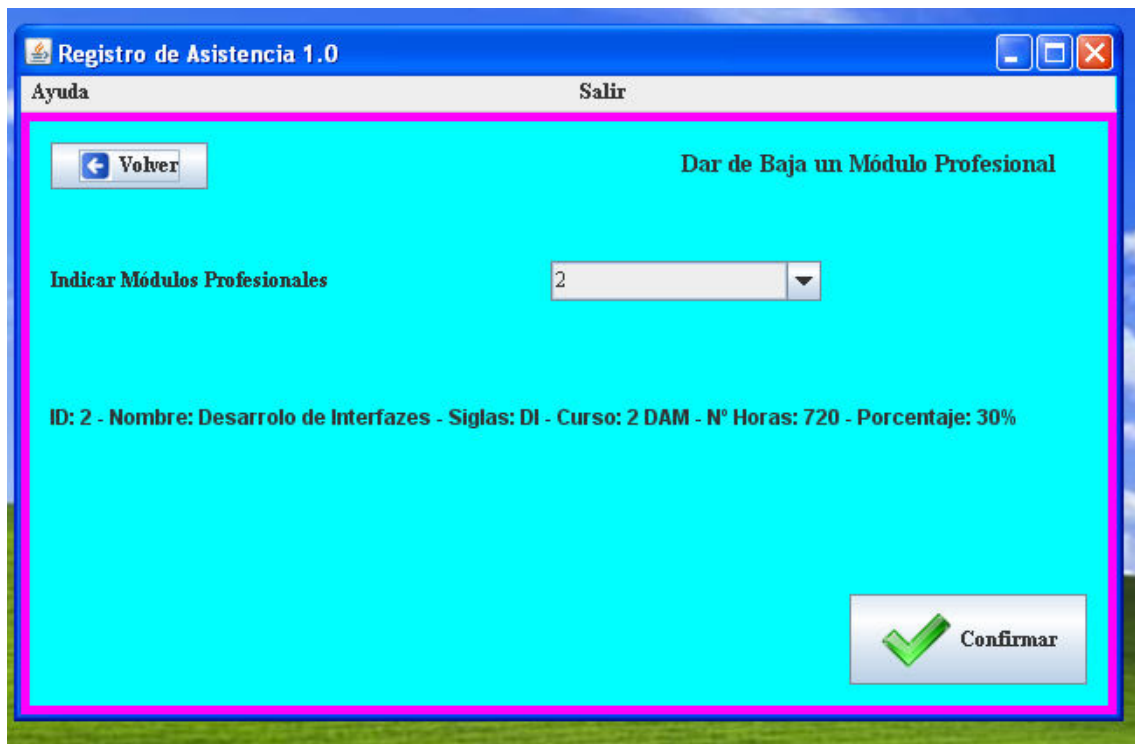
Identificador	<input type="text"/>	Curso	<input type="text" value="1 DAM"/>
Nombre	<input type="text"/>	N° Horas	<input type="text"/>
Siglas	<input type="text"/>	Porcentaje	<input type="text"/> %

Dar de Baja

Para dar de baja a un módulo profesional solo deberemos seleccionar el identificador del módulo profesional a eliminar.

Una vez seleccionado, automáticamente se mostrara abajo toda la información correspondiente a ese módulo profesional.

Si queremos eliminar el módulo profesional seleccionado solo deberemos pulsar el botón confirmar. Si la aplicación ha eliminado el registro correspondiente a través del identificador del módulo profesional seleccionado, la aplicación mostrara un mensaje de información al usuario.




The screenshot shows a Windows-style application window titled "Registro de Asistencia 1.0". The window has a menu bar with "Ayuda" and "Salir". The main content area has a light blue background and is titled "Dar de Baja un Módulo Profesional". In the top left corner of the content area is a button labeled "Volver" with a left-pointing arrow. Below this is a label "Indicar Módulos Profesionales" followed by a text box containing the number "2" and a dropdown arrow. Further down, the following text is displayed: "ID: 2 - Nombre: Desarrollo de Interfases - Siglas: DI - Curso: 2 DAM - N° Horas: 720 - Porcentaje: 30%". In the bottom right corner of the content area is a button labeled "Confirmar" with a green checkmark icon.

Matricular Alumno

Para poder matricular a un alumno en un módulo profesional es necesario que el alumno pertenezca al mismo curso que el módulo profesional, para ello al introducir el identificador del alumno o el identificador del módulo profesional la aplicación mostrará el curso del alumno y del módulo profesional, si no lo encuentra mostrará que ese identificador no es válido.

En caso de que no pertenezcan al mismo curso al pulsar sobre el botón confirmar se deshabilitará automáticamente impidiendo realizar la operación errónea.

