CURSO ACADÉMICO 2021-2022

**FAMILIA INFORMÁTICA Y COMUNICACIONES**

CICLO FORMATIVO GRADO SUPERIOR:

**DESARROLLO DE APLICACIONES**

**MULTIPLATAFORMA (DAM)**

**PROYECTO DE DESARROLLO DE APLICACIONES**

**MULTIPLATAFORMA**

**RECURSOS HUMANOS APLICACIÓN.**

AUTOR/ES: AYOUB AGHMIR

CARLOS SALOMÓN MANGUE MBA AVOMO

JOSE PLATA ALMEIRA

TUTOR/A: ROBERTO SIMAL MARTÍNEZ

Fecha de entrega: 30-6 junio 2022

ÍNDICE

1.Introducción.................................................................................................................3

1.1 Sobre la aplicación de recursos humanos…..............................................................3

1.2 Objetivos en la aplicación...........4

2. English Introduction................4

2.1.About the application of human resources ...5

2.2 Objectives in the application....5

3. Faseas y tareas (Diseño y desarrollo de interfaces)...

4. Programación del proyecto.

4.1. Diseño de la base de datos.

4.2. Diseño y estructura de la aplicación RRHH.

4.3 Los métodos y atributos implementados en java.

4.3.1 Recursos humanos app.

4.3.2 Vistas (Nuevo trabajador, Listado trabajador, Menú)

4.3.3 Base de datos (Conexión)…........

4.3.4 Entidades (Trabajador)….....

4.3.5 DAO (Trabajador DAO)…

4.3.6 Internacionalización (Principal)…...

5. Fututos posibles trabajos.

6. Conclusión

7. Opiniones personales.

8. Bibliografía.

**1. INTRODUCCIÓN**

**1.1 SOBRE LA APLICACIÓN DE RECURSOS HUMANOS**

La aplicación de escritorio para recursos humanos es más bien un sistema de control de personal en java o un sistema de gestión de empleados java con base de datos hechos en java eclipse y XAMPP MySQL, que este es un programa básico para controlar al personal de una empresa como su sueldo, fecha de ingreso y otras funcionalidades que se pueden agregar.

Es una aplicación en Java los tamaños que desean modernizar sus servicios y mejorar la productividad del personal. Este software se puede utilizar tanto en la web como en el host local. Esta aplicación es una solución digital para gestionar el personal a distancia con todas las ventajas y funcionalidades de un software de recursos humanos, tal y como lo hemos mencionado arriba. Una herramienta fácil e intuitiva que extiende las características esenciales para gestionar un departamento de RRHH, que puede ser desde el móvil o tableta, pero en nuestro caso es desde ordenador, es decir, que tenemos un programa ya preparado para poder hacer estas funcionalidades. Una de las ventajas de nuestra app es la mejora de la coordinación de los equipos, como veremos en la aplicación, la rapidez y fluidez de la información es esencial para la buena coordinación.

Veremos que en nuestra app tenemos unas herramientas esenciales, como la suprimir y poder consultar por un trabajador, estas funcionalidades nos ayudan, a fin de que no perdamos datos y poder almacenar lo que nosotros registramos; Esta app nos facilita la organización de tareas, responsabilidades, y otra cosa es que esta app tiene la virtud de adaptarse a los tiempos que corren y estará o puede estar en continua actualización, en el que podremos agregar nuevas funcionalidades, para que se adapte a las necesidades que vayan surgiendo para mejorar las tareas que se vayan a implementar en nuestra aplicación de cara al futuro, ya que nuestra aplicación es para gestión de personal, se necesitará muchísimo.

**2. ENGLISH INTRODUCCIÓN.**

**2.1. ABOUT THE APPLICATION OF HUMAN RESOURCES**

The desktop application for human resources is more of a java personnel control system or a java employee management system with a database made in java eclipse and XAMPP MySQL, that this is a basic program to control the personnel of a company such as salary, date of entry and others features that can be added.

It is a Java application for sizes that want to modernize their services and improve staff productivity. This software can be used both on the web and on the local host. This application is a digital solution to manage remote personnel with all the advantages and functionalities of a human resources software, as we have mentioned above. An easy and intuitive tool that extends the essential features to manage an HR department, which can be from a mobile or tablet, but in our case it is from a computer, that is, we have a program already prepared to be able to carry out these functions. One of the advantages of our app is the improvement of the coordination of the teams, as we will see in the application, the speed and fluidity of the information is essential for good coordination.

