

Universidad ORT Uruguay

Facultad de Ingeniería

Escuela de Tecnología

Obligatorio de Diseño y Desarrollo de Aplicaciones

Juego de Dominó

Segunda Entrega: Segunda versión del programa

Entrega: 12/12/2016

Bruno DIAZ – 203056
Federico SPERONI – 165357

Carrera AP – Turno Matutino
2016

Indice

| | |
|---|----|
| Indice | 2 |
| Autoevaluación | 4 |
| Autocalificación | 4 |
| Datos de Prueba..... | 5 |
| Diagrama Conceptual del Dominio del Problema..... | 6 |
| Diagramas de Diseño | 7 |
| Diagrama de Diseño - Lógica | 7 |
| Diagrama de Diseño – Interfáz de Usuario | 8 |
| Diccionario de Clases..... | 9 |
| Clase: Usuario | 9 |
| Atributos | 9 |
| Métodos..... | 9 |
| Clase: Administrador..... | 9 |
| Métodos..... | 9 |
| Clase: Ficha | 10 |
| Atributos | 10 |
| Métodos..... | 10 |
| Clase: Apuesta..... | 10 |
| Atributos | 10 |
| Métodos..... | 10 |
| Clase: Movimiento | 11 |
| Atributos | 11 |
| Métodos..... | 11 |
| Clase: Jugador..... | 12 |
| Atributos | 12 |
| Métodos..... | 12 |
| Clase: Partida..... | 13 |
| Atributos | 13 |
| Métodos..... | 13 |
| Clase: ProcesoContadorTiempo | 16 |
| Atributos | 16 |
| Métodos..... | 16 |

| | |
|------------------------------------|----|
| Clase: SubSistemaUsuario | 16 |
| Atributos | 16 |
| Métodos | 16 |
| Clase: SubSistemaPartida | 17 |
| Atributos | 17 |
| Métodos | 17 |
| Clase: Sistema | 17 |
| Atributos | 17 |
| Métodos | 17 |
| Clase: MapeadorPartida | 18 |
| Atributos | 18 |
| Métodos | 18 |
| Clase: MapeadorAdministrador | 19 |
| Atributos | 19 |
| Métodos | 19 |
| Clase: MapeadorJugador | 20 |
| Atributos | 20 |
| Métodos | 20 |
| Clase: BaseDatos | 21 |
| Atributos | 21 |
| Métodos | 21 |
| Clase: Persistencia | 21 |
| Atributos | 21 |
| Métodos | 21 |

Autoevaluación

Consideramos que cumplimos con todos los requerimientos solicitados en el sistema, no habiendo sido necesario tocar nada de lo implementado en la primer entrega del programa debido a que ya se había cumplido también con todos los requerimientos solicitados.

Se creó un nuevo proyecto ObligatorioWeb para poder tener los casos de uso del usuario administrador vía web, realizando el acceso a través de un servidor web Tomcat integrado a NetBeans.

Autocalificación

| CONCEPTO | PTS. POSIBLES | CUMPLIMIENTO ? | AUTOCALIFICACIÓN |
|---|------------------|-------------------|------------------|
| Diagramas de clases (Detalle y concordancia del código) | 3 | SI | 3 |
| Time Out | 12 | SI | 12 |
| Persistencia | 12 | SI | 12 |
| Interfaz Web | 12 | SI | 12 |
| Auto Evaluación | 1 | SI | 1 |
| Total: | 20 | | 40 |

Datos de Prueba

| JUGADORES | | | | |
|-----------|-------------------|------------|------------------|-------|
| Cantidad | Nombre de Usuario | Contraseña | Nombre Completo | Saldo |
| 1 | a | a | Alvaro Fernández | 450 |
| 2 | b | b | Bruno Díaz | 400 |
| 3 | c | c | Carlos González | 350 |
| 4 | d | d | Darío Pérez | 300 |
| 5 | e | e | Emiliano Lasa | 250 |
| 6 | f | f | Federico Speroni | 200 |
| 7 | g | g | Gustavo Valverde | 150 |
| 8 | h | h | Hugo Collazo | 100 |
| 9 | i | i | Ismael Espósito | 50 |
| 10 | j | j | Javier Montero | 0 |

Nota: En nuestra solución, la apuesta inicial de todas las partidas es de 100 por jugador, por lo que todos los que poseen saldo menor a 100, no podrán ingresar al juego.

