SISTEMA DE INFORMACIÓN PARA SEGUIMIENTO DE LOS BENEFICIARIOS DE TALENTO DIGITAL

JUAN EDUARDO SALAZAR ARCILA

KONRAD LORENZ FUNDACIÓN UNIVERSITARIA

FACULTAD DE MATEMÁTICAS E INGENIERÍAS

PROGRAMA DE INGENIERIA DE SISTEMAS

BOGOTÁ

2015

SISTEMA DE INFORMACIÓN PARA SEGUIMIENTO DE LOS BENEFICIARIOS DE TALENTO DIGITAL

JUAN EDUARDO SALAZAR ARCILA  
Código 506131025

Trabajo de grado para optar por el título de Ingeniero de Sistemas

Director:

JORGE ELIECER CAMARGO MENDOZA, PhD

Asesor metodológico

GUSTAVO ANDRÉS CAMPOS AVENDAÑO PhD.

KONRAD LORENZ FUNDACIÓN UNIVERSITARIA

FACULTAD DE MATEMÁTICAS E INGENIERÍAS

PROGRAMA DE INGENIERIA DE SISTEMAS

BOGOTÁ

2015

Nota de aceptación:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del presidente del jurado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del jurado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del jurado

Bogotá D.C, \_\_ de \_\_\_\_\_\_\_\_ de 2015.

**DEDICATORIA**

*“Quiero dedicar este trabajo de grado a la memoria de mi Nona Amalia Castillo y a mi Madre Carmen Omaira Arcila Castillo quienes con cada una de sus enseñanzas y consejos a lo largo de mi vida me han formado como la persona que soy hoy y la que quiero ser en el futuro, seguramente sin todo el apoyo que ellas me han dado no hubiera llega a esta importante etapa de mi vida y a las que están por venir. ”*

AGRADECIMIENTOS

Agradezco de antemano, al pHD Jorge Eliecer Camargo Mendoza, por ser mi tutor guiándome con su conocimiento y experiencia para completar con éxito este trabajo.

Agradezco de antemano, al Ingeniero Nelson Vargas, por ser quien me enseño a programar en el primer semestre de la carrera dándome unas buenas bases para formar mi camino a lo largo de la carrrera.

Agradezco a todos los docentes y compañeros que a lo largo de esta etapa me enseñaron sus conocimientos y a habilidades para afrontar la vida laboral con otro punto de vista como Ingeniero de sistemas.

**INTRODUCCIÓN**

1. **FORMULACIÓN DEL PROBLEMA**
   1. **DESCRIPCIÓN DEL PROBLEMA**

En la actualidad en nuestro país tenemos un déficit de 15.000 ingenieros y trabajadores de las Tecnologías de la Información y Comunicaciones (TIC), lo que muestra la amplia oportunidad laboral en este campo. A partir de esta necesidad surge Talento Digital que es el resultado del trabajo conjunto del Ministerio TIC en alianza con Icetex y la Asociación Colombiana de Ingenieros de Sistemas (ACIS).

[[1]](#footnote-1)Talento Digital es un fondo que busca formar talento humano en TIC por medio de convocatorias que ofrecen créditos condonables hasta el 100% de la matrícula para animar a la formación de los colombianos en carreras TIC, para impulsar la competitividad, la investigación, la innovación y la proyección internacional del sector TIC de Colombia.

Las TIC están constantemente transformando a Colombia y el mundo, ellas apoyan al crecimiento de la calidad de vida de las personas, así mismo, para encontrar un proyecto de vida encaminado a un desarrollo laboral y profesional.

El programa de Talento Digital en la actualidad controla el seguimiento de sus beneficiarios por medio de formatos de Excel lo que hace que sea una labor tediosa al momento de generar reportes y estadísticas con estos datos.

* 1. **ÁREA DE CONOCIMIENTO**

Las áreas de conocimiento en las cuales se va enfocar el proyecto son la ingeniería de software, bases de datos y el análisis de sistemas.

* 1. **ALCANCES Y DELIMITACIONES**
* Se entregará una aplicación web en la que los funcionarios del programa de Talento Digital puedan realizar seguimiento a los estudiantes beneficiarios.
* La aplicación podrá ser accedida desde un computador con acceso a internet.
* Solo se permitirá el acceso a usuarios previamente verificados por el programa de Talento Digital.
* Cualquier estudiante beneficiario de Talento Digital podrá crear su perfil.

1. **JUSTIFICACIÓN**

Debido a la gran cantidad de beneficiarios con los que cuenta el programa talento Digital, el proceso de seguimiento es complejo dado que no se cuenta con un sistema de información que permita almacenar, procesar y analizar la información. Por esta razón el Ministerio TIC necesita de un sistema que mejore dicha labor de seguimiento por parte de los funcionarios de Talento Digital; actualmente, existen más de 6.000 beneficiarios en toda Colombia. Este nuevo sistema será una herramienta de gran utilidad para los estudiantes que se encuentran realizando sus estudios en las áreas de tecnologías TIC gracias al programa de Talento Digital impulsado por el Gobierno colombiano y el Ministerio de TIC. Este proyecto será desarrollado utilizando tecnologías web y frameworks de desarrollo actualizados, para que desde cualquier dispositivo que tenga conexión a Internet pueda beneficiarse con sus múltiples características.

1. **OBEJTIVOS GENERAL Y ESPECÍFICOS**
   1. **OBJETIVO GENERAL**

Desarrollar una aplicación web mediante la cual los funcionarios de Talento Digital puedan llevar el control y seguimiento de todos los beneficiarios en todo el país.

* 1. **OBJETIVOS ESPECÍFICOS**
* Desarrollar una aplicación web que se adapte a cualquier dispositivo.
* Brindar una nueva herramienta que permita mejorar el seguimiento de los beneficiarios Talento Digital.
* Ofrecer a los funcionarios de Talento Digital un módulo de reportes para extraer información de los beneficiarios de Talento Digital.

1. **MARCO REFERENCIAL**
   1. **MARCO HISTÓRICO**

En el año 2012 es lanzada la primera convocatoria del programa Talento Digital con el objetivo de incentivar a los jóvenes colombianos a estudiar carreras relacionadas con las tecnologías de información y comunicaciones (TIC), hoy 3 años después son más de 6.000 beneficiarios en formación técnica, tecnológica, ingeniería y posgrado.

Los beneficiarios deben llenar un formato en Excel con toda su información sobre el estudio que están realizando y el proyecto que desean realizar para retribuir el conocimiento adquirido a la sociedad, dicho formato se envía por medio de correo electrónico y los funcionarios de Talento Digital no cuentan con una herramienta para unificar toda esta información para dar un seguimiento a los estudiantes.

* 1. **MARCO TEÓRICO**
     1. **Sistema de información**

Un sistema de información es un conjunto de componentes que están relacionados entre sí que permiten capturar, procesar, interpretar, guardar y distribuir la información sirviendo como un soporte al momento de tomar decisiones. Existe una gran cantidad de tipos de sistemas de información en su gran mayoría son representados por medio de bloques, elementos de entrada, elementos de salida, transformación de la información, mecanismos de control y objetos.

