

**Proyecto: Sistema Informativo del Servicio Social.**

Documento de Diseño Detallado

11 de Diciembre de 2015

**Índice**

[1. Requerimientos para el desarrollo 3](#_Toc445997670)

[2. Diccionario de Datos 4](#_Toc445997671)

[3. Dependencias 6](#_Toc445997672)

[4. Definición de Procesos 7](#_Toc445997673)

[5. Pantallas 11](#_Toc445997674)

**Control de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha Modificación** | **Autor** | **Ajustes Realizados** |
| 1.0 | 11/Diciembre/2015 | Manuel Calzado | Creación documento |
|  |  |  |  |

# Requerimientos para el desarrollo

**Herramientas para el Desarrollo**

* BD Central: Oracle 10g
* Programación Pantalla: Java, Eclipse

**Plataformas de Desarrollo (Servidores a utilizar)**

* Servidor de Base de Datos Oracle 10g
* Servidor de Aplicaciones WebLogic

**Instancias de Base de Datos**

* Ambiente Desarrollo
  + BD SPE: localhost:1521:XE.SIADM
* Ambiente Producción
  + BD SPE:

**Permisos Requeridos Base de Datos**

|  |  |
| --- | --- |
|  | BD: XE/  Usuario: SIADM |
| Tablas | CREACIÓN, MODIFICACIÓN, BORRADO, SELECCIÓN |
| Vistas | CREACIÓN, MODIFICACIÓN, BORRADO, SELECCIÓN |
| Índices | CREACIÓN, MODIFICACIÓN, BORRADO |
| Funciones/Procedimientos/Packages | CREACIÓN, MODIFICACIÓN, BORRADO, EJECUCIÓN |

# Diccionario de Datos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | USERS | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Catálogo de Usuarios. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_USER | | NUMBER(4) | **PK** | NOT NULL | Identificador del usuario. |
| USERNAME | | VARCHAR2(20 BYTE) |  | NOT NULL | Nombre del usuario. |
| PASSWORD | | VARCHAR2(20 BYTE) |  | NOT NULL | Contraseña del usuario. |
| ID\_PROFILE | | NUMBER(1) |  | NOT NULL | Identificador del Perfil. |
| ACTIVE | | NUMBER(1) |  | NOT NULL | 0 – Inactivo.  1 – Activo. |
| USER\_MODIFY | | VARCHAR2(20 BYTE) |  | NULL | Usuario que modifica. |
| CREATION\_DATE | | DATE |  | NULL | Fecha de creación. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | PROFILES | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Catálogo de Perfiles. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_PROFILE | | NUMBER(4) | **PK** | NOT NULL | Identificador del Perfil. |
| DESC\_PROFILE | | VARCHAR2(30 BYTE) |  | NOT NULL | Descripción del Perfil. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | STUDENTS | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Catálogo de Estudiantes. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_STUDENT | | NUMBER(10) | **PK** | NOT NULL | Identificador del Estudiante. |
| STUDENT\_NUMBER | | VARCHAR2(10 BYTE) |  | NOT NULL | Matricula del Estudiante. |
| FIRST\_NAME | | VARCHAR2(50 BYTE) |  | NOT NULL | Primer nombre del Estudiante. |
| LAST\_NAME | | VARCHAR2(50 BYTE) |  | NOT NULL | Apellido del Estudiante. |
| BACHELOR | | VARCHAR2(50 BYTE) |  | NULL | Carrera del Estudiante. |
| EMAIL | | VARCHAR2(50 BYTE) |  | NULL | Correo electrónico del Estudiante. |
| PHONE\_NUMBER | | NUMBER(20) |  | NULL | Número telefónico del Estudiante. |
| COMMENTS | | VARCHAR2(4000 BYTE) |  | NULL | Comentarios acerca del Estudiante. |
| ACTIVE | | NUMBER(1) |  | NOT NULL | 0 – Inactivo.  1 – Activo. |
| USER\_MODIFY | | VARCHAR2(20 BYTE) |  | NULL | Usuario que modifica. |
| CREATION\_DATE | | DATE |  | NULL | Fecha de creación. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | ASSOCIATIONS | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Catálogo de Asociaciones. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_ASSOCIATION | | NUMBER(10) | **PK** | NOT NULL | Identificador de la Asociación. |
| ASSOCIATION\_NAME | | VARCHAR2(50 BYTE) |  | NOT NULL | Nombre de la Asociación. |
| EMAIL | | VARCHAR2(50 BYTE) |  | NULL | Correo electrónico de la Asociación. |
| PHONE\_NUMBER | | NUMBER(20) |  | NULL | Número telefónico de la Asociación. |
| COMMENTS | | VARCHAR2 (4000 BYTE) |  | NULL | Comentarios acerca de la Asociación. |
| ACTIVE | | NUMBER(1) |  | NOT NULL | 0 – Inactivo.  1 – Activo. |
| USER\_MODIFY | | VARCHAR2(20 BYTE) |  | NULL | Usuario que modifica. |
| CREATION\_DATE | | DATE |  | NULL | Fecha de creación. |

