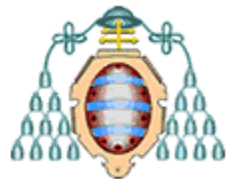


Ticket Run - Diseño

Ingeniería del Software

José Ramón de Diego



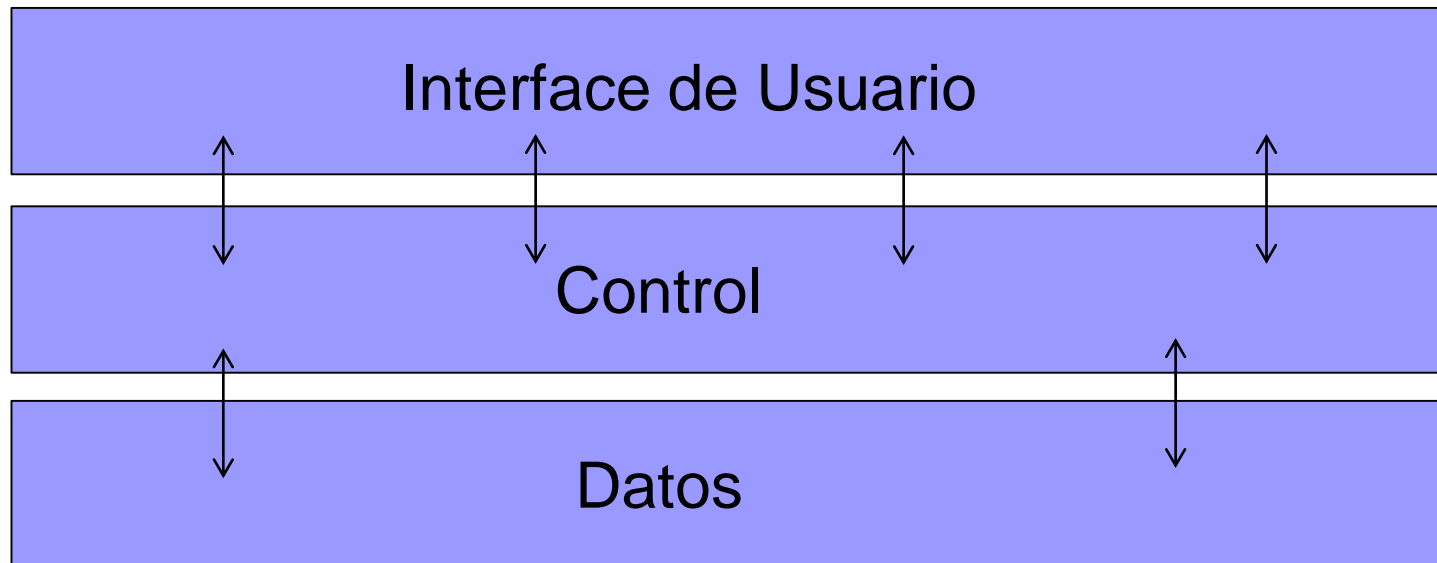


Contenido

- Diagramas de Componentes
- Diagramas de Despliegue
- Diagramas de interacción
- Transformación a tablas

Sistema Ticket Run.

■ Capas del sistema



Sistema envío de paquetes.