We will see that in our app we have some essential tools, such as deleting and being able to consult by a worker, these functionalities help us, so that we do not lose data and be able to store what we register; This app makes it easier for us to organize tasks, responsibilities, and another thing is that this app has the virtue of adapting to current times and will be or may be continuously updated, in which we can add new features, so that it adapts to the needs that arise to improve the tasks that are going to be implemented in our application for the future, since our application is for personnel management, a lot will be needed.

**1.2 Objetivos en la aplicación.**

Creación de una aplicación de escritorio para la empresa “x” con el objetivo del uso y gestión de una base de datos.

La cual permitirá el almacenamiento persistente, así como la gestión de sus datos mediante operaciones *CRUD*.

El sistema se utilizará para la gestión en los recursos humanos de la empresa “x” especializada en “x”.

**2.2 Objectives in the application.**

Creation of a desktop application for company "x" for the purpose of using and managing a database.

Which will allow persistent storage as well as the management of your data through CRUD operations.

The system will be used for the human resources management of the company "x" specialized in "x".

**3. Fases y tareas (Diseño y desarrollo de interfaces)**

**Fases:**

**1** Definición de procesos a seguir y tecnologías a utilizar para el desarrollo de la aplicación.

**2** Creación de la BBDD y diseño de los campos y tipo de datos de los mismos.

**3** Diseño de la interfaz gráfica que facilite el manejo de la base de datos y sus tablas para la obtención de la información requerida.

**4** Creación de la clase de conexión la cual nos permitirá conectarnos con la base de datos MySQL.

**5** Construcción de una clase *DAO* (*Data access object)* la cual nos dará acceso a la base de datos desde nuestro lenguaje java para su desarrollo (proceso de “mapeo”)

**6** Unión de la clase DAO con la interfaz gráfica para el uso real de formularios como listados y búsquedas.

**7** Creación de validaciones.

**8** Internacionalización del aplicativo.

**Tareas:**

**1** Diseño y desarrollo de interfaces que permitan al usuario interactuar con la base de datos.

**2** Creación de la base de datos e implementación del ORM para el uso de la base de datos.

**3** Internacionalización del sistema.

En primer lugar, hemos decidido crear una aplicación de recursos humanos, con el que empezaremos creando un paquete llamado **“recursoshumanosapp”**, es donde crearemos su clase que tendrá el mismo nombre que el paquete, que será la capa visual del proyecto, que también tendrá una conexión con la DAO.

En segundo lugar, crearemos la clase conexión, que proviene del paquete **bd, lugar** para realizar la clase de conexión, tendremos que crear una clase llamada **“Conexión”**, que será con el programa Eclipse por que será el programa que usaremos, esa clase de conexión se ocupará de tener todas las credenciales para poder entrar la IP también en donde queda el servidor de base de datos y poder conectarse a ella. Primero en esta clase tendremos que importar unas clases y que también nos ayudarán para hacer la conexión.

**3.1. INTERFAZ GRÁFICA**

Aquí es donde partiremos para crear nuestra querida aplicación de recursos humanos, en el que se incluyen la clase de recursos humanos, la conexión que hace referencia a la BBDD, la de **TrabajadorDAO,** Trabajador en entidades, la de vistas que contiene doble clase y por último la de Internacionalización con la clase llamada Principal.

**3.2 VARIABLE PARA LA GESTIÓN DE LA CONEXIÓN**

A parte de todo esto tenemos que saber muy a qué base de datos nos vamos a conectar, ya que el código nos puede servir para diferentes bases de datos, y nos conectaremos desde MySQL.

Entrando en código, necesitaremos poner estas variables:

private Connection conexión

private String host = “localhost:3306”

private String usuario = “root” para poder entrar al phpMyAdmin.

private String password = “”; no agregaremos contraseña, lo dejamos vacío.

private String nombreBD = “recursoshumanos”; nombre de la BBDD.

private String url = “jdbc: mysql:// “+ host + ”/ ” + bd ;

**3.2 CREACIÓN DE MÉTODOS PARA ABRIR Y CERRAR LA CLASE**

Tendremos un método público que va a retornar la conexión y la conexión será de tipo Connection: **public Connection conectar ()**, que normalmente es la instancia de conexión.

