

Universidad Andres Bello  
Facultad de Ingeniería

**Proyecto Semestral**

**Carrera: Ingeniería en Computación e Informática**

**Asignatura: Programación II (INS124)**

**Profesor: José Souza**

Alumnos:

* Rodrigo Gonzalez A.
* Jorge Sepúlveda S.

Jueves, 15 de junio de 2017

**ÍNDICE**

**OBJETIVOS 3**

**INSTRUCCIONES 3**

**INTEGRANTES 3**

**ENUNCIADO 3**

**REQUERIMIENTOS 3**

**SOLUCIÓN 4**

**BASE DE DATOS, TABLAS Y RELACIONES 5**

**AMBIENTE GRÁFICO (VENTANAS) Y ACCESO A BASE DE DATOS 7**

**VISTAS 13**

**CUADRATURAS 14**

1. **OBJETIVO**
   1. Aplicar metodología de implementación de Bases de Datos física.
   2. Hacer uso de lenguaje SQL y Java con acceso a Base de Datos para satisfacer requerimientos planteados.
2. **INSTRUCCIONES:**
3. Desarrollar la aplicación en Java para satisfacer los requerimientos planteados. cada ítem usando el motor de Base de Datos. Por cada ítem en el lugar de la respuesta copiar y pegar la imagen correspondiente al desarrollo.
4. Presente el informe impreso con portada, índice enunciado y desarrollo.
5. En una carpeta con el nombre de los integrantes, guarde el proyecto y copia del informe.
6. Las copias serán evaluadas con nota 2.0, independiente de quien copia. Cada grupo es responsable de resguardar su trabajo.
7. **INTEGRANTES**
8. Nombre: Rodrigo Gonzalez A.  
   Rut:
9. Nombre: Jorge Sepúlveda S.  
   Rut:
10. **ENUNCIADO**

La empresa “Viaje Feliz SA” requiere una aplicación para controlar la venta de pasajes interprovinciales.

1. **Los atributos de los buses son:**
   * Patente, modelo año, cantidad\_asientos.
2. **Los viajes se planifican considerando:**
   * CodigoViaje, fecha, hora, bus, conductor, origen, destino, precio, CantAsientos, PasajeDisponibles, estado. (p: pendiente, r: realizado, a: abortado)
3. **Venta de pasajes:**
   * Folio, CodigoViaje, FechaVta, Rut\_Vendedor, Precio.
4. **Vendedor:**
   * Rut\_Vendedor, nombre, dirección.
5. **REQUERIMIENTOS:**

Se solicita lo siguiente:

1. **Ítem I.-** Crear la base de datos, las tablas y sus relaciones. (10 pts)
2. **Ítem II.-** Desarrollar en java el ambiente gráfico con acceso a la base de datos. (5 pts)
3. **Ítem III.-** Implementar un menú principal. (5 pts)
4. **Ítem IV.-** Implementar ventanas y métodos para:
   1. Mantención de Tablas:
      * CRUD Vendedor, (validar rut) (15 pts)
      * Crear Bus (5 pts)
      * Crear Viaje (5 pts)
      * Venta de Pasajes (10 pts)

No se pueden vender más pasajes que los asientos del bus. Por cada venta se actualiza la disponibilidad de asientos.

Las ventanas hijas deben retornar a la principal.

1. **Ítem V.-** Vistas para generar los siguientes requerimientos: (45 pts)
2. Presentar los resultados de las vistas en el proceso (programa).
3. Cuadratura de Caja: Listar todas las ventas de un vendedor en un día determinado, Totalizar el $$. (Usar tablas: Viaje, Venta\_Pasajes, Vendedor).
   1. Listar todos los pasajes vendidos de un viaje determinado. (Usar tablas: Viaje, Venta\_Pasajes).
   2. Listar las ventas totales en un mes determinado. Imprimir subtotales por Viaje. (Usar tablas: Viaje, Venta\_Pasajes)
4. **SOLUCIÓN**

**ANTECEDENTES**

Dado que el desarrollo del proyecto consideraba trabajar con un motor de base de datos y Java como lenguaje de programación, utilizamos las siguientes aplicaciones para desarrollar dicho proyecto:

**XAMPP (versión 5.6.30)**

* Es un paquete de instalación independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

