

Documentación del proyecto.

Club Social y Deportivo

[Introducción.](#)

[Vista General.](#)

[Organización del Proyecto.](#)

[Gestión del proceso.](#)

[Documento de diseño](#)

[Prototipo](#)

Plan de desarrollo del Proyecto.

HISTORIAL DE REVISIONES DEL MISMO

Fecha	Versión	Descripción	Autor
1/3/2013	0.1	Versión Preliminar del documento.	Aarón Soria
6/3/2013	0.8	Modificación de documento adaptándolo según la estructura adecuada.	Aarón Soria
7/3/2013	0.9	Realizando la documentación.	Aarón Soria
12/3/2013	1.0	Presentación de la documentación.	Aarón Soria

1. Introducción.

Este es el documento correspondiente a la planificación y organización del proyecto.

Propósito

El propósito del plan de desarrollo es dar a conocer y controlar el desarrollo del proyecto. Explicar qué se ha hecho, el modo y con qué se ha realizado.

Alcance

El plan de desarrollo quiera explica como se ha realizado el proyecto “Club Social y Deportivo”. El avance del proyecto y el seguimiento en cada una de las iteraciones ocasiona el ajuste de este documento produciendo nuevas versiones actualizadas.

2. Vista General.

Aquí se explicará la idea del proyecto explicada según las exigencias y del enunciado del mismo.

El club social y deportivo pretende tener la gestión digitalizada de:

- Datos de los socios, altas y bajas, bajas temporales.
- Registro de entrada de los socios al club.
- De datos de los personal, altas y bajas.
- De Actividades.
- Alquileres de instalaciones.
- Organización de eventos y otros servicios.
- Facturas y deudas a pagar
- Informes, recibos y facturas

El proyecto consiste en una aplicación para la gestión de una base de datos desde una interfaz gráfica utilizando distintas tecnologías de bases de datos, además de un generador de informes.

3. Organización del Proyecto.

Una vez leídas las especificaciones del proyecto y entendidas y definidas las exigencias del mismo.

Participantes del Proyecto

Joan Sebastian, Aarón Soria y Adrián Tello.

Joan y Aarón se encargan de la parte del esquema entidad-relación y están realizando la parte correspondiente a el sql, así como de la documentación inicial.

Aarón se encarga de realizar la documentación de la planificación.

Adrian Tello realiza el trabajo correspondiente a la estructura básica de la interfaz, realizando un diseño previo mediante bocetos a modo de idea preliminar.

Una vez definidas las tablas podremos realizar la distribución de las tecnologías(XML, Hibernate, Objeto relacional y Base de Datos Relacional) para cada tabla. Y así poder ya decidir qué tabla realizamos cada uno y desarrollar la parte funcional de la interfaz ya una vez preparada una base de datos básica a modo de referencia. Para ver la distribución de tecnologías se ha realizado un documento específico titulado "TABLAS". Cada uno realizaremos con cada tecnología una tabla, salvo la correspondiente al XML que la haremos posiblemente entre todos.

Adrian Tello realizará la base de la interfaz y la parte funcional de su parte para que los demás sigamos el modo de cómo la ha hecho.

Adrian crea un repositorio en github y aloja todo el código. Utilización de Gradle para compilar y pruebas. Se realizan los primeros pull requests.

Después Joan y Adrián desarrollarán en más detalle la parte SQL.

Día	Especificaciones
1	Se inicia el Proyecto se realiza una idea general y se comienza a esquematizar relaciones y ha hacer diagramas. También se hará diseño previo de interfaz. Comenzar documentación
6	Creates hechos. Además la estructura básica está definida. Hay que realizar la parte funcional teniendo en cuenta la parte sql.
7	Se crea un repositorio en github. Y se empieza a utilizar.
11	Primera entrega. Implantar definitivamente sin errores el plugin de eclipse de Gradle en windows.
15	Usar tecnologías de base de datos y empezar a realizar funcionalidad del resto de la base de datos y interfaz.
17	Parte funcional del Gestor finalizada. Retocar Interfaz según el tema elegido. Y corregir errores.
18	Entrega final.