■ Capas del sistema

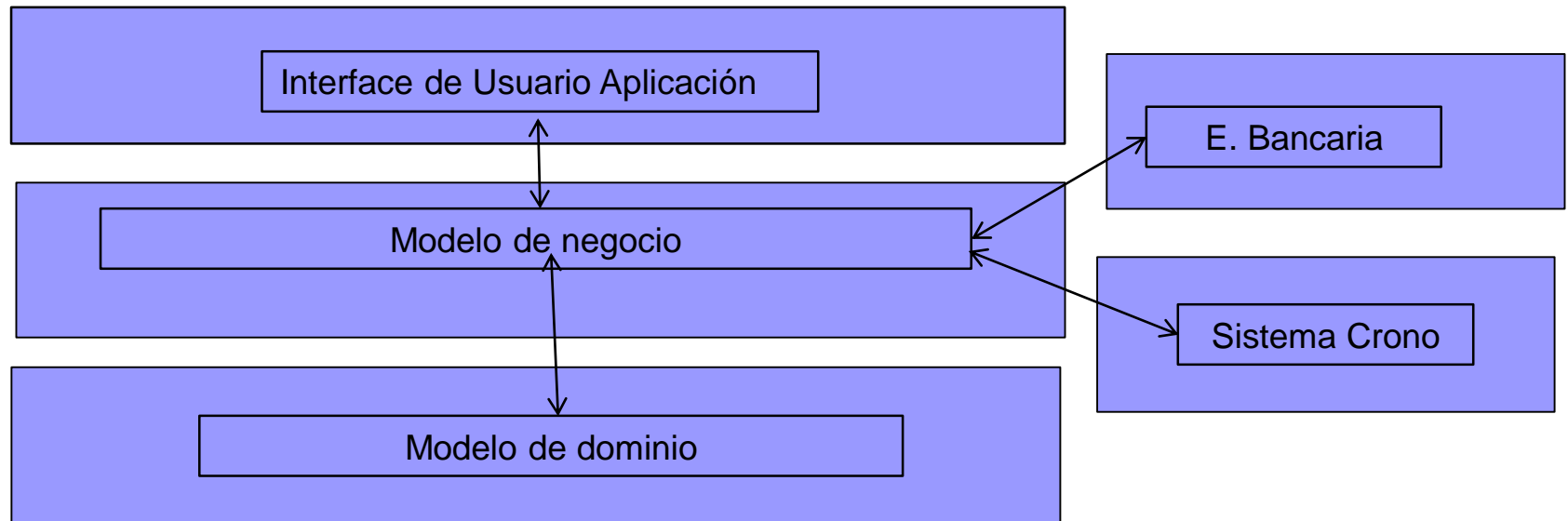
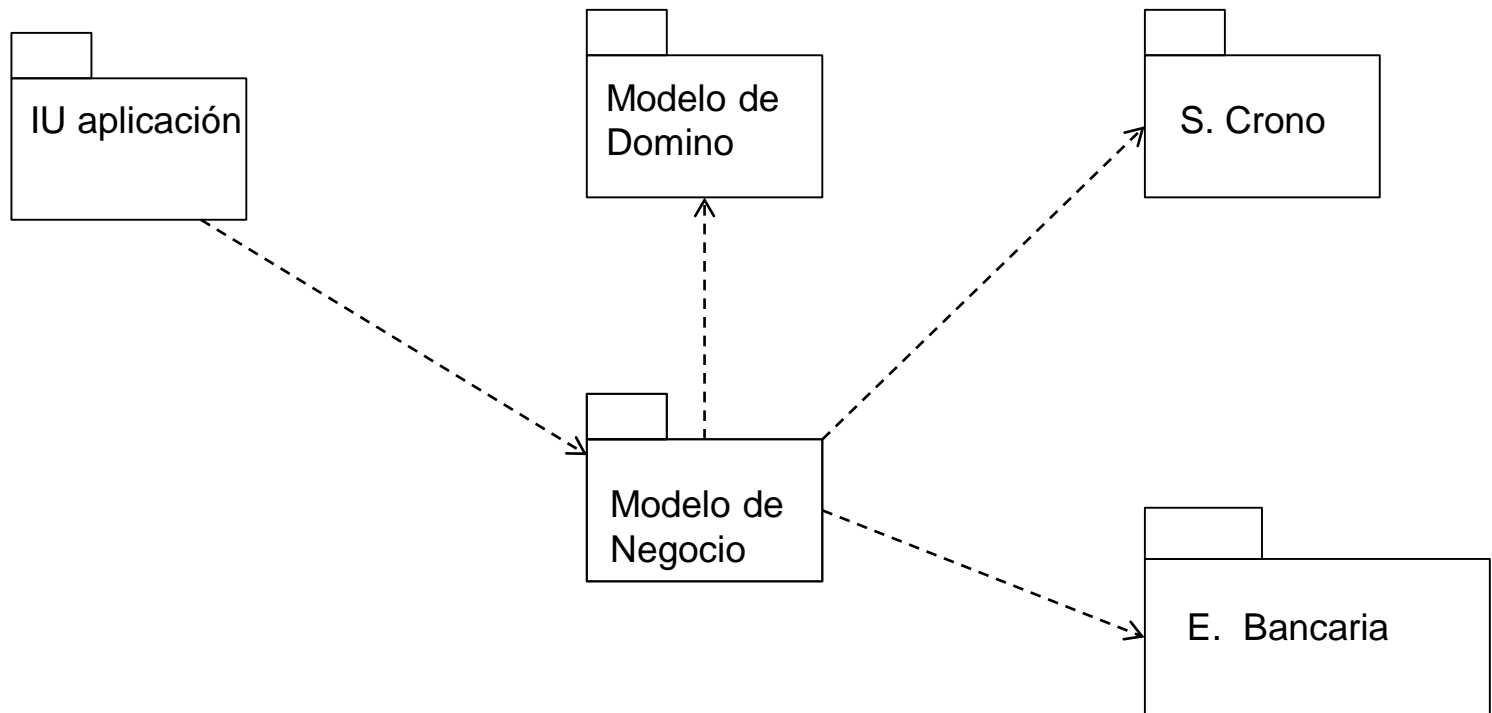