| ADMINISTRADORES | | | |
|-----------------|-------------------|------------|-----------------|
| Cantidad | Nombre de Usuario | Contraseña | Nombre Completo |
| 1 | a | a | Analía Pereyra |
| 2 | b | b | Blanca Moreira |
| 3 | c | c | Claudia Tabárez |
| 4 | d | d | Dilma Rousseff |
| 5 | e | e | Emilia Suárez |
| 6 | f | f | Fabiana Guerra |
| 7 | g | g | Graciela García |
| 8 | h | h | Heidy Montero |
| 9 | i | i | Ilda De León |
| 10 | j | j | Judith Barsi |

Diagrama Conceptual del Dominio del Problema

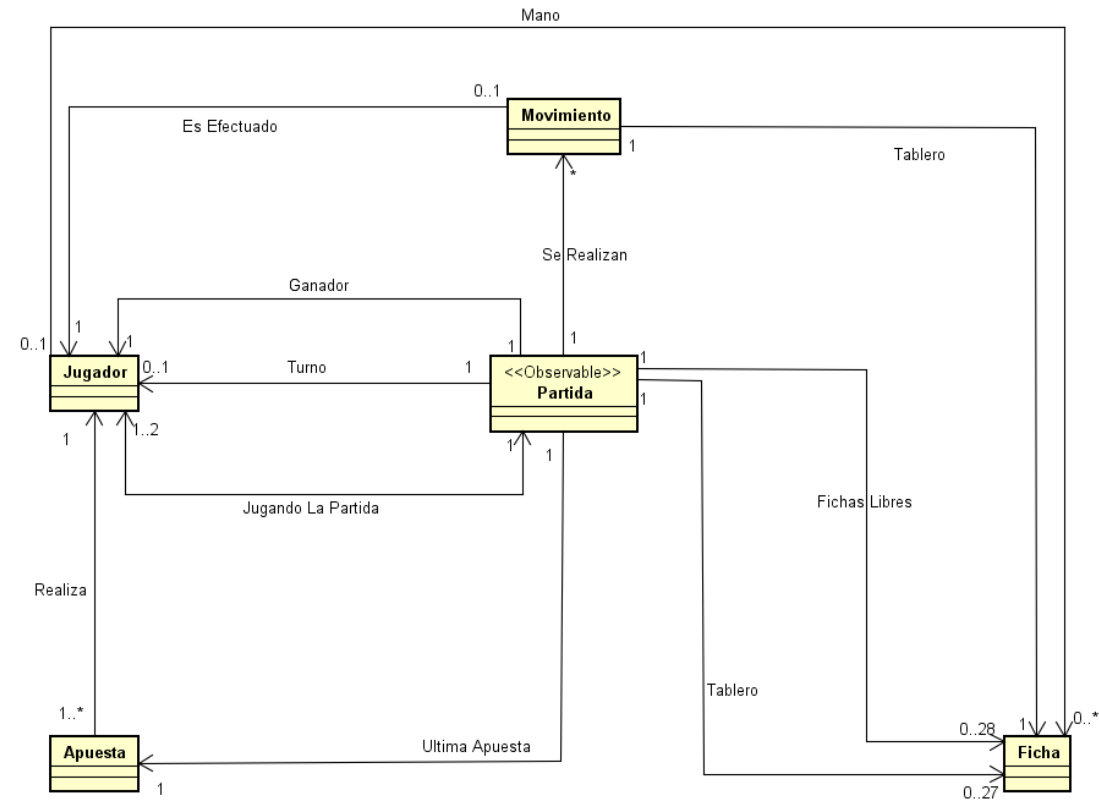


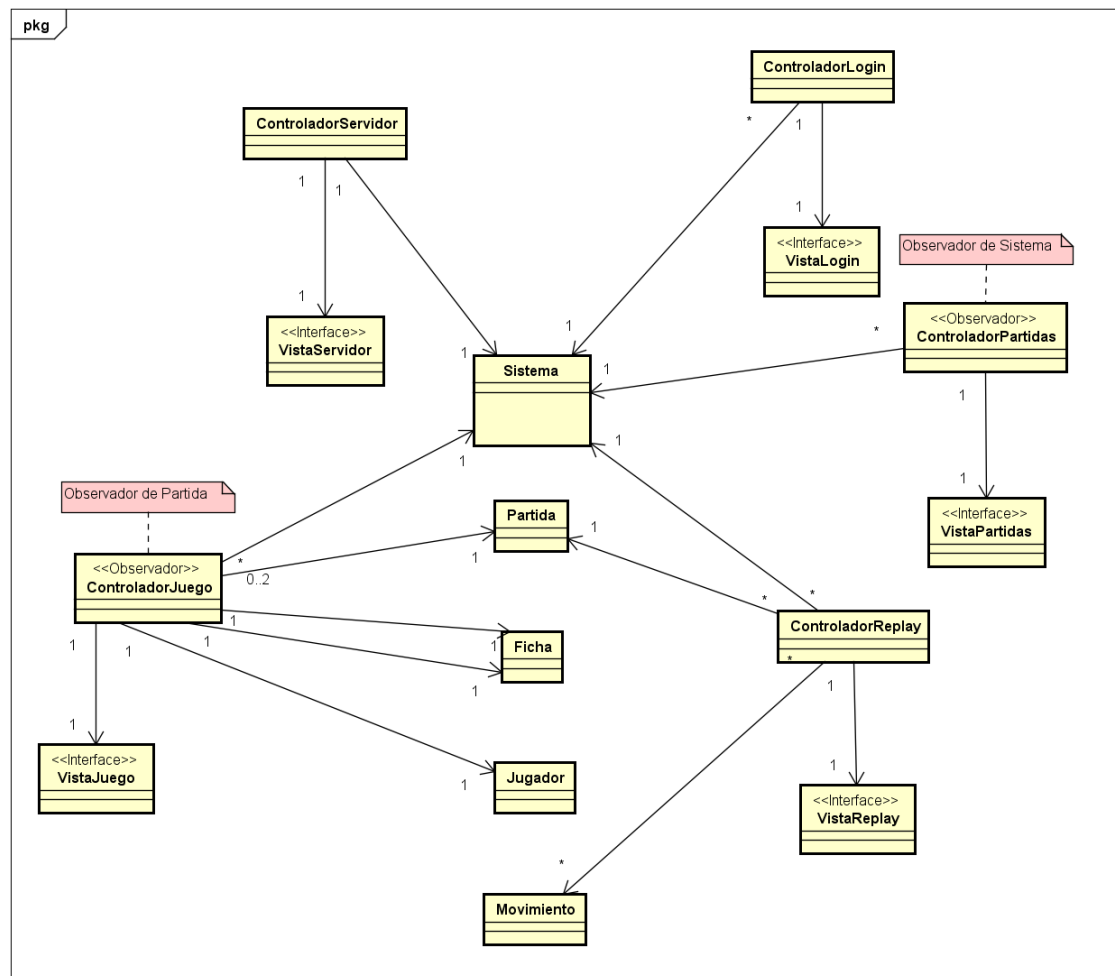
Diagrama de Diseño - Lógica

```

classDiagram
    class SubSistemaUsuario
    class SubSistemaPartida
    class Sistema {
        <<Singleton>>
        <<Fachada>>
        <<Observable>>
    }
    class Usuario
    class Administrador
    class Jugador
    class Partida {
        <<Observable>>
    }
    class Mapeador
    class MapeadorPartida
    class BaseDatos {
        <<Singleton>>
    }
    class Persistencia
    class MapeadorJugador
    class MapeadorAdministrador
    class Ficha
    class Movimiento
    class Runnable
    class ProcesoContadorTiempo

    SubSistemaUsuario "1" -- "1" Sistema
    SubSistemaPartida "1" -- "1" Sistema
    SubSistemaPartida "1" -- "1" MapeadorPartida
    MapeadorPartida "1" -- "1" BaseDatos
    BaseDatos ..> Persistencia
    Persistencia --> Mapeador
    Mapeador "1" -- "0..*" Administrador
    Administrador "1" -- "0..*" MapeadorAdministrador
    MapeadorAdministrador --> Mapeador
    Mapeador "1" -- "0..*" Jugador
    Jugador "0..*" -- "0..*" SubSistemaUsuario : Logueados
    Jugador "0..*" -- "0..*" SubSistemaPartida : Todos
    Jugador "0..*" -- "0..*" Partida : Ultima Apuesta
    Jugador "0..1" -- "0..1" Partida : Proxima Partida
    Jugador "0..1" -- "0..1" Partida : Turno
    Jugador "0..1" -- "0..1" Partida : Ganador
    Jugador "0..2" -- "0..1" Partida : Jugando
    Jugador "1" -- "0..1" MapeadorJugador
    MapeadorJugador --> Mapeador
    MapeadorJugador "1" -- "0..1" Jugador
    MapeadorJugador "1" -- "0..1" Ficha