Fases del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto. Será bastante flexible debido a una serie de cuestiones de compatibilización de este proyecto con otros. Pero se tendrán en cuenta las fechas de entrega establecidas
El día 11 hay que tener la planificación, doc. de diseño y prototipo.

4. Gestión del proceso.

Tecnologías utilizadas para el desarrollo

Hemos decidido usar la plataforma de Google Drive para almacenar la documentación, además utilizaremos bitbucket para guardar el código.

Entorno de desarrollo de Eclipse programar.

SQLDeveloper para realizar más cómodamente.

Herramienta Dia, para digitalizar todos los esquemas de la base de datos.

Repositorio en github, plugin de Gradle para compilar y pruebas. Se utiliza el sistema de integración continua Travis CI.

Seguimiento y Control del Proyecto

Gestión de Requisitos

Los requisitos del sistema son especificados en la documentación, tanto el enunciado como los de los profesores mismos.. Los cambios en los requisitos serán gestionados y guardados en la plataforma de Google de eventos.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal entre los compañeros.

Y se realizará eventos con el Google+ con fechas para tener fácil poder comunicar a los compañeros cambios de fecha o de proyecto

Control de Calidad

Cada cierto tiempo los participantes se mostrarán los avances mutuamente y compartirán sus dudas y discutirán soluciones.

5. Referencias

Documento pdf de modelo a seguir "Plan de Desarrollo Software.pdf"

Documento de diseño

Requisitos

Los requisitos son los siguientes:

Acceso a bases de datos relacionales desde java directamente con dos tablas relacionadas por clave primaria/ajena. Debe tener toda la gestión de los datos en modo consola o en modo gráfico, Swing o FX (insertar, modificar, borrar , consultar y buscar)

- Acceso a bases de datos relacionales con Hibernate con dos tablas relacionadas por clave primaria/ajena.

Debe tener toda la gestión de los datos en modo consola o en modo gráfico, Swing o FX (insertar, modificar, borrar , consultar y buscar)