Necesitaremos una herramienta que nos permita los errores, porque la conexión a veces puede fallar, la herramienta será el **try catch**, donde contendrá el **Class.forName** donde pondremos la ruta de la clase que definirá MySQL para el paquete de driver que tiene la siguiente estructura: “**com.mysql.jdbc.Driver”** que el acompañara una instancia y donde se almacenará la conexión será el **this.conexión = DriverManager.getConnection();** Ésta será la clase de conexión.

**3.3 BBDD Y CLASE CONEXIÓN**

Aquí es donde comenzaremos con la bd, pero antes de hacer eso, debemos tener una base de datos, que lo podemos visualizar de varias formas, con **XAMPP, phpMyAdmin,**etc; cuando tengamos eso, seremos capaces de poder acceder a esas plataformas mediante una interfaz gráfica, tendremos que iniciar el programa que usemos o vayamos a tener para poder importar un archivo que contendrá varios registros de trabajadores.

**3.4 CLASE DAO**

Ahora tendremos que hacer la parte de entidades y, para tal efecto vamos a partir primero desde las entidades y para eso tendremos que crear otro paquete, que se llamará **entidades** y que contendrá la clase **Trabajador** que tendrá variables y constructores.

Private String DNI.

Private String Nombre.

Private String Apellido.

Private int Edad.

Private String Cargo.

Private boolean Sexo.

Private boolean Extranjero;

Tendremos que hacer un constructor sin parámetros de Trabajador y luego un constructor de Trabajador con todos los parámetros y luego agregar los métodos **Getter y Setter,** y por último el **toString ()**.

Luego crearemos un paquete que se llamará **DAO** y contendrá la clase **TrabajadorDAO,** que contendrá todas las consultas que tengan que ver con la tabla trabajador.  
   
Antes no lo dije, pero también tendremos que haber creado un paquete llamado recursoshumanosapp y que contendrá la clase **RecursosHumanoaApp(main),** donde desde entidades vamos a importar el trabajador, desde DAO vamos a importar el TrabajadorDAO y luego crearemos un trabajador con datos.

Lo que tendremos que hacer ahora es volver a la ventana y programar algunos botones, como el de guardar y el de limpiar o borrar, en el que importaremos el DAO (TrabajadorDAO), el de entidades. Trabajador en el que vamos a recoger los datos. Normalmente aquí es unir el

DAO con la interfaz gráfica, de manera que el formulario tenga una funcionalidad real y aparte de eso, la clase DAO es que el contendrá las consultas SQL, esas dos clases nos permitirán hacer la conexión final con BD pasando por las consultas.

**3.5 UNIR INTERFAZ CON DAO**

Aquí tendremos que unir el DAO con la interfaz gráfica, y con eso tendremos que volver a la ventana de vistas, donde vamos a reprogramar algunos botones ya puestos, como en el caso del botón **“Guardar”** y también hacer la unión con el DAO, importando en esa clase (import entidades. Trabajador e import DAO. TrabajadorDAO), será muy necesario y también con la bbdd.

**3.6 LISTADO Y BÚSQUEDA**

Aquí daremos paso a lo que es el listado de los datos, para que se guarden y que se vieran en la bbdd, y que el usuario lo pueda ver, por eso tendremos que crear una clase JFrame, es decir, otra ventana más que se llamará “Listado trabajadores", desde el paquete vistas, luego procederemos a generar cómo se va a ver la ventana del listado primero.

Tendremos que crear una tabla, aquí es listar datos en una tabla, además indagaremos en él cómo se puede realizar un filtro por la clave primaria de la tabla y que también tendremos que acudir con la clase TrabajadorDAO, en que nos devolverá todos los trabajadores con un método **ArrayList.**

**3.6 ELIMINAR REGISTROS**

En este capítulo veremos cómo eliminar, que es una de las cosas más fáciles, y tendremos que volver a la clase DAO, que desde podremos crear un método para que pueda eliminar al trabajador y crear un **boolean** para saber si el trabajador fue eliminado o no y obtener la conexión, que nos ayudará a conectar con la bbdd y poder así suprimir al trabajador elegido y agregando varias cosas y datos del trabajador.