    MapeadorJugador "1" -- "0..1" Movimiento
    MapeadorJugador "1" -- "0..1" Partida : Es Realizado
    MapeadorJugador "1" -- "0..1" Partida : Tiempo Apuesta
    MapeadorJugador "1" -- "0..1" Partida : Tiempo Turno
    MapeadorJugador "1" -- "0..1" Partida : Libres
    MapeadorJugador "1" -- "0..1" Partida : Tablero
    MapeadorJugador "1" -- "0..1" Partida : Primera del Tablero
    MapeadorJugador "1" -- "0..1" Partida : Ultima del Tablero
    MapeadorAdministrador "1" -- "0..*" Ficha
    Ficha "0..1" -- "0..1" Jugador : Mano
    Ficha "0..1" -- "0..1" Partida : Es Efectuada
    Ficha "0..1" -- "0..1" Partida : Turno
    Ficha "0..1" -- "0..1" Partida : Ganador
    Ficha "0..1" -- "0..1" Partida : Tablero
    Ficha "0..1" -- "0..1" Partida : Libres
    Ficha "0..1" -- "0..1" Partida : Tablero
    Ficha "0..1" -- "0..1" Partida : Primera del Tablero
    Ficha "0..1" -- "0..1" Partida : Ultima del Tablero
    Movimiento "0..1" -- "0..1" Partida : Es Realizado
    Movimiento "0..1" -- "0..1" Partida : Tiempo Apuesta
    Movimiento "0..1" -- "0..1" Partida : Tiempo Turno
    Movimiento "0..1" -- "0..1" Partida : Libres
    Movimiento "0..1" -- "0..1" Partida : Tablero
    Movimiento "0..1" -- "0..1" Partida : Primera del Tablero
    Movimiento "0..1" -- "0..1" Partida : Ultima del Tablero
    Runnable "1" -- "0..1" Partida : Es Realizado
    Runnable "1" -- "0..1" Partida : Tiempo Apuesta
    Runnable "1" -- "0..1" Partida : Tiempo Turno
    Runnable "1" -- "0..1" Partida : Libres
    Runnable "1" -- "0..1" Partida : Tablero
    Runnable "1" -- "0..1" Partida : Primera del Tablero
    Runnable "1" -- "0..1" Partida : Ultima del Tablero
    ProcesoContadorTiempo "1" -- "0..1" Partida : Es Realizado
    ProcesoContadorTiempo "1" -- "0..1" Partida : Tiempo Apuesta
    ProcesoContadorTiempo "1" -- "0..1" Partida : Tiempo Turno
    ProcesoContadorTiempo "1" -- "0..1" Partida : Libres
    ProcesoContadorTiempo "1" -- "0..1" Partida : Tablero
    ProcesoContadorTiempo "1" -- "0..1" Partida : Primera del Tablero
    ProcesoContadorTiempo "1" -- "0..1" Partida : Ultima del Tablero
  