- Bases de datos Objeto relacionales con Oracle, debe tener lo siguiente

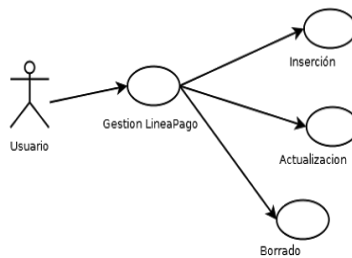
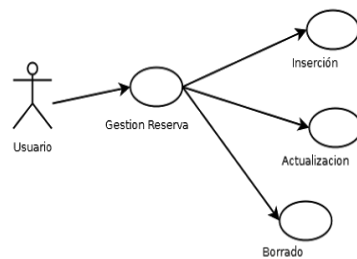
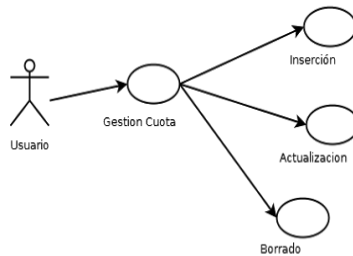
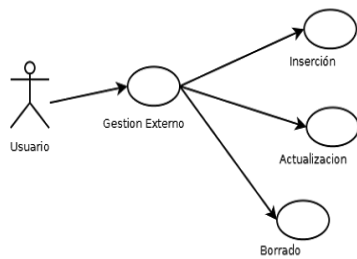
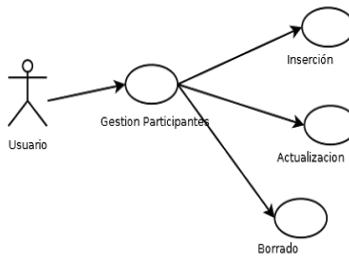
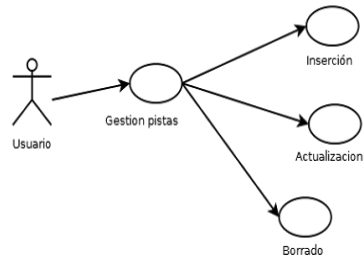
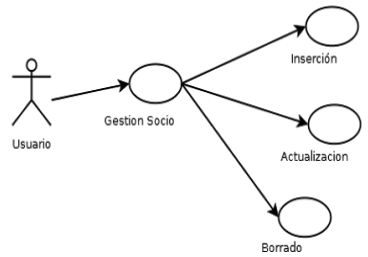
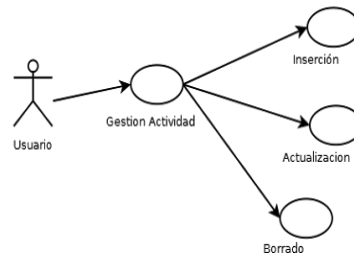
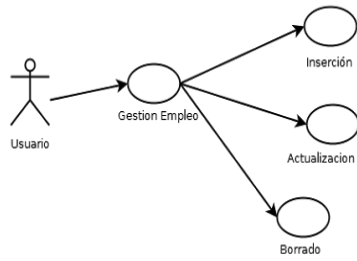
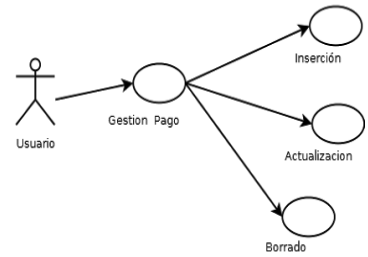
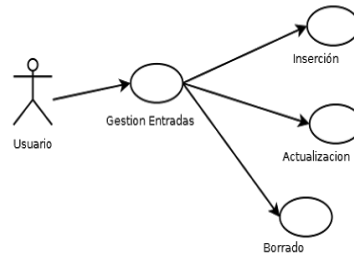
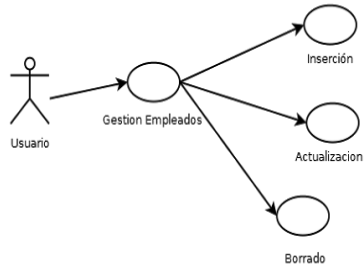
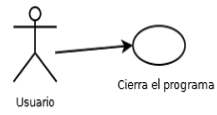
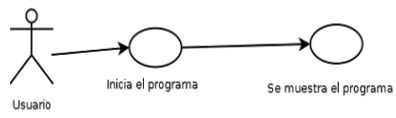
- * Referencias (REF)
- * Clases (type) incluidas en otras clases como atributos
- * Insertar, modificar y borrar de todas las tablas de objetos.
- * Diferentes consultas, pero no es necesario que la parte objeto relacional realice todas las funciones de gestión de los datos afectados
- * Se valorará la utilización de tablas anidadas
- * Se valorará la inclusión en java de esta parte por medio de CallableStatement.

- Bases de datos en XML, deberá tener creacion esquema (con DTD o XSD) y consultas y acceso desde java para cambios. Los objetos XML pueden ser tablas del esquema de bases de datos, o bien datos obtenido del esquema relacional , como por ejemplo preparación para impresión o envía vía correo electrónico o web.

Además, se ha de aportar:

Una planificación, documento de diseño un prototipo, Aplicación, Memoria, Manual de usuario.

Casos de uso



```

    graph LR
      Empleado -- 1 --> Empleo
      Empleo -- 2 --> Empleado
      Empleo -- 1 --> PersonalExterno
      PersonalExterno -- 1 --> Empleo
      PersonalExterno -- 1 --> Reserva
      Reserva -- 2 --> Pistas
      Pistas -- 1 --> Reserva
      Reserva -- 1 --> Entrada
      Entrada -- 2 --> Participantes
      Participantes -- 2 --> Actividad
      Actividad -- 1 --> Participantes
      Cuota -- 1 --> LineaPago
      LineaPago -- 1 --> Cuota
      LineaPago -- 1 --> Pago
      Pago -- 1 --> LineaPago
      Pago -- 1 --> Socio
      Socio -- 1 --> Pago
      Socio -- 1 --> Reserva
      Socio -- 1 --> Entrada
      Socio -- 1 --> Participantes
      Socio -- 1 --> Actividad
  