**3.7 VALIDACIÓN Y MENÚ**

En éste penúltimo proceso veremos cómo hacer validaciones y crear un menú. Las validaciones tienen que estar hechas en la entidad correspondiente a la clase Trabajador, concretamente en los **sets** de las variables que hayan creado en dicha clase, en este punto veremos que, al agregar caracteres, habrá ciertas condiciones que cumplir que se hayan establecido en el código, me refiero a las validaciones, por ejemplo, podemos cambiar la condición que los menores de 18 no pueden ingresar para poder trabajar, eso estaría detrás de la ventana de la clase **NuevoTrabajador.** Y en cuanto al menú que también es creada con una clase JFrame, donde la ventana contendrá unos ítems para que se pueda visualizar lo que se quiere ver, por ejemplo, el de listar y agregar, pincharíamos en uno de los dos y nos mostraría una venta de listar.

**3.8 INTERNACIONALIZACIÓN**

Llegamos al punto final donde agregaremos lo que es la internacionalización que en java consiste en hacer nuestro código de forma de que funcione correctamente en varios idiomas, en nuestro caso hemos añadido solo el castellano y el inglés;

Aquí crearemos una clase JFrame y con esto lo que vamos a conseguir es que cuando el idioma del sistema operativo, el idioma de la configuración regional de un ordenador esté por ejemplo en castellano, francés, ruso, y con esto pues que los textos aparezcan en el idioma correcto; La clase creada se llama **“Principal”,** que proviene del paquete internacionalización, y en ello tendremos que agregar algunos controles.

**4.Programación del proyecto**.

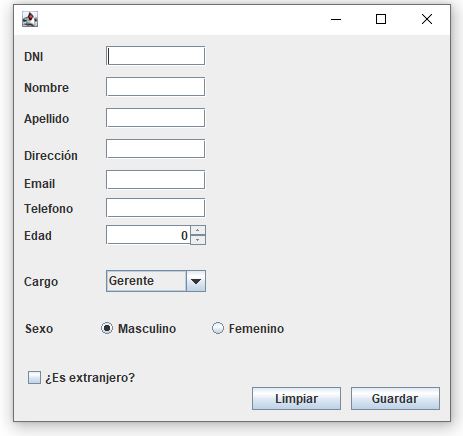
**4.1. Diseño de la base de datos**

**4.2 Diseño y estructura de la aplicación RRHH.**

**1. Contenido 1: Selección entre librería java Swing y AWT**

Hemos elegido esta opción, porque con ello nos será más fácil poder programar con ello, para poder crear la aplicación, esta sería la parte fundamenta, que se filtre la parte principal de la aplicación, que es donde recogemos los datos de los trabajadores.

**Registro**



Nuestras interfaces gráficas se componen de cajas de texto, labels, desplegables, botones, etc. El tipo de programación que se utiliza es programación basada en eventos, Vista para el listado de datos del CRUD.

Usamos un **GroupLayout** y un **JScrollPane** que nos permite hacer scroll. Después añadimos una **JTable** y la configuramos a nuestro gusto. Lo más importante darle el número de filas y columnas que vayamos a utilizar, esto lo podremos hacer mediante la función “model”.

**Arquitectura General.**

**USUARIO → INTERFAZ GRÁFICA → DAO(sql) → CONEXION → BBDD**

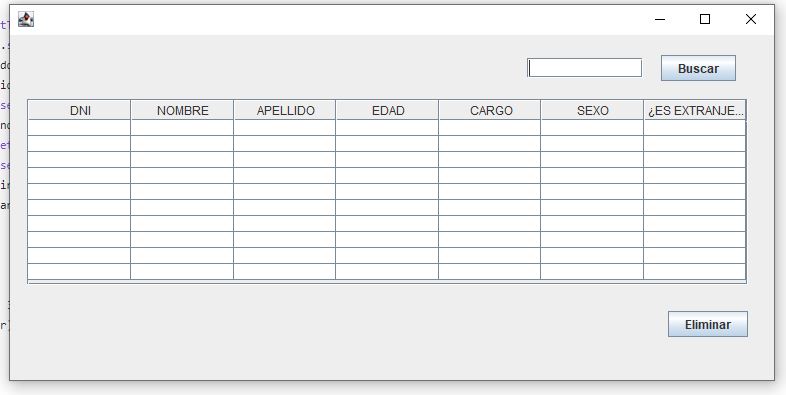
**Código para los controles gráficos.**

Cada control que usaremos tendrá un código, de esta manera no habrá colisión con ninguna variable que tengamos y son fácilmente identificables.