Para volver a matricular a un alumno en un módulo profesional es necesario pulsar sobre el botón actualizar para borrar todos los campos introducidos y volver a habilitar el botón confirmar.



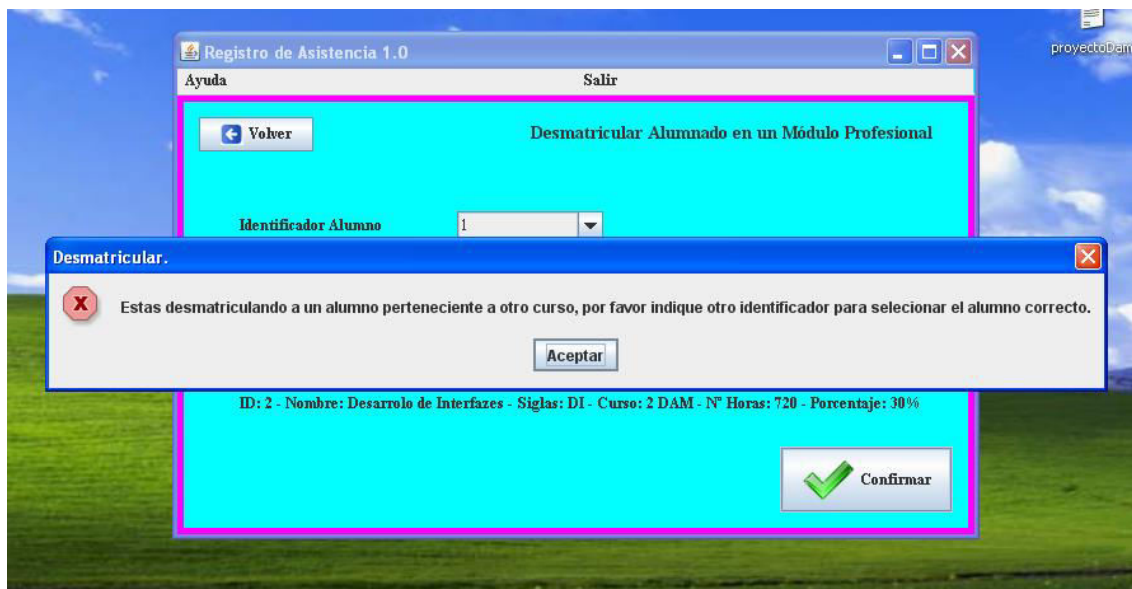
The screenshot shows a Windows-style application window titled "Registro de Asistencia 1.0". It has a menu bar with "Ayuda" and "Salir". The main area has a cyan background and is titled "Matricular Alumnado en un Módulo Profesional". In the top left corner of the main area is a "Volver" button with a left arrow. The form contains two rows of input fields. The first row has "Identificador Alumno" with a text box containing "1" and "Curso Alumno: 1 DAM". The second row has "Identificador Módulo" with a text box containing "2" and "Curso Módulo: 2 DAM". At the bottom are three buttons: "Actualizar" with a green circular arrow icon, "Limpiar Campo" with a blue eraser icon, and "Confirmar" with a grey checkmark icon.

Desmatricular Alumno

Para poder desmatricular a un alumno en un módulo profesional es necesario que el alumno pertenezca al mismo curso que el módulo profesional, para ello al seleccionar el identificador del alumno o el identificador del módulo profesional posibles en la base de datos la aplicación mostrara el curso del alumno y del módulo profesional, si no lo encuentra mostrará que ese identificador no es válido.

En caso de que no pertenezcan al mismo curso al pulsar sobre el botón confirmar la aplicación mostrara un mensaje de error informando al usuario que esta seleccionado a un alumno perteneciente a otro curso.

En caso de que no haya ningún tipo de problemas la aplicación mostrara al usuario un mensaje de confirmación al usuario sobre el registro introducido en la base de datos.



Una vez explicado el botón de gestionar módulos profesionales, explicaremos el botón de gestionar faltas de asistencia, en el cual se ofrecen todas las funcionalidades que abarca el área de faltas de asistencia como registrar, modificar, consultar o imprimir.



Registrar

Para registrar una falta de asistencia es necesario haber matriculado antes a algún alumno en un algún módulo profesional del mismo curso.

Deberemos seleccionar el identificador del módulo profesional, para poder seleccionar solo los alumnos a los que pertenecen ese curso.

Si el usuario selecciona un alumno que pertenezca a un curso diferente del módulo profesional, la aplicación mostrara un mensaje de error al usuario y deshabilitara el botón de confirmación hasta no haber seleccionado a un alumno que pertenece al mismo curso que el módulo profesional el botón no volverá a estar disponible.