Como se ve en la figura 1, los recursos ingresan al sistema de información por medio de los elementos de entrada, seguido de esto son modificados en la sección de transformación. Este proceso es administrado por el mecanismo de control. Una vez la información en el sistema cumple su objetivo sale a través de los elementos de salida.

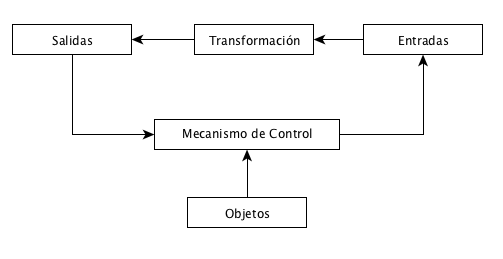
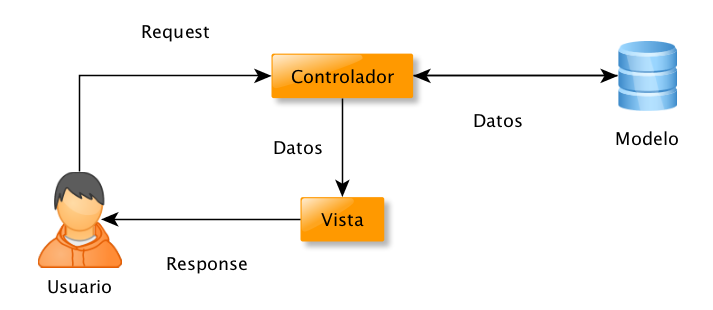


Figura 1. Modelo General Sistema de Información

* + 1. **MVC (modelo vista controlador)**

MVC es un patrón de arquitectura de software que se encarga de separar los datos de una aplicación, la interfaz gráfica de usuario (GUI) y la lógica de control en tres componentes distintos, es muy utilizado en aplicaciones web, donde la vista seria la página HTML y el código que entrega los datos a la página. El modelo es el sistema de gestión de bases de datos y la lógica del negocio, por último, el controlador es el responsable de recibir los eventos de entrada de la vista.

* **Modelo:** Este es una representación de la información la cual es utilizada en el sistema. El modelo se limita a lo que tiene que ver con la vista y su controlador facilitando las presentaciones visuales complejas. El sistema a su vez puede operar con más datos no relativos a la prestación, haciendo uso integrado de otras lógicas de negocio y de datos que tengan que ver con el sistema en cuestión.
* **Vista:**Esta es usualmente la interfaz de usuario en la cual se muestran los datos y se puede interactuar con ellos.
* **Controlador:** Este es el encargado de escuchar los eventos que son acciones del usuario invoca peticiones al modelo y en ocasiones a la vista.

  
Figura 2 Modelo vista controlador.

Utilizando MVC la aplicación el desarrollo es rápido, de una forma modular y sostenible en el tiempo. Aislar las funciones de la aplicación en modelos, vistas y controladores nos ayuda a que la aplicación sea mucho más ligera. Cuando hay características nuevas en la aplicación estas se agregan con gran facilidad a las ya existentes.

Al tener un diseño modular permite que los desarrolladores y los diseñadores puedan trabajar conjuntamente, así como construir prototipos muy rápidamente. Esto también permite que se hagan cambios en la aplicación y en la vista sin que los demás componentes se vean afectados.

* + 1. **Lenguaje unificado de modelado (UML 2.0)**

Para cualquier rama de la ingeniería es mucho más fácil representar los diseños en una forma gráfica. Desde los inicios de la informática se han estado utilizando diversas formas de representación gráfica. Sin embargo, no había un estándar de estas representaciones gráficas y que se pudieran compartir fácilmente y la gente los entendiera.

Por esta razón se necesitaba una norma para poder comunicar las ideas a otros desarrolladores y a su vez que sirviera de apoyo al momento de analizar un problema. Con este objetivo se creó el Lenguaje Unificado de Modelado (UML: Unified Modeling Language). UML se ha convertido en la mejor norma para representar y modelar la información con la que se trabaja en las fases de análisis y diseño.

UML es un lenguaje que tiene una notación gráfica que es muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto que se esté construyendo, desde el análisis con casos de uso, el diseño con los diagramas de clases, objetos, secuencia, etc. hasta la implementación y configuración con los diagramas de despliegue.

* + 1. **Bases de datos NoSql**

Las llamadas bases de datos NoSql, también llamadas No Solo SQL, es un nuevo enfoque que va hacia la gestión de datos y el diseño de base de datos que es una herramienta para grandes conjuntos de datos distribuidos.

NoSql, busca resolver los problemas de escalabilidad y rendimiento de big data que las bases de datos relacionales no fueron diseñadas para abordar. NoSql es especialmente útil cuando se necesitan analizar una gran cantidad de datos no estructurados o datos que se almacenan de forma remota en varios servidores virtuales en la nube.

Por otro lado hay una idea falsa de que por su nombre las bases de datos NoSQL prohíben el lenguaje estructura de consultas (SQL). Si bien es cierto que algunos sistemas NoSQL son 100% no-relacionales, otros simplemente evitan funcionalidades relacionales seleccionadas como esquemas de tablas fijas y operaciones conjuntas. Por ejemplo, no se utilizan tablas, una base de datos NoSQL podría organizar los datos en objetos (JSON o XML), pares clave/valor o tuplas.

Se podría afirmar que la base de datos más popular NoSQL es Apache Cassandra. Cassandra, que una vez fue la base de datos propietaria de Facebook, fue liberada como código abierto en 2008. Otras implementaciones NoSQL incluyen SimpleDB, Google BigTable, Apache Hadoop, MapReduce, MemcacheDB y Voldemort. Las empresas que utilizan NoSQL incluyen NetFlix, LinkedIn y Twitter.

* + 1. **Firebase**

Firebase es un nuevo servicio online multiplataforma para construir aplicaciones ricas y potentes en tiempo real, Firebase es lo que se conoce como “Backend como servicio”, que básicamente provee de una API para guardar y sincronizar datos en la nube en tiempo real con todos los dispositivos que estén conectados, Firebase ayuda a los desarrolladores a que se preocupen más por sus aplicaciones y clientes que por la sincronización de sus aplicaciones y el backend, Entre sus características están:

* Tiene una sencilla implementación.
* Cuando el valor de una variable cambia se actualiza en cada uno de los dispositivos conectados.
* Cuenta con una forma de trabajar offline y se sincroniza cuando el servidor esta en línea.
* Los datos se guardan en un JSON standard, por ello es 100% multiplataforma mediante API REST, además existen SDKs para JavaScript, Objective-C y Java que aportan algunas funcionalidades extras.
* Mediante una librería llamada Firebase SimpleLogin tenemos soporte de autenticación para Facebook Login, Twitter oAuth y GitHub oAuth.
  + 1. **AngularJS**

AngulasJS es un framework MVC de javascript utilizado en el desarrollo Front End que permite crear aplicaciones SPA (Single page applications). Anteriormente la alternativa más común para manipular los datos en la página web era Jquery pero esta solución no tenía un patrón a seguir. Todo el código estaba en funciones que se iban creando según la necesidad, con el tiempo esto se volvía difícil de manejar.

AngularJS está pisando fuerte. Aunque su primera versión es de 2009, se ha hecho muy popular a finales de 2012 y ahora en 2013 está en pleno auge. AngularJS permite extender el vocabulario HTML con directivas y atributos, manteniendo la semántica y sin necesidad de emplear librerías externas como jQuery o Underscore.js para que funcione.