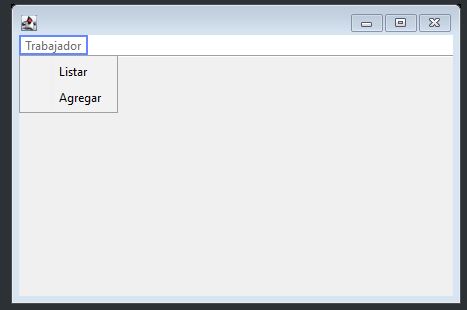
|  |  |
| --- | --- |
| **Control** | **Código** |

|  |  |
| --- | --- |
| TextField | txt |
| Label | lbl |
| ComboBox | cbo |
| CheckBox | chk |
| Spinner | Spn |
| Tabla | Tbl |
| RadioButton | rbt |

**Listado**



**Menú**



**Internacionalización**

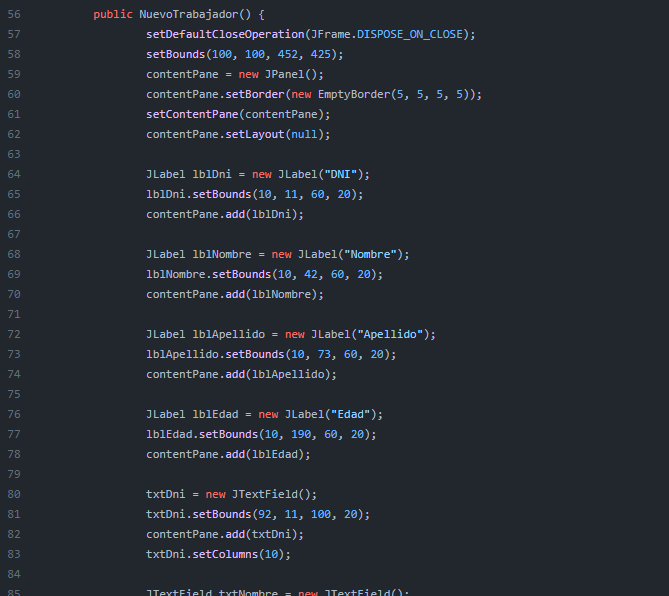
**4.3. Los métodos y atributos implementados en java.**

**4.3.1 Recursos humanos app**



**4.3.2 Vistas (Nuevo trabajador, Listado trabajador y Menú)**

**Nuevo Trabajador**

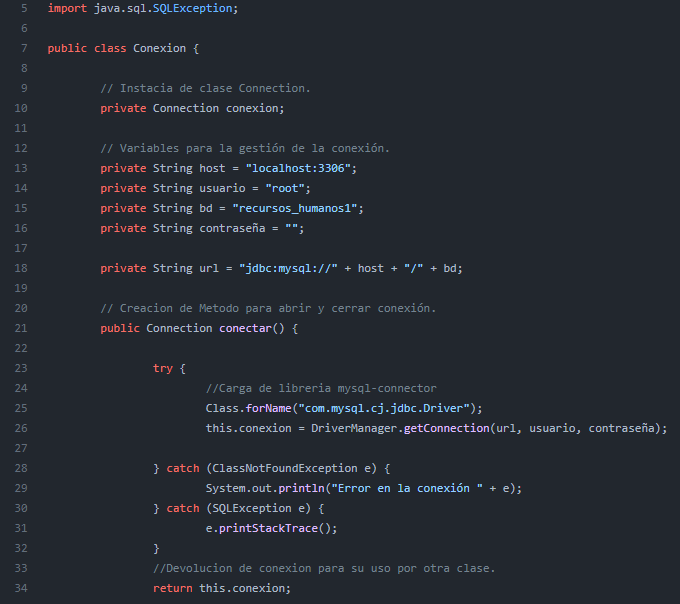


**Listado Trabajadores**

**Menú**



**4.3.3 Base de datos (Conexión)**



**4.3.4 Entidades (Trabajador)**



**4.3.5 DAO (Trabajador DAO)**



**4.3.6 Internacionalización (Principal)**

**Métodos:**

1. Public Connection conectar: Hará la conexión.
2. Public boolean agregar Trabajador: Se guardará aquí el trabajador añadido.
3. Public ArrayList<Trabajador> listadoTrabajadores: Devuelve una lista de trabajadores.
4. Public ArrayList<Trabajador> buscarPorDni: Devuelve la lista DNI.
5. Public void actionPerformed: método que va a ejecutar el código una vez que pulsemos el botón.
6. String []idiomas={“es\_Es”,”us\_Us”}: Agregaran los idiomas a la app.