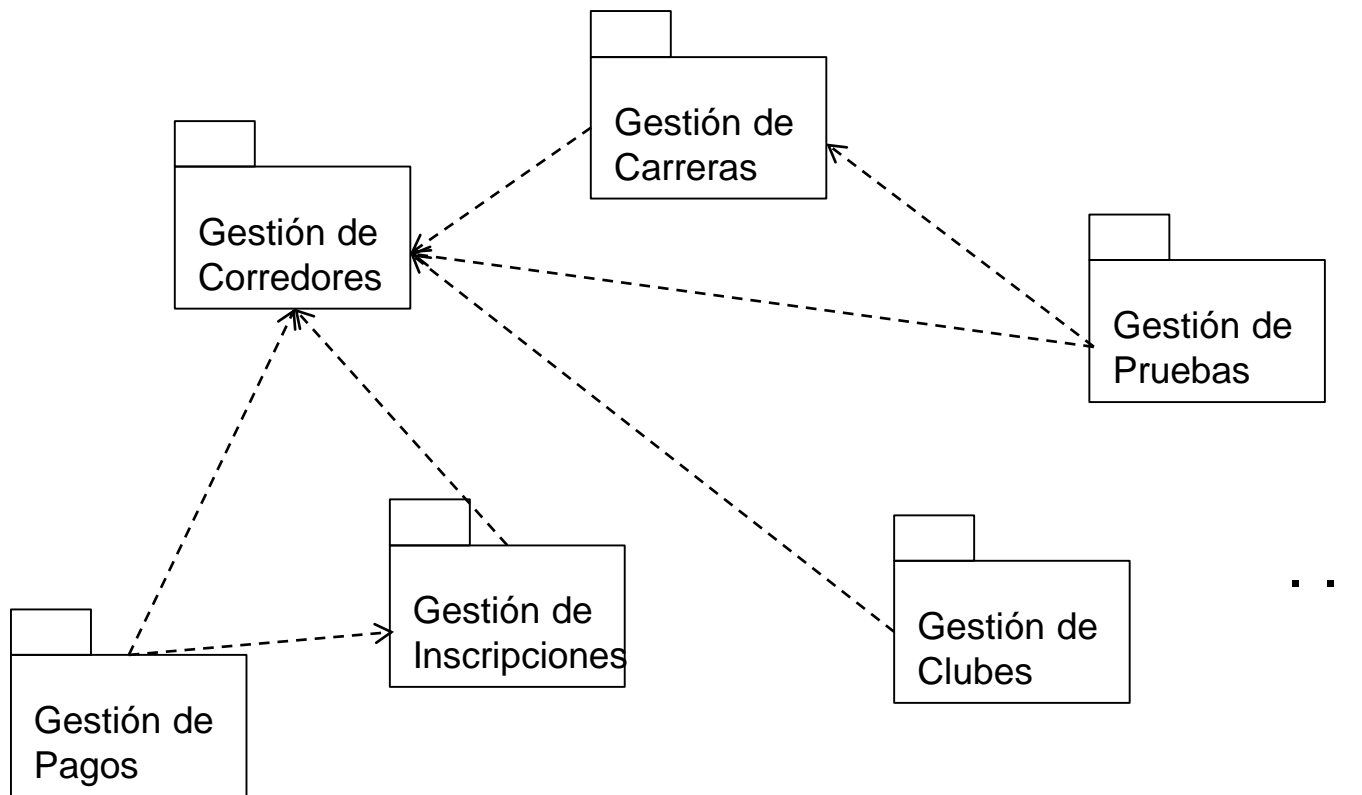
Diagrama de paquetes 1.