1. **DISEÑO METODOLÓGICO**
   1. **LEVANTAMIENTO DE REQUERIMIENTOS**
      1. **Actores**

|  |  |  |
| --- | --- | --- |
| ID | Nombre | Función |
| Actor1 | Administrador | Es el encargado de crear los usuarios para cada uno de los funcionarios, administrar los parámetros del sistema. |
| Actor2 | Funcionario | Es la persona con la posibilidad de consultar toda la información de los beneficiarios, generar reportes con dicha información. |
| Actor3 | Beneficiario | Es el estudiante que es beneficiario del programa de talento digital, el cual puede ingresar sus datos personales, datos de contacto e información a cerca de su trabajo de grado. |

Tabla 1. Lista de Actores del sistema.

* + 1. **Requerimientos funcionales**

Luego de tener varias reuniones con los funcionarios de Talento Digital, se realizo la etapa de análisis para el proyecto, de este análisis se lograron identificar los siguientes requerimientos funcionales, que son:

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ1 | Talento Digital |
| Descripcion | |
| El sistema debe contar con un módulo para poder administrar usuarios para los funcionarios. | |
| Detalle | |
| El usuario administrador podra acceder a este modulo y administrar usuarios para los funcionarios de talento digital con los siguientes datos:   * Correo * Clave * Nombre | |
| Restricciones | |
| Este modulo solo puede ingresar el usuario administrador del sistema. | |

Tabla 2. Especificación RQ1

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ2 | Talento Digital |
| Descripcion | |
| Los beneficiarios del programa Talento Digital podrán crear su cuenta en el sistema, y podrán ingresar su información personal, sus datos de contacto, información académica e información sobre el proyecto de grado que desea realizar. | |
| Detalle | |
| Los beneficiarios podran ingresar la siguiente información:   * Información Personal   + Nombre completo   + Tipo de indentificación   + Número de indentificación   + Dirección   + Departamento   + Ciudad   + Correo electrónico   + Convocatoria. * Datos de la Universidad   + Nombre de la Universidad   + Ciudad   + Departamento   + Nivel de formación   + Nombre del programa   + Semetre que cursa   + Semestre que inicio con Talento Digital * Datos de la entidad de naturaleza publica del proyecto a realizar   + Nombre de la entidad   + Ciudad   + Departamento   + Dependencia responsable del proyecto   + Coordinador y responsable del proyecto   + Dirección de contacto   + Teléfono fijo   + Correo electronico   + Teléfono celular * Datos del proyecto   + Nombre del proyecto   + Descripcion del proyecto   + Objetivos   + Relación del proyecto con la estrategia de gobierno en línea   + Avance del proyecto   + Metodologia a seguir   + Observaciones   + Contacto con la entidad   + Necesita carta de presentación   + Duración (Meses). | |
| Restricciones | |
|  | |

Tabla 3. Especificación RQ2

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ3 | Talento Digital |
| Descripcion | |
| Los usuarios que deseen recuperar su clave por medio del sistema en caso de olvido lo pueden hacer. | |
| Detalle | |
| Los usuarios que olviden su clave podran solicitarla al sistema ingresando su correo electronico al cual se les enviara los pasos para recuperar dicha clave. | |
| Restricciones | |
| Solamente usuarios registrados podran recuperar su clave | |

Tabla 4. Especificación RQ3

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ4 | Talento Digital |
| Descripcion | |
| Los funcionarios deben poder generar reportes gráficos con la información que exista de los beneficiarios. | |
| Detalle | |
| Los funcionarios podran generar reportes que se podran ver en el sistema con la informacion que este en ese momento en el sistema. | |
| Restricciones | |
| Los reportes son unicamente visuales y no se podran descargar. | |

Tabla 5. Especificación RQ4

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ5 | Talento Digital |
| Descripcion | |
| El sistema debe permitir realizar búsquedas en la información de los beneficiarios así como también aplicar filtros. | |
| Detalle | |
| El funcionario podra realizar filtros en los listados de beneficiarios por los siguientes criterios:   * Nombre * Cedula * Correo electronico | |
| Restricciones | |
| Los filtros son unicamente visuales para ayudar a buscar beneficiarios. | |

Tabla 6. Especificación RQ5

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ6 | Talento Digital |
| Descripcion | |
| Los funcionarios pueden visualizar la información detallada de un beneficiario. | |
| Detalle | |
| Los funcionarios podran seleccionar un beneficiario y poder ver en detalle toda su información. | |
| Restricciones | |
|  | |

Tabla 7. Especificación RQ6

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ7 | Talento Digital |
| Descripción | |
| El administrador podrá cambiar su dirección de correo. | |
| Detalle | |
| Para poder cambiar la cuenta de correo se deben ingresar los siguientes valores.   * Correo Actual * Correo Nuevo * Clave Actual. | |
| Restricciones | |
|  | |

Tabla 8. Especificación RQ7

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ8 | Talento Digital |
| Descripcion | |
| Los funcionarios podrán visualizar un listado completo de los Beneficiarios. | |
| Detalle | |
| Se debe mostrar un listado completo de los beneficiarios registrados en el sistema. | |
| Restricciones | |
|  | |

Tabla 9. Especificación RQ8

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ9 | Talento Digital |
| Descripcion | |
| Todos los usuarios deben iniciar sesión. | |
| Detalle | |
| Todos los usuarios deben autenticarse ante el sistema con usuario y clave. | |
| Restricciones | |
|  | |

Tabla 10. Especificación RQ9

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ10 | Talento Digital |
| Descripcion | |
| Todos los usuarios podrán cambiar su clave. | |
| Detalle | |
| Todos los usuarios para poder cambiar la clave deben ingresar la clave actual y una nueva clave. | |
| Restricciones | |
|  | |

Tabla 11. Especificación RQ10

|  |  |
| --- | --- |
| ID RQ | Solicitante |
| RQ11 | Talento Digital |
| Descripcion | |
| Todos los usuarios podrán cerrar la sesión. | |
| Detalle | |
| Todos los usuarios podrán cerrar la sesión. | |
| Restricciones | |
|  | |

Tabla 12. Especificación RQ10

* + 1. **Requerimientos no funcionales**

De acuerdo al levantamiento de requerimientos que se realizó, el sistema de información para el seguimiento de los beneficiarios del programa de talento digital debe tener los siguientes requerimientos no funcionales:

* Debe ser responsive desing.
* Debe ser un sistema seguro.
* Debe estar disponible desde cualquier computador con acceso a internet.
* Debe tener interfaces amigables para el usuario.
  1. **MODELADO DEL SISTEMA**
     1. **Casos de uso**

En la fase de análisis y diseño, se modelaron los casos de uso basándose en los requerimientos funcionales que se identificaron durante las reuniones que se tuvieron con los funcionarios del programa Talento Digital.