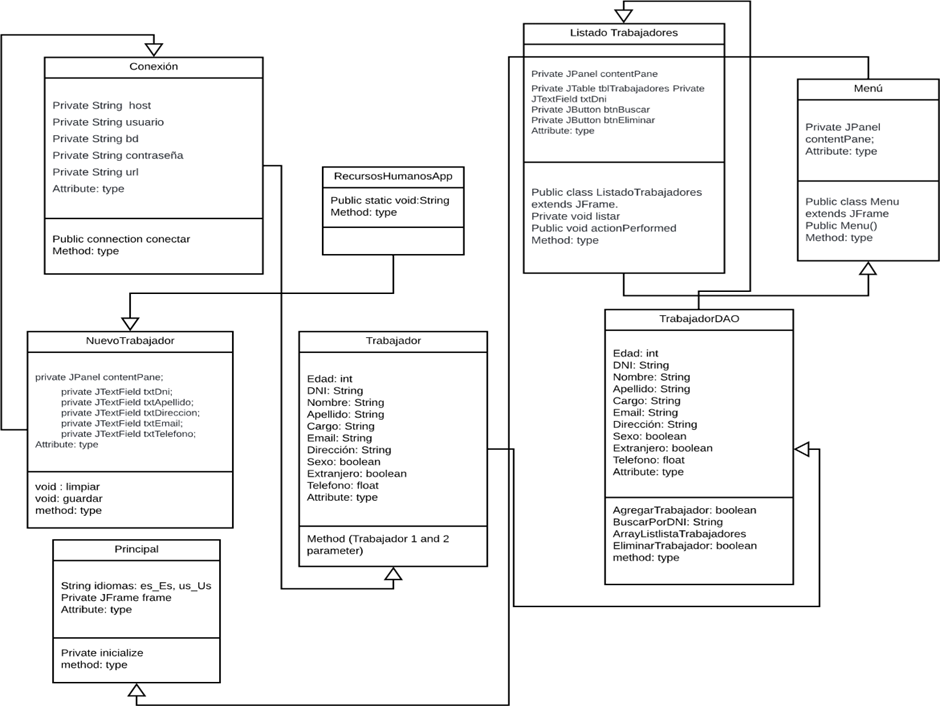
Estos son los métodos que más o menos hemos utilizado ejecutar la app y en cuanto a los atributos o variables hay varios:

1. Private String DNI, nombre, apellido, cargo.
2. Private int edad.
3. Private boolean sexo, extranjero.
4. Private String host, usuario, BD, contraseña.

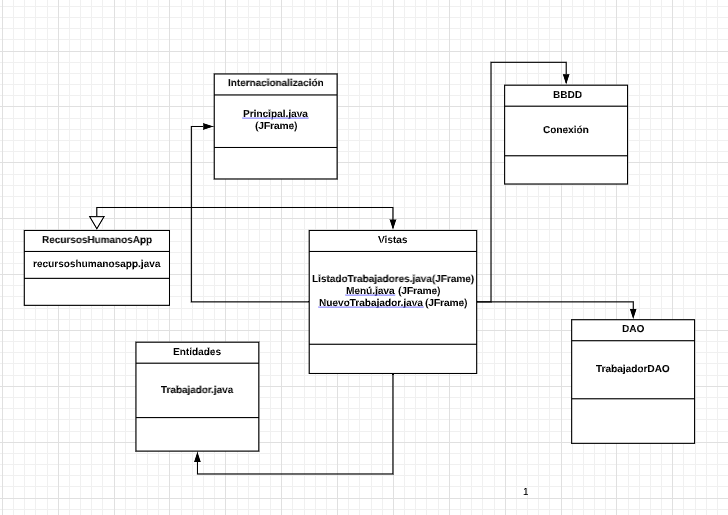
En el JFrame tenemos los atributos:

1. Private JPanel contentPane.
2. Private JTextField txtDni.
3. Private JTextField txtApellido.

Que son los atributos que tenemos detrás del código de la ventana principal. Éstos últimos atributos vienen por defecto según vayas creando la ventana te irán saliendo por el código. Ahora mostraremos algunas imágenes de las clases, métodos y atributos que hemos añadido para nuestra aplicación de recursos humanos (RRHH).