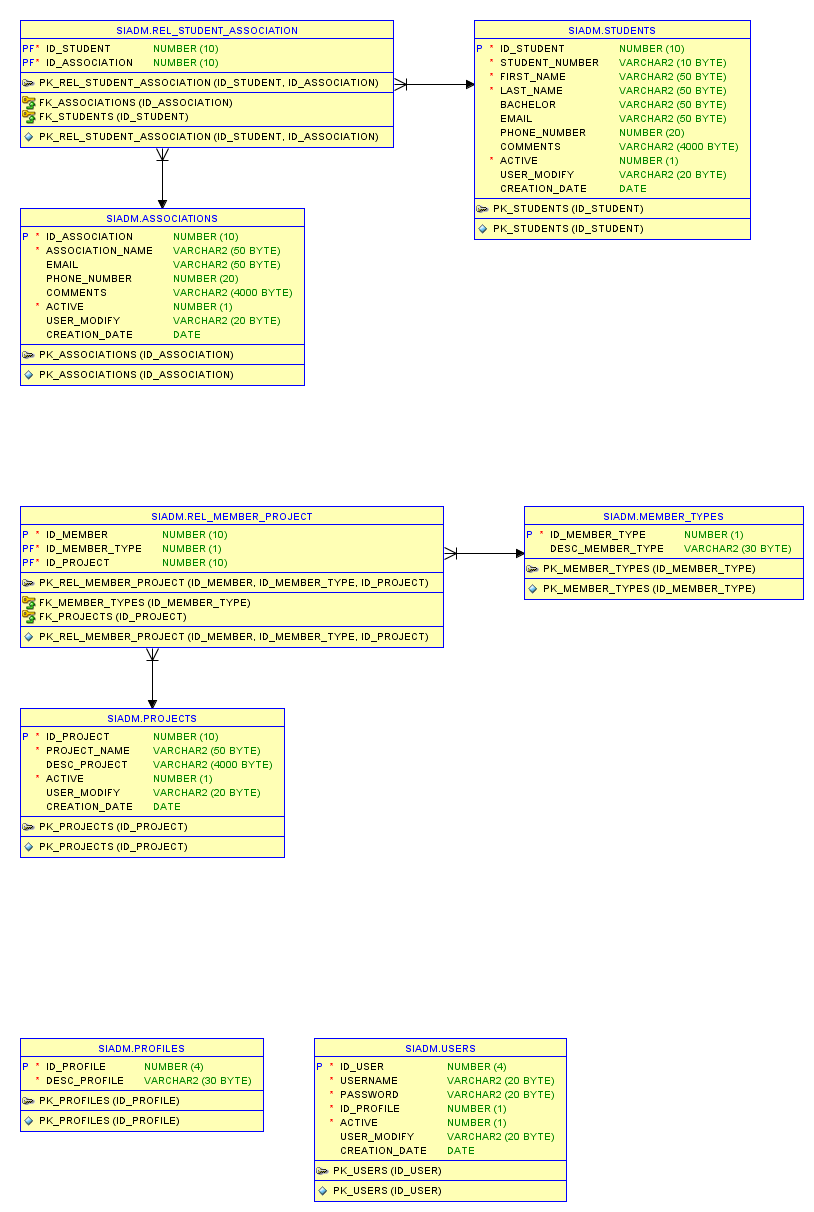
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | REL\_STUDENT\_ASSOCIATION | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Relaciones entre Estudiantes y Asociaciones. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_STUDENT | | NUMBER(10) | **PK,FK** | NOT NULL | Identificador del Estudiante. |
| ID\_ASSOCIATION | | NUMBER(10) | **PK,FK** | NOT NULL | Identificador de la Asociación. |