Nodos

Subsistemas (D. de paquetes)





Componentes Inscripción

TicketRUN

Seleccionar una opción:

Competiciones (Organiz

Inscripciones (Atleta/Clu

Recepcion de Pagos (Or

Registro de Tiempos (Or

Clasificaciones

Errores:

TicketRUN - Inscripción en Pruebas

Lista de pruebas populares

idCompeticion	descripcion	fechaCelebracion	fechalnicio1	fechalnicio2	fechaC
1	Maraton de Madrid	27-abr-2014	01-sep-2013	31-ene-2014	27-mar-

Fecha Inscripción (provisión para pruebas)

2014-02-04

Modificar Fecha Inscripcion

☐ Pruebas Abiertas
☐ Pruebas Cerradas
☒ Todos las Pruebas

Seleccionar Atleta

NIF

Buscar Atleta

Modificar Atleta

Nuevo Atleta

Inscripción Individual

Inscripción Club

Datos del Atleta

IDAtleta

NIF

Nombre

Apellido

Fecha Nacimiento

Club

País

Sexo

Nuevo Atleta

IDAtleta2

NIF

Nombre

Apellido


Fecha Nacimiento

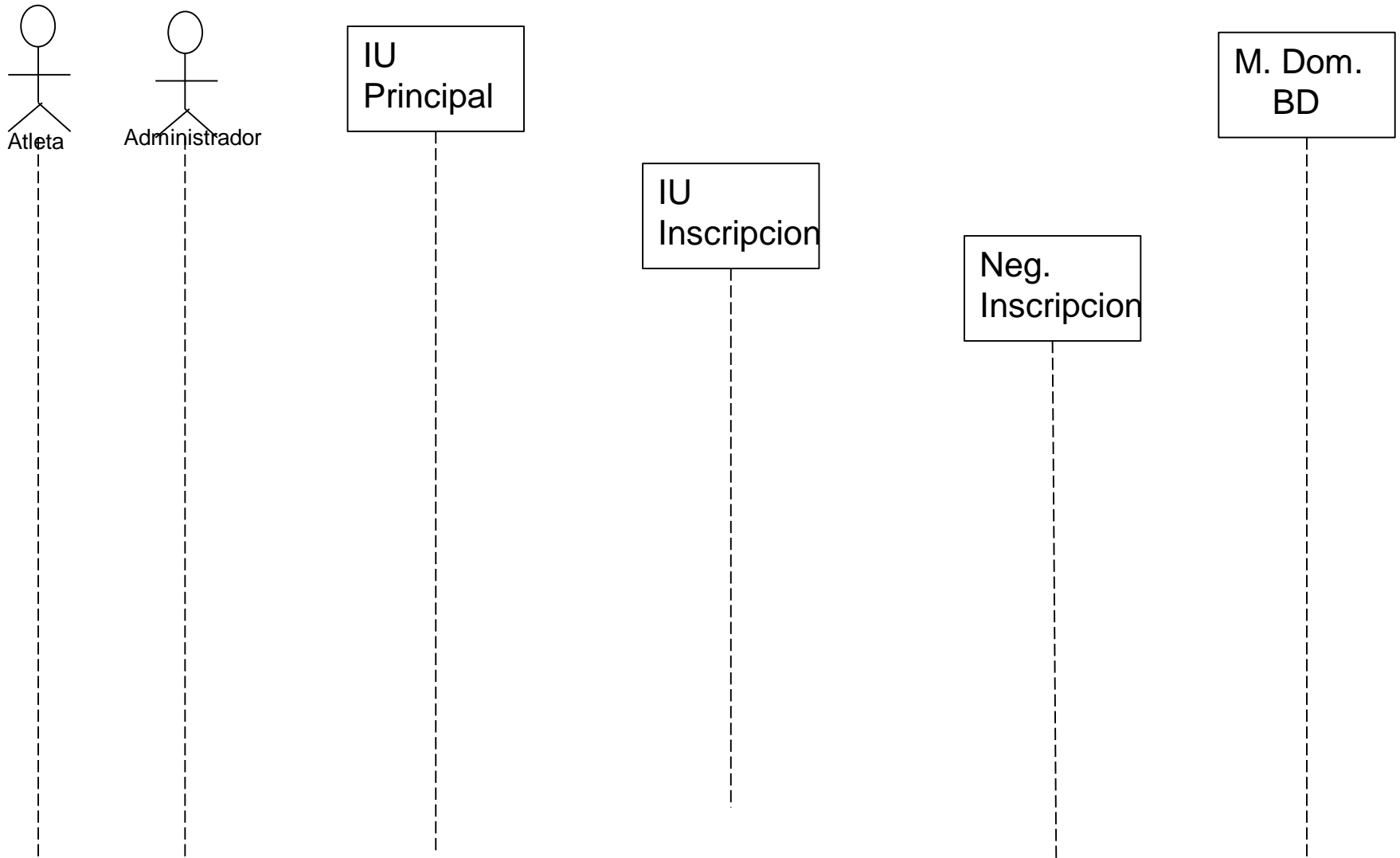
Club

País

Sexo

Errores:

- 
- 1.El atleta le indica al administrador que quiere inscribirse
 - 2.El administrador selecciona la opción de inscripción en prueba.
 - 3.El sistema muestra la pantalla de inscripción
 - 4.El sistema obtiene la lista de pruebas y se la muestra al administrador.
 - 5.El administrador introduce el NIF del atleta y le da a la opción de buscar atleta.
 - 6.El sistema le muestra al administrador los datos del atleta
 - 7.El administrador selecciona la prueba a la que se quiere apuntar el atleta
 - 8.El sistema comprueba que hay plazas disponibles
 - 9.El sistema realiza la inscripción del atleta en la prueba indicada.



IU - Inscripción Pruebas

```
package interfaz;
```

```
import java.awt.BorderLayout;
```

```
...
```

```
import negocio.Inscripcion;
```

```
import fact.util.DJPanel;
```

```
import fact.util.DJTable;
```

```
...
```

```
public class InscripcionPruebas extends DJPanel {
```

```
    static final long serialVersionUID=0;
```

```
    ...
```

[...Atributos, descripción de los elementos del interfaz]

```
