HACK X CRACK

Java:bases + sql

By VaNcHoXcHk

hack crack
WWW.HACKXCRACK.ES

JAVA FOR HACK X CRACK

Que es Java?

Lenguaje de programación nacido en 1991 su padre *SUN MICROSYSTEMS,* Empresa la cual ha sido la promotora de este desde el principio de los tiempos.

Debido al avance en las tecnologías de computadoras, versatilidad en los sistemas y diferencias entre los mismos Dicha empresa desarrolla un código "neutral" el cual puede correr en diferentes tipos de plataformas sin importar la maquina y el tipo de sistema que maneje, corriendo las aplicaciones en una "maquina virtual" *JAVA VIRTUAL MACHINE*, La función en particular de esta JVM es convertir el código java al formato i/o lenguaje correcto de la maquina anfitriona haciendo de este un lenguaje apreciado hoy día por muchos programadores gracias a su multiplataforma.

Un breve repaso

A continuación algunas características las cuales hacen de java marcar la diferencia entre muchos lenguajes de programación:

SIMPLICIDAD:

Que sea simple no quiere decir que sea menos potente que otros lenguajes de programación no!
Solo que java posee las funciones de un lenguaje potente y robusto, eliminando funciones muy poco usadas y confusas al usuario.

DISTRIBUIDO:

En cuanto a su distribución aplica al manejo de librerías que este incorpora (java) librerías para el manejo de conexiones TCP/IP y protocolos como http y ftp para acceder a bases de datos, ficheros y funciones que se encuentren en ordenadores remotos.

ROBUSTICIDAD

Lo que lleva a java a ser un lenguaje robusto directamente es su protección al programador en cuanto a eventuales errores de programación, verificando los mismos en tiempo de compilación como en tiempo de ejecución.

NEUTRALIDAD

Su neutralidad se basa en la generación de un código objeto el cual es independiente de la arquitectura de cualquier maquina, el fichero generado por un compilador de java actualmente tiene soporte para sistemas Solaris, Linux, Windows, Mac, Apple Etc.

SEGURIDAD

El código Java pasa muchos *tests* antes de ejecutarse en una máquina. El código se pasa a través de un verificador de byte-codes que comprueba el formato de los fragmentos de código y aplica un probador de teoremas para detectar fragmentos de código ilegal -código que falsea punteros, viola derechos de acceso sobre objetos o intenta cambiar el tipo o clase de un objeto-.

Si los byte-codes pasan la verificación sin generar ningún mensaje de error, entonces sabemos que:

- **El código no produce desbordamiento de operandos en pilas**
- **El tipo de los parámetros de todos los códigos de operación son conocidos y correctos**
- **No ha ocurrido ninguna conversión ilegal de datos, tal como convertir enteros en punteros**
- **El acceso a los campos de un objeto se sabe que es legal: public, private, protected**
- **No hay ningún intento de violar las reglas de acceso y seguridad establecidas**

PORTABILIDAD

La portabilidad va desde su poder de ejecución en múltiples plataformas, hasta facilitar un entorno desarrollado idéntico para todos los usuarios en diversas maquinas.

INTERPRETACION

Sistema Run-Time es el interprete de java encargado de la ejecución del código objeto comúnmente el interprete consume menos recursos que al compilar el code por lo que nosotros como desarrolladores java tendremos más tiempo para crear, al ser java un lenguaje interpretado lo hace más lento que otros lenguajes de programación compilados ya que este (java) debe ser interpretado y no compilado ya que no cuenta hasta ahora con un compilador propio para cada plataforma.

ORIENTADO A OBJETOS

Soporta los tres paradigmas fundamentales para la orientación a objetos:
Encapsulación, herencia y polimorfismo.

ENTORNOS DE DESARROLLO

Existen diversos programas para el desarrollo de código java estos son los muy populares IDE "INTEGRATED DEVELOPMENT ENVIRONMENT" la compañía SUN nos ofrece gratuitamente el kit para poder desarrollar java "JDK" JAVA DEVELOPMENT KIT dotado de una serie de librerías y programas para desarrollar, compilar y ejecutar en java y los IDE anteriormente mencionados como siempre lo he dicho nos facilitan la vida =) en un mismo programa podrás ejecutar y compilar en java.

Actualmente hay muchos IDE disponibles para java entre muchos tenemos, entre otros:

- **NETBEANS**
- **ECLIPSE**
- **JDEVELOPER**
- **JBUILDER**
- **VISUAL J++**

EMPECEMOS: COMPILAR Y EJECUTAR TU PRIMER PROGRAMA EN JAVA

Primero que todo nos Instalamos JDK(Java Development Kit) "Descarga"

http://www.4shared.com/file/MkCwFiMb/jdk-6u20-windows-i586.html

Después de Instalado JDK Procedemos a su configuración

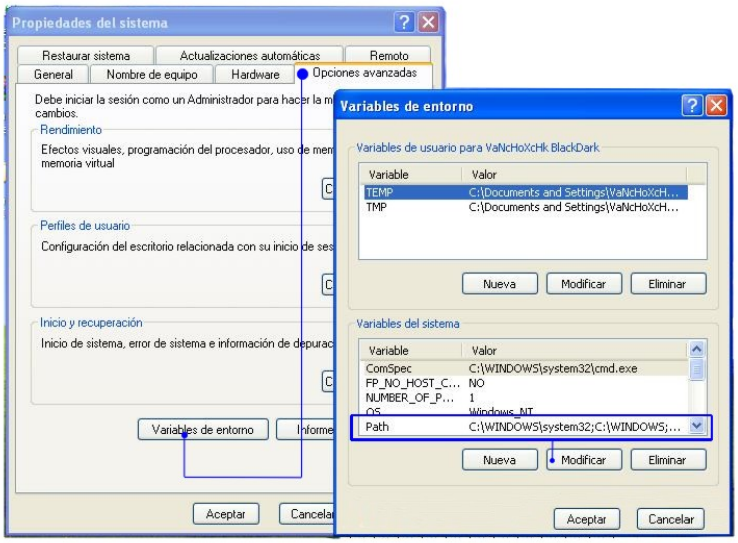
Nos Ubicamos en el directorio donde este se instala Por defecto es (C:\Archivos de Programa\Java) tal vez tengas otras versiones instaladas seleccionas la más reciente que tengas instalada lo más probable es que sea la que acabamos de instalar xDD bue.. Abrimos la carpeta "bin" y copiamos la ruta

Ejemplo: C:\Program Files\Java\jdk1.6.0_20\bin

Configuramos el PATH

Nos Ubicamos en "MI PC" damos Clic derecho seleccionamos propiedades, picamos en la pestaña Opciones avanzadas, luego en variables de entorno, Seleccionar Path y pincha en modificar, en los valores si la cadena no termina con Punto y coma " ; " se lo pones y a continuación pegas la ruta que copiaste que hace referencia a los binarios de JAVA, Cerramos la cadena con punto y coma ; Clic en OK y Aceptar Creo que no está de más decir que no tienes que borrar nada =D

Una Imagen..



Abrimos la SHELL!! O CMD o la Pantallita negra..

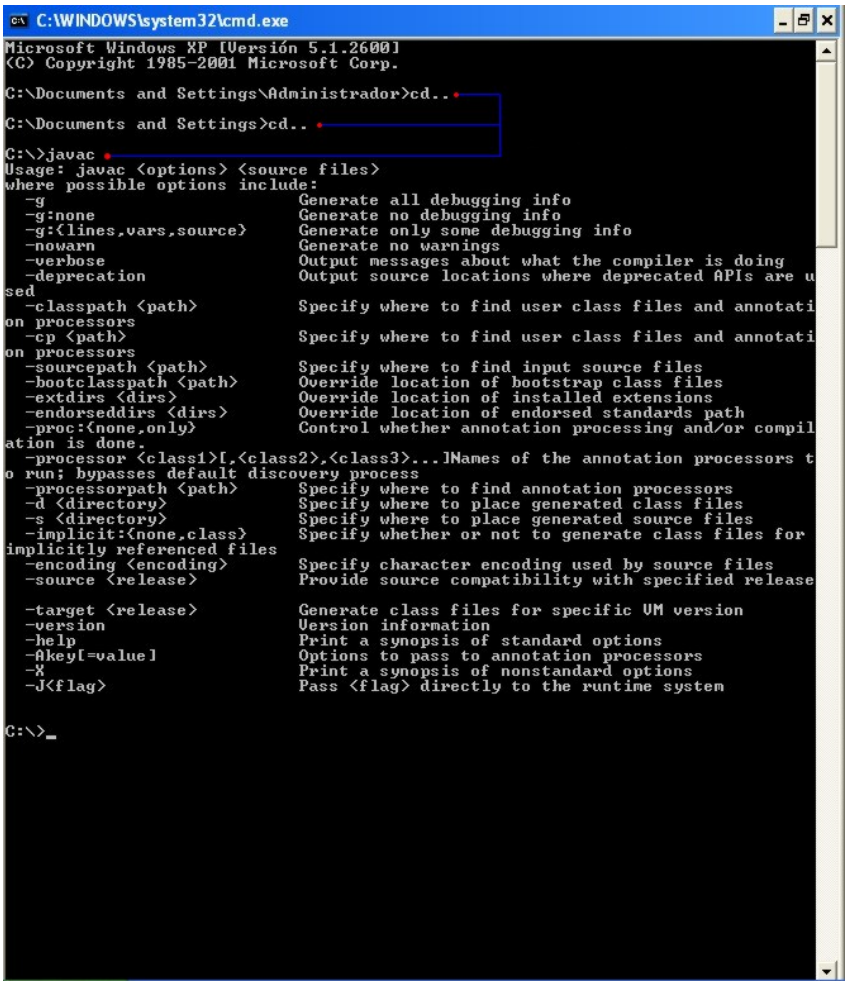
Inicio → Ejecutar.. → Tipeamos CMD → Enter

Tecla Windows + R → Tipeamos CMD → Enter

Hacemos una escala de directorios hasta llegar al Directorio Raíz C:\ con el comando CD..

Cuando estemos en c:\ tipeamos javac Si Nos muestra algún error de no reconocer el comando es porque no está bien configurado nuestro path pero si recibimos una respuesta como la de la siguiente imagen es porque somos triunfantes xD

De esta manera tendremos nuestro path configurado correctamente
Pero se estarán preguntando
Y el path para qué?¿
Pues el path nos permitirá
Hacer referencia a
componentes
Y funciones de JAVA
Para nuestro Uso ;D



Con esto podemos con plena certeza compilar y Ejecutar programas Escritos en JAVA, Como primera medida compilemos nuestro primer programa, Al Igual que en muchas ocasiones nos lanzaremos al estrellato programando el popular "HOLA MUNDO"

PROGRAMANDO Y COMPILANDO NUESTRO PRIMER PROGRAMA EN JAVA "HOLA MUNDO"

Conceptos a tener en cuenta antes de iniciar:

QUE ES UNA CLASE?

Una Clase JAVA es la "plantilla"sobre la cual trabajaremos y estas pueden contener variables y métodos
Dicha clase debe tener el mismo nombre del archivo .java

QUE ES UN MÉTODO?