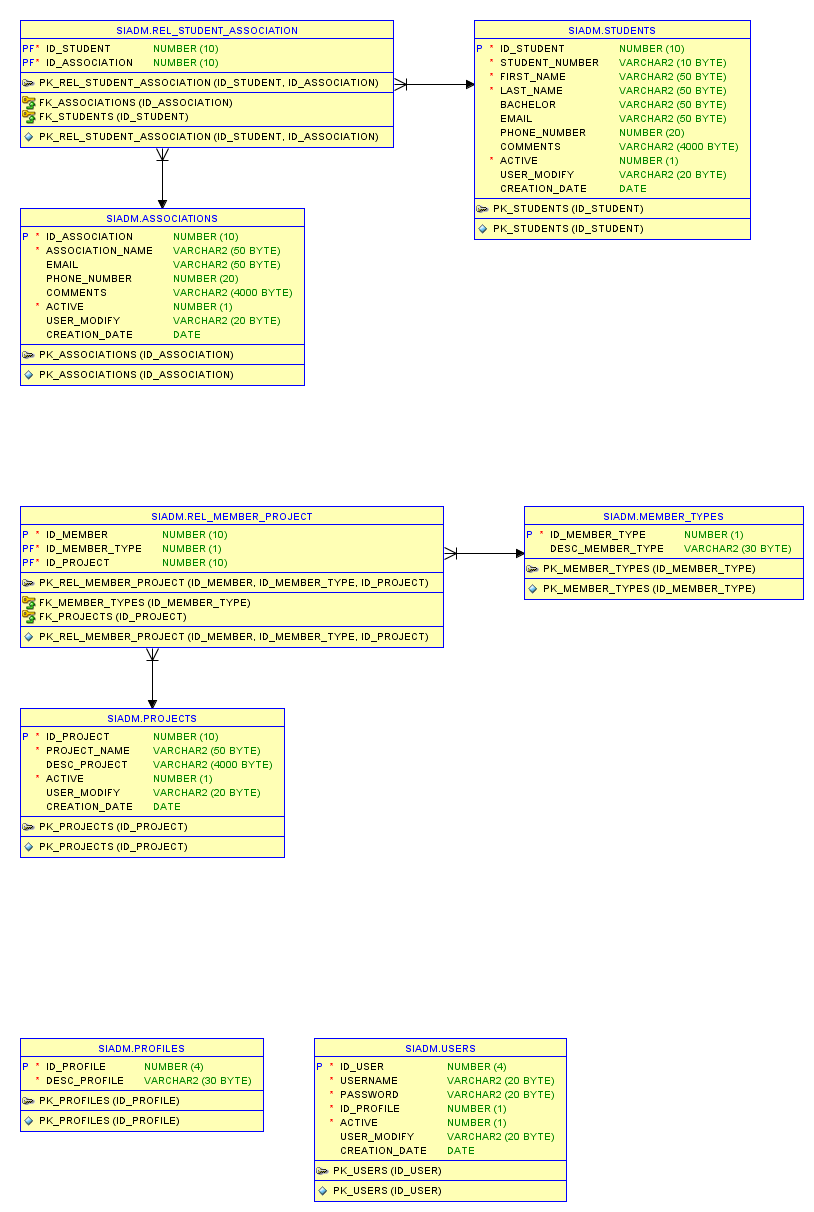
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | PROJECTS | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Catálogo de Proyectos. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_PROJECT | | NUMBER(10) | **PK** | NOT NULL | Identificador del Proyecto. |
| PROJECT\_NAME | | VARCHAR2(50 BYTE |  | NOT NULL | Nombre del Proyecto. |
| DESC\_PROJECT | | VARCHAR2(4000 BYTE |  | NULL | Descripción del Proyecto. |
| ACTIVE | | NUMBER(1) |  | NOT NULL | 0 – Inactivo.  1 – Activo. |
| USER\_MODIFY | | VARCHAR2(20 BYTE |  | NULL | Usuario que modifica. |
| CREATION\_DATE | | DATE |  | NULL | Fecha de creación. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | MEMBER\_TYPES | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Catálogo de Tipo de Miembros. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_MEMBER\_TYPE | | NUMBER(1) | **PK** | NOT NULL | Identificador del Tipo de Miembro. |
| DESC\_MEMBER\_TYPE | | VARCHAR2(30 BYTE) |  | NULL | Descripción del Tipo de Miembro. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre:** | REL\_MEMBER\_PROJECT | | | | |
| **Esquema:** | SIADM | | | | |
| **Objetivo:** | Relaciones entre Miembros y Proyectos. | | | | |
| **Campo** | | **Tipo de Dato** | **PK** | **Nulos** | **Descripción** |
| ID\_MEMBER | | NUMBER(10) | **PK,FK** | NOT NULL | Identificador del Miembro. |
| ID\_MEMBER\_TYPE | | NUMBER(1) | **PK,FK** | NOT NULL | Identificador del Tipo de Miembro. |
| ID\_PROJECT | | NUMBER(10) | **PK,FK** | NOT NULL | Identificador del Proyecto. |