```

Diagrama de Diseño – Interfáz de Usuario



Diccionario de Clases

Clase: Usuario

Atributos

- oid : int
- nombre : String
- password : String
- nombreCompleto : String

Métodos

- + getNombre(): String
- + getPassword(): String
- + getNombreCompleto(): String
- + getOid(): int
- + setOid(oid : int) : void
- + setNombre(nombre : String) : void
- + setPassword(password : String) : void
- + setNombreCompleto(nombreCompleto : String) : void
- + Usuario()
- + Usuario(String n, String p, String nc)

Clase: Administrador

Métodos

- + Administrador(n : String, p : String, np : String)
- + Administrador()
- + actualizarEnBase() : void

Clase: Ficha

Atributos

- valorDerecha : int
- valorIzquierda : int

Métodos

- rotar() : void
- + getValorDerecha() : int
- + getValorIzquierda() : int
- + sePuedeUnir(lado : String, fichaDescartada : Ficha) : void
- + Ficha(valorIzquierda : int, valorDerecha : int)

Clase: Apuesta

Atributos

- valor : double
- jugador : Jugador

Métodos

- + getValor() : double
- + getJugador() : Jugador
- + setValor(valor : double) : void
- + setJugador(jugador : Jugador) : void
- + Apuesta(valor : double)