Por último se muestra el contenido del número de horas que dicho alumno ha faltado o no ha asistido al módulo profesional indicado, y si introducimos el nuevo valor este se sumara automáticamente al valor visualizado anteriormente tras presionar el botón de confirmar.

Si se ha registrado la falta de asistencia la aplicación mostrara un mensaje de información al usuario.

Modificar

Para modificar una falta de asistencia es necesario seleccionar el identificador del módulo profesional y del alumno posible, solo se mostrará la información que pertenezca al mismo curso.

Por último se muestran el número de horas de faltas de asistencia que pertenecen al alumno, para modificarla sobre es necesario introducir un nuevo valor y pulsar sobre el botón de confirmar.

Si la modificación se ha realizado correctamente la aplicación mostrara un mensaje de información al usuario para confirmar la modificación.

The screenshot shows a Windows-style application window titled 'Registro de Asistencia 1.0'. It has a menu bar with 'Ayuda' and 'Salir'. The main area has a cyan background and is titled 'Modificar Faltas de Asistencia'. It contains a 'Volver' button with a left arrow, a dropdown for 'Identificador Módulo' (value 2), a text display 'ID: 2 - Nombre: Desarrollo de Interfazes - Siglas: DI - 2 DAM - N° Horas: 720 - Porcentaje: 30%', another dropdown for 'Identificador Alumnos' (value 3), a text display 'ID: 3 - Nombre: Santiago - Apellidos: Córdoba Ramos - Curso: 2 DAM', a text input for 'N° Horas' (value 45), and a 'Confirmar' button with a green checkmark icon.

Imprimir

Muestra un listado de faltas de asistencia de un determinado módulo profesional.

Id Módulo	Nombre Módulo	N° Horas	Porcentaje	Id Alumno	Nombre Alumno	Horas
2	Desarrollo de Interfazes DI	720	30%	3	Santiago Córdoba Ramos	45

Diseño de Pantallas.

En cuanto al diseño de pantallas atendiendo a las pautas de usabilidad, estudiadas en clase, de la aplicación comentaremos porque nuestras pantallas reúnen estas ciertas características.



En todas nuestras pantallas existe una relación entre los elementos. Nuestros elementos son los más sencillos e identificativos posibles, gracias a las imágenes utilizadas como icono de los botones que describen perfecta y adecuadamente las acciones. Todos nuestros elementos que componen nuestras pantallas cumplen perfectamente una cierta alineación, incluso proporcionando unos tamaños similares o idénticos a otros elementos del mismo tipo.

Registro de Asistencia 1.0

Ayuda Salir

Volver

Matricular Alumnado en un Módulo Profesional

Identificador Alumno 1 Curso Alumno: 1 DAM

Identificador Módulo 2 Curso Módulo: 2 DAM

Actualizar Limpiar Campo Confirmar

El color empleado para la aplicación guarda una cierta armonía y no se utilizan una serie de colores que no guarden una cierta armonía, ya que nos proporciona una visión clara de la aplicación. El estilo de fuente utilizado para la aplicación es el empleado por defecto en la aplicación y siempre es el mismo, que es Times New Roman.

6. Diseño de Base de Datos.

En esta sección se explicará detalladamente el planteamiento sobre el diseño de la base de datos de la aplicación del proyecto.

Nuestra base de datos se compone de tres tablas, que son tabla alumnos, tabla módulos, tabla faltasAsistencia. Cada una está compuesta por una serie de campos determinados, que son:

-Tabla_Alumno(Id_Alumno, Curso, Nombre, Apellidos).