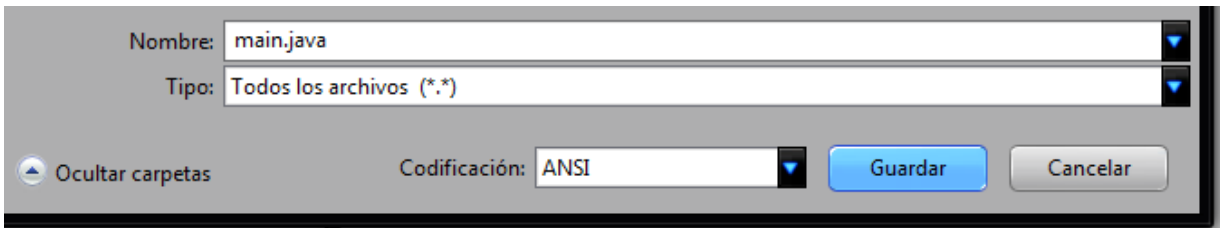
Un método JAVA es un Bloque de código que se le asigna un nombre para ser llamado luego en cualquier evento del programa.
Listo después seguimos con más conceptos
Empezamos a programar en JAVA nada mas basta con abrir un editor de texto y que mejor opción que el tradicional, viejo, muy conocido, estupendo y poderoso notepad "block de notas", Que pensabas que el notepad solo servía para hacer bat's y autorun's Bue.. Ya con nuestro blog de notas abierto tiramos code JAVA!
Para este ejemplo Crearemos una Carpeta en C:\ yo la llame "hola mundo" ósea que este ejercicio lo guardare en " C:\hola mundo\ "y le pondré como nombre main.java haber si le pones otro nombre he?

```
public class main
{
public static void main(String[] args)
{
System.out.println("Hola Mundo");
}
}
```

Nota: El Nombre de la Clase debe ser el Mismo Que el nombre del archivo
Ejemplo:
Nombre archivo: suma.java
Clase: public class suma

Recordar! JAVA Diferencia entre mayúsculas y minúsculas
Ósea que es diferente "Variable1" a "variable1"

Le damos Archivo → Guardar como → Buscamos nuestra ruta " C:\hola mundo\ " ya saben nombre del archivo main.java y abajo en tipo de archivo seleccionamos todos los archivos



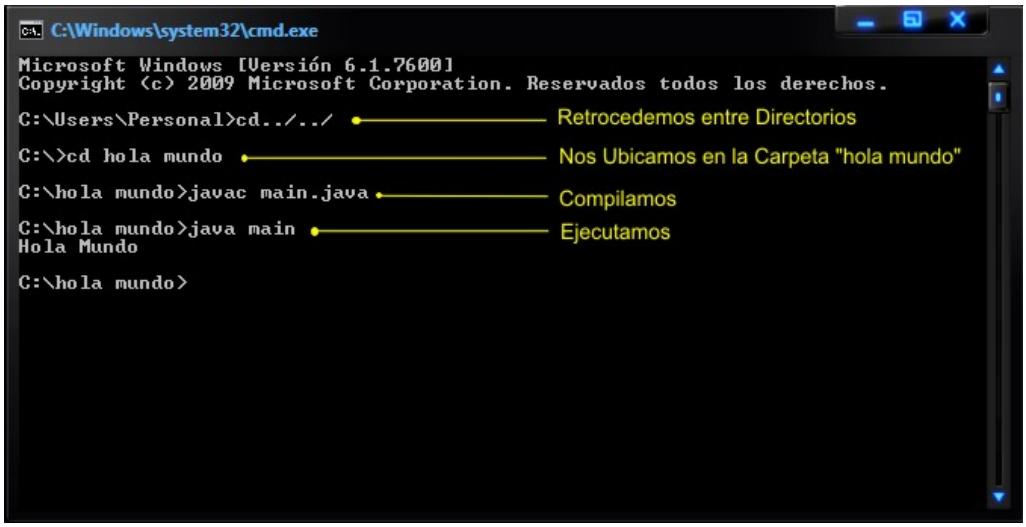
Listo hemos creado el archivo JAVA

Abrimos La Shell nos ubicamos en la carpeta donde guardamos el archivo

Para compilar javac [nombre archivo].java

Para ejecutar java [nombre archivo]

Cabe Recalcar que estos comandos de ejecución y compilación se puden ejecutar desde cualquier directoro en el que estemos ubicados en la consola, solo en el directorio raiz



Tal Vez si nunca has programado en java te preguntaras Por Dios Que es este código :S ?

Listo explicare..

Al declarar

```
public class main
{
```

Si sabemos un poco de ingles Estamos declarando una clase pública ósea que esta clase puede ser accesible desde otras clases

Aquí estamos programando el cuerpo de nuestra clase y las funciones del programa

```
public static void main(String[] args)
{
```

Con esta Instrucción Mostramos Un mensaje en pantalla, en este caso ** Hola Mundo******

```
System.out.println("Hola Mundo");
```

Y por ultimo cerramos

```
}
}
```

Con esto hemos terminado el primer programa en java de nuestras vidas si.. Si.. Vendrán muchos más..

Como por ejemplo hagamos otro programa típico al iniciar en cualquier lenguaje de programación Sumemos 2 números! =D

Pero primero lo primero …

VARIABLES:

Con el fin de retener y poder utilizar tanto los datos iníciales como los resultados de operaciones efectuadas durante la ejecución de un programa se requiere la utilización de variables, que representan un lugar de almacenamiento, cuyo contenido podrá variar durante el proceso de ejecución del programa.

Variables de tipo Enteras:

```
int variable1 = 500;
```

Declaramos que variable1 es un entero que equivale a 500

Los datos de tipo int representan el conjunto de los números enteros positivos y el conjunto de los números enteros negativos

Variables de Tipo Cadena:

```
String nombre = "hackxcrack";
```

Declaramos que nombre es una cadena de carácter equivalente a hackxcrack
El string se define como una cadena de caracteres determinada por una secuencia de caracteres encerrados entre los apostrofes “”

Variables de tipo decimal

```
double res = 3.5;
double numero1;
```

Declaramos una variable res con valor de 3.5 y una variable numero1 sin ningún valor tal vez la declaramos para asignarle algún valor durante la ejecución del programa

```
float variable1,variable2,variable3;
float nn = 050;
```

Declaramos inicialmente 3 variables sin valor y mas abajo una variable nn con valor 050.

Variables de tipo Switch Falso o Verdadero

```
boolean respuesta1 = true;
variable respuesta1 equivalente a true “verdadero”
```

Los valores que puede tomar una variable de tipo BOOLEAN, son valores lógicos, es decir, TRUE (verdadero) o FALSE (falso)

También Encontramos otros tipos de datos primitivos como char, long, byte que serán tratados con posteridad por ahora trabajaremos con estos tipos de datos; D

IDENTIFICADORES:

Un identificador es un nombre que otorgamos a una variable para diferenciarla dentro del programa, dichos identificadores están regidos por algunas reglas expuestas por el lenguaje estas son:

- **Debe Comenzar con letra**
- **No puede contener espacio en blanco**
- **No se podrá utilizar palabras reservadas del lenguaje como identificadores**
- **Como dije arriba se distinguirá entre MAYUSCULAS y minúsculas**
- **No Incluir caracteres especiales**

Correcto:	Incorrecto:
✓ Variable1	✘ 1nombre → Comienza por Numero
✓ Saldo2	✘ Cuenta tres → Contiene espacio en blanco
✓ NOTAFINAL	✘ new → es una palabra reservada
	✘ nueva# → Contiene Meta Carácter

COMENTARIOS:

Los Comentarios son los textos que generalmente se hacen en la cabecera o al final del programa, también se suelen colocar comentarios para explicar líneas de código y guiarse en programas extensos por lo menos yo uso los comentario para boludear nomas =)

Los Comentarios En Java tienen la siguiente estructura.

```
// Hola Soy Un Comentario en Java y Mi Presencia aquí no afecta en lo mas mínimo al programa

/*

 * Hola Soy Un Comentario en Java

 * De Varias Lineas =D

 */
```

Bien ahora teniendo unos cuantos conceptos básicos tiramos el code:

```
public class suma{
   public static void main (String [] args)
   {
 int num1 = 20;
 int num2 = 20;
 int suma;
 suma = num1 + num2;
   System.out.println("El resultado es"+suma);
      }
}
```

```
C:\>java suma
El resultado es40

C:\>
```

Explicacion

Bien Compilamos como ya sabemos =) y tendremos como resultado 40 en este ejercicio vemos la manera de declarar variables asignadas como num1 que es un entero con valor de 20 y la variable suma que es un entero que no le asignamos valores ya que el valor de este vendrá de la suma de num1 + num2, nos topamos también con instrucción:

System.out.println("El resultado es"+suma);

Si..si.. ya se es para mostrar un mensaje en pantalla y también resultados obtenidos en operaciones realizadas dentro del programa.

Muy Bien! Así es.. ;D

Pero No quiero Asignar los Números quiero Digitarlos Yo Mismo!! :S

Muy Bien En el Siguiente Ejercicio veremos como capturar datos digitados desde el teclado, asignarlos a variables para luego realizar operaciones pertinentes.

```
import java.util.Scanner;
public class sumaj{
public static void main(String Args[]){
      int num1;
      int num2;
      int suma;
Scanner reader = new Scanner(System.in);
```

```
System.out.print ("Digite primer numero:");
num1 = reader.nextInt();
System.out.print ("Digite segundo numero:");
num2 = reader.nextInt();
suma = num1+num2;
System.out.println("El resultado es:"+suma);
 }
}
```

```
C:\>java sumaj
Digite primer numero:45
Digite segundo numero:45
El resultado es:90

C:\>
```

Nos topamos que hemos usado la instrucción import con la que.. si si.. importamos la librería java.util.Scanner para poder recuperar en nuestras variables los datos digitados en la consola .D
Mediante num1 = reader.nextInt(); de resto igual una vez tenemos los datos realizamos operaciones y mostramos resultados en el transcurso de la programación en java podrás darte cuenta que deberás importar algunas librerías para poder utilizar ciertas funciones y controlar eventos.

INSTRUCCIONES DE CONDICION

Estas Instrucciones evalúan la veracidad en cuanto a eventos dados en el programa, exacto te estoy hablando del muy conocido "SI" el cual abunda en la mayoría de los lenguajes de programación ya que es una instrucción muy poderosa.
sintaxis

```
if [Condición] {
    [sentencias 1]
  }
else
{
    [sentencias 2]
  }
```

Un ejemplo: Deseamos llevar a juan, carlos y a miguel a una Disco pero solo admiten solo mayores de 18 Tendríamos que saber la edad de los 3 no? Listo Pedimos la edad de cada uno y evaluamos si estas son mayores a 18 con un si.
Comenzamos.

SOLUCION:

```
import java.util.Scanner;
public class van{
   public static void main(String Args[]){
        int edaddejuan, edaddecarlos, edaddemiguel;
        Scanner reader = new Scanner(System.in);
        System.out.print ("Digite la edad de juan");
        edaddejuan = reader.nextInt();
        System.out.print ("Digite la edad de carlos");
        edaddecarlos = reader.nextInt();
        System.out.print ("Digite la edad de miguel");
        edaddemiguel = reader.nextInt();
        if (edaddejuan >= 18) {
              System.out.println("JUAN PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddejuan);
         }
         else
         {
              System.out.println("JUAN NO PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddejuan);
         }
         if (edaddecarlos >= 18) {
              System.out.println("CARLOS PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddecarlos);
         }
         else
         {
               System.out.println("CARLOS NO PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddecarlos);
         }
         if (edaddemiguel >= 18) {
              System.out.println("MIGUEL PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddemiguel);
         }
         else
         {
               System.out.println("MIGUEL NO PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddemiguel);
          }
     }
}
```

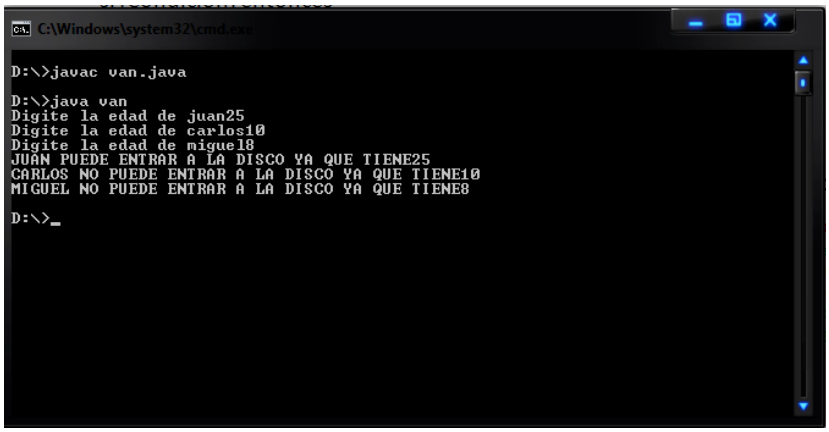
Explicación del código:

Como en el ejercicio anterior pedimos los datos, pero con la particularidad que evaluamos los datos recuperados con el condicional *if:*

```
if (edaddecarlos >= 18) {
      System.out.println("CARLOS PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddecarlos);
}
else {
      System.out.println("CARLOS NO PUEDE ENTRAR A LA DISCO YA QUE TIENE"+edaddecarlos);
}
```

Si edaddecarlos es mayor o igual (>=) a 18 escriba CARLOS PUEDE ENTRAR A LA DISCO
Si no es así: CARLOS NO PUEDE ENTRAR A LA DISCO
Fácil verdad?