Clase: Movimiento

Atributos

- ganador : Jugador
- fechaHora : Date
- pozoApuestas : double
- jugador : Jugador
- tablero : ArrayList<Ficha>

Métodos

- + setJugador(jugador : Jugador) : void
- + setFechaChora(fechaHora : Date) : void
- + setPozoApuestas(pozoApuestas : double) : void
- + getGanador() : Jugador
- + getFechaHora() : Date
- + getPozoApuestas() : double
- + getJugador() : Jugador
- + getTablero() : ArrayList<Ficha>
- + setGanador(ganador : Jugador) : void
- + Movimiento(ganador : Jugador, pozoApuestas : double, jugador : Jugador, tablero : ArrayList<Ficha>)

Clase: Jugador

Atributos

- saldo : double
- mano : ArrayList<Ficha>
- partidaJugando : Partida

Métodos

- + setSaldo(saldo : double) : void
- + getSaldo() : double
- + verificarSaldo(monto : double) : void
- + agregarFicha(f : Ficha) : void
- + getFichas() : ArrayList<Ficha>
- + quitarApuesta(valor : double) : void
- + eliminarFicha(ficha : Ficha) : void
- + vaciarMano() : void
- + actualizarEnBase() : void
- + Jugador(n : String, p : String, np : String, s : double)
- + Jugador()

Clase: Partida

Atributos

- oid : int
- partidaActiva : boolean
- pozoApuestas : double
- jugadores : ArrayList<Jugador>
- turno : Jugador
- ganador : Jugador
- movimientos : ArrayList<Movimiento>
- ultimaApuesta : Apuesta
- libres : ArrayList<Ficha>
- tablero : ArrayList<Ficha>
- primera : Ficha
- ultima : Ficha
- id : String
- segTurno : int
- segApuesta : int
- tiempoTurno : ProcesoContadorTiempo
- tiempoApuesta : ProcesoContadorTiempo

Métodos

- + getSegTurno() : int
- + getTiempoTurno() : ProcesoContadorTiempo
- + getTiempoApuesta() : ProcesoContadorTiempo
- + iniciarTiempoTurno() : void
- + iniciarTiempoApuesta() : void
- + pausarTiempoTurno() : void
- + terminarTiempoApuesta() : void
- + continuarTiempoTurno() : void
- + setMovimientos(movimientos : ArrayList<Movimiento>) : void

- + setJugadores(jugadores : ArrayList<Jugador>) : void
- + setPozoApuestas(pozoApuestas : double) : void
- + setGanador(ganador : Jugador) : void
- + setId (id : String) : void
- + getId() : String
- + setOid (oid : int) : void
- + getOid() : int
- + jugadorAbandonando(jugador : Jugador) : void
- + tiempoFinalizado() : void
- + getLibres() : ArrayList<Ficha>
- + getTurno() : Jugador
- + getTablero() : ArrayList<Ficha>
- + getGanador() : Jugador
- + getPozoApuestas() : double
- + getJugadores() : ArrayList<Jugador>
- + getUltimaApuesta() : Apuesta
- + getMovimientos() : ArrayList<Movimiento>
- + agregarJugador(jugador : Jugador) : void
- + mover(jugador : Jugador, fichaTablero : Ficha, fichaDescartada : Ficha) : void
- controlesAntesJugar(jugador : Jugador) : void
- + robar(j : Jugador) : void
- + apostar(apostador : Jugador, monto : double) : void
- + confirmarApuesta(confirmación : boolean) : void
- + primerJugada(fichaDescartada : Ficha) : void
- + segundaJugada(fichaTablero : Ficha, fichaDescartada : Ficha) : void
- unirFicha(fichaTablero : Ficha, fichaDescartada : Ficha) : void
- restarMontoJugadoresSumarApuestaEnPartida() : void
- repartirFichas() : void
- mezclarFichas() : void

- cambiarTurno() : void
- finalizarPartida(ganador : Jugador) : void
- verificarTurno(jugador : Jugador) : void
- verificarSiSeDescartoTodas() : void
- verificarSiTieneMovimientos(j : Jugador) : boolean
- verificarApuesta(monto : double) : void
- verificarUltimoEnApostar(apostador : Jugador) : void
- sePuedeJugar() : void
- agregarMovimiento() : void
- crearFichas() : void
- actualizarSaldoJugadores() : void
- actualizarEnBase() : void
- + Partida()

Clase: ProcesoContadorTiempo

Atributos

- hilo : Thread
- correr : boolean
- nombre : String
- segundosInicial : int
- segundos : int
- partida : Partida