```

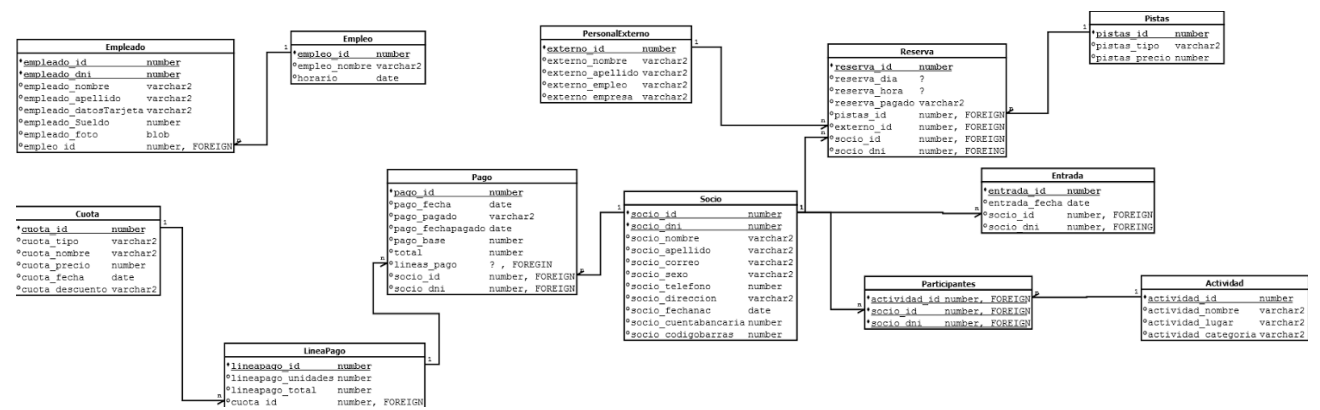


Diagrama de clases