La siguiente es una lista preliminar de casos de uso:

* Usuario
  + CU01 - Autenticarse.
  + CU02 - Cambiar clave.
  + CU03 - Recuperar clave.
  + CU04 - Cerrar sesión.
* Administrador
  + CU05 - Agregar Usuario.
  + CU06 - Consultar Usuarios.
  + CU07 - Modificar Usuario.
  + CU08 - Eliminar Usuario.
  + CU09 - Cambiar Correo.
* Funcionario
  + CU10 - Consultar beneficiarios.
  + CU11 - Filtrar listado de beneficiarios.
  + CU12 - Consultar detalle de un beneficiario.
  + CU13 - Generar reportes de los beneficiarios.
* Beneficiario
  + CU14 - Registrarse en el sistema.
  + CU15 - Actualizar datos personales.
  + CU16 - Actualizar datos de la Universidad.
  + CU18 - Actualizar datos del proyecto.
  + CU19 - Actualizar datos de la entidad pública.
    1. **Relación requerimientos casos de uso**

En la siguiente tabla se muestra la relación que tiene cada caso de uso con uno o varios requerimientos funcionales del sistema.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actor** | **Caso de uso** | **RQ1** | **RQ2** | **RQ3** | **RQ4** | **RQ5** | **RQ6** | **RQ7** | **RQ8** | **RQ9** | **RQ10** | **RQ11** |
| Usuario | CU01 |  |  |  |  |  |  |  |  |  |  |  |
| CU02 |  |  |  |  |  |  |  |  |  |  |  |
| CU03 |  |  |  |  |  |  |  |  |  |  |  |
| CU04 |  |  |  |  |  |  |  |  |  |  |  |
| Administrador | CU05 |  |  |  |  |  |  |  |  |  |  |  |
| CU06 |  |  |  |  |  |  |  |  |  |  |  |
| CU07 |  |  |  |  |  |  |  |  |  |  |  |
| CU08 |  |  |  |  |  |  |  |  |  |  |  |
| CU09 |  |  |  |  |  |  |  |  |  |  |  |
| Funcionario | CU10 |  |  |  |  |  |  |  |  |  |  |  |
| CU11 |  |  |  |  |  |  |  |  |  |  |  |
| CU12 |  |  |  |  |  |  |  |  |  |  |  |
| CU13 |  |  |  |  |  |  |  |  |  |  |  |
| Beneficiario | CU14 |  |  |  |  |  |  |  |  |  |  |  |
| CU15 |  |  |  |  |  |  |  |  |  |  |  |
| CU16 |  |  |  |  |  |  |  |  |  |  |  |
| CU17 |  |  |  |  |  |  |  |  |  |  |  |
| CU18 |  |  |  |  |  |  |  |  |  |  |  |

Tabla 10. Relación requerimientos casos de uso

* + 1. **Especificación casos de uso**

Para la especificación de los casos de uso se utilizó el formato conocido como especificación de caso de uso, dentro de su estructura permite relacionar de manera específica las actividades que podrá realizar el usuario en el sistema. A continuación, se enunciarán cada uno de los campos que componen dicho formato y se agregará una descripción para cada uno de ellos:

* Número: numeración que se utiliza para identificar cada uno de los casos de uso.
* Identificador: nombre que se le asigna al caso de uso.
* Descripción: un breve resumen que introduce lo que se podrá  realizar por medio del caso de uso.
* Actores: usuario del sistema involucrado en la ejecución del caso de uso.
* Entradas: variables de entrada que debe recibir el caso de uso para ejecutarse de manera satisfactoria.
* Salidas: respuesta que mostrará el sistema luego de haberse ejecutado el caso de uso.
* Precondición: hace referencia a una serie de condiciones que se deben cumplir antes de poder ejecutar el caso de uso.
* Post-condición: hace referencia al estado en el cual permanece el sistema luego de haber ejecutado de manera exitosa el listado de actividades del caso de uso.
* Flujo normal de eventos: es el listado de actividades que se ejecutan de manera secuencial tanto por parte del usuario como del sistema.
* Flujos alternos: hacen referencia a las actividades que no están presentes dentro del flujo normal de eventos del caso de uso.
* Caminos de excepción: hacen referencia a los errores que pueden presentarse durante la ejecución de un caso de uso y definen como debe comportarse el sistema ante los mismos.

La especificación de los casos de uso se muestra a continuación:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Número** | | | **CU01** | | |
| **Identificador** | | | Autenticarse | | |
| **Descripción:**  El actor podrá modificar sus credenciales de ingreso al sistema de administración de los planes analíticos. | | | | | |
| **Actores:**  Usuario | | | | | |
| **Entradas:**  Correo, Clave. | | | | **Salidas:**  Se permite el acceso al sistema. | |
| **Precondición:**  El usuario debe ingresar correo y clave que sean validados en el sistema. | | | | **Postcondición:**  Al validarse el usuario y la clave se le permite el acceso al sistema. | |
| **Flujo normal de eventos** | | | | | |
|  | **Acción del actor** | |  | | **Acción del sistema** |
| **1** | El usuario ingresa su correo y clave. | | **2** | | El sistema valida los datos del usuario y permite el acceso al sistema para el usuario. |
|  |  | |  | |  |
| **Flujos Alternos:**  **2** El sistema muestra el mensaje de error correspondiente al validar la información del usuario. | | | | | |
| **Caminos de excepción:**  **2** El sistema debe mostrar el mensaje de error y permitir que el usuario lo intente nuevamente. | | | | | |
| **Autor** | **Fecha** | | **Versión** | | **Creación/Modificación** |
| Juan Salazar | | 10/10/2015 | 1.0 | |  |

Tabla especificación casos de uso - Autenticarse

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Número** | | | **CU02** | | |
| **Identificador** | | | Cambiar clave | | |
| **Descripción:**  El actor podrá cambiar la clave para autenticarse con el sistema. | | | | | |
| **Actores:**  Usuario | | | | | |
| **Entradas:**  Correo, Clave Actual, Clave Nueva. | | | | **Salidas:**  Mensaje de confirmación cambio clave.. | |
| **Precondición:**  El usuario debe estar autenticado en el sistema. | | | | **Postcondición:**  Se actualiza la nueva clave en la base de datos. | |
| **Flujo normal de eventos** | | | | | |
|  | **Acción del actor** | |  | | **Acción del sistema** |
| **1** | El usuario ingresa su correo, clave Actual y Clave Nueva. | | **2** | | El sistema valida los datos del usuario y cambia la clave en el sistema. |
|  |  | |  | |  |
| **Flujos Alternos:**  **2** El sistema muestra el mensaje de error correspondiente al validar la información del usuario. | | | | | |
| **Caminos de excepción:**  **2** El sistema debe mostrar el mensaje de error y permitir que el usuario lo intente nuevamente. | | | | | |
| **Autor** | **Fecha** | | **Versión** | | **Creación/Modificación** |
| Juan Salazar | | 10/10/2015 | 1.0 | |  |