    private Inscripcion inscripcion;           // objeto de negocio utilizado para la capa de control
```

```
    public InscripcionPruebas(String v) {super(v);}
}
```

Resto de los componentes que necesita
Componentes de soporte y librerías

...

```
/** * Configuración de la ventana de diálogo */
```

```
public void inicializaVentana() {
```

```
    ...
```

```
    //formulario para el manejo campo de fecha de inscripción
```

```
    ...
```

```
    //formulario para el manejo de la búsqueda del atleta por nif
```

```
    ...
```

```
    //tabla para visualizar la lista de inscritos de una prueba seleccionada
```

```
    ...
```

```
    //formulario para visualizar/modificar los datos de un atleta
```

```
    ...
```

```
    //formulario para insertar los datos de un atleta
```

```
    ...
```

```
    //Botones de modificar la fecha de inscripción
```

```
    // Botones de opciones y acciones del usuario encerrados en una Box
```

```
    //botones para seleccionar los pruebas a mostrar en el panel de pruebas
```

```
    //botón para buscar el atleta
```

```
    //botón para modificar los datos del atleta
```

```
    //boton para insertar un nuevo atleta
```

```
    // Encierra ambos Box en otra zona horizontal añadiendo el conjunto a la ventana
```

```
}
```

...

```
/**
 * Muestra las inscripciones (correctas o incorrectas) de una inscripción por club en una ventana
 (JFrame) de texto (JTextArea)
 * Las inscripciones a visualizar van en el ArrayList (primer parámetro). El segundo parámetro
 es el título que se verá en la ventana
 */

public void muestraInscripcionesClub (){
    ...
    //Configura la ventana (JFrame) donde estará contenido el área de texto
    ...
    //Configura el área de texto donde se mostrarán las inscripciones
    ...
    //Coloca el area de texto dentro de la ventana principal
    ...
    //Añade los textos que se mostrarán

    //Añade los datos de cada inscripcion correcta
    ...
    //Lo visualiza
}
```