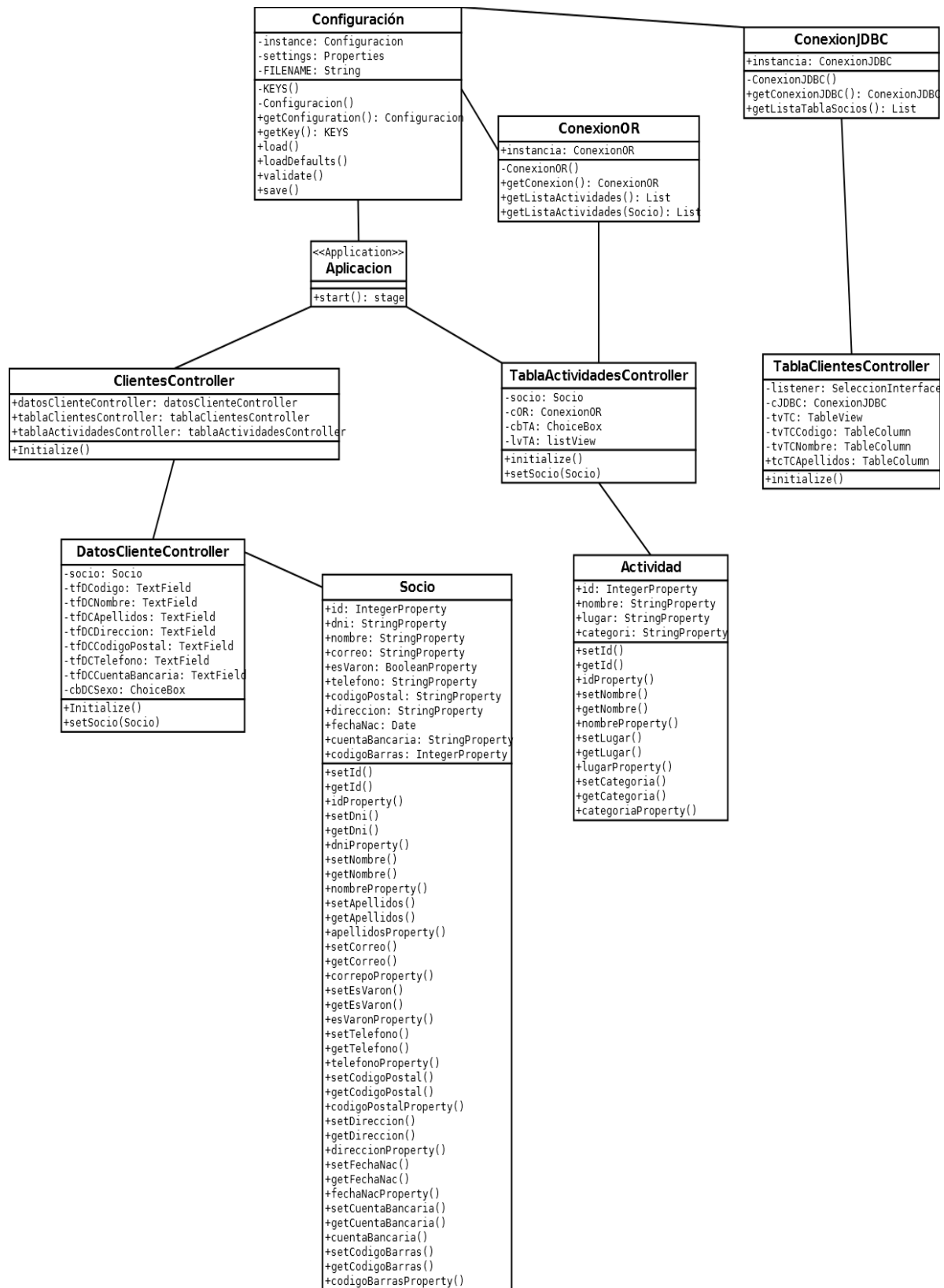