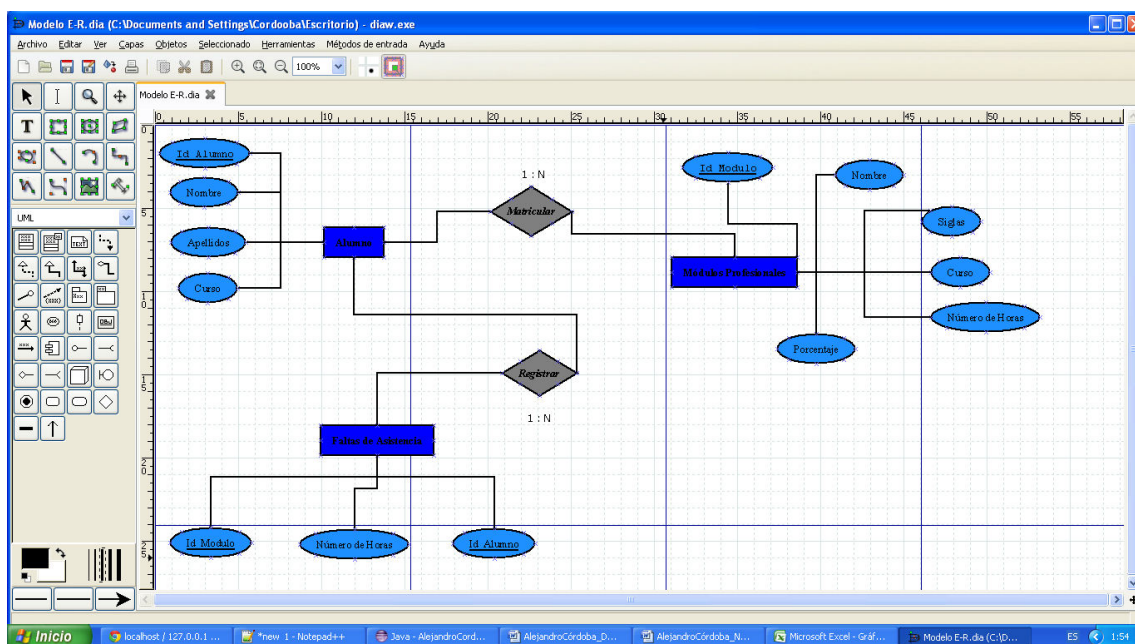
-Tabla_Modulos(Id_Modulo, Nombre, Siglas, Curso, N° horas, Porcentaje).

-Tabla_FaltasAsistencia(Id_Modulo, Id_Alumno, N° horas).

Es importante comentar el contenido de la tabla faltasAsistencia, en concreto reflejar que nos encontramos con una clave primaria compuesta, que son Id_Modulo y Id_Alumno, a su también son claves foráneas apuntando al campo identificador en su correspondiente tabla identificada con el nombre del campo.

Nos encontramos con una relación 1:N relacionada con la tabla alumnos y la tabla modulos, ya que un único alumno solo puede ser matriculado en varios módulos profesionales mientras que pertenezcan al mismo curso. También existe otra relación 1:N relacionada con la tabla alumnos y faltasAsistencia, debido a que un único alumno puede registrar varias faltas de asistencia.

A continuación se muestra el diagrama entidad-relación sobre la base de datos diseñada para la aplicación.



Una vez definido el diagrama entidad-relación el siguiente paso es la normalización de nuestra base de datos para que no se produzcan errores como redundancia, ambigüedad, perdida de restricciones de integridad, entra otros.

A continuación se comentara la normalización de la 3FN detalladamente.

Es necesario principalmente nuestra base de datos se encuentre en la 2FN y también que los atributos que no forman parte de ninguna clave no dependan de forma completa de algún atributo que no sea clave.

En la Tabla_Alumno los campos Curso, Nombre y Apellidos dependen solamente de la clave principal o primaria (Id_Alumno).

Tabla Alumnos

ID	Nombre	Apellidos	Curso
1	Alejandro	Cordoba Muñoz	2 DAM
2	Santiago	Pérez Gomez	1 SMR

En la Tabla_Modulos los campos Nombre, Siglas, Curso, N° horas, Porcentaje dependen solamente de la clave principal o primaria (Id_Modulo).

ID	Curso	Nombre	Siglas	Horas	Porcentaje
1	1 DAM	Programacion	PR	420	18
2	2 SMR	Redes	R	385	21

Tablas Módulos

En la Tabla_FaltasAsistencia el campo N° horas dependen solamente de la clave principal o primaria (Id_Alumno y Id_Modulo).