IU - Inscripción Pruebas

...

```
/**
 * Inicializa los datos a mostrar: Visualiza los pruebas populares existentes
 */
public void inicializaDatos() throws DataException, java.sql.SQLException {
    inscripcion=new Inscripcion();

    //se muestran todos los socios al entrar
    this.listaPruebas.cargaResultSet(inscripcion.leeCompeticionesTodas());

    //Mostrar la fecha de inscripción que existe por defecto
    inFechaInscripcion.setText(inscripcion.getFechaInscripcion());

    //se busca un nuevo id de atleta para una posible insercion (nuevo atleta)
    inIdAtleta.setText(Integer.toString(inscripcion.nuevaClave("Atleta","idAtleta")));
}
```

IU - Inscripción Pruebas

...

```
/**
```

```
 * Respuesta a los eventos de botones y seleccion de tipo de operacion
```

```
 */
```

```
protected void procesaEvento(ActionEvent e) throws DataException,SQLException,Exception {
```

```
    if (eventoActivado(e,btnPruebasAbiertas)) {
```

```
        listaPruebas.cargaResultSet(inscripcion.leeCompeticionesAbiertas());
```

```
    } else if (eventoActivado(e,btnPruebasTodas)) {
```

```
        listaPruebas.cargaResultSet(inscripcion.leeCompeticionesTodas());
```

```
    }
```

```
    ...
```

```
    else if (eventoActivado(e,btnNuevaInscripcion)) {
```

```
        //Atleta a inscribir: el que está en el formulario de visualización
```

```
        String nifAtleta = outNIF.getText();
```

```
        //Prueba: la marcada en la lista de pruebas
```

```
        int i = this.listaPruebas.getSelectedRow();
```

```
        if (i!=-1 && nifAtleta.length() >0 ) { //Se ha seleccionado alguna fila y  
                                                se ha indicado el atleta
```

```
        int idCompeticion = listaPruebas.getIntAt(i, 0);
```

```
        String msgInscripcion =
```

```
        inscripcion.nuevaInscripcion(nifAtleta, idCompeticion, inscripcion.getFechaInscripcion(), "I");
```

```
        if (msgInscripcion == ""){ //Inscripcion OK, mostrar datos
```

```
            ResultSet rs = inscripcion.leeInscripcion(nifAtleta, idCompeticion);
```

```
    ... }
```


Modelo Negocio Inscripción

```
package negocio;

import java.io.IOException;

...
import fact.util.CsvReader;
import in2test.util.DateTime;

...
public class Inscripcion {
    private Data db;
    public String fechaInscripcion;
    public String msgCabecera;
    public ArrayList<String> inscripcionesOK;
    public ArrayList<String> inscripcionesKO;

    public static final String PROVISIONAL = "Provisional";
    public static final String OK_PLAZO1 = "1";

    ...
    public static final double DESCUENTO_CLUB = 20.0;

    ...
}
```

//subsistemas de soporte

// Datos propios por parte de la clase

Modelo Negocio Inscripción

...

```
protected Data getData () throws DataException {
    DataConnection dc= new DataConnection();           //conexión a la base de datos
    try {
        dc.conn = DriverManager.getConnection("jdbc:ucanaccess://TICKETRUN.accdb");
    } catch (SQLException e) {
        e.printStackTrace();
    }
    db=new Data("TICKETRUN",dc);
    return db;
}
```

```
public Inscripcion () throws DataException {
    //Pone fecha por defecto de la inscripcion la fecha actual
    DateTime hoy = new DateTime();
    fechaInscripcion = hoy.toSQLString();
    //Establece la conexión a la BD
    db = getData();
}
```

...