IMAGEN DEL EJERCICIO ANTERIOR



INSTRUCCIONES DE REPETICION "CICLO FOR"

Un Poco mas de Teoria.. es bueno por que muchos "Programadores" solo conocen el nombre de las funciones, no se muchas veces usan instrucciones en un lenguaje por "inercia" bue.. Sigamos..

El Bucle FOR es una estructura de control disponible en casi todos los lenguajes de programación.. Ofreciéndole al programador la posibilidad de definir cuantas veces desea generar determinada acción

Sintaxis

```
for variable1 ← variable2 hasta variable3 incrementos de variable4 {
Instrucciones…
}
```

Stop!!! **Antes de tirar el code Veamos los operadores de relación:**

OPERADORES DE RELACION

Operador	Nombre
=	Igual
<	Menor
>	Mayor
<=	Menor O Igual
>=	Mayor O igual
!=	Diferente

Míralos Bien Que serán vuestros Nuevos aliados de ahora en adelante =D

Un Ejemplo: Hacer un programa que genere las tablas de multiplicar, Pidiendo de entrada la tabla que deseamos revisar.

```
import java.util.Scanner;

public class tabla{
```

```
public static void main(String Args[]){
int tabla;
Scanner reader=new Scanner(System.in);
System.out.println("Que Tabla de Multiplicar Deseas Revisar?");
tabla=reader.nextInt();
System.out.println("****** TABLA DE MULTIPLICAR GENERADA ****** ");
for (int var1=1;var1<11;var1++)
System.out.println("Multiplicacion de: "+tabla+ "x" +var1+ "=" +(var1*tabla));
}
}
```

Como vemos pedimos el numero de la tabla que deseamos ver y mediante el ciclo "for" iniciamos la variable "variable1" en un valor de "1" e incrementándola hasta que sea menor a 11 luego mostramos la salida por pantalla con un mensaje en el cual también lleva la operación de multiplicación.

INSTRUCCIÓN DE REPETICION CICLO "WHILE" Mientas QUE

Esta es otra instrucción poderosa al momento de repetir bloques de código, su función esta centrada en ejecutar bloques de código mientras una instrucción sea verdadera.

Sintaxis

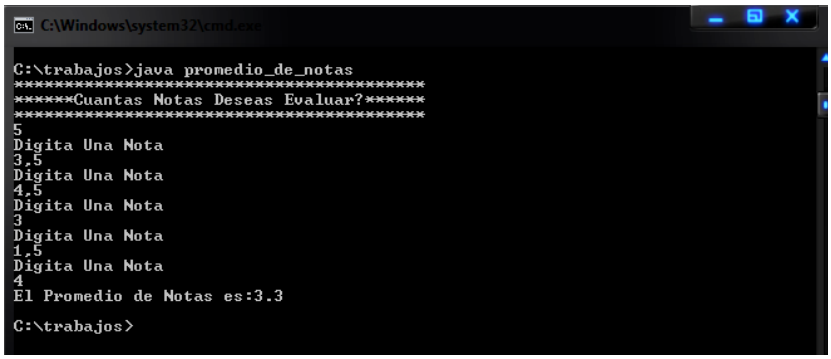
```
While  (Condición)
        {
          Instrucciones..
        }
```

Sale Ejemplo...
Hacer Un programa que pida el número de notas a trabajar y las promedie.

```
/*PROMEDIO DE NOTAS
*UTILIZANDO CICLO WHILE EN JAVA "Recuerda Soy Un Comentario =D"
*/
import java.util.Scanner;
```

```
public class promedio_de_notas  {
public static void main (String Args[]){
//Declaracion de Variables
int nn;
int tope = 1;
float n;
float notas = 0;
Scanner reader=new Scanner(System.in);
//Mensaje al Usuario
System.out.println("****************************************");
System.out.println("******Cuantas Notas Deseas Evaluar?******");
System.out.println("****************************************");
//Recojemos La Cantidad de Notas a Promediar
nn=reader.nextInt();
while(tope<=nn) //Mientras que Tope Sea Menor ó Igual a nn Pidamos Notas
{
System.out.println("Digita Una Nota");
n=reader.nextFloat();
tope=tope+1; //Incrementamos el valor de la variable tope para controlar el ciclo
notas=notas+n; //Vamos Recolectando el Valor de las Notas Que Digitamos y las sumamos
}//cerramos el ciclo
System.out.println("El Promedio de Notas es:"+(notas/nn)); // Mostramos Resultados
}
}
```

IMAGEN DEL EJERCICIO ANTERIOR



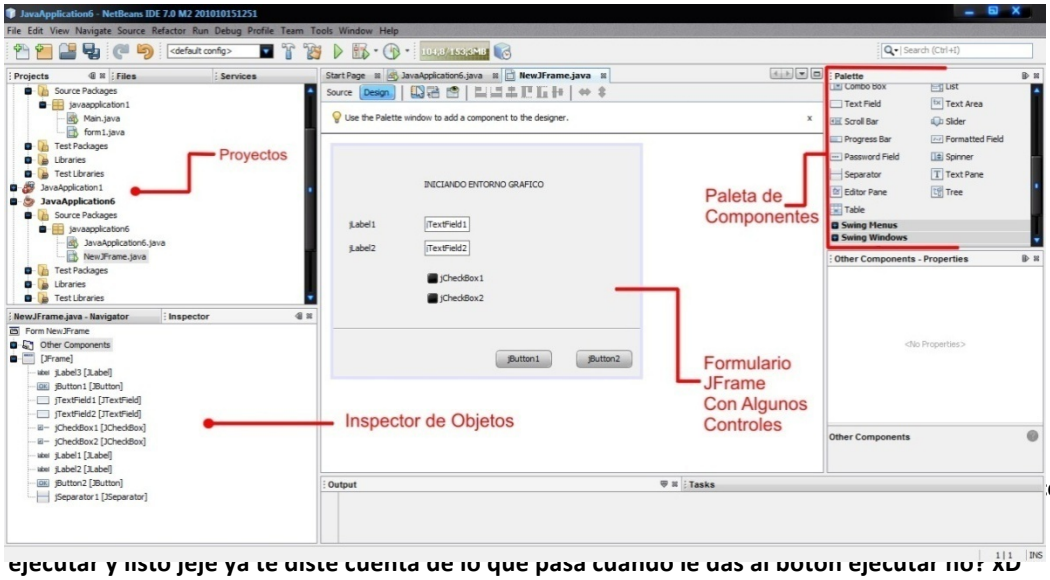
Aquí Terminamos Con Java En modo consola ahora pasamos a un entorno de trabajo mucho más amigable y cómodo el cual nos facilitara mucho nuestro trabajo "NETBEANS" Hay muchos IDE Lo sé pero para mí uno de los más completos y robustos es NETBEANS de la misma compañía de java, SUN que actualmente cursa en su versión 7 =)

Podemos Acceder a la descargar Aca: http://netbeans.org/community/releases/70/

En la parte de arriba explique lo que era un IDE pero tal vez te estés preguntando será que el IDE programara por mi? Aja como acabas de decir que me facilitara el trabajo.. :$

Está bien aclaremos los IDE "INTEGRATED DEVELOPMENT ENVIRONMENT" Es un programa el cual nos pone en bandeja de plata un lenguaje para que programemos solo las funciones de los programas que hagamos y no todo el proyecto en si. Ejemplo:
Por consola os tocaría desde diseñar las ventanas de tus formularios hasta crear cada botón a mano code por code.
Mientras que un IDE te permite crear formularios arrastrar componentes "botones, etiquetas, tablas" etc
Vez la ventaja de tener un IDE =) pero cabe recalcarte que un buen programador debe saber hacer desde arrastrar un componente a un formulario, hasta saber el code que se ha generado al momento de realizar esa acción.
Exacto! Que las cosas es mejor hacerlas uno mismo que a que otro te las de echas pero sabemos que este campo de la programación se mueve muy rápidamente y debemos estar a la par y sacar proyectos para vender y obtener más clientes así que el espabila pierde!! .

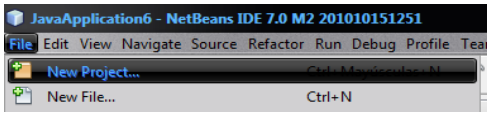
ECHANDO NUESTRO PRIMER VISTAZO AL IDE =)



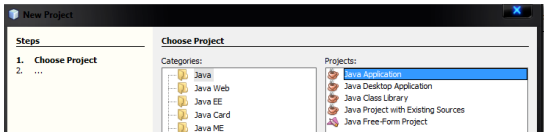
omo los que solo damos ejecutar y listo jeje ya te diste cuenta de lo que pasa cuando le das al boton ejecutar no? xD

También tienes a tu disposición una paleta de componentes con botones, labels, campos de texto entre muchos otros componentes más que solo será dar clic y llevarlo a tu formulario ;) El Inspector de Objetos muestra un árbol de todos los componentes que estés utilizando en algún formulario, se que estas ansioso por hacer tu primera aplicación haber qué tal te va listo hagámosla!!

Como primera medida vamos a crear un proyecto
File→ New Proyect…



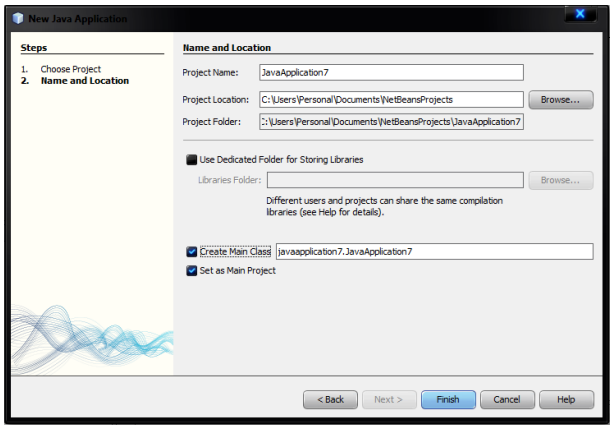
Elegimos Java Application como en la siguiente imagen .)

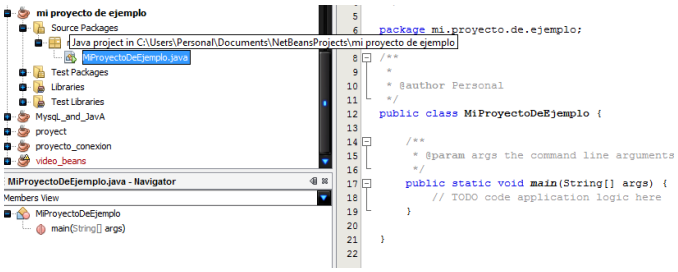


Next >>

Ponemos el nombre del proyecto es el nombre que quieras man ¬¬ yo le puse "mi proyecto de ejemplo"

Veamos que tanto vuela tu imaginación :D Y finish

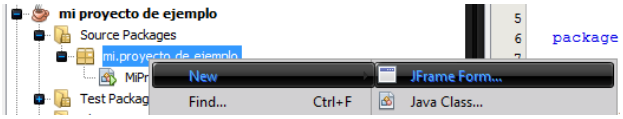




=O se crearon el paquete y la clase (H), Exacto se crean solitos y como lo que vamos a trabajar es entorno grafico crearemos un formulario para empezar a realizar tareas sencillas, cabe decir que nuestra finalidad es llegar a realizar un proyecto con conexiones a bases de datos =) pero para llegar haya tendremos que practicar algunas cosillas básicas, "Paciencia Virtud de Sabios"

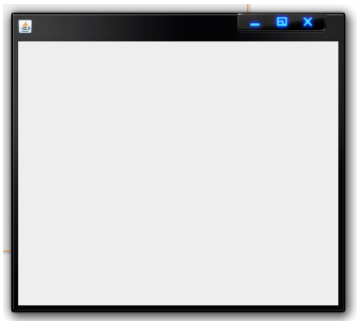
Bien Creamos el Formulario Clic Derecho Sobre nuestro Paquete → new → jframeform

Se.. Creativo cambia el nombre xDD



Sé que eres impaciente y quieres ejecutar el formulario así en blanco ¬¬

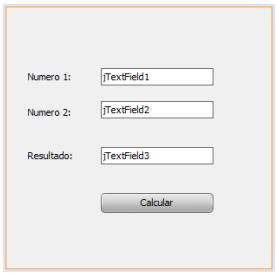
Ejecutas con Shift + F6 o te vas al Explorador de proyectos clic derecho sobre el archivo que quieres ejecutar y escoges run file



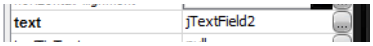
Y Lo Ejecutaste.. =D
Bueno un gran programador
Es una persona inquieta
Que le gusta experimentar
Hasta las cosas más sencillas
De las cuales aprende mucho más de lo que imagina

Creando El Primer Entorno Grafico Con Controles

Ya que tenemos el formulario listo solo tenemos que arrastrar controles de la paleta de controles que se encuentra a la derecha.
Hagamos un primer ejercicio básico arrastramos 3 jtextFied's 3 JLabel's y un JButton
Exacto Sumaremos :$



Para Cambiar el texto de las etiquetas de las cajas de texto o del botón solo seleccionamos el control y nos vamos a las propiedades del objeto y buscamos la propiedad text y modificamos.