Id_Alumno	Id_Modulo	Horas
1	1	30
2	2	15

Tabla_FaltasAsistencia

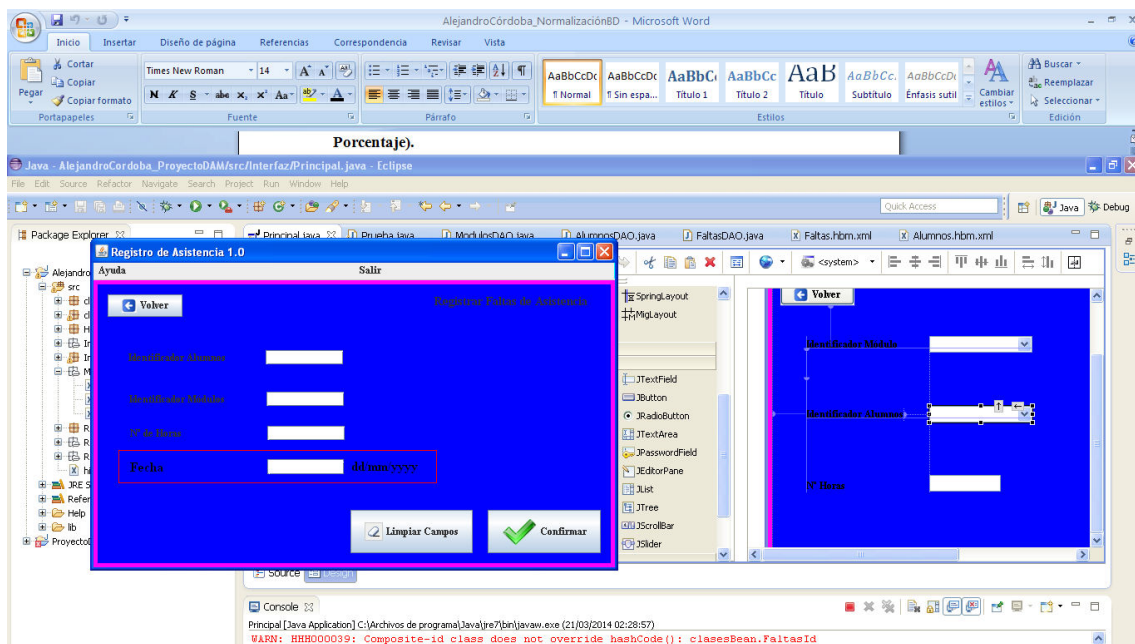
7. Propuestas de Mejora para la Aplicación.

En este apartado del documento se definen una serie de propuestas para poder mejorar la aplicación y presentar un diseño y una funcionalidad un poco más profesional.

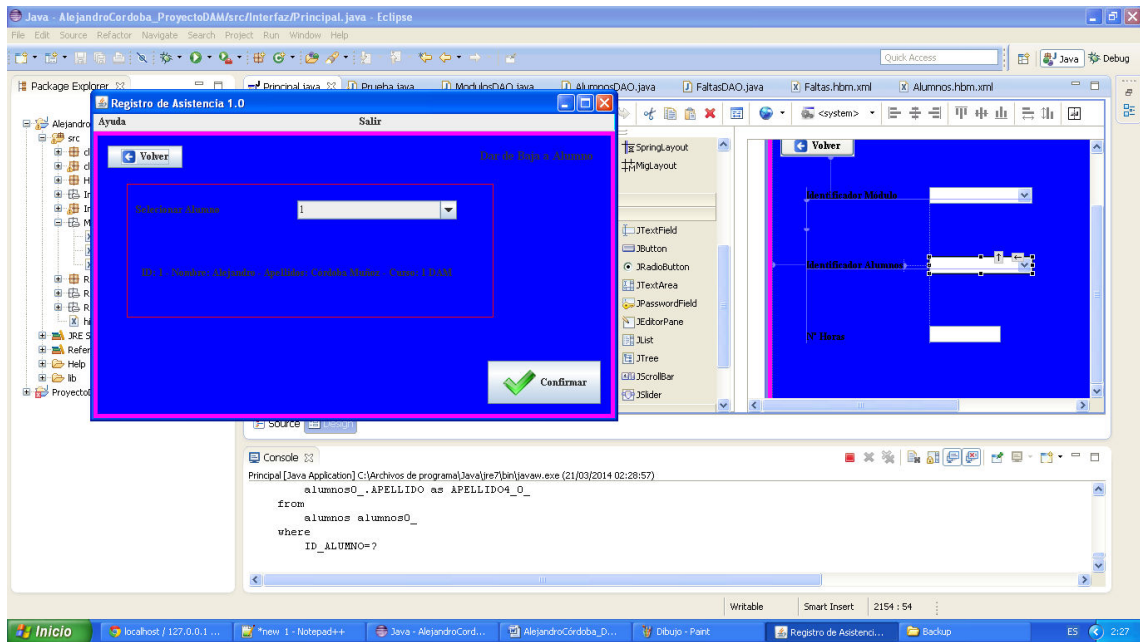
Estas mejoras han intentado ser implementadas en la aplicación, pero debido al tiempo empleado para el desarrollo de la aplicación no ha sido posible el caso de su implementación.

La primera mejora que se intento llevar a cabo fue la introducción de un campo más en la base de datos de tipo VARCHAR O DATE por ejemplo, para poder introducir la fecha del registro de una falta de asistencia de un alumno en un módulo profesional especificado, así podríamos desarrollar muchas funcionalidades más como por ejemplo implementar un informe realizado cada mes sobre el registro de faltas de asistencia de los alumnos matriculados en varios módulos profesionales o incluso poder llevar un control más correcto sobre las faltas de asistencia de los alumnos.