Tabla especificación casos de uso – Cambiar Clave.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Número** | | | **CU03** | | |
| **Identificador** | | | Recuperar clave | | |
| **Descripción:**  El actor podrá recuperar la clave para autenticarse con el sistema. | | | | | |
| **Actores:**  Usuario | | | | | |
| **Entradas:**  Correo. | | | | **Salidas:**  Mensaje de confirmación de envió clave al correo. | |
| **Precondición:**  El usuario debe existir en el sistema. | | | | **Postcondición:**  Se actualiza la nueva clave en la base de datos y se envía por correo al usuario. | |
| **Flujo normal de eventos** | | | | | |
|  | **Acción del actor** | |  | | **Acción del sistema** |
| **1** | El usuario ingresa su correo, clave Actual y Clave Nueva. | | **2** | | El sistema valida los datos del usuario y cambia la clave en el sistema. |
|  |  | |  | |  |
| **Flujos Alternos:**  **2** El sistema muestra el mensaje de error correspondiente al validar la información del usuario. | | | | | |
| **Caminos de excepción:**  **2** El sistema debe mostrar el mensaje de error y permitir que el usuario lo intente nuevamente. | | | | | |
| **Autor** | **Fecha** | | **Versión** | | **Creación/Modificación** |
| Juan Salazar | | 10/10/2015 | 1.0 | |  |

Tabla especificación casos de uso – Recuperar Clave.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Número** | | | **CU04** | | |
| **Identificador** | | | Cerrar Sesión. | | |
| **Descripción:**  El actor podrá cerrar su sesión en el sistema. | | | | | |
| **Actores:**  Usuario | | | | | |
| **Entradas:** | | | | **Salidas:**  Se le informa al usuario que la sesión fue cerrada y se lo envía a la pagina de login. | |
| **Precondición:**  El usuario debe tener la sesión activa en el sistema. | | | | **Postcondición:**  Se borran los datos de la sesión actual y se envía a la pagina de login. | |
| **Flujo normal de eventos** | | | | | |
|  | **Acción del actor** | |  | | **Acción del sistema** |
| **1** | El usuario da click en el botón cerrar sesión. | | **2** | | El sistema borrar los datos correspondientes a esa sesión y dirige al usuario a la pagina de login. |
|  |  | |  | |  |
| **Flujos Alternos:**  No hay | | | | | |
| **Caminos de excepción:**  No hay | | | | | |
| **Autor** | **Fecha** | | **Versión** | | **Creación/Modificación** |
| Juan Salazar | | 10/10/2015 | 1.0 | |  |

Tabla especificación casos de uso – Cambiar Clave.

* + 1. **Diagramas casos de uso**

**Usuario:**

En el siguiente diagrama, se muestran lo que pueden realizar los usuarios del sistema cuando hayan ingresado al mismo. Estos casos de uso pertenecen a cualquier usuario que este autenticado en el sistema.

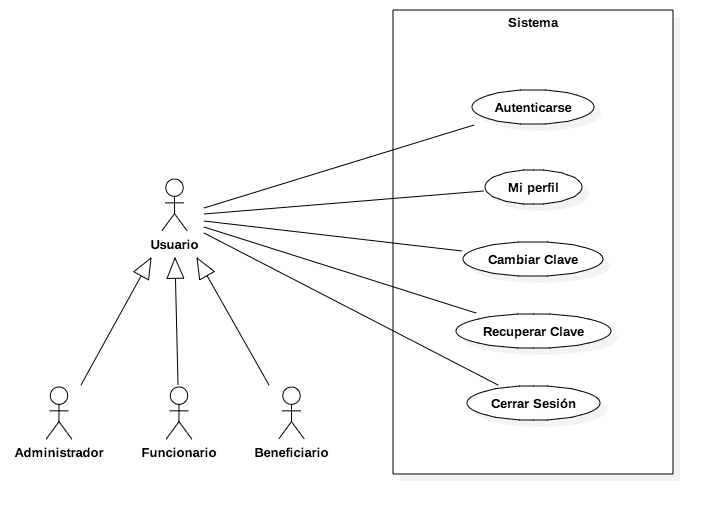
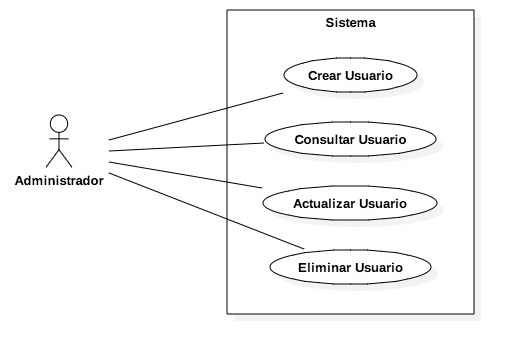
****

Diagrama casos de uso – Usuario.

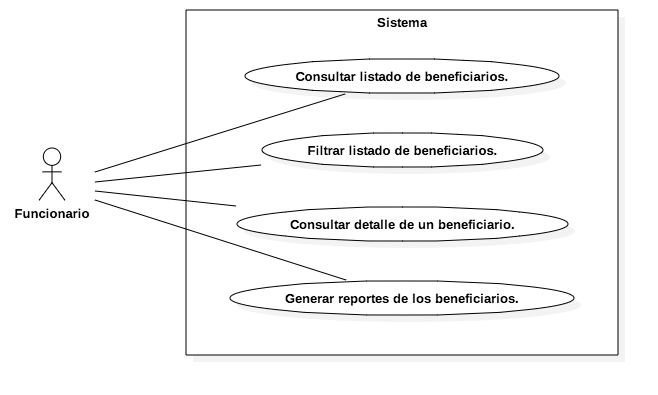
**Administrador:**

En el siguiente diagrama se muestran las actividades que puede hacer un usuario autenticado con el rol de administrador:

  
Diagrama caso de uso – Administrador

**Funcionario:**

En el siguiente diagrama se puede observar las actividades que puede realizar un usuario con el rol de funcionario.

****Diagrama de caso de uso – Funcionario.

**Beneficiario:**

En el siguiente diagrama se puede observar las actividades que puede realizar un usuario con el rol de beneficiario.