# Dependencias





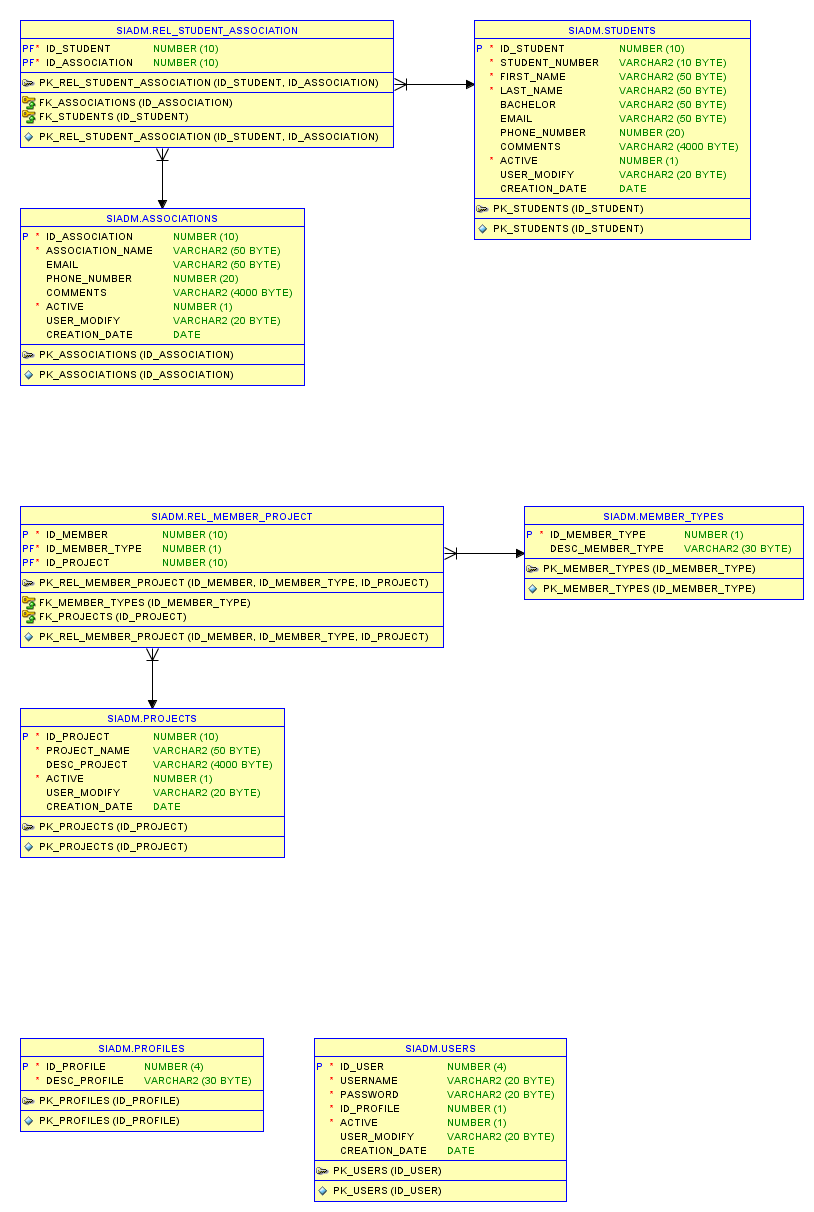


Fig. 1. Dependencias de las tablas del SISS.