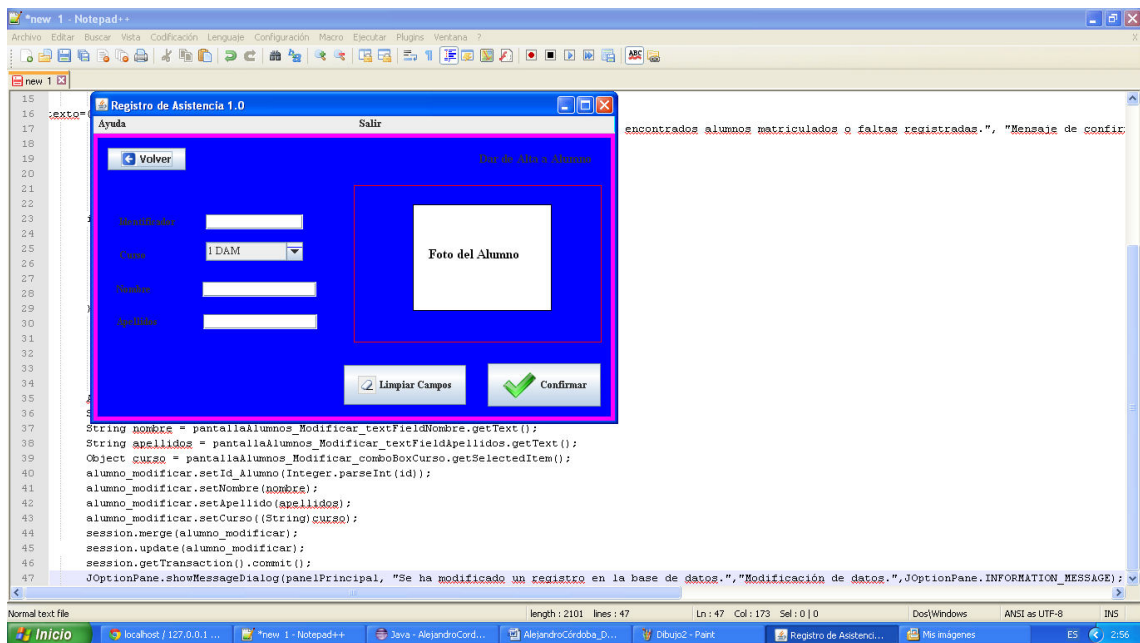
A continuación se muestra una imagen sobre una posible posibilidad de añadir la funcionalidad de registrar en nuestra aplicación una fecha en nuestra falta de asistencia.



Otra propuesta llevada a cabo es sustitución, en las pantallas de nuestra aplicación, de los elementos tipo JComboBox y JLabel para poder seleccionar y mostrar la información del alumno por un elemento tipo JTable, ya que es mucho más cómodo y visible para el usuario mostrar la información al completo y a través de su selección poder realizar la acción deseada sobre uno de los registros que compone el elemento o componente de tipo JTable.



Otra de las mejoras propuestas en la aplicación está relacionada con la administración y la gestión de los alumnos y puede convertirse muy útil en situaciones para identificar o imprimir un informe con unos datos concretos, es crear un componente para poder introducir en él una foto donde se visualice una imagen para identificar al alumno para presentar un diseño más profesional, y sobre todo para poder desarrollar e implementar otras funcionalidades, relacionadas con la aplicación, con la impresión de informes.



Al boceto presentado, se le podría añadir un botón para poder seleccionar la imagen o poder acceder a la selección de la imagen al pinchar sobre el rectángulo gris que es la posición del componente o elemento sobre el que se cargara la imagen del alumno.

8. Conclusión y Valoración Personal del Proyecto.

En este apartado del documento se explica detalladamente la conclusión final obtenida por el desarrollador del proyecto y luego la realización de una pequeña valoración personal sobre el proyecto.

Se ha llegado a la conclusión de que una vez realizado el proyecto y habiendo cumplido todos los puntos exigidos en la aplicación, se ha tenido una visión más concreta sobre lo que en realidad es una aplicación de gran envergadura en java, respecto a la estructura de sus paquetes, la implementación de la ayuda, la estructura del proyecto en general, etc. Esta visión de la elaboración de cualquier proyecto java de importancia para el desarrollador requiere el conocimiento sobre métodos, funciones o procedimientos para poder implementarlas en el código o incluso hay veces que el conocimiento no puede ir mas allá y en necesario buscar fuentes de consultas alternativas, en las cuales se aumenta más el conocimiento.

También comentar que con la implementación de una base de datos en nuestra aplicación se ha obtenido una cierta idea sobre como es el funcionamiento de casi todas las aplicaciones que utilicen o empleen bases de datos para la administración o gestión de sus datos.