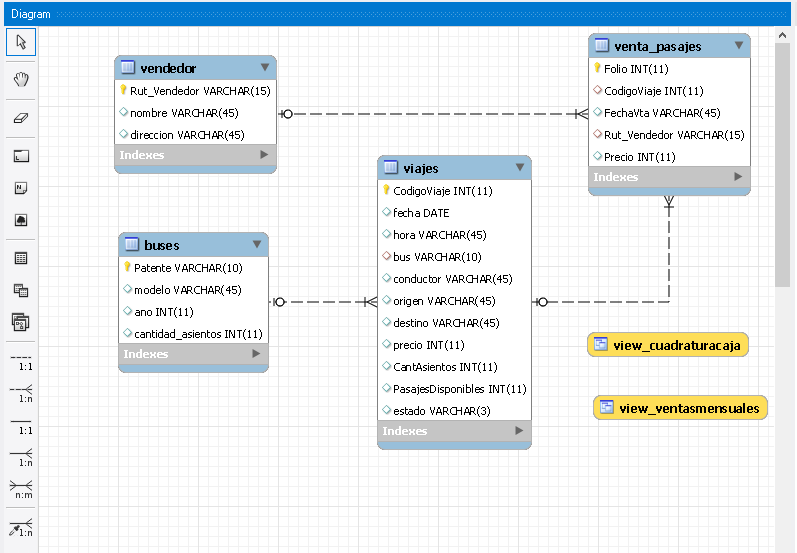
**Netbeans IDE (versión 8.2)**

* Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. Gracias a que soporta otros lenguajes de programación, puede realizar aplicaciones también para ellos, como HTML,PHP, C/C+\*, etc.

**MySQL Workbench 6.1.7 CE**

* Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL. Es el sucesor de DBDesigner 4 de fabFORCE.net, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle

1. **ITEM I – Crear la base de datos, las tablas y sus relaciones**
2. Para desarrollar el proyecto utilizamos las siguientes tablas:
   * vendedor, viajes, venta\_pasajes y buses.
3. El diagrama de entidad relación entre las tablas es la siguiente:



1. Una vez definida las relaciones entre las tablas a utilizar se crearon estas directamente en el motor de bases de datos con los siguientes comandos o bien, utilizando un script SQL para automatizar dicha tarea:

--

-- Table structure for table `Vendedor`

--

DROP TABLE IF EXISTS `Vendedor`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `Vendedor` (

`Rut\_Vendedor` varchar(15) NOT NULL,

`nombre` varchar(45) DEFAULT NULL,

`direccion` varchar(45) DEFAULT NULL,

PRIMARY KEY (`Rut\_Vendedor`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `buses`

--

DROP TABLE IF EXISTS `buses`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `buses` (

`Patente` varchar(10) NOT NULL,

`modelo` varchar(45) DEFAULT NULL,

`ano` int(11) DEFAULT NULL,

`cantidad\_asientos` int(11) DEFAULT NULL,

PRIMARY KEY (`Patente`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `venta\_pasajes`

--

DROP TABLE IF EXISTS `venta\_pasajes`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `venta\_pasajes` (

`Folio` int(11) NOT NULL,

`CodigoViaje` int(11) DEFAULT NULL,

`FechaVta` varchar(45) DEFAULT NULL,

`Rut\_Vendedor` varchar(15) DEFAULT NULL,

`Precio` int(11) DEFAULT NULL,

PRIMARY KEY (`Folio`),

KEY `fk\_CodigoViaje\_idx` (`CodigoViaje`),

KEY `fk\_Rut\_Vendedor\_idx` (`Rut\_Vendedor`),

CONSTRAINT `fk\_CodigoViaje` FOREIGN KEY (`CodigoViaje`) REFERENCES `viajes` (`CodigoViaje`) ON DELETE NO ACTION ON UPDATE NO ACTION,

CONSTRAINT `fk\_Rut\_Vendedor` FOREIGN KEY (`Rut\_Vendedor`) REFERENCES `Vendedor` (`Rut\_Vendedor`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Table structure for table `viajes`

--

DROP TABLE IF EXISTS `viajes`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `viajes` (

`CodigoViaje` int(11) NOT NULL,

`fecha` date DEFAULT NULL,

`hora` varchar(45) DEFAULT NULL,

`bus` varchar(10) DEFAULT NULL,

`conductor` varchar(45) DEFAULT NULL,

`origen` varchar(45) DEFAULT NULL,

`destino` varchar(45) DEFAULT NULL,

`precio` int(11) DEFAULT NULL,

`CantAsientos` int(11) DEFAULT NULL,

`PasajesDisponibles` int(11) DEFAULT NULL,

`estado` varchar(3) DEFAULT NULL,

PRIMARY KEY (`CodigoViaje`),

KEY `fk\_buses\_idx` (`bus`),

CONSTRAINT `fk\_buses` FOREIGN KEY (`bus`) REFERENCES `buses` (`Patente`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