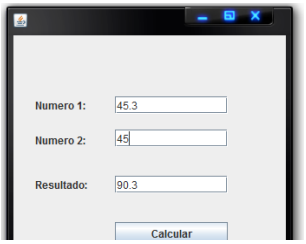


Ó simplemente clic derecho al componente "Edit Text"
Bien Ahora a programar!! Antes de empezar recordar que los jtextField's recuperan todo en formato texto :S Que haremos!!
Bien Pasos a seguir
1. declararemos variables String para caputar los datos de los jtextfield's
2. declararemos variables de tipo numero.
3. realizaremos una conversión de las variables tipo String a numero
4. realizamos una suma común y corriente
5. mostramos el resultado en el tercer JtextField
Facil!!
Con el formulario abierto da clic en el botón code jeje si lo tienes en ingles en source y nos ubicamos bajo package mi.proyecto.de.ejemplo;
E importamos la librería javax.swing.JOptionPane; para poder lanzar mensajes.

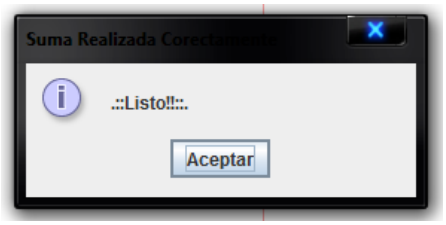
```
import javax.swing.JOptionPane;
```

Bien damos doble clic al botón "calcular" y tiramos el code.

```
String a,b; //Declaramos Variables String
double c,d,e; // Declaramos variables Tipo Double
a = jTextField1.getText(); // Capturamos Los Valores del JtextFiel1 en "a"
```



```
        b = jTextField2.getText();       // Capturamos Los Valores del JtextFiel2 en "b"
        c = Double.parseDouble(a);   //Conversion de Texto a Double
        d = Double.parseDouble(b);   //Conversion de Texto a Double
        e = c + d; //Ya podemos sumar normalmente
        jTextField3.setText(String.valueOf(e)); //Mostramos Valores en el JTextFiel3
JOptionPane.showMessageDialog(this,        ".::Listo!!::.",        "Suma        Realizada        Corectamente",
JOptionPane.INFORMATION_MESSAGE);     //Listo!
```

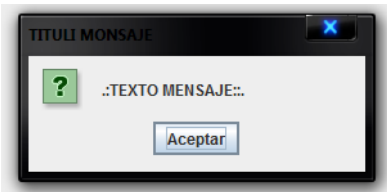


TIPOS DE MENSAJES CON JOPTIONPANE

El code siempre será el mismo lo único que muchas veces debemos mostrar diferentes tipos de mensajes ejemplos:

Mensaje de Error

```
JOptionPane.showMessageDialog(this, ".:TEXTO MENSAJE::.", "TITULI MONSAJE", JOptionPane.ERROR_MESSAGE);
```



Mensaje de Pregunta

```
JOptionPane.showMessageDialog(this, ".:TEXTO MENSAJE::.", "TITULI MONSAJE", JOptionPane.QUESTION_MESSAGE);
```

Mensaje de Advertencia

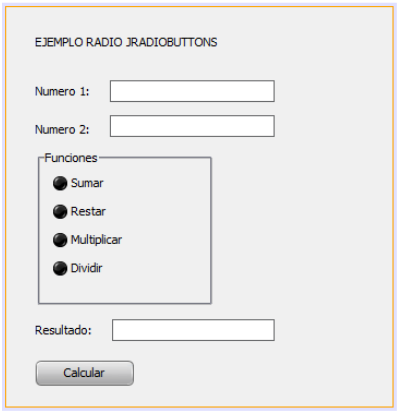
```
JOptionPane.showMessageDialog(this, ".:TEXTO MENSAJE::.", "TITULI MONSAJE", JOptionPane.CANCEL_OPTION);
```

Estos son algunos ejemplos, solo es colocar
JOptionPane.showMessageDialog(this, ".:TEXTO MENSAJE::.", "TITULI MONSAJE", JOptionPane. Y saldrán todas las opciones para el manejo de mensajes ya sea de información o de solicitud de datos muy configurables.

EJERCICIOS PROPUESTOS PARA PRACTICAR FUNCIONES DE ALGUNOS COMPONENTES

COMPONENTE A ESTUDIAR JRADIOBUTTON

Los RadioButtons son componentes para selección de opciones un ejemplo:



Digitamos los 2 números y escogemos sumar sumamos digitamos los 2 números y escogemos restar restamos y así sucesivamente =)

Bien lo que debemos hacer es decirle al botón calcular que cuando Se le dé clic evalué el JRADIOBUTTON seleccionado y haga alguna Acción Pertinente en este caso sumar, restar, multiplicar o dividir

Nota: Cada JRadioButton lleva consigo puesto el nombre equivalente a su función. Iniciamos:
Doble clic al Botón calcular:

```
String n1,n2;
 if (sumar.isSelected()) { //aqui es donde verificamos
    Double numero1,numero2,resultado;
    n1 = jTextField1.getText();
    n2 = jTextField2.getText();
    numero1 = Double.parseDouble(n1);
    numero2 = Double.parseDouble(n2);
    resultado = numero1 + numero2;
    jTextField3.setText(String.valueOf(resultado));
    //del resto igual, conversion y muestra deresultado
 }

 if (restar.isSelected()) {
    Double numero1,numero2,resultado;
    n1 = jTextField1.getText();
    n2 = jTextField2.getText();
    numero1 = Double.parseDouble(n1);
    numero2 = Double.parseDouble(n2);
    resultado = numero1 - numero2;
     jTextField3.setText(String.valueOf(resultado));
 }
if (multiplicar.isSelected()) {
    Double numero1,numero2,resultado;
    n1 = jTextField1.getText();
    n2 = jTextField2.getText();
    numero1 = Double.parseDouble(n1);
```

```
        numero2 = Double.parseDouble(n2);
        resultado = numero1 * numero2;
         jTextField3.setText(String.valueOf(resultado));
    }
 if (dividir.isSelected()) {
        Double numero1,numero2,resultado;
        n1 = jTextField1.getText();
        n2 = jTextField2.getText();
        numero1 = Double.parseDouble(n1);
        numero2 = Double.parseDouble(n2);
        resultado = numero1 / numero2;
         jTextField3.setText(String.valueOf(resultado));
    }
  }
```

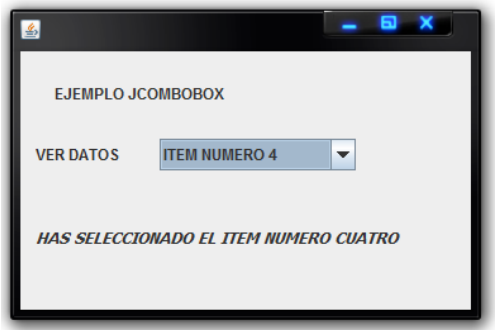
COMPONENTE A ESTUDIAR JCOMBOBOX

El uso habitual de un Jcombobox o listas desplegables es de seleccionar un ítem y se genere una acción o muchas veces también has podido darte cuenta en los típicos formularios de registro en cualquier web cuando vas a seleccionar el país le das clic a la lista desplegable "combobox" y seleccionas el tuyo bueno lo que se hace en esos formularios es guardar el registro de ese combobox según lo que vos selecciones =)
Practiquemos un poco con los JCOMBOBOX:

EJERCICIO PROPUESTO:

Asignar 5 opciones a un JComboBox y al seleccionar algún ítem un JLabel diga que ítem seleccionamos:
Doble clic al JComboBox y codeamos:

```
String var; //Declaramos una variable string para guardar el dato del JCombobox
 var=((String) jComboBox1.getSelectedItem()); //Asignamos el valor seleccionado a la variable
 if (var.equals("ITEM NUMERO 1")){  //evaluamos qué valor tiene la variable var
   jLabel3.setText("HAS SELECCIONADO EL ITEM NUMERO UNO");
// mostramos que ítem seleccionamos
}
 if (var.equals("ITEM NUMERO 2")){
   jLabel3.setText("HAS SELECCIONADO EL ITEM NUMERO DOS");
}
 if (var.equals("ITEM NUMERO 3")){
   jLabel3.setText("HAS SELECCIONADO EL ITEM NUMERO TRES");
}
 if (var.equals("ITEM NUMERO 4")){
   jLabel3.setText("HAS SELECCIONADO EL ITEM NUMERO CUATRO");
}
 if (var.equals("ITEM NUMERO 5")){
   jLabel3.setText("HAS SELECCIONADO EL ITEM NUMERO CINCO");
}
```



COMPONENTE A ESTUDIAR JTEXTFIELD

Las JtextField osea las tradicionales cajas de texto, su funcion directamente es recibir datos ya sea para guardar en una BD, hacer operaciones matematicas en fin lo que el programador estime pertinente con un dato escrito en estas.

EJERCICIO PROPUESTO

Digitar un valor Cualquiera en un JTextField y mostrar este en un Mensaje en otro JtextField y en un JLabel

Importamos la librería jOptionpane

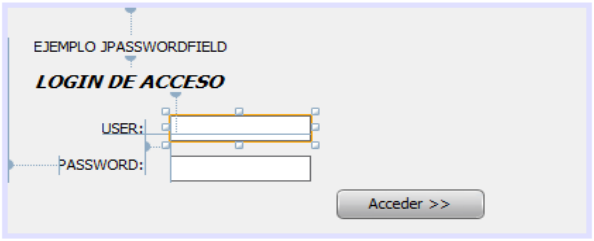
```
package mi_proyecto_de_ejemplo; //Recuerda este es el paquete de tu proyecto
import javax.swing.JOptionPane; //importamos
```

Haber damos Doble clic en el boton OK

```
    String texto;
    texto = jTextField1.getText(); //Recuperamos el valor escrito en el jTextField1
    jTextField2.setText(String.valueOf(texto));//Asignamos el valor recuperado a jTextField2
    jLabel5.setText(String.valueOf(texto));//Asi mismo a Label5
    JOptionPane.showMessageDialog(this, ""'+jTextField1.getText()+"", "Texto En JTextField1",
JOptionPane.INFORMATION_MESSAGE);// Asi mismo en un mensaje
```

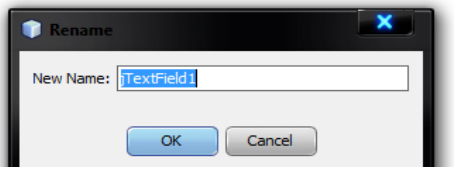
COMPONENTE A ESTUDIAR: JPASSWORDFIELD

Los JpasswordField son los componentes que reciben datos confidenciales en la mayoria de los casos contraseñas esas cajitas de texto que cuando escribes solo salen los típicos asteriscos ******



EJERCICIO PROPUESTO

Bien para el ejemplo haremos un formulario login muy sencillo se pedira para el acceso a la aplicación nombre de usuario y password haber de usuario colocare "Administrator" y pass colocare "12345678910" HUM.. que colocas vos he? Bien empezamos!