El desarrollo del proyecto ha sido de gran utilidad tanto para el desarrollador como para el usuario destinado a utilizar o emplear la aplicación, ya que por una parte el desarrollador inexperto ha aprendido como es la gestión y administración de una aplicación soportando una base de datos, en el cual la aplicación utiliza los datos de la bases de datos, para esto es necesario el tratamiento de los datos con librerías específicas de java como Hibernate teniendo una cierta base, también la modificación de datos a través de componentes desarrollados con Swing, etc.

Es un proyecto muy interesante que abarca ciertas funcionalidades basadas en el tratamiento de los datos de la base de datos que a simple vista pueden resultarnos un poco fácil, pero en realidad puede resultar algo compleja.

9. Bibliografía y Fuentes de Consulta.

Maria Desirée Pérez García: Tema1.Interfaces de Usuario.

Maria Desirée Pérez García: Tema4.Usabilidad

Maria Desirée Pérez García: Tema5.Confección de Informes

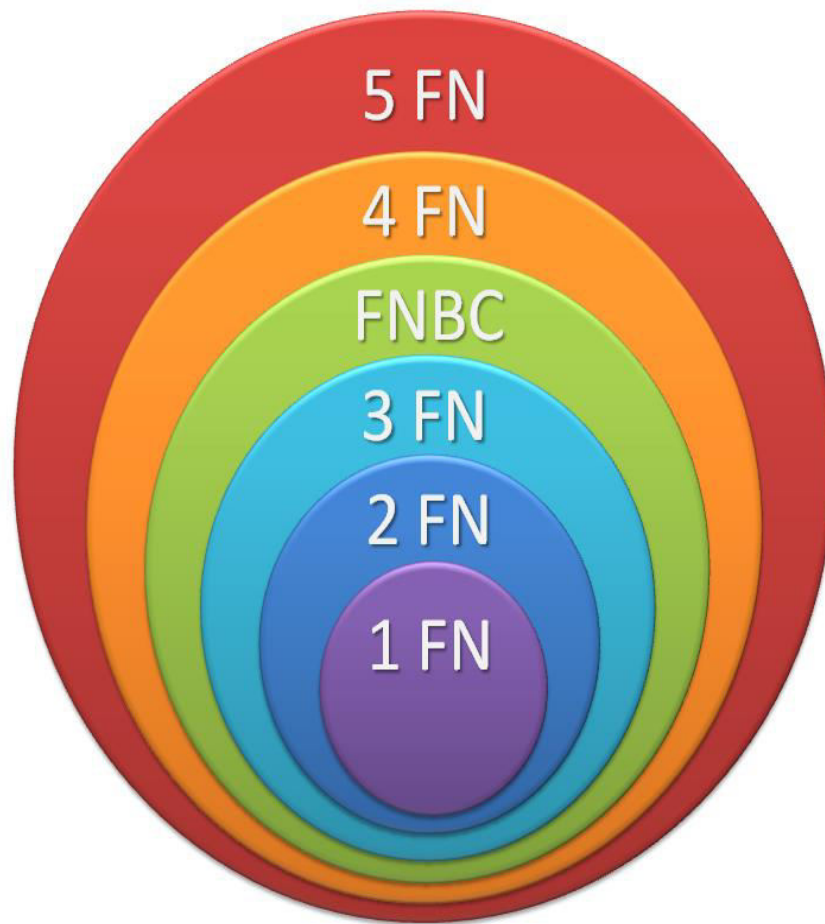
Maria Desirée Pérez García: Tema6.Documentación de Aplicaciones

M Carmen Vera Balbuena: Documentación aportada sobre Hibernate.

Curso Hibernate: <http://cursohibernate.es/doku.php>



Proyecto Desarrollo de **Aplicaciones Multiplataforma:** **Gestion de Faltas de Asistencia.**



Normalización Base de Datos (Gestión de Faltas de Asistencia)

Índice

1. <u>Introducción.</u>	<u>Pág. 3</u>
2. <u>Primera Forma Normal (1FN).</u>	<u>Pág. 4</u>
3. <u>Segunda Forma Normal (2FN).</u>	<u>Pág. 6</u>
4. <u>Tercera Forma Normal (3FN).</u>	<u>Pág. 7</u>
5. <u>Forma Normal de Boyce-Codd (FNBC).</u>	<u>Pág. 8</u>

1. Introducción.

En la siguiente documentación que a continuación procederemos con su desarrollo, podremos visualizar el proceso por el cual el resultado obtenido del modelo entidad-relación de nuestra base de datos para la aplicación de gestión de faltas de asistencia es procesado por un proceso para detectar posibles fallos en el diseño como puede ser el caso de **redundancia, ambigüedad, pérdida de restricciones de integridad, anomalías en operaciones de modificación o borrado de datos** o problemas indetectables.