# Definición de Procesos

**PROCEDIMIENTOS Y FUNCIONES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Base de datos:** | XE/ | **DBMS** | Oracle DB |
| **Esquema:** | SIADM | **PAQUETE** | PKG\_SISTEMA\_INFORMATIVO\_SS |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | **Tipo** |
| SP\_GET\_LIST\_USERS | Obtiene listado de Usuarios.  Parámetros de entrada:  NA  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_USER\_BY\_ID | Obtiene Usuario en base a su Identificador.  Parámetros de entrada:  p\_id\_user IN NUMBER  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_ADD\_USER | Agrega un nuevo Usuario  Parámetros de entrada:  p\_userId IN NUMBER,  p\_username IN VARCHAR2,  p\_profileId IN NUMBER,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_DEL\_USER | Inactiva un Usuario.  Parámetros de entrada:  p\_userId IN NUMBER,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_GET\_LIST\_STUDENTS | Obtiene listado de Estudiantes.  Parámetros de entrada:  NA  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_STUDENT\_BY\_ID | Obtiene Estudiante en base a su Identificador.  Parámetros de entrada:  p\_id\_student IN NUMBER,  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_ADD\_STUDENT | Agrega un nuevo Estudiante.  Parámetros de entrada:  p\_studentId IN NUMBER,  p\_studentNumber IN VARCHAR2,  p\_firstName IN VARCHAR2,  p\_lastName IN VARCHAR2,  p\_bachelor IN VARCHAR2,  p\_email IN VARCHAR2,  p\_phoneNumber IN NUMBER,  p\_comments IN VARCHAR2,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_DEL\_STUDENT | Inactiva un Estudiante.  Parámetros de entrada:  p\_studentId IN NUMBER,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_GET\_LIST\_ASSOCIATIONS | Obtiene listado de Asociaciones.  Parámetros de entrada:  NA  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_ASSOCIATION\_BY\_ID | Obtiene Asociación en base a su Identificador.  Parámetros de entrada:  NA  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_ADD\_ASSOCIATION | Agrega nueva Asociación.  Parámetros de entrada:  p\_associationId IN NUMBER,  p\_associationName IN VARCHAR2,  p\_email IN VARCHAR2,  p\_phoneNumber IN NUMBER,  p\_comments IN VARCHAR2,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_DEL\_ASSOCIATION | Inactiva nueva Asociación.  Parámetros de entrada:  p\_associationId IN NUMBER,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_GET\_LIST\_PROJECTS | Obtiene listado de Proyectos.  Parámetros de entrada:  NA  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_PROJECT\_BY\_ID | Obtiene Proyecto en base a Identificador.  Parámetros de entrada:  p\_id\_project IN NUMBER,  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_ADD\_PROJECT | Agrega nuevo Proyecto.  Parámetros de entrada:  p\_projectId IN NUMBER,  p\_projectName IN VARCHAR2,  p\_projectDesc IN VARCHAR2,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_DEL\_PROJECT | Inactiva un Proyecto.  Parámetros de entrada:  p\_projectId IN NUMBER,  p\_userModify IN VARCHAR2  Parámetros de salida:  NA | Procedure |