1. **Ítem II - Desarrollar en java el ambiente gráfico con acceso a la base de datos**

Para el proyecto utilizamos la clase llamada ClassMysql.java que realiza la conexión al motor de base de datos MySQL y posteriormente carga la base de datos “Proyecto”

1. La clase es la siguiente:

package proyecto;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author rgonzalez

\*/

public class ClassMysql {

static Statement stmt;

static Connection con;

static String url=("jdbc:mysql://127.0.0.1:3306/Proyecto?user=root&password=@-JASS-newPsrotax\*21");

public ClassMysql() {

this.con= getConnection();

}

public Connection getCon (){

return this.con;

}

public static Connection getConnection()

{

try {

Class.forName("com.mysql.jdbc.Driver"); // conectamos a la base de datos

} catch(java.lang.ClassNotFoundException e) {

System.err.print("ClassNotFoundException, no se encuentra el Driver: ");

System.err.println(e.getMessage());

}

try {

con =DriverManager.getConnection(url);

} catch(SQLException ex) {

System.err.println("SQLException: " + ex.getMessage());

}

return con;

}

public void EjecutarQuery(String Query)

{

Connection con = getConnection();

try {

stmt = con.createStatement();

stmt.executeUpdate(Query);

stmt.close();

con.close();

} catch(SQLException ex) {

System.err.println("SQLException: " + ex.getMessage());

JOptionPane.showMessageDialog(null, ex.getMessage());

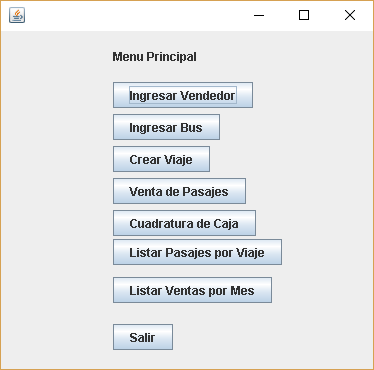
}

}

}

1. **Ítem III - Implementar un menú principal**

El proyecto consta de la siguiente pantalla principal:

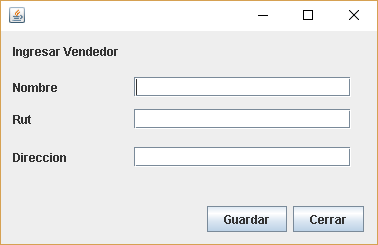


Cada botón tiene un método que permite realizar una acción dependiendo de lo que se necesite realizar.

1. **Ítem IV - Implementar ventanas y métodos**

El proyecto consta de las siguientes ventanas para solucionar los requerimientos iniciales

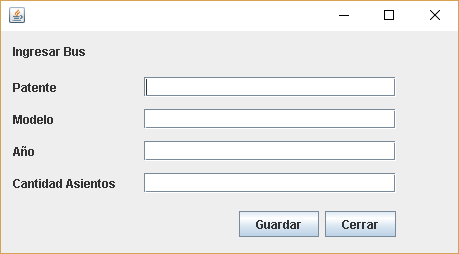
1. Ventana para ingresar los datos de un nuevo vendedor:



Los métodos utilizados son:

* + btnGuardarActionPerformed()
  + validaRut()
  + validaVendedor()

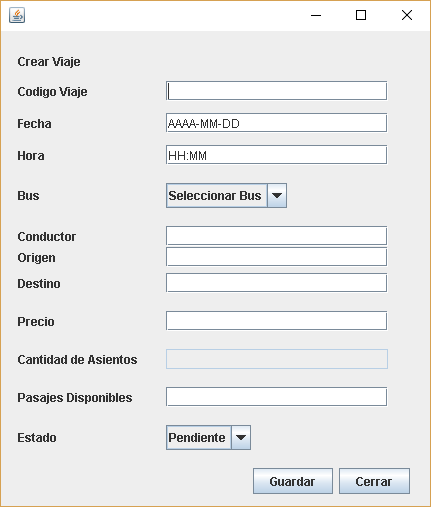
1. Ventana para ingresar los datos de un nuevo bus:



Los métodos utilizados son:

* btnGuardarActionPerformed()
* validarPatente()

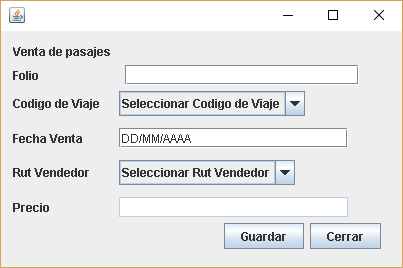
1. Ventana para crear un nuevo viaje:



Los métodos son:

* btGuardarActionPerformed()
* GetEstado()
* LlenarBuses()
* ValidarCodViaje()
* GetFechaActual()
* GenerarCodigoViaje()

1. Ventana para vender un pasaje:



Los métodos utilizados son:

* btnGuardarActionPerformed()
* ValidarCampos()
* ValidarFolio(Folio)
* ValidarCantAsientos()
* LlenarCodigosViajes()
* LlenarRutVendedores()
* GetCantAsientosDisponibles()
* ObtenerPrecio()
* GenerarFolio()
* Ponerfolio()
* GetFechaActual()

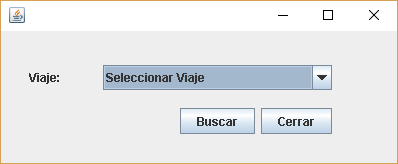
1. Ventana para la cuadratura de la caja:



Los métodos utilizados son:

* jButton1ActionPerformed()
* LlenarRutVendedores()
* LlenarDias()
* CuadraturaCaja()
* LlenarTabla()

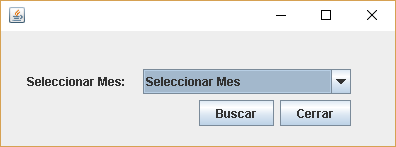
1. Ventana para listar los pasajes por viaje:



Los métodos utilizados son:

* jButton1ActionPerformed()
* LLenarCodViaje()
* PasajesPorViajes()

1. Ventana para listar las ventas por mes:



Los métodos utilizados son:

* jButton1ActionPerformed()
* LlenarDias()
* VentasMensuales()

.

1. **Ítem V - Vistas para generar los requerimientos**

Para el ítem de las vistas se utilizaron las siguientes estructuras de vistas:

1. Estructura “view\_cuadraturacaja”:

--

-- Temporary view structure for view `view\_cuadraturacaja`

--

DROP TABLE IF EXISTS `view\_cuadraturacaja`;

/\*!50001 DROP VIEW IF EXISTS `view\_cuadraturacaja`\*/;

SET @saved\_cs\_client = @@character\_set\_client;

SET character\_set\_client = utf8;

/\*!50001 CREATE VIEW `view\_cuadraturacaja` AS SELECT

1 AS `Folio`,

1 AS `CodigoViaje`,

1 AS `FechaVta`,

1 AS `Rut\_Vendedor`,

1 AS `precio`,

1 AS `nombre`,

1 AS `fecha`,

1 AS `hora`,

1 AS `bus`,

1 AS `estado`,

1 AS `DiaVenta`\*/;

SET character\_set\_client = @saved\_cs\_client;

1. Estructura “view\_ventasmensuales”

--

-- Temporary view structure for view `view\_ventasmensuales`

--

DROP TABLE IF EXISTS `view\_ventasmensuales`;

/\*!50001 DROP VIEW IF EXISTS `view\_ventasmensuales`\*/;

SET @saved\_cs\_client = @@character\_set\_client;

SET character\_set\_client = utf8;

/\*!50001 CREATE VIEW `view\_ventasmensuales` AS SELECT

1 AS `Folio`,

1 AS `CodigoViaje`,

1 AS `FechaVta`,

1 AS `Rut\_Vendedor`,

1 AS `Precio`,

1 AS `Mes`\*/;

SET character\_set\_client = @saved\_cs\_client;

1. **CUADRATURAS**
2. Para mostrar las ventas que ha realizado un vendedor en un día, se ha utilizado la siguiente consulta SQL:
   * select \* from view\_CuadraturaCaja WHERE `Rut\_Vendedor`='"+Rut+"' AND DiaVenta='"+DiaVenta+"'
3. Para mostrar los pasajes por viajes, se ha utilizado la siguiente consulta SQL:
   * select \* from `venta\_pasajes` WHERE `CodigoViaje`='"+CodViaje+"'
4. Para mostrar las ventas del mes con el subtotal, se ha utilizado la siguiente consulta SQL:
   * (select Folio, CodigoViaje, FechaVta, Rut\_Vendedor, Precio , Mes from `view\_VentasMensuales` where mes LIKE '%"+Mes+"') UNION (SELECT NULL,CodigoViaje,NULL,'Subtotal', CodigoViaje,SUM(precio) as Total from `view\_VentasMensuales` where mes LIKE '%"+Mes+"' GROUP BY CodigoViaje) ORDER BY CodigoViaje,Folio DESC