Este proceso es denominado como **teoría de normalización o formas normales** desarrollado por Boyce-Cood, este proceso es desarrollado en nuestra aplicación para obtener un **diseño correcto y profesional** en nuestra base de datos.

2. Primera Forma Normal (1FN).

Para empezar nuestra base de datos se compone de tres tablas principalmente, que son Tabla_Alumno, Tabla_Modulos, Tabla_FaltasAsistencia, cuyo contenido es:

-Tabla_Alumno(Id_Alumno, Curso, Nombre, Apellidos).

-Tabla_Modulos(Id_Modulo, Nombre, Siglas, Curso, N° horas, Porcentaje).

-Tabla_FaltasAsistencia(Id_Modulo, Id_Alumno, N° horas).

Tabla Alumnos

ID	Nombre	Apellidos	Curso
1	Alejandro	Cordoba Muñoz	2 DAM
2	Santiago	Pérez Gomez	1 SMR

ID	Curso	Nombre	Siglas	Horas	Porcentaje
1	1 DAM	Programacion	PR	420	18
2	2 SMR	Redes	R	385	21

Tablas Módulos

Id_Alumno	Id_Modulo	Horas
1	1	30
2	2	15

Tabla_FaltasAsistencia

Para que nuestras tablas se encuentren en la Primera Forma Normal es necesario que todos los atributos sean atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.

Como podemos observar la información que se muestra en las tablas de nuestra base de datos, son solo de ejemplo, sirven para mostrar que los valores de nuestros campos son atómicos. Es imposible que algún campo muestre diversos valores.

3. Segunda Forma Normal (2FN).

Para que nuestras tablas se encuentren en la 2FN, es necesario principalmente que se encuentre en la 1FN y también si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal.

Es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal).

Como podemos observar en nuestras tablas, todos los campos que la componen dependen únicamente y solamente de la clave primaria o también denominada como primary key, no existe ningún campo que no cumpla esta restricción.

En la Tabla_Alumno los campos Curso, Nombre y Apellidos depende únicamente de la clave principal o primaria (Id_Alumno).

-Tabla_Alumno(Id_Alumno, Curso, Nombre, Apellidos).

En la Tabla_Modulos los campos Nombre, Siglas, Curso, N° horas, Porcentaje depende únicamente de la clave principal o primaria (Id_Modulo).

-Tabla_Modulos(Id_Modulo, Nombre, Siglas, Curso, N° horas, Porcentaje).

En la Tabla_FaltasAsistencia el campo N° horas depende únicamente de la clave principal o primaria (Id_Alumno y Id_Modulo).

-Tabla_FaltasAsistencia(Id_Modulo, Id_Alumno, N° horas).

4. Tercera Forma Normal (3FN).

Para que nuestras tablas se encuentren en la 3FN, es necesario principalmente que se encuentre en la 2FN y también que los atributos que no forman parte de ninguna clave no dependan de forma completa de algún atributo que no sea clave.

Como podemos observar en nuestras tablas, todos los campos o atributos que la componen dependen únicamente y solamente de la clave primaria o principal también denominada como primary key, no existe ningún campo que no dependa de algún otro atributo que no sea clave primaria.

En la Tabla_Alumno los campos Curso, Nombre y Apellidos dependen solamente de la clave principal o primaria (Id_Alumno).

-Tabla_Alumno(Id_Alumno, Curso, Nombre, Apellidos).

En la Tabla_Modulos los campos Nombre, Siglas, Curso, N° horas, Porcentaje dependen solamente de la clave principal o primaria (Id_Modulo).

-Tabla_Modulos(Id_Modulo, Nombre, Siglas, Curso, N° horas, Porcentaje).

En la Tabla_FaltasAsistencia el campo N° horas dependen solamente de la clave principal o primaria (Id_Alumno y Id_Modulo).

-Tabla_FaltasAsistencia(Id_Modulo, Id_Alumno, N° horas).

5. Forma Normal de Boyce-Codd (FNBC).

Es muy raro este caso ya que si nuestras tablas se encuentran o reúnen las especificaciones de la 3FN cumplen también la FNBC, aunque es posible que pueda darse el caso de que no la cumpla.

Una tabla se encuentra en la FNBC si cada determinante, atributo que determina completamente a otro, es clave candidata.

En nuestro caso podríamos decir que nuestras tablas, cumple la FNBC debido a que todo campo del que dependa otro es una clave candidata, y como se puede apreciar se cumple.

-Tabla_Alumno(Id_Alumno, Curso, Nombre, Apellidos).

-Tabla_Modulos(Id_Modulo, Nombre, Siglas, Curso, N° horas, Porcentaje).

-Tabla_FaltasAsistencia(Id_Modulo, Id_Alumno, N° horas).