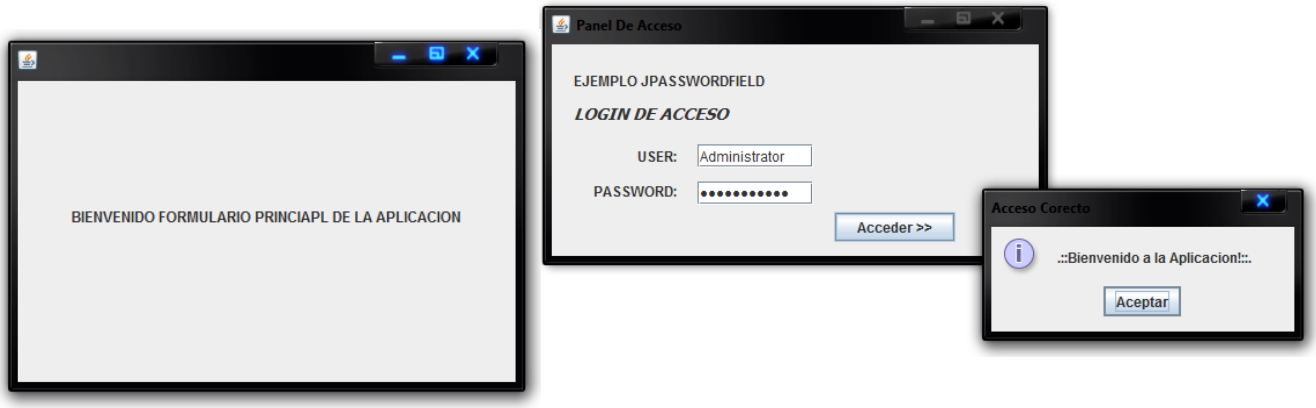


Diseñamos un Formulario donde pidamos user y pass. Cambiemos los nombres de los campos clic derecho sobre el jTextField → change variable name yo le puse "user" y al jPasswordField le puse "pwd" .

Como sabemos que validaremos los datos en el boton Acceder damos doble clic sobre el y programamos: Me Imagino que no tengo que recordar importar la librería joptionpane xDD

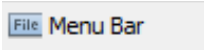
```
String Usuario = user.getText();
char passArray[] = pwd.getPassword();
String pass = new String(passArray);
//capturamos los datos digitados por el usuario
if (Usuario.equals("Administrator") && pass.equals("12345678910")) {
   JOptionPane.showMessageDialog(this, ".::Bienvenido a la Aplicacion!::.", "Acceso Correcto", JOptionPane.INFORMATION_MESSAGE);
    //verificamos si estos equivalen a nuestro login user Administrator y pass 12345678910
     mi_proyecto_de_ejemplo.principal principal= new mi_proyecto_de_ejemplo.principal();
    principal.show();
    //De ser cierto Mostramos el formulario principal
    setVisible(false);
    //y ocultamos el login
} else {
    // de ser erroneo nuestros datos digitados mostramos un error de acceso
    JOptionPane.showMessageDialog(this, "Usuario o Contraseña Incorrectos Verifica", "Error de Acceso", JOptionPane.ERROR_MESSAGE);

    user.setText(null);
    pwd.setText(null);
    //y limpiamos los campos para un nuevo intento

}
```

IMAGEN DEL EJERCICIO ANTERIOR



COMPONENTE A ESTUDIAR: JMENUBAR

Bien un menu para ofrecer opciones a diferentes parte de un software asi como cuando el profe de informatica decia en el cole, bueno nos vamos a archivo → nuevo xDD exacto esta haciendo referencia al item nuevo que esta en el menu archivo bien hagamos uno en netebeans creamos un formulario y seleccionamos de la seccion Swing Menus de la paleta de componentes el componente MENUBAR



Y lo colocamos en nustro formulario como vez enseguida se situa en la parte superior como cualquier menu.

Armando el menu asi..

ARCHIVO ACERCA DE..

Opcion 1 hacerca de hack x crack

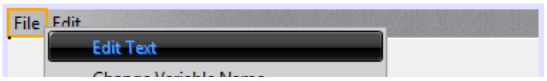
Opcion 2

Opcion 3

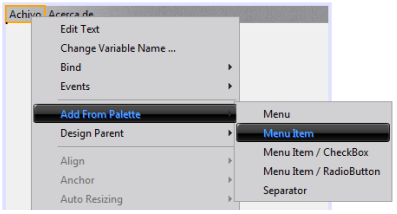
Salir

Al picar en opcion1 abriremos un formulario puede ser en blanco y asi con opcion 2,3 y tambien en el menu acerca de..

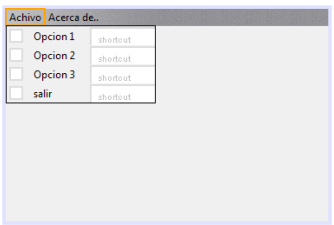
Clic derecho sobre el primer item → edit text para modificar el texto pones Archivo y en el otro pones Acerca de..

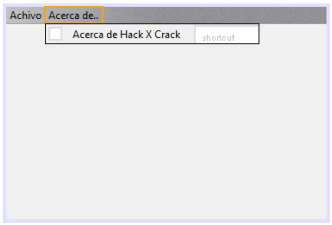


Ahora agregamos los items de cada menu clic derecho add from palette → menu item



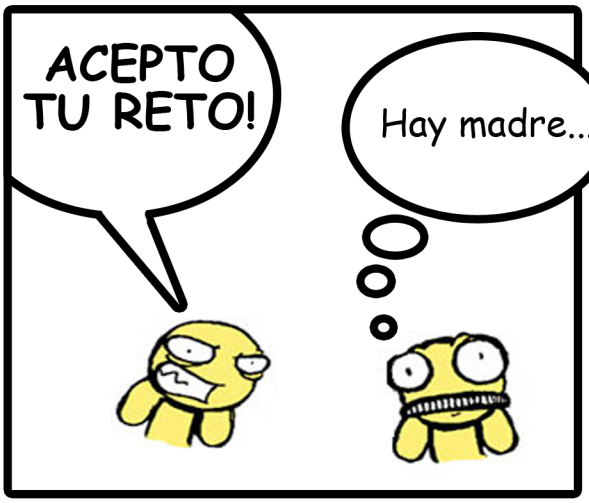
Agregamos las 4 opciones y la opcion de acerca de.. deberiamos ir asi..





Si eres Buen observador puedes dar cuenta que os da la opcion de colocarle a los items shortcut osea asignarle una tecla de acceso rapido y un icono solo es darle doble clic y configurarla =)

judgeNow!

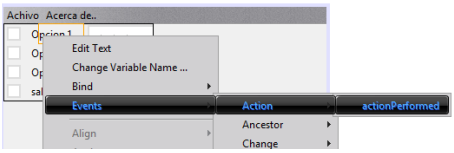


... continuará

www.judgenow.org

ASIGNANDO FUNCIONES A CADA ITEM

Ubiquemos Opcion 1 del menu archivo y demos clic derecho → events→action→actionPerformed



En este caso abrire un formulario que se le llame opcion1

```
mi_proyecto_de_ejemplo.opcion1 variable= new mi_proyecto_de_ejemplo.opcion1();
    variable.show();
```

asi mismo con los demas items

y el item salir? Asi me gusta que estes atento =D

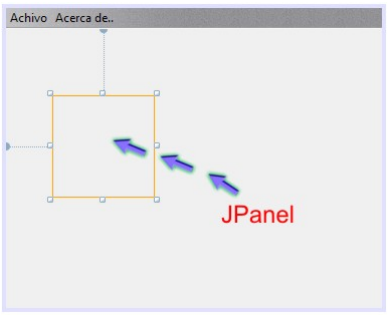
hacemos igual que en los demas pero el code seria:

```
System.exit(0);//Un boton cerrar :$
```

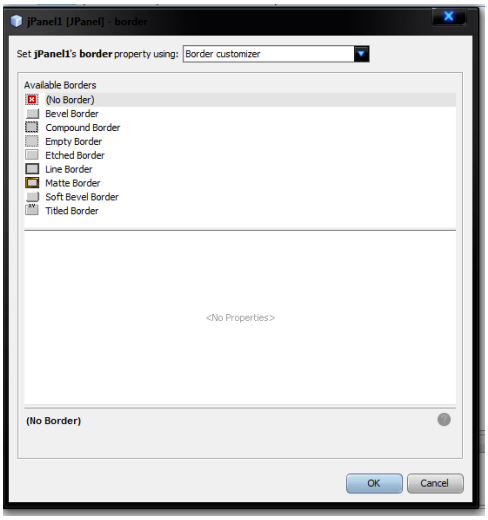
COMPONENTE A ESTUDIAR JPANEL

Los panel son utilizados para agrupar componentes o en muchas ocaciones yo los tomo solo para colocarle una imagen de fondo a un formulario.

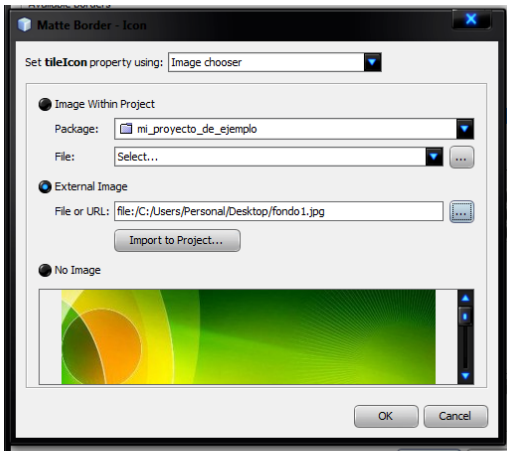
Sale ejemplo coloquemos una imagen de fondo a un formulario mediante un jpanel



Expandemos el jpanel de tal manera que cubra todo el formulario y nos bamos a propiedades.. no se olvide que se encuentra en la parte derecha del programa escojes border.

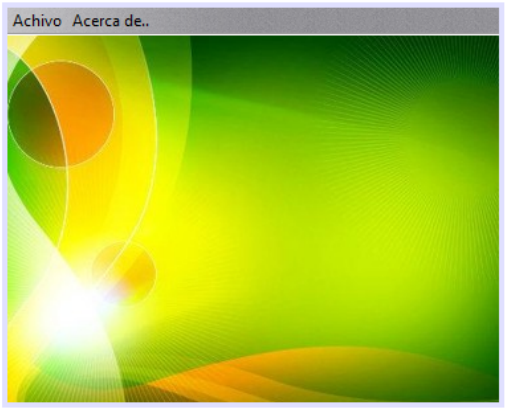


Aquí seleccionas Matte Border y luego clic en icon seleccionas external image para buscarla en tu Pc



Clic en OK y listo habremos puesto una imagen de fondo al programa

En mi caso me quedo asi.



Bien si has llegado hasta aquí quiere decir que tienes muchos conseptos basicos en programacion java y del manejo del IDE NETBEANS, ahora haremos la tan anhelada conexión a BASE DE DATOS para el primer paso en las conexiones, utilizaremos como "Motor de Base de Datos" Microsoft Access, Quizas si has leido sobre access muchos no lo catalogan como un verdadero motor y gestor de bases de datos opinion que comparto con el que lo dice ;) pero sera el primer escalon a subir.

Pero me dices que no es catalogado como un verdadero motor de bases de datos :$ por que no hacemos una conexión con un verdadero motor de bases de datos?

Paciencia.. decierto os digo que al terminar la conexión con access realizaremos una con MYSQL =)

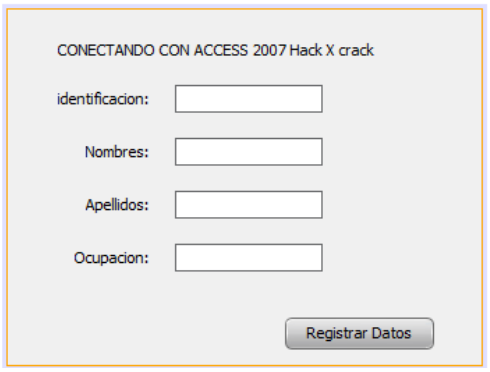
No pongas esa cara ningun conocimiento esta de mas tal vez algun dia necesites una basesita de datos en access para algun programa pequeño.. o bien utilizar tablas temporales en access vez.. de algo puede servir :D

Pero ten en cuenta las palabras de un profesor que tuve " DESPUES QUE LA EMPRESA TENGA UN AVISO DE MAS DE 4 METROS DE ANCHO YA ACCESS NO LE SIRVE COMO MOTOR DE BASE DE DATOS XD"

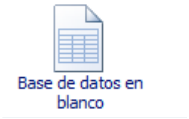
Sin mas ni mas empezamos

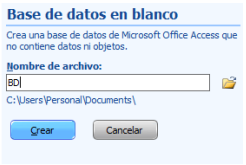
CONECTANDO A NUESTRA PRIMERA BASE DE DATOS EN ACCESS

Montamos un escenario con 4JttextField's 4 Jlabel's y un JButton Asi:

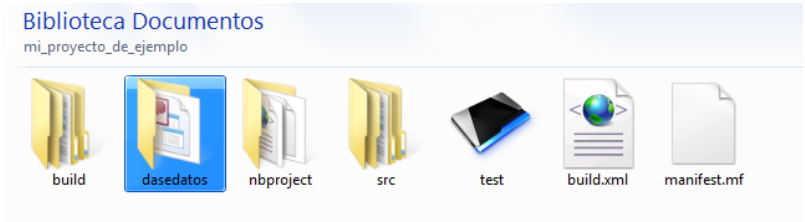


Cambiamos las variables de las cajas de texto como ya sabemos clic derecho change variable name
Y pongamos nombres alusivos identificacion,nombres,apellidos,ocupacion como ya sabes que la finalidad es guardar esos campos digitados en una base de datos en access abrimos access para crear la BD.