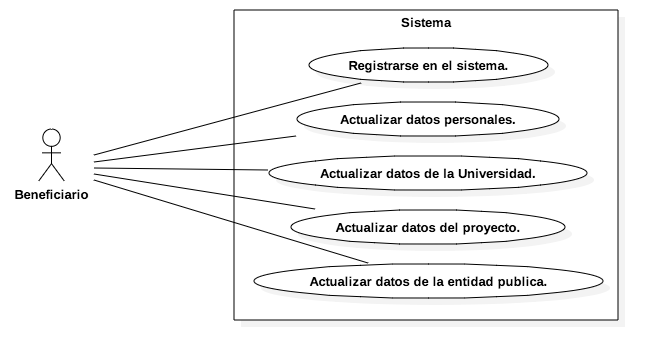
****

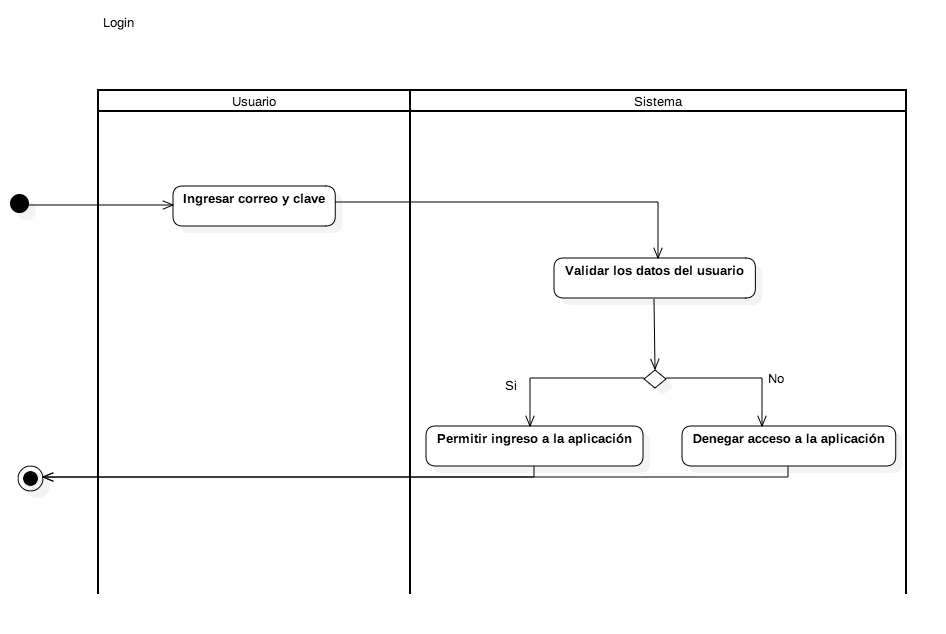
Diagrama de caso de uso – Beneficiario.

* + 1. **Diagramas de actividad**

El objetivo principal de la decisión de usar diagramas de actividad, de esta manera se puede mostrar en un alto nivel la interacción que van a tener cada uno de los usuarios del sistema con cada uno de los casos de uso, para realizar estos diagramas de actividad es necesario tener como prerrequisito los diagramas de casos de uso ya que son un punto de partida clave para estos.

Para mantener la misma línea de diseño de los diagramas de actividades, se utilizó la herramienta StarUML.

A continuación se muestran los diagramas de actividad del sistema de información :

  
Diagrama de actividad – Login.

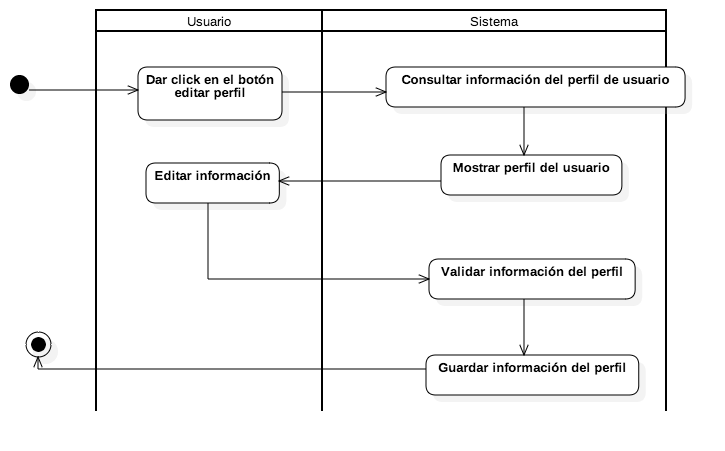
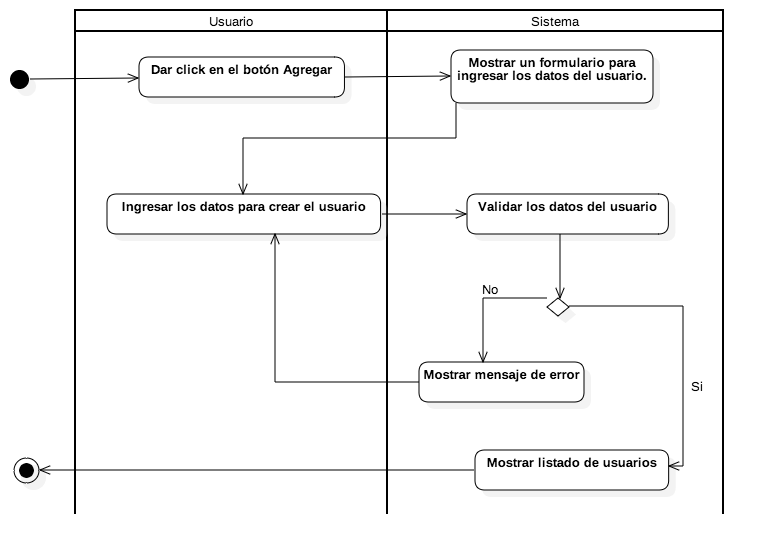
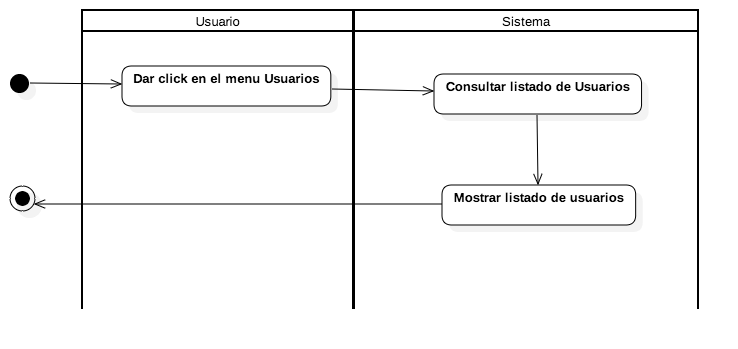
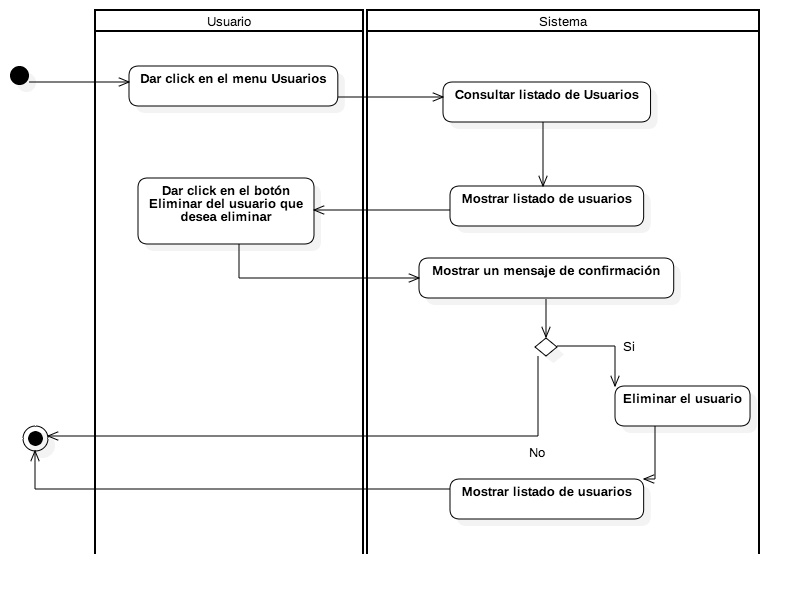


Diagrama de actividad – Editar información de perfil.