**5. Esquema de las clases de la aplicación RRHH.**



**6. Futuros posibles trabajos**

Una vez finalizado el proyecto, en la aplicación de recursos humanos, en un futuro “x” vamos a querer implementar otras nuevas herramientas, la idea final es poder llevar el proyecto al siguiente nivel, a la vida real. Por ello queremos continuar desarrollando nuestra queridísima aplicación otorgándole a ésta una mayor funcionalidad y optimizarla aún mejor para ofrecer al cliente un sinfín de posibilidades a la hora de poder utilizar nuestra app.

**6. Conclusión**

Una vez finalizado el proyecto es momento de analizarlo y mirarlo con perspectiva e identificar aciertos y errores.

Todo empezó con una idea, hacer una aplicación de escritorio de algo que realmente nos gustara, así todo sería más sencillo. Pronto vimos que técnicamente habíamos subestimado el proyecto, a pesar de lo atractivo de la idea para nosotros, iba a costar, pero con el tiempo vimos que la cosa era muy fácil de procesar. Primero tuvimos que compaginarnos y repartir tareas, aunque fue un poco difícil, ya que por tema de horarios parecía complicado encontrar ese punto de encuentro para trabajar. Al final lo conseguimos y todo funcionó de una mejor manera, aunque en el último momento le dimo más vueltas al proyecto. Llegamos a organizarnos de tal forma que no siempre fue necesario trabajar los dos juntos a la vez. En términos organizativos supimos revertir una situación complicada y eso ha sido un gran aprendizaje, ya que saber medir los tiempos es algo fundamental, porque sin ello nos costará trabajar ejercer nuestras funciones.

Técnica y temporalmente hablando, pensamos que todo era bastante más asequible de lo que después ha sido, debido a nuestra falta de experiencia a la hora de realizar un proyecto así, ya que es nuestra primera que realizamos un proyecto de fin de grado. En cuanto al código más bien nos referimos a que hemos sabido adaptarnos y aprender cómo debe ser por dentro una aplicación de este tipo, utilizando el SWING (biblioteca gráfica) que con ello hemo logrado crear ventanas sobre los cuales hemos añadido distintos objetos con los que se podrá interactuar o no con el usuario. y también adaptarnos y buscar soluciones en ocasiones quizás por encima de los conocimientos que ya habíamos adquirido durante el ciclo formativo, algo normal a la hora de realizar un proyecto por otra parte. Por otra parte, no hemos podido desarrollar más allá nuestros conocimientos sobre el SWING, con la aplicación esta hemos adquirido mucha información de la que desconocíamos, porque hemos utilizado más el término JFrame y el JPanel que el SWING, porque la diferencia está en que las dos primeras poseen algunas nociones típicas de una ventana como minimizar, cerrar, maximizar y poder moverla

**7. Opinión personal**

**Carlos Salomón Mangue Mba Avomo**

La creación de la aplicación de recursos humanos en Java ha sido una experiencia muy positiva para mí, porque tanto como para mí como para mis compañeros ha sido el primer contacto que tenemos en un entorno igual y en este sentido hemos comprobado lo complicado que es entrar en un mundo que desconocemos y tener que buscarnos la vida para desarrollar este proyecto. Además, hemos podido entender la importancia de la gran mayoría de las asignaturas que hemos cursado durante estos dos años.

Es más, tras haber hecho este trabajo tanto mis compañeros como yo hemos ampliado de forma muy notable nuestros conocimientos sobre algunos de los campos más importantes de la informática, al haber realizado el Grado Superior de Desarrollo de Aplicaciones Multiplataforma, que hemos culminado bien este curso.

En este sentido me gustaría agradecer a mis compañeros que desde el primer momento han estado siempre conmigo para poder sacar este proyecto adelante, que sin ellos, nos hubiéramos llegado a este paso, que es crear este proyecto basado en una aplicación de escritorio.

**Jose Plata Almeira**

**8.Bibliografía.**