Modelo Negocio Inscripción

```
...  
public ResultSet leeCompeticionesTodas() throws DataException {  
    String query="SELECT  
        idCompeticion,descripcion,fechaCelebracion,fechaInicio1,fechaInicio2,fechaCierre,  
        numMaxInscripciones,precioReducido1,precioReducido2,precioInscripcion";  
    query=query + " FROM Competicion";  
  
    return (db.query(query));  
}  
  
public ResultSet leeCompeticionesAbiertas () throws DataException {  
    String query="SELECT  
        idCompeticion,descripcion,fechaCelebracion,fechaInicio1,fechaInicio2,fechaCierre,  
        numMaxInscripciones,precioReducido1,precioReducido2,precioInscripcion ";  
    query=query + "FROM Competicion WHERE ";  
    query=query + "fechaCierre >= " + "#" + fechaInscripcion + "#" + " AND " + "fechaInicio1 <= "  
        + "#" + fechaInscripcion + "#";  
  
    return (db.query(query));  
}  
...
```

Modelo Negocio Inscripción

```
...

public ResultSet leeCompeticionesCerradas () throws DataException {
    ...
}

public ResultSet leeInscritos (int id) throws DataException {
    ...
}

public ResultSet leeInscripcion (String nif, int idComp) throws DataException {
    ...
}

public ResultSet leeSoloInscripcion (String nif, int idComp) throws DataException {
    ...
}

public void insertarCompeticion(String query) throws DataException{
    db.execute (query);
}
```

Modelo Negocio Inscripción

```
...
public String nuevaInscripcion (String nifAtleta, int idComp, String fechaInscripcion,
                                String tipo) throws Exception {

    ResultSet rsAtleta = leeAtleta (nifAtleta);
    ResultSet rsComp = leeCompeticion (idComp);
    String msg=""; //Para devolver el mensaje

    if (rsAtleta.next()){
        if (rsComp.next()){
            //Crear la inscripcion en la competición para el atleta
            DateTime fecha = new DateTime (fechaInscripcion);
            String codValidacion = validarInscripcion (rsAtleta,rsComp,fecha);
            System.out.println("Codigo validación:" + codValidacion);
            if (codValidacion == "1" || codValidacion == "2" || codValidacion == "3"){
                int idInscripcion = nuevaClave ("Inscripcion","idInscripcion");
                int idAtleta = rsAtleta.getInt("idAtleta");
                //Computa la cantidad a pagar en función del plazo de inscripción
                double cantidadPlazo1 = rsComp.getDouble ("precioReducido1");

                ...
                //Se obtiene la categoría de la inscripción del atleta en función de la edad cuando sea
                la fecha de la competición

                ...

            } else msg=KO_COMPETICION;
        } else msg=KO_ATLETA;
    }
    return msg;
}
```

Modelo Negocio Inscripción

```
...
protected String validarInscripcion (ResultSet rsAtleta, ResultSet rsCompeticion, DateTime
                                     fechaInscripcion) throws SQLException{
...

public void inscripcionClub (String fromFile, int idComp) throws Exception{
...

public ResultSet leeAtleta (String nif ) throws DataException {
...

public void mostrarAtleta (ResultSet rs) throws SQLException{
...

public ResultSet leeCompeticion (int idComp) throws DataException {
...

public String calculaCategoria (int edad, String sexo ){
...

```

Atleta		
	Nombre del campo	Tipo de datos
🔑	idAtleta	Número
	NIF	Texto corto
	fechaNacimiento	Fecha/Hora
	Campo1	Texto corto
	club	Texto corto
	nombre	Texto corto
	apellido1	Texto corto
	pais	Texto corto
	sexo	Texto corto

Inscripcion		
	Nombre del campo	Tipo de datos
🔑	IdInscripcion	Número
	IdAtleta	Número
	IdCompeticion	Número
	tipo	Texto corto
	categoria	Texto corto
	fechaInscripcion	Fecha/Hora
	cantidad	Número
	cantidadAbonada	Número
	dorsal	Número
	estado	Texto corto
	incidencias	Texto corto

Competicion		
	Nombre del campo	Tipo de datos
🔑	idCompeticion	Número
	descripcion	Texto corto
	fechaCelebracion	Fecha/Hora
	fechaInicio1	Fecha/Hora
	fechaInicio2	Fecha/Hora
	fechaCierre	Fecha/Hora
	numMaxInscripciones	Número
	precioReducido1	Número
	precioReducido2	Número
	precioInscripcion	Número