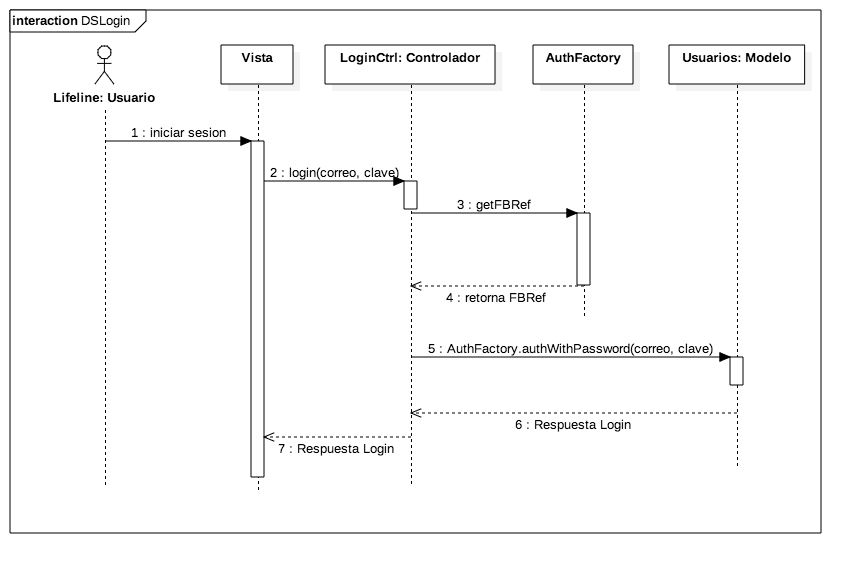
****Diagrama de actividad- Agregar Usuario.

****Diagrama de actividad – Consultar usuario**.**

****Diagrama de actividad – Eliminar Usuario.

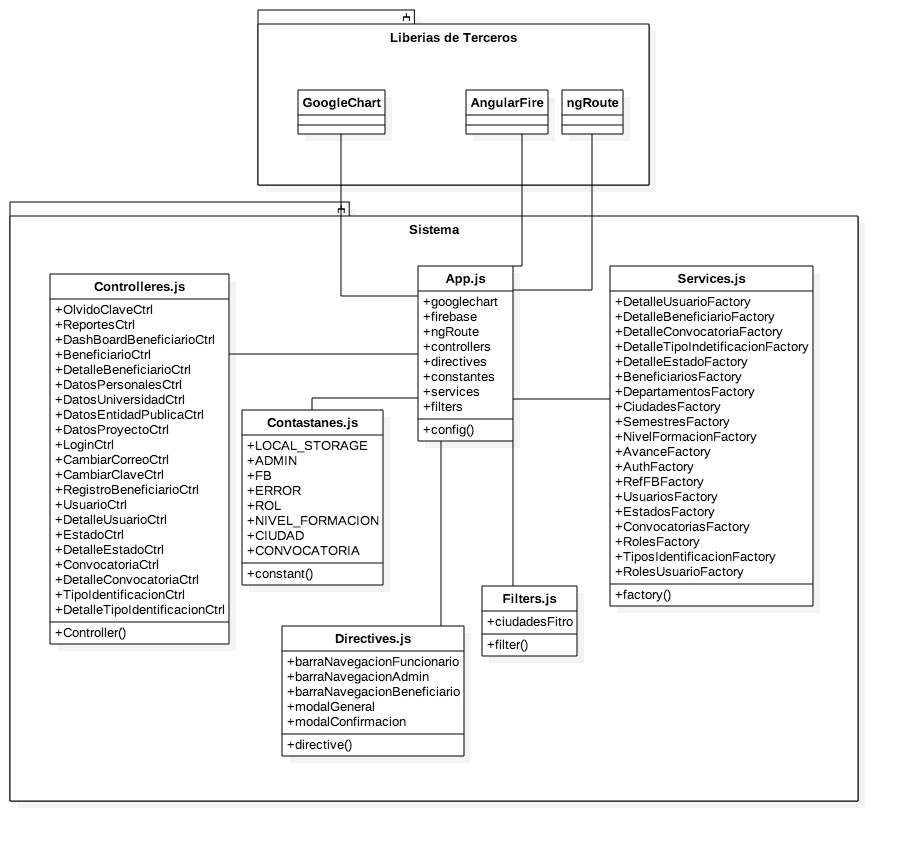
* + 1. **Diagramas de secuencia.**

Los diagramas de secuencia son usados en este caso para ilustrar la interacción del actor con los objetos dentro del sistema, el diagrama con tiene al detalle de los objetos y métodos que están presentes, en el siguiente diagrama se muestra como será la secuencia de interacción en el caso de uso Autenticarse:

  
Diagrama secuencia – login.

* + 1. **Arquitectura del sistema.**

Para este sistema se uso AngularJs, es un framework en Javascript para el desarrollo de aplicaciones web con lógica del lado del cliente, es desarrollo por Google, para la arquitectura del sistema se va usar el patrón estereotipo, en este patrón se tiene un fichero con una función en especifico, el cual es almacenar y agrupar los elementos en común siguiendo la siguiente agrupación como se muestra en la figura:



Arquitectura del sistema.

**App.js**

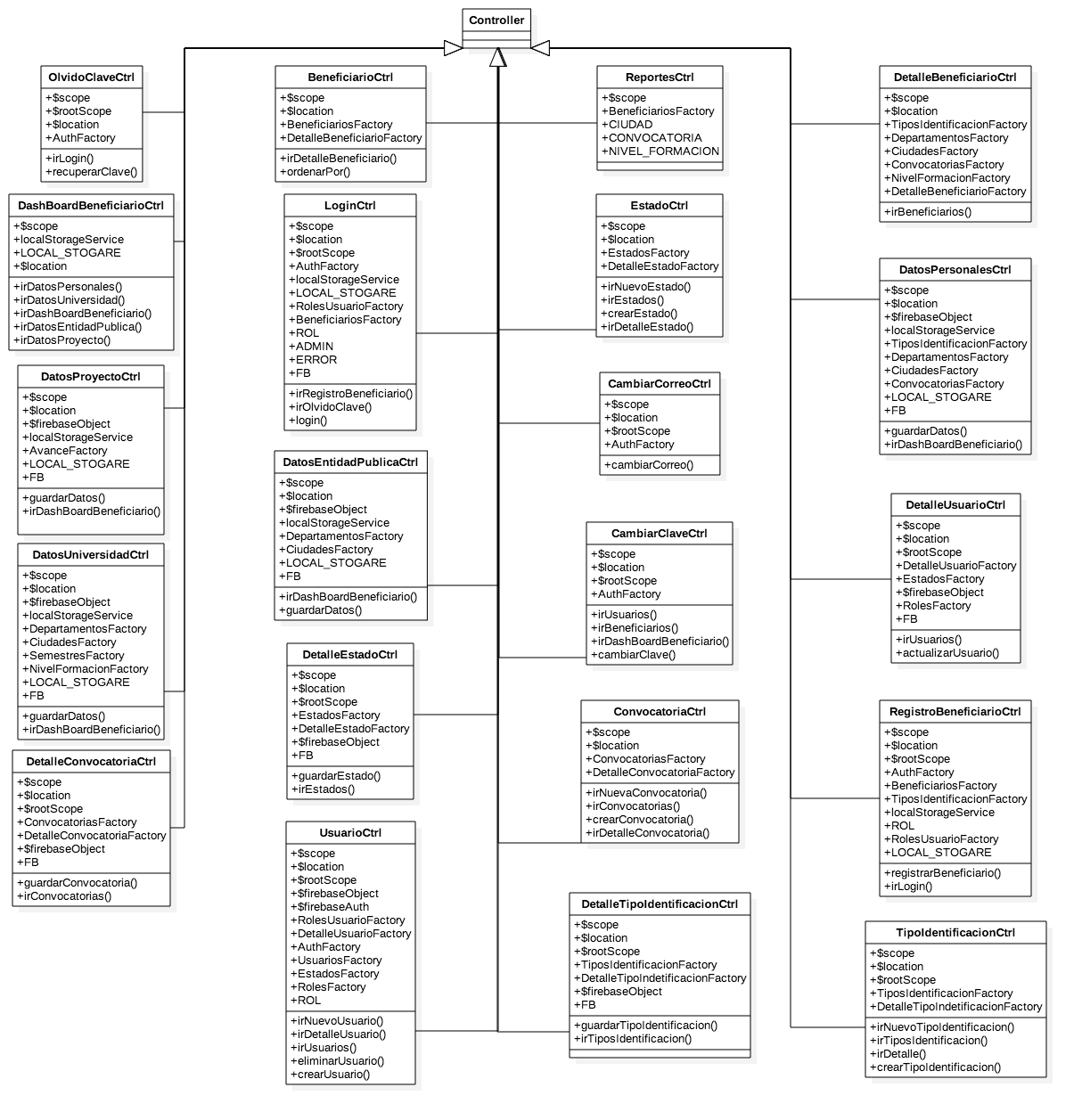
Este fichero contiene la configuración general de la aplicación y también agrupa los demás ficheros, filtres, directives, controllers y services.