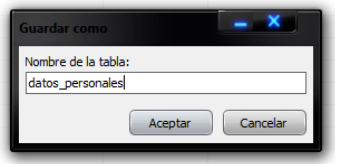




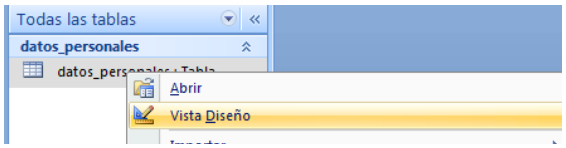
Como puedes ver le coloque de nombre BD.mdb, OJO no la guardes aun el paso a seguir es guardarla en la misma carpeta de tu proyecto.
¿y donde estan guardados los proyectos de NETBEANS?
En Mis Documentos en la carpeta llamada NetBeansProjects hay se encuentran todos yo en mi caso quice ser un poco organizado y dentro de la carpeta del proyecto cree otra con nombre basededatos y la guarde hay dentro =)



Ahora trabajamos en access damos doble clic sobre cada columna y colocamos los nombres de estas asi:



Cierras la tabla y pedira un nombre yo le puse datos_personales
Segimos clic derecho a la tabla ps digo la por que solo tenemos una xD
Y escojes vista diseño

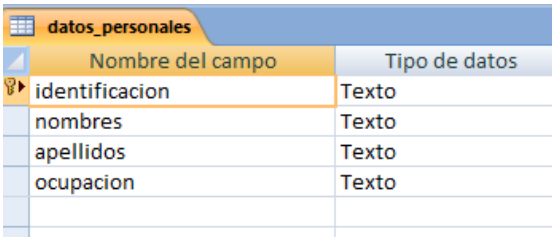


Vemos como estan constituidos los datos de la tabla y tambien vemos que la llave primaria es el campo identificacion

LLAVE PRIMARIA: es la columna que representara los datos almacenados en la tabla y no debe repetirse

Ejemplo: Insertaste en identificacion: 123, en nombre: jose, en apellidos: mendez, y en ocupacion: desempleado al buscar la identificacion con numero 123 sabras que haces referencia a jose medes que es un desocupado xDD

Bien vemos como esta conformada la tabla y nos damos cuenta que el campo identificacion tiene tipo de dato autonumerico lo pasamos a texto tambien ,)



Guardamos, hemos terminado con access volvamos a NetBeans.

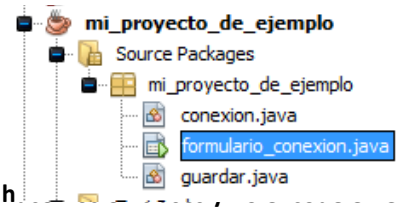
Creamos Una clase para la conexión

Recuerda clic derecho sobre el paquete de tu proyecto → new → Java class de nombre puse conexión

Creamos otra clase llamada guardar.

La clase conexión hara la referencia o la conexión con la base de datos de access y la clase guardar tendra las funciones "obvias" de guardar utilizando como referencia la clase conexión

Asi va el paquete



Y por que todo lo h como he visto en otros lenguajes?

Pues si podemos hacerlo con el code en el boton pero piensa si tienes un software con mil botones guardar, modificar, eliminar etc cuantas conexiones tocaria hacer? Este ejemplo es pensado para personas que quieran seguir programando en java a lo grande, hay que prepararse para la destruccion de mundo y asi se empieza =)

Empezaremos con el code, ya sabemos que donde veas un Joptiompane debes importar la librería ¬¬

Abrimos la clase conexión e importamos algunas librerias necesarias para el control de errores "de haberlos" y drivers de conexión

```
package mi_proyecto_de_ejemplo;// Ya sabes que este es el paquete
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class conexion {
private Connection conexion;
public Statement sentencia;
```

```
    public void conectarBaseDeDatos() {
      try {

         final String CONTROLADOR = "sun.jdbc.odbc.JdbcOdbcDriver"; //Aqui se declara el controlador a utilizar
         Class.forName( CONTROLADOR );
         final String RUTA_BD = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ="+
         System.getProperty( "user.dir" )+"/dasedatos/BD.mdb"; // hacemos referencia al archivo access BD.mdb
         conexion = DriverManager.getConnection(RUTA_BD);
         sentencia = conexion.createStatement();
        // se declara una variable sentencia para la manipulacion de la base de datos
      }
      catch (ClassNotFoundException ex1) {
         ex1.printStackTrace();
         javax.swing.JOptionPane.showMessageDialog(null,"Error Carga Driver");
         System.exit(1);
       //En caso de errores de driver
      }
      catch (SQLException ex) {
         ex.printStackTrace();
         javax.swing.JOptionPane.showMessageDialog(null,"Error Creacion Statement");
         System.exit(1);
      //en caso de no acceder a la base de datos o no encontrar ruta correcta
      }
   }
}
```

SEGUIMOS AHORA CON LA CLASE GUARDAR

```
package mi_proyecto_de_ejemplo;

public class guardar {
   mi_proyecto_de_ejemplo.conexion bd = new mi_proyecto_de_ejemplo.conexion();
// Asiganmos un alias a la clase conexión en este caso bd
   public void registrar_datos(String id,String n,String a, String o){
 //halamos las variables declaradas en el formulario principal.. tranquilo ya las declaramos... xD
  try{
        bd.conectarBaseDeDatos();
//conectamos..
        bd.sentencia.execute("INSERT   INTO   datos_personales(identificacion,nombres,apellidos,ocupacion)
VALUES('"+id+"','"+n+"','"+a+"','"+o+"'')");
//insertamos los datos un SQL normal..
//PD no explico SQL por que hay una revista expresamente para eso =D
     }catch (Exception e){}
  }
}
```

Ahora seguimos con el FORMULARIO en mi caso se llama formulario_conexion y añadimos un alias para hacer referencia a la clase guardar

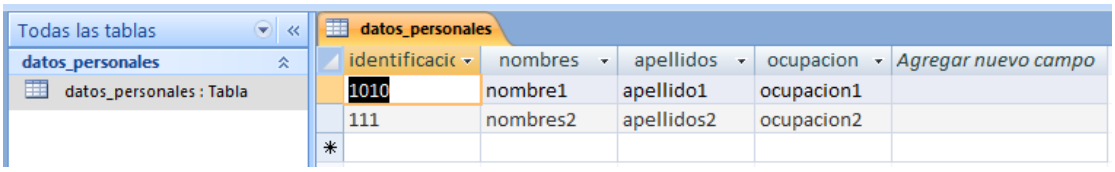
```
public class formulario_conexion extends javax.swing.JFrame { //esto es para que os guie ;D
   mi_proyecto_de_ejemplo.guardar guardar = new mi_proyecto_de_ejemplo.guardar();
```

Listo seguimos con el boton, doble clic y programamos..

```
        String  id = identificacion.getText();
        String n = nombres.getText();
         String a = apellidos.getText();
         String o = ocupacion.getText();
         //Recuperamos en variables los valores digitados en las JtextField
         guardar.registrar_datos(id, n, a, o);
         //Mandamos las variables a la clase guardar.. ya ella sabra que hacer con ellas xDD
         JOptionPane.showMessageDialog(null,"Datos Guardados Correctamente ","Mensaje de
Confirmacion",JOptionPane.PLAIN_MESSAGE);
         //Decimos que ya todo esta heco (H)
         identificacion.setText("");
         nombres.setText("");
         apellidos.setText("");
         ocupacion.setText("");
  // y limpiamos los campos para seguir digitando ☺
```

Listo Corremos la aplicación

Verificamos la Base de datos..



Efectivamente Los datos son enviados correctamente a los campos de la BD
Aquí hemos terminado nuestra primera conexión a bases de datos muy simple ☺
Me imagino que debes estar pensando que si puedes guardar.. podras…
Eliminar, Modificar, buscar Exacto ya has dado un gran paso hacer la conexión y guardar datos

Acontinuacion una verdadera conexión a bases de datos utilizando como SGBD "Sistema Gestor de Bases de Datos" MYSQL 5.1

Si no lo tienes instalado puedes descargarlo aca: http://dev.mysql.com/get/Downloads/MySQL-5.1/mysql-5.1.51-win32.msi

Proyecto a Realizar:

Software Con Conexión MYSQL "Modo Local"
Funciones:

- **Registrar datos (guardar)**
- **Actualizar Registros**
- **Eliminar Registros**
- **Buscar Registros**
- **Mostrar registros Almacenados en la Base de Datos**

Utilizaremos los mismos mecanisnos ubicando cada funcion en clases diferentes para un mejor entendimiento y manejo de clases empezamos!!

PARA EL EJEMPLO HE DECIDIDO QUE CREEMOS UN PROYECTO PARA REGISTRO DE ESTUDIANTES DE UN COLEGIO "X"

Creamos nuestra Base de Datos en MYSQL La llamaremos "colegio"

Abrimos La Consola de MYSQL Por si no tenemos un IDE Para Mysql como SQL JOG ó Navicat

```
CREATE DATABASE colegio;
```

Para Activarla y que Todo lo que Hagamos en esa Base de datos

```
USE colegio
```

Creamos la tabla donde guardaremos todos los registros

```
CREATE TABLE  alumnos (
id VARCHAR(20) NOT NULL,
nombres VARCHAR(30) NOT NULL,
apellidos VARCHAR(30) NOT NULL,
edad VARCHAR(10) NOT NULL,
dirección VARCHAR(50) NOT NULL,
primary key(id)
);
```

RECIBIMOS..

Query OK, 0 rows affected (0.25 sec)

PERO aun asi.. verificamos que la tabla se haya creado.. necio yo? xDD

SHOW TABLES;

```
+-------------------+
| Tables_in_colegio |
+-------------------+
| alumnos           |
+-------------------+
```

Okey hay tenemos nuestra tabla creada en la base de datos colegio

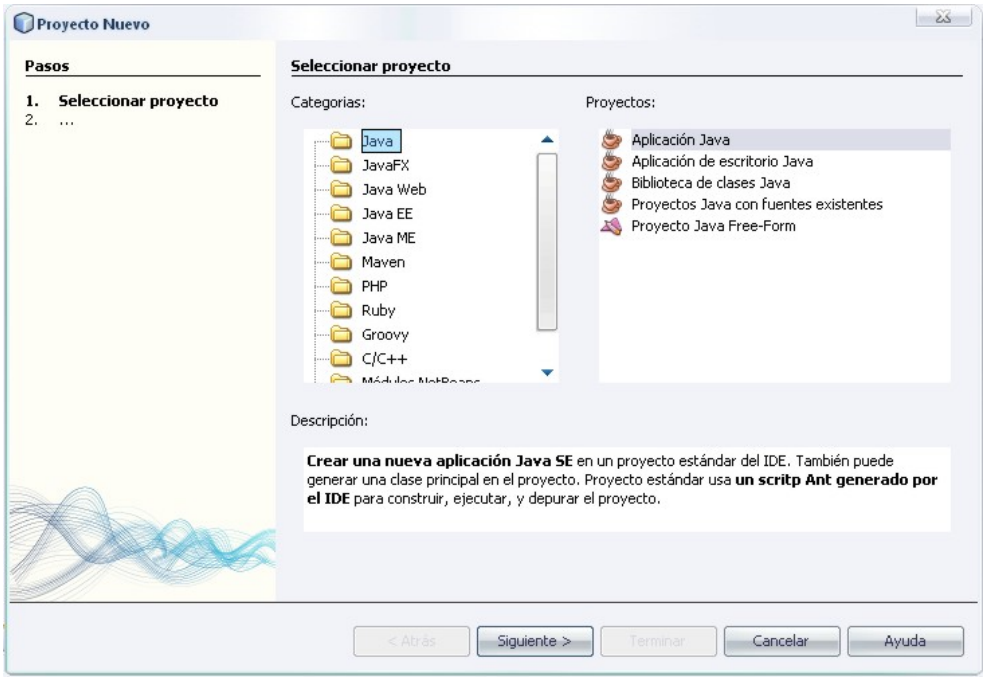
Otro tip SQL de paso miremos como está compuesta la tabla..