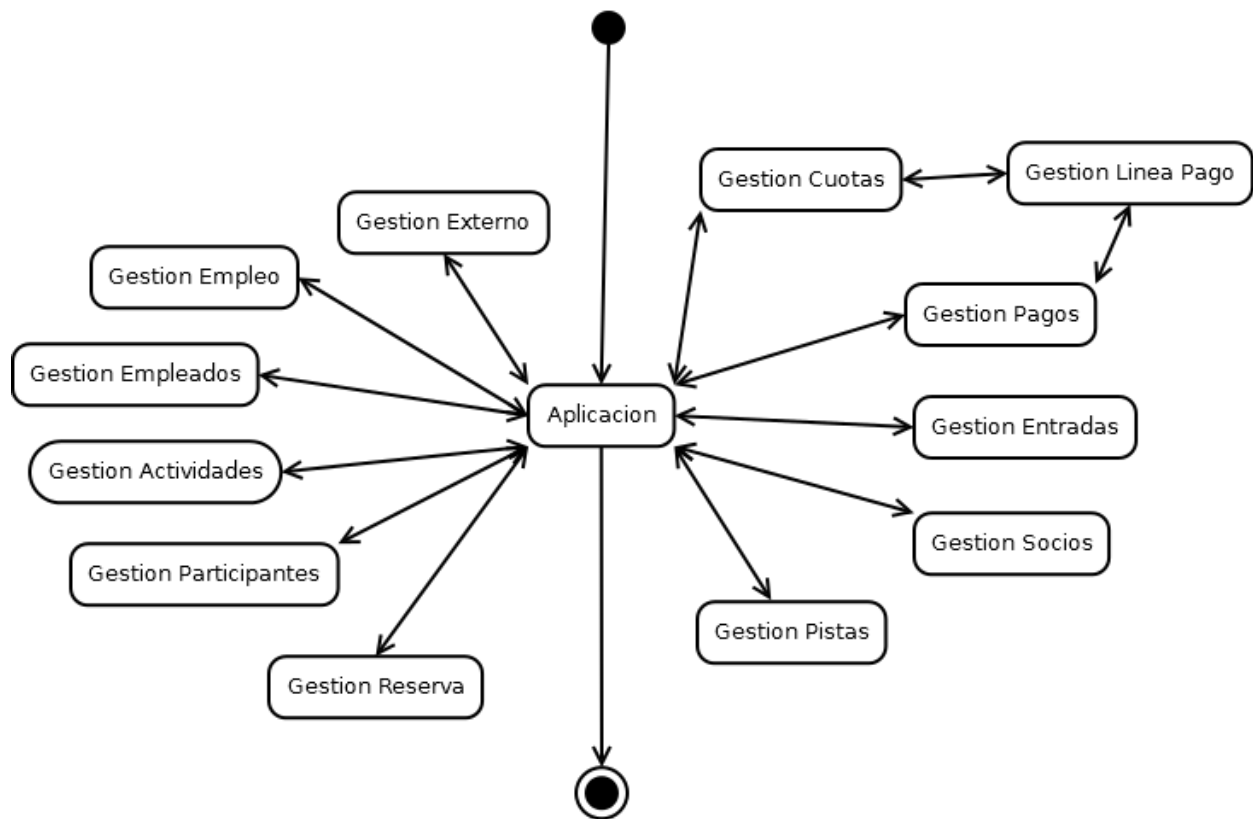
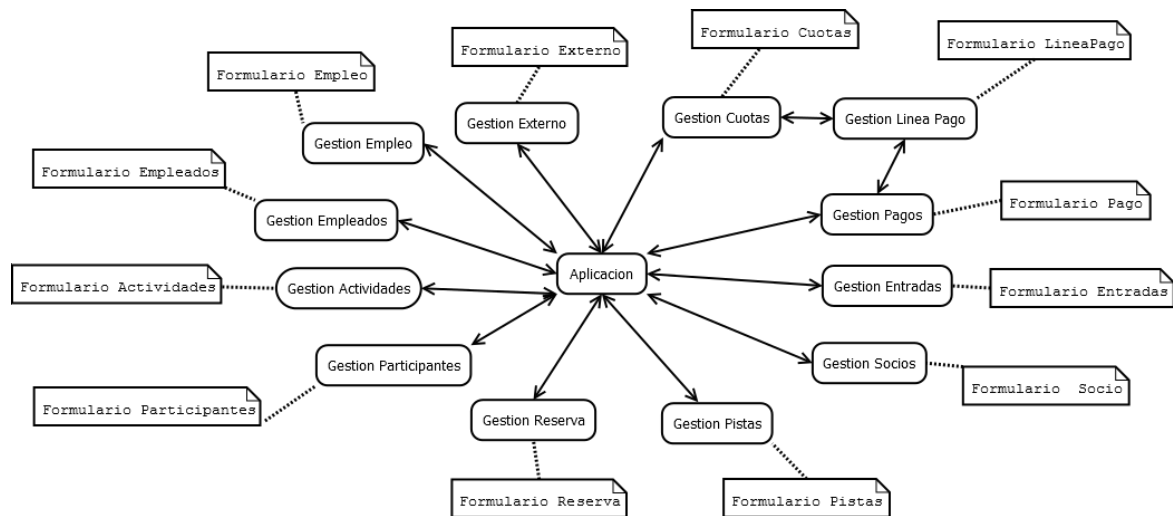