| SP\_VALIDATE\_USER | Valida si las credenciales proporcionadas son válidas.  Parámetros de entrada:  p\_username IN VARCHAR2,  p\_password IN VARCHAR2,  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_UPDATE\_USER\_PASSWORD | Actualiza contraseña del usuario.  Parámetros de entrada:  p\_username IN VARCHAR2,  p\_password IN VARCHAR2  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_ASSOC\_BY\_STU\_ID | Obtiene las Asociaciones de un Estudiante.  Parámetros de entrada:  p\_id\_student IN NUMBER  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_STU\_BY\_ASSOC\_ID | Obtiene los Estudiantes de una Asociación.  Parámetros de entrada:  p\_id\_association IN NUMBER  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_GET\_MEMB\_BY\_PROJ\_ID | Obtiene los Estudiantes y Asociaciones que participan en un Proyecto.  Parámetros de entrada:  p\_id\_project IN NUMBER  Parámetros de salida:  c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_ADD\_REL\_STU\_ASSOC | Agrega nueva relación Estudiante - Asociación.  Parámetros de entrada:  p\_studentId IN NUMBER,  p\_associationId IN NUMBER  Parámetros de salida:  NA | Procedure |
| SP\_DEL\_REL\_STU\_ASSOC | Elimina una relación Estudiante - Asociación.  Parámetros de entrada:  p\_studentId IN NUMBER,  p\_associationId IN NUMBER  Parámetros de salida:  NA | Procedure |
| SP\_GET\_PROY\_BY\_MEMB\_ID | Obtiene los Proyectos en los que participa un Miembro.  Parámetros de entrada:  p\_memberId IN NUMBER,  p\_memberType IN NUMBER,  Parámetros de salida:   * c\_resultados OUT REF\_CURSOR | Procedure |
| SP\_ADD\_REL\_MEMB\_PROJ | Agrega nueva relación Miembro – Proyecto.  Parámetros de entrada:  p\_memberId IN NUMBER,  p\_memberType IN NUMBER,  p\_projectId IN NUMBER  Parámetros de salida:   * NA | Procedure |
| SP\_DEL\_REL\_MEMB\_PROJ | Elimina relación Miembro – Proyecto.  Parámetros de entrada:  p\_memberId IN NUMBER,  p\_memberType IN NUMBER  Parámetros de salida:  NA | Procedure |

# Pantallas

* **Pantalla de Usuarios.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | usuarios.jsp | |
| **Descripción de la clase** | | Su propósito es mostrar el listado de los usuarios registrados en el sistema. Se permiten Altas, Bajas y Cambios. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | usersView.js | |
| **Descripción de la clase** | | Java Script cuyo propósito es llenar la pantalla con los filtros, campos y botones necesarios. Crea las tablas y las llena con los valores obtenidos. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | User.java | |
| **Descripción de la clase** | | La clase User en donde se encuentran los objetos de la pantalla con sus métodos “gets” y “sets” | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | UsersDao.java | |
| **Descripción de la clase** | | Recibe los parámetros y ejecuta el stored procedure del método invocado. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  | No Aplica | | No Aplica |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |
|  | getRow() | | Obtiene los valores de la lista de resultados y los inserta en las variables correspondientes. |
|  | getSPMap() | | Crea la definición del stored procedure, define el nombre del procedure, sus parámetros de entrada y parámetros de salida. |
|  | preProcess() | | Llena los parámetros necesarios para el procedure a ejecutar con los valores correspondientes. |
|  | postProcess() | | Se regresan los resultados. |