DESCRIBE alumos;

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| id        | varchar(20) | NO   | PRI |         |       |
| nombres   | varchar(30) | NO   |     |         |       |
| apellidos | varchar(30) | NO   |     |         |       |
| edad      | varchar(10) | NO   |     |         |       |
| direccion | varchar(50) | NO   |     |         |       |
+-----------+-------------+------+-----+---------+-------+
```

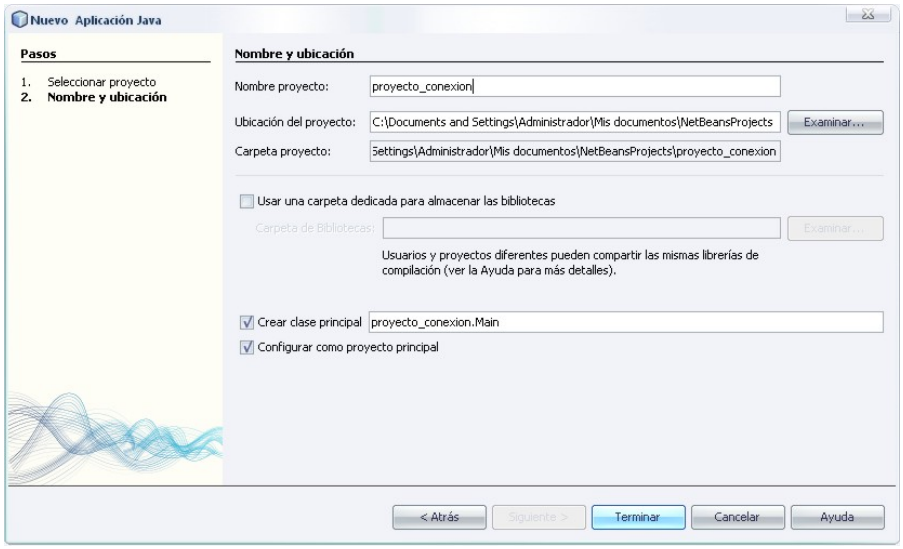
Okey hay tenemos la descripción de la tabla que hicimos anteriormente,, bue,, dejamos el SQL por ahora..

Abrimos Nuestro IDE de NetBeans y Creamos el Proyecto para Realizar las Conexiones..



Ya esta parte de creacion de proyectos la explique pero por lo general hay alumnos que llegan tarde a las clases xDD sigamos..

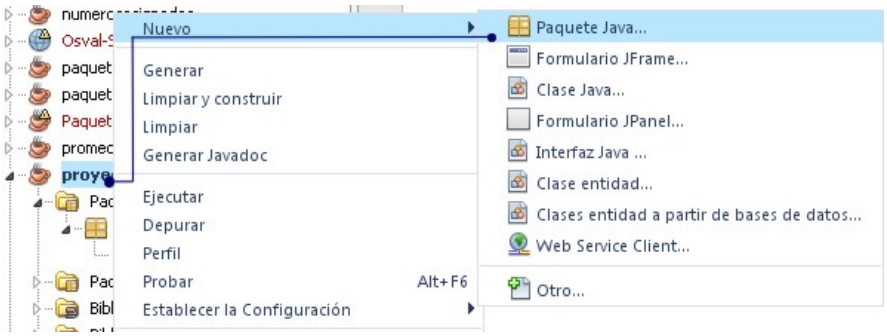
En este caso yo lo llame proyecto_conexion ps alusivo a lo que estamos haciendo



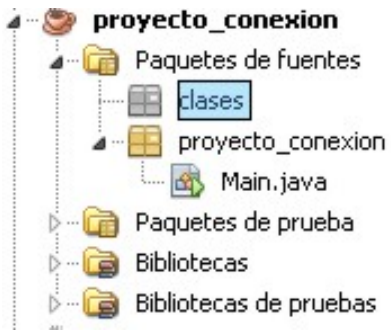
Para organizar los elementos en nuestro proyecto separamos las clases de los formularios

Agregamos otro paquete al proyecto como en la siguiente imagen..

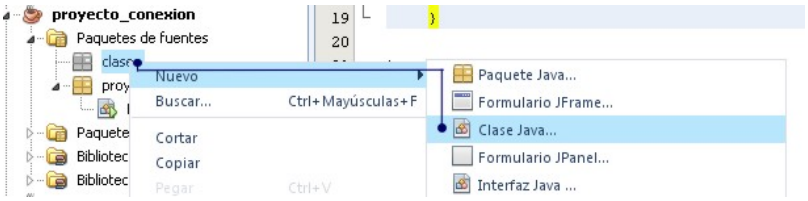
Clic derecho sobre el proyecto → nuevo → paquete java



Listo entonces en el paquete clase si… guardaremos todo los que son clases y en el paquete proyecto_conexion guardaremos todo lo que serán formularios =)



Agregamos Una clase—en el paquete clases la cual se encargara de realizar la conexión a la base de datos.



Yo la he llamado conexionMySQL

Clic en terminar.

CODIGO DE LA CLASE conexionMySQL:

```
package clases;
import java.security.Principal;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import java.sql.Statement;
public class conexionMySQL {
public static Connection conexion;
public Statement sentencia;
private Principal prin;
public ResultSet resultado;
public void conectarBaseDeDatos() {

try {

final String CONTROLADOR = "org.gjt.mm.mysql.Driver";
//Al igual que en el ejemplo anterior Declaramos que driver vamoa a utilizar

Class.forName( CONTROLADOR );

conexion = DriverManager.getConnection("jdbc:mysql://localhost/colegio","root","root");
//hacemos la conexion en este caso en localhost por que la bd esta en nuestro pc
//especificando usuario y contraseña
sentencia = conexion.createStatement();
//variable sentencia encargada de las funciones hacia la db
}

catch (ClassNotFoundException ex1) {

ex1.printStackTrace();
javax.swing.JOptionPane.showMessageDialog(null,"Error Carga Driver");
//encaso de errores referentes a driver error de carga, no encontrarlo etc
System.exit(1);
}
catch (SQLException ex)
{
ex.printStackTrace();
javax.swing.JOptionPane.showMessageDialog(null,"Error Creacion Statement");
//encaso de no encontrar la bd
System.exit(1);
}
}
 public void desconectarBaseDeDatos() {
// metodo de desconexion
try {
if (conexion != null ) {
if(sentencia != null) {
//si la conexion devuelve valores nulos
sentencia.close();
}
conexion.close();
```

```
}
}
 catch (SQLException ex) {
ex.printStackTrace();
}
}
}
```

Recordarte Guarda muy bien este code ya que para realizar conexiones en un futuro te sera muy util solo bastaria cambiar el nombre de la base de datos a la que ballas a utilizar

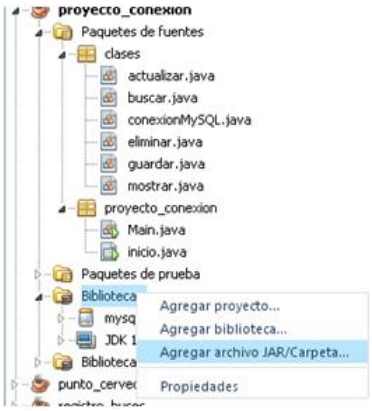
Luego nos vamos a Bibliotecas y añadimos nuestro driver

Clic derecho en el paquete bibliotecas → añadir Biblioteca

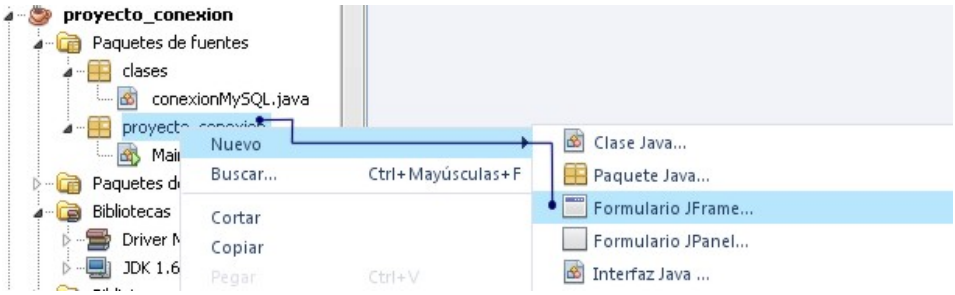
Encaso de No tener el Driver Instalado no sería nada raro #D

http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.13.tar.gz

Lo descargamos obvio que ya no sería clic derecho sobre Bibliotecas añadir bibliotecas
Ahora seria clic derecho sobre bibliotecas à Añadir archivo JAR/Carpeta



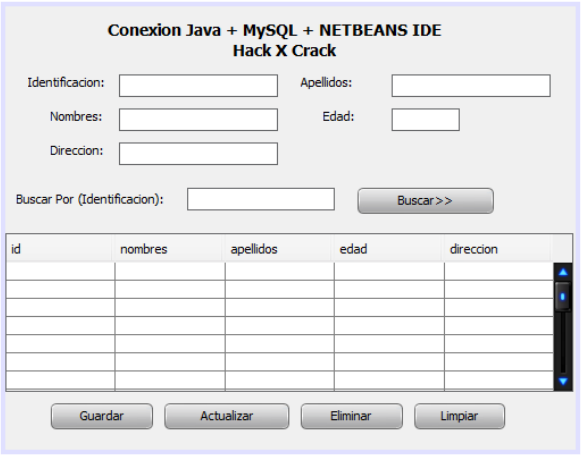
Buscamos La carpeta mysql-connector-java-5.1.13 y añadimos el archivo JAR
Ahora agregamos El formulario el cual tendrá las funciones directas Guardar, Eliminar,Buscar,Modificar y cualquier otra que se nos ocurra xDD



Yo lo llame inicio.

Y lo diseñamos de la siguiente manera acorde a nuestros datos de la bd Asi:

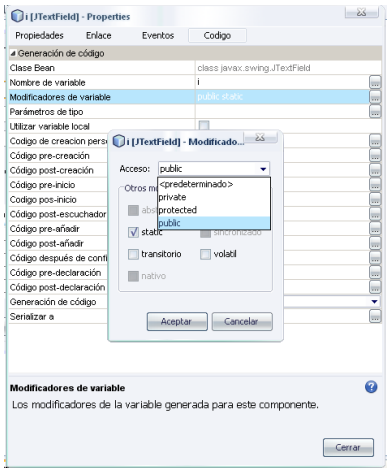
Nombre Tipo_Componente
i JTextField
n JTextField
a JTextField
e JTextField
buscar_ident JTextField
Mitabla JTable
5 JButton's



Se Explica Solo no?
Al crear toda nuestra linda interfaz se ...s.."Todos he?"

Damos Clic Derecho al Componente en este ejemplo a "i" Que es un Jtextfield nos vamos a propiedades luego a la pestaña código luego nos ubicamos en modificadores de variables y escogemos public static

Como en la siguiente imagen >

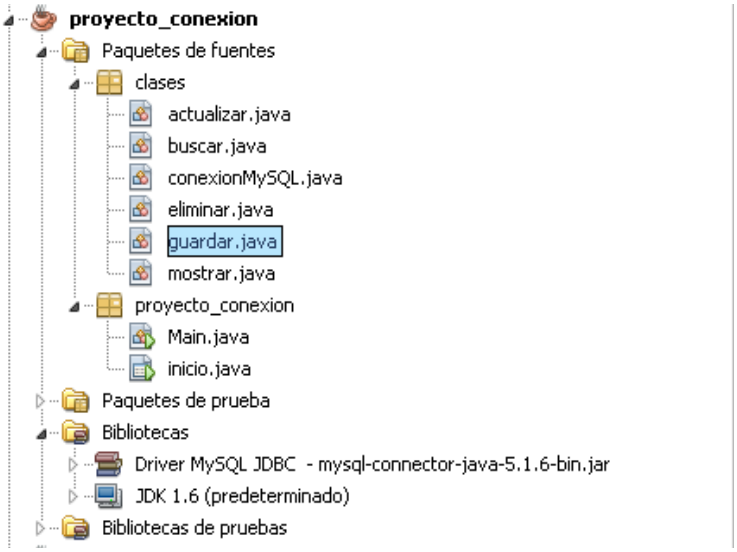


Igual Con Los Demás Componentes..

Nota:

Un Acceso público es para Que desde Cualquier parte de nuestro proyecto podamos acceder a estos componentes ya sea asignándole valores o extrayendo valores de estos.

Creando Las Clases Guardar, Eliminar, Actualizar, buscar y Mostrar



CODIGO DE LA CLASE GUARDAR