Métodos

- + getNombre() : String
- + getSegundos() : int
- + continuar() : void
- + terminar() : void
- + pausar() : void
- + iniciar() : void
- + run() : void
- + toString() : String
- + ProcesoContadorTiempo(partida : Partida, nombre : String, segundos : int)

Clase: SubSistemaUsuario

Atributos

- administradores : ArrayList<Administrador>
- jugadores : ArrayList<Jugador>
- jugadoresLogueados : ArrayList<Jugador>

Métodos

- + loginJugador(n : String, p : String) : Jugador
- + loginAdministrador(n : String, p : String) : Administrador
- + logoutJugador(jug : Jugador) : void
- boolean estaLogueado(n : String) : boolean
- cargarUsuarios() : void
- + SubSistemaUsuario()

Clase: SubSistemaPartida

Atributos

- partidas : ArrayList<Partida>
- proximaPartida : Partida

Métodos

- + partidaParaJugar() : Partida
- + partidasFinalizadas() : void
- + getPartidas() : ArrayList<Partida>
- + agregarJugador(j : Jugador) : void
- cargarPartidas() : void
- + SubSistemaPartida()

Clase: Sistema

Atributos

- ssu : SubSistemaUsuario
- ssp : SubSistemaPartida
- instancia : Sistema <<static>>

Métodos

- + getInstancia() : Sistema <<static>>>
- + loginJugador(nombre : String, pass : String) : Jugador
- + loginAdministrador(nombre : String, pass : String) : Administrador
- + partidasFinalizadas() : void
- + getPartidas() : ArrayList<Partida>
- + agregarJugador(j : Jugador) : void
- + getPartidaParaJugar() : Partida
- + logoutJugador(jug : Jugador) : void
- Sistema()

Clase: MapeadorPartida

Atributos

- partida : Partida

Métodos

+ setPartida(partida : Partida) : void

+ setOid(oid : int) : void

+ getOid() : int

+ getSqlInsert() : String[]

- generarMovimientos(sqls : String[], desde : int) : void

- generarTableros(sqls : String[], desde : int) : void

+ getSqlUpdate() : String[]

+ getSqlDelete() : String[]

+ getSqlRestaurar() : String[]

+ leer(rs: ResultSet) : void

- cargarMovimientosFichas(rs : ResultSet) : void

- cargarPartida(rs : ResultSet) : void

+ getSqlSelect() : String

+ crearNuevo() : void

+ getObjeto() : Object

+ MapeadorPartida()

+ MapeadorPartida(partida : Partida)

Clase: MapeadorAdministrador

Atributos

- a : Administrador

Métodos

+ setAdministrador(a : Administrador) : void

+ setOid(oid : int) : void

+ getOid() : int

+ getSqlInsert() : String[]

+ getSqlUpdate() : String[]

+ getSqlDelete() : String[]

+ getSqlRestaurar() : String[]

+ leer(rs: ResultSet) : void

+ getSqlSelect() : String

+ crearNuevo() : void

+ getObjeto() : Object

Clase: MapeadorJugador

Atributos

- j : Jugador

Métodos

+ setJugador(j : Jugador) : void

+ setOid(oid : int) : void

+ getOid() : int

+ getSqlInsert() : String[]

+ getSqlUpdate() : String[]

+ getSqlDelete() : String[]

+ getSqlRestaurar() : String[]

+ leer(rs: ResultSet) : void

+ getSqlSelect() : String

+ crearNuevo() : void

+ getObjeto() : Object

Clase: BaseDatos

Atributos

- instancia : BaseDatos <<static>>
- conexión : Connection
- sentencia : Statement

Métodos

- + getInstancia() : BaseDatos <<static>>
- BaseDatos()
- conectar(url : String, usuario : String, pass : String) : void
- + desconectar() : void
- + consultar(sql : String) : ResultSet
- + modificar(sql : String) : int
- + transaccion(sqls : String[]) : boolean

Clase: Persistencia

Atributos

- bd : BaseDatos

Métodos

- + proximoOid : int
- + guardar(p : Mapeador) : void
- insertar(p : Mapeador) : void
- actualizar(p : Mapeador) : void
- + borrar(p : Mapeador) : void
- + restaurar(p : Mapeador) : void
- + obtenerTodos(p : Mapeador) : ArrayList
- + buscar(p : Mapeador, where : String) : ArrayList