* **Pantalla de Alumnos.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | alumnos.jsp | |
| **Descripción de la clase** | | Su propósito es mostrar el listado de los alumnos registrados en el sistema. Se permiten Altas, Bajas y Cambios. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | studentsView.js | |
| **Descripción de la clase** | | Java Script cuyo propósito es llenar la pantalla con los filtros, campos y botones necesarios. Crea las tablas y las llena con los valores obtenidos. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | Student.java | |
| **Descripción de la clase** | | La clase Student en donde se encuentran los objetos de la pantalla con sus métodos “gets” y “sets” | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | StudentsDao.java | |
| **Descripción de la clase** | | Recibe los parámetros y ejecuta el stored procedure del método invocado. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  | No Aplica | | No Aplica |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |
|  | getRow() | | Obtiene los valores de la lista de resultados y los inserta en las variables correspondientes. |
|  | getSPMap() | | Crea la definición del stored procedure, define el nombre del procedure, sus parámetros de entrada y parámetros de salida. |
|  | preProcess() | | Llena los parámetros necesarios para el procedure a ejecutar con los valores correspondientes. |
|  | postProcess() | | Se regresan los resultados. |

* **Pantalla de Grupos Estudiantiles (Asociaciones).**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | grupos.jsp | |
| **Descripción de la clase** | | Su propósito es mostrar el listado de los grupos estudiantiles registrados en el sistema. Se permiten Altas, Bajas y Cambios. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | associationsView.js | |
| **Descripción de la clase** | | Java Script cuyo propósito es llenar la pantalla con los filtros, campos y botones necesarios. Crea las tablas y las llena con los valores obtenidos. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | Association.java | |
| **Descripción de la clase** | | La clase User en donde se encuentran los objetos de la pantalla con sus métodos “gets” y “sets” | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | AssociationsDao.java | |
| **Descripción de la clase** | | Recibe los parámetros y ejecuta el stored procedure del método invocado. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  | No Aplica | | No Aplica |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |
|  | getRow() | | Obtiene los valores de la lista de resultados y los inserta en las variables correspondientes. |
|  | getSPMap() | | Crea la definición del stored procedure, define el nombre del procedure, sus parámetros de entrada y parámetros de salida. |
|  | preProcess() | | Llena los parámetros necesarios para el procedure a ejecutar con los valores correspondientes. |
|  | postProcess() | | Se regresan los resultados. |

* **Pantalla de Proyetos.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | proyectos.jsp | |
| **Descripción de la clase** | | Su propósito es mostrar el listado de los proyectos registrados en el sistema. Se permiten Altas, Bajas y Cambios. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | projectsView.js | |
| **Descripción de la clase** | | Java Script cuyo propósito es llenar la pantalla con los filtros, campos y botones necesarios. Crea las tablas y las llena con los valores obtenidos. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | Project.java | |
| **Descripción de la clase** | | La clase User en donde se encuentran los objetos de la pantalla con sus métodos “gets” y “sets” | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la clase** | | ProjectDao.java | |
| **Descripción de la clase** | | Recibe los parámetros y ejecuta el stored procedure del método invocado. | |
| **Clase Padre** | | NA | |
| **Interfaces** | | NA | |
| **Paquete** | | NA | |
| **Atributos** | | | |
|  | **Declaración** | | **Descripción** |
|  | No Aplica | | No Aplica |
|  |  | |  |
| **Métodos** | | | |
|  | **Declaración** | | **Descripción** |
|  | getRow() | | Obtiene los valores de la lista de resultados y los inserta en las variables correspondientes. |
|  | getSPMap() | | Crea la definición del stored procedure, define el nombre del procedure, sus parámetros de entrada y parámetros de salida. |
|  | preProcess() | | Llena los parámetros necesarios para el procedure a ejecutar con los valores correspondientes. |
|  | postProcess() | | Se regresan los resultados. |