```
package clases;

import javax.swing.JOptionPane;

public class guardar {
   clases.conexionMySQL bd = new clases.conexionMySQL();//se declara un alias bd que es igual que decir conexionMySQL
 public void registrar_alumno(String id, String nombres, String apellidos, String edad, String direccion){
   //aqui se halan las variables declaradas en el formulario de inicio que traen los valores de las cajas de texto
      //hehe ahora las declaramos trankis
      try{
    bd.conectarBaseDeDatos();
    //conectamos
    bd.sentencia.execute("insert into alumnos(id,nombres,apellidos,edad,direccion) values ("+id+",'"+nombres+"','"+apellidos+"','"+edad+"','"+direccion+"')");
    //insertamos los datos
    JOptionPane.showMessageDialog(null, "El Alumno Fue Registrado Correctamente", " .::MENSAJE::.", JOptionPane.INFORMATION_MESSAGE);

   }catch (Exception e){}
  }
}
```

CODIGO DE LA CLASE eliminar

```
package clases;
import javax.swing.JOptionPane;

public class eliminar {
    clases.conexionMySQL bd = new clases.conexionMySQL();
//Denuevo el alias a la conexion obvio hay que llamarla para conectar #D
   public void eliminar_alumno(String ide){
try{
     bd.conectarBaseDeDatos();
     bd.sentencia.execute("Delete From alumnos WHERE id= ('"+ide+"')");
 //ejecutamos un simple SQL borrando de la tabla alumno donde la ide sea igual a la ide
     JOptionPane.showMessageDialog(null, "El Alumno Fue Eliminado Correctamente", " .::MENSAJE::.",
JOptionPane.INFORMATION_MESSAGE);
     }catch (Exception e){}
  }
}
```

CODIGO DE LA CLASE ACTUALIZAR

```
package clases;
import javax.swing.JOptionPane;
public class actualizar {
    clases.conexionMySQL bd = new clases.conexionMySQL();
   public void actualizar_alumno(String elegir_id, String no, String ape, String ed, String dir){
//Estas variables vienen delformulario de inicio pronto las declararemos /D
try{
     bd.conectarBaseDeDatos();
     bd.sentencia.execute("UPDATE alumnos SET
nombres='"+no+"',apellidos='"+ape+"',edad='"+ed+"',direccion='"+dir+"' WHERE id= '"+elegir_id+"'");
//hey!! Actualiza los datos donde id sea igual a elegir_id (H)
     JOptionPane.showMessageDialog(null, "Los Datos Del Alumno Fueron Actualizados Correctamente", "
.::MENSAJE::.", JOptionPane.INFORMATION_MESSAGE);
     }catch (Exception e){}
  }
}
```

Como vemos la estructura de las clases van siendo identicas solo vamos cambiando las instrucciones SQL y algunas nuevas funciones .)

CODIGO DE LA CLASE BUSCAR

```
package clases;

import javax.swing.JOptionPane;
public class buscar {

 clases.conexionMySQL bd = new clases.conexionMySQL();//Esta linea es la protagonista de la pelicula xD
```

judgeNow!

juez de programación online

Operativo a partir de enero de 2011

Aprende programación en 3 sencillos pasos

Con judgeNow!, el aprendizaje de los principales lenguajes de programación, queda al alcance de todos..y gratis! Estos son los 3 unicos pasos necesarios para pode r gozar de todaslas ventajas del nuevo juez de programación de Hack x Crack.

Además, si vas muy perdido y lo deseas, puedes convertirte en usuario premium por unas pequeñas cuotas mensualeso anuales y obtener así ayuda en línea y/o por mail.

1. Registrate

Elige el nombre que te acompañará durante tu aprendizaje!

2. Practicas

Tenemos los mejores tutoriales de cada lenguaje en línea!

Y todo gratis! Solo loguéate y empieza a empaparte de conocimientos.

Pronto estarás listo para ponerte aprueba con los cientos de ejercicios que te propondremos!

3. Examínate

Tu eliges que ejercicios quieres realizar, de que nivel quieres que sean, y cuando hacerlos!

Una vez terminados, basta con enviarlo a alguno de nuestros servidores para que lo analicen y te den su veredicto: el juez nunca se equivoca!

Exprime tu cerebro e intenta llegar a los ejercicios propuestos para los más veteranos!

Mucha suerte!

Tutoriales, ejercicios, soporte en línea, soporte vía mail...que más se puede pedir??

Y además **podrás participar en nuestros torneos y concursos** de programación y en torneos externos que iremos publicando !

Y si todo va bien, podremos impartir cursos de iniciación de **programación en el campo de la seguridad informática!**

*EL servicio está fase beta en estos momentos, estará 100% operativo a principios de 2011

www.judgenow.org

```
  public void buscar_alumno(String buscar_id) {
   try{
       bd.conectarBaseDeDatos();
        bd.resultado = bd.sentencia.executeQuery( "SELECT * FROM alumnos WHERE id = '"+buscar_id+"'");
        if( bd.resultado != null ) {
        while(bd.resultado.next() ) {
          proyecto_conexion.inicio.i.setText(""+bd.resultado.getString("id"));
          proyecto_conexion.inicio.n.setText(""+bd.resultado.getString("nombres"));
          proyecto_conexion.inicio.a.setText(""+bd.resultado.getString("apellidos"));
          proyecto_conexion.inicio.e.setText(""+bd.resultado.getString("edad"));
          proyecto_conexion.inicio.d.setText(""+bd.resultado.getString("direccion"));
         }
        }
         else
        JOptionPane.showMessageDialog(null,"Ningun dato Encontrado","Mensaje de
Informacion",JOptionPane.PLAIN_MESSAGE);
     }catch (Exception e){}
  }
}
```

Fijar que

Aquí buscamos en la tabla alumnos la identificacion que este almacenada en buscar_id

y decimos mediante esta cadena

```
 if( bd.resultado != null ) {
```

en palabras rusticas si encuentra algo, osea si la consulta no biene vacia “null”

mostramos los datos relacionados con la consulta en los campos de texto

```
proyecto_conexion.inicio.i.setText(""+bd.resultado.getString("id"));
```

vemos que especificamos la ruta proyecto.formulario.componente

CODIGO DE LA CLASE MOSTRAR

Encargada de cargar los datos de la tabla de Mysql en la Tabla del Programa

```
package clases;

public class mostrar {
 clases.conexionMySQL bd = new clases.conexionMySQL();
     //el famoso alias alusivo a la conexion
  int i=0; //variable i para el control de ciclo while
  public void mostrar_datos(){
   try{
       bd.conectarBaseDeDatos();
      i=0;
       bd.resultado = bd.sentencia.executeQuery( "SELECT * FROM alumnos" );
      //seleccionamos todos de la tabla alumnos
       if( bd.resultado != null ) {
      //si la consulta no biene vacia
           while(bd.resultado.next() ) {
```

```
        proyecto_conexion.inicio.mitabla.setValueAt(""+bd.resultado.getString("id"),i, 0);
        proyecto_conexion.inicio.mitabla.setValueAt(""+bd.resultado.getString("nombres"),i, 1);
        proyecto_conexion.inicio.mitabla.setValueAt(""+bd.resultado.getString("apellidos"),i, 2);
        proyecto_conexion.inicio.mitabla.setValueAt(""+bd.resultado.getString("edad"),i, 3);
        proyecto_conexion.inicio.mitabla.setValueAt(""+bd.resultado.getString("direccion"),i, 4);
       //Asignamos los valores de la consulta a la tabla
        i=i+1;
       }
      }
    }catch (Exception e){}
  }
}
```

SEGUIREMOS AHORA CON LA ASIGNACIÓN DE FUNCIONES A LOS BOTONES DEL FORMULARIO DE INICIO

pero primeramente será asignar un alias a las clases para que estas puedan ser llamadas desde el formulario junto a sus funciones asi:

damos clic en el boton souce en el formulario inicio

```
public class inicio extends javax.swing.JFrame { //Esto es para que sepas donde debes ubicarte =)
clases.guardar guardar= new clases.guardar();
clases.buscar buscar= new clases.buscar();
clases.eliminar eliminar= new clases.eliminar();
clases.mostrar mostrar= new clases.mostrar();
clases.actualizar actualizar= new clases.actualizar();
/*Listo las tenemos referenciadas. Ahora Ahora para Que Cuando el Formulario Corra ( Se Ejecute ) la
tabla carge los datos de la tabla de MYSQL */
public inicio() { //Esto es para que te guies Recuerda que estamos en el formulario inicio
      initComponents();
      mostrar.mostrar_datos();
 }
//Declaramos variables para realizar todas las funciones:
//Debajo de @SuppressWarnings("unchecked")
String id;
String nombres;
String apellidos;
String edad;
String direccion;
String elegir_id;
```

SEGUIMOS CON LAS FUNCIONES DE LOS BOTONES

BOTON GUARDAR

```
        id = i.getText();
        nombres = n.getText();
        apellidos = a.getText();
        edad = e.getText();
        direccion = d.getText();
//capturamos los datos digitados en los campos de texto
        guardar.registrar_alumno(id, nombres, apellidos, edad, direccion);
//Aquí lo que hacemos es enviar los datos de los campos de texto en variables a la clase guardar
 i.setText("");
n.setText("");
a.setText("");
e.setText("");
d.setText("");
//Despues de Guardar Limpiamos los campos
```

BOTON ACTUALIZAR

```
 elegir_id = buscar_ident.getText();
 nombres = n.getText();
 apellidos = a.getText();
 edad = e.getText();
 direccion = d.getText();
 actualizar.actualizar_alumno(elegir_id, nombres, apellidos, edad, direccion);
```

BOTON ELIMINAR

```
eliminar.eliminar_alumno(buscar_ident.getText());
i.setText("");
n.setText("");
a.setText("");
e.setText("");
d.setText("");
buscar_ident.setText(""); //Solo se limpian los campos
```

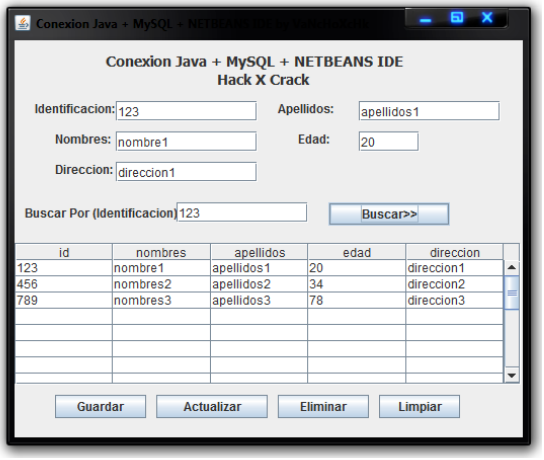
BOTON BUSCAR

```
 buscar.buscar_alumno(buscar_ident.getText());
```

BOTON LIMPIAR

```
i.setText("");
n.setText("");
a.setText("");
e.setText("");
d.setText("");
buscar_ident.setText("");
```

Una ultima screen del programa en ejecución:



Aqui termina la conexion con MYSQL, con esto hemos logrado hacer las funciones primordiales de un gran proyecto como lo son guardar,eliminar,actualizar,buscar de seguro ha sido una gran ayuda para todo aquel que quiere iniciarse en java y sacarle el mayor provecho a un lenguaje tan versatil y poderoso como este.
Recuerda..

"El que AMA lo que hace, aun arrodillado alcanza las estrellas"
VaNcHoXcHk

Gracias.