Diagrama de estados



Esquema de navegación



Diseño de las pruebas, unitarias y funcionales

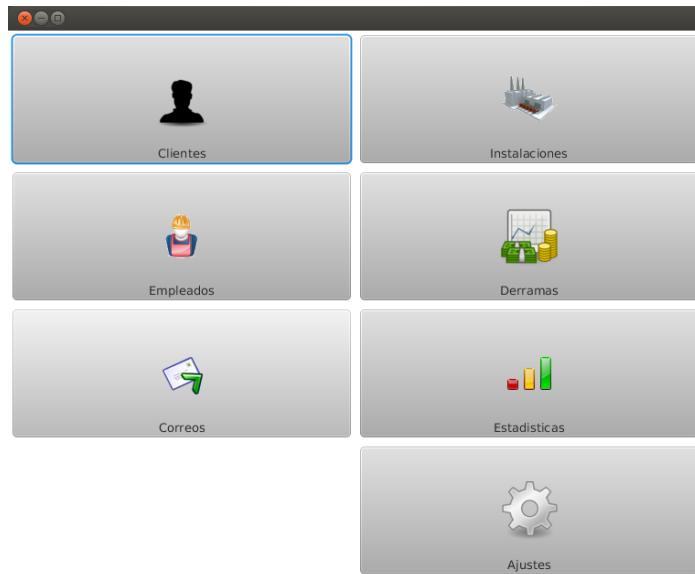
Para la realización de las pruebas unitarias y funcionales, por cada “Gestión”, se realizará un Insert, un Update y un Delete en sus tablas correspondientes con su consecuente SELECT * de la tabla para comprobar que la operación ha sido realizada correctamente, esto a cuanto funcionalidad, de forma unitaria probaremos que al poner valores erróneos en los campos, de error.

Se programará además unas pruebas para todas las clases que se puedan usar sin interfaz. Comprobando su inicialización y sus métodos.

Las pruebas están integradas en nuestro proceso de desarrollo, puesto que antes de fusionar un pull request, este ha de superar las pruebas anteriores y en caso de ser posible, incluir pruebas nuevas.

Prototipo

Diseño de los formularios



Pantalla principal

A main form screen for client management. At the top, there's a header bar with window control icons and four buttons: 'Nuevo', 'Guardar', 'Eliminar', and 'Filtrar'. Below this, there are several input fields: 'Codigo:', 'Nombre:', 'Apellidos:', 'Direccion:', 'C.P.:', 'F. nac:' (with a dropdown), 'Telefono:', 'Cuenta bancaria:', 'Sexo:' (with a dropdown), and a 'Ver' checkbox. There are also 'Asignar Tarjeta' buttons. Below the input fields, there are four tabs: 'Cuotas', 'Actividades', 'Entradas', and 'Derramas'. The 'Cuotas' tab is selected. Below the tabs, there are two tables. The first table has columns 'Fecha', 'Meses', and 'Cantidad', and it contains the text 'Tabla sin contenido'. The second table has columns 'Codigo', 'Nombre', and 'Apellidos', and it also contains the text 'Tabla sin contenido'. At the bottom of the screen, there is a 'Volver' button.

Pantalla clientes

Instalacion Socio Tiempo restante

Tabla sin contenido

Volver

Pantalla instalaciones

Nuevo Guardar Filtar

ID:

Nombre:

Apellidos:

DNI:

Empleo:

Imagen

Nombre	Apellidos
Tabla sin contenido	

Volver

Pantalla empleados

Concepto	Cantidad	% Pagado
Tabla sin contenido		

Crear

Volver

Pantalla derramas

Titulo:

Contenido:

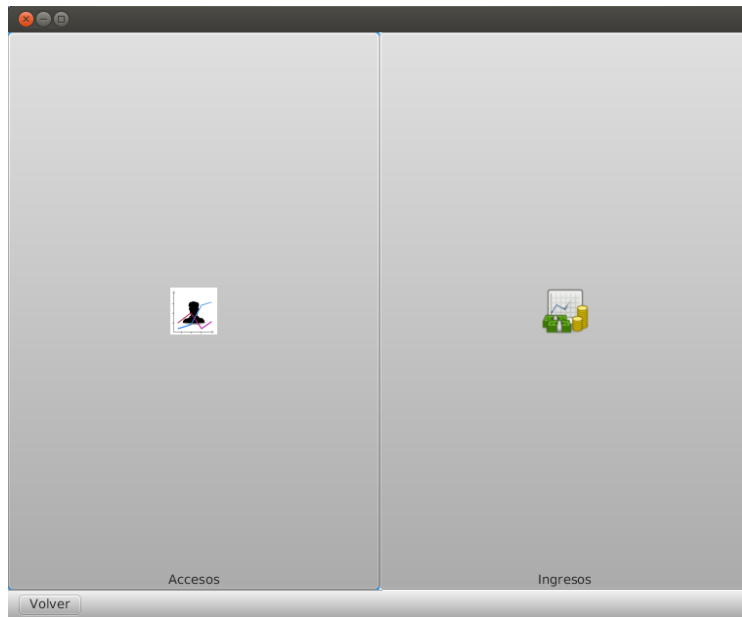
Actividades:

Añadir

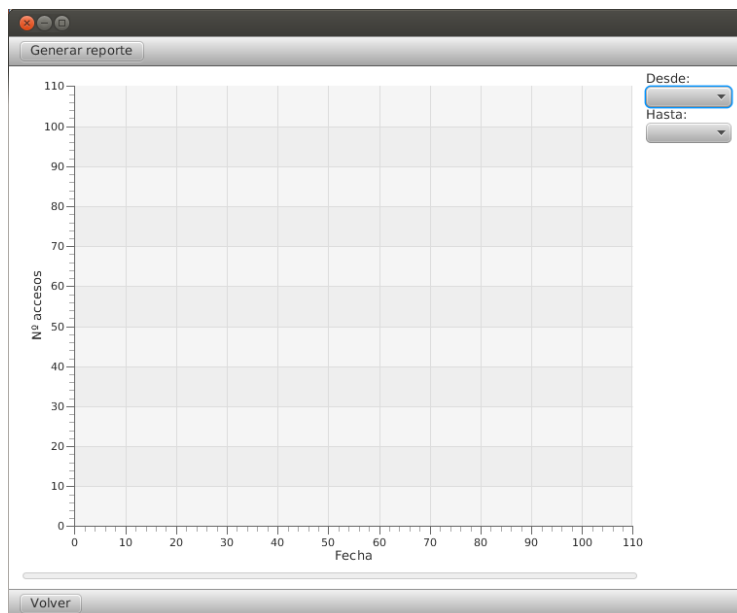
Eliminar

Volver

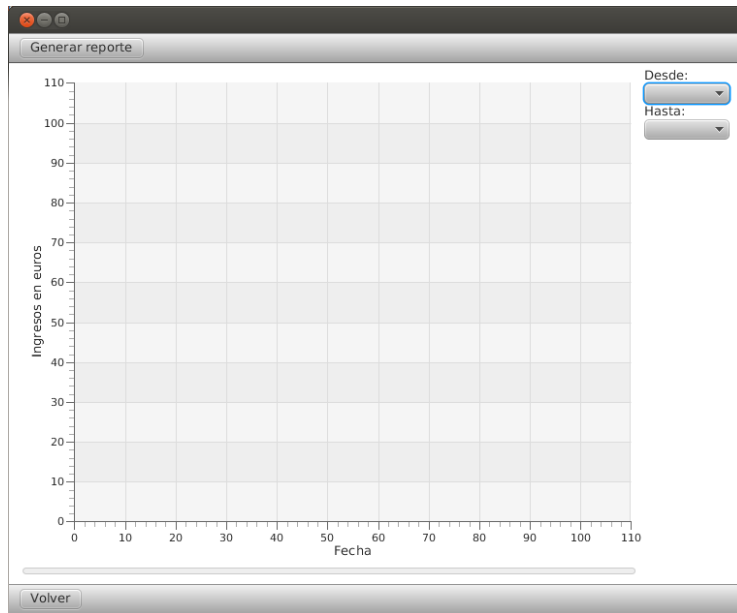
Pantalla correos



Pantalla estadísticas



Pantalla estadísticas accesos



Pantalla estadísticas

Nombre	Horario	Sueldo	Eliminar
Tabla sin contenido			

Volver

Añadir

Pantalla ajustes

Tema visual

Solo contendrá el tema por defecto de JavaFX. Se estiliza por medio de css dentro de la aplicación el la carpeta **main/resources/css/**