**Controllers.js**

Los controllers son los encargados de inicializar y modificar la información que contienen los scopes (Los scopes son los distintos contextos de ejecución sobre los que trabajan las expresiones de AngularJS) en función de las necesidades de la aplicación.

También se pueden declarar funciones en el scope que se podrán utilizar más tarde o ser llamadas desde la vista.

En la imagen se muestra un diagrama en alto nivel cada uno de los controladores que se van a usar en el sistema, los controladores en AngularJs son los encargados de procesar la lógica para cada una de las vistas sin embargo puede existir un controlador para varias vistas de esta manera se reutilizan funciones en común que tengan las vistas.

  
Diagrama de clases controladores.

**Directives.js**

Mediante el uso de las directivas se pueden extender sintaxis de HTML y darle el comportamiento según el objetivo que se quiera llegar a tener. Se pueden crear directivas a nivel de elemento o de un atributo en particular, de clase y de comentario.

**Filters.js**

En este fichero estarán ubicados los filtros que se vayan a crear para modificar la información que se esta mostrando en la vista.

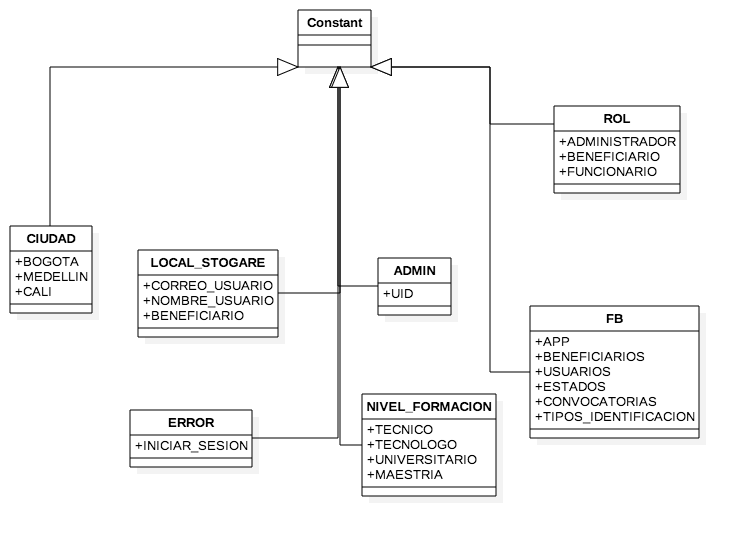
**Services.js**

En este fichero se encuentran ubicados los services y factories, utilizados para comunicarse con el servidor y compartir y reutilizar estos mismos en los controladores, de esta manera se pueden reutilizar funciones.

**Constantes.js**

En este fichero se van almacenar todas las constantes que se van a usar en el sistema con el objetivo de optimizar el uso de ellas en los diferentes ficheros que las necesiten usar.

En el siguiente diagrama se muestra un diagrama en alto nivel de cómo están agrupadas las constantes de acuerdo a sus objetivos en común para ser usados mas adelante en cada uno de los controladores de la aplicación.

  
Imagen - Diagrama de clases constantes.

* + 1. **Modelo de datos**

Para este sistema se utilizara una base de datos en formato JSON ya que es una base de datos noSQL. Utilizando objetos, arreglos y datos nativos como String y Number en su estructura, a continuación se muestra el diseño de base de datos para cada una de las entidades.

**Entidad Beneficiario**

*"beneficiarios" : {*

*"ID\_BENEFICIARIO" : {*

*"ciudad" : ,*

*"convocatoria" : ,*

*"correo" : ,*

*"departamento" : ,*

*"nombre" : ,*

*"numeroIdentificacion" : ,*

*"tipoIdentificacion" :,*

*"datosEntidadPublica" : {*

*"celular" : ,*

*"ciudad" : ,*

*"coordinadorResponsable" : ,*

*"correo" : ,*

*"departamento" : ,*

*"dependenciaResponsable" : ,*

*"direccionContacto" : ,*

*"nombre" : ,*

*"telefono" :*

*},*

*"datosProyecto" : {*

*"avance" : ,*

*"cartaPresentacion" : ,*

*"contactoEntidad" : ,*

*"descripcion" : ,*

*"duracion" : ,*

*"metodologia" : ,*

*"nombre" : ,*

*"objetivos" : ,*

*"observaciones" : ,*

*"relacion" :*

*},*

*"datosUniversidad" : {*

*"ciudad" : ,*

*"departamento" : ,*

*"nivelFormacion" : ,*

*"nombre" : ,*

*"nombrePrograma" : ,*

*"semestre" : ,*

*"semestreInicioTD" :*

*}*

*},*

*}*

**Entidad Configuración**

*"configuracion" : {*

*"convocatorias" : {*

*"ID\_CONVOCATORIA" : {*

*"descripcion" : ,*

*"id" : ,*

*"nombre" :*

*}*

*},*

*"estados" : {*

*"ID\_ESTADOS" : {*

*"descripcion" : ,*

*"id" : ,*

*"nombre" :*

*}*

*},*

*"tiposIdentificacion" : {*

*"ID\_TIPO\_IDENTIFICACION" : {*

*"descripcion" : ,*

*"id" : ,*

*"nombre" :*

*}*

*}*

*}*

**Entidad Roles Usuario**

*"rolesUsuario" : {*

*"ID\_ROL\_USUARIO" : {*

*"rol" : ,*

*"uid" :*

*}*

*}*

**Entidad Usuarios**

*"usuarios" : {*

*"ID\_USUARIO" : {*

*"correo" : ,*

*"estado" : ,*

*"nombre" :*

*}*

*}*

* + 1. **Diccionario de datos**

Con el objetivo de describir cada uno de los datos que están en el modelo de datos JSON que se describió anteriormente se busco un formato que tuviera esta descripción, A continuación se muestra la documentación que se realizo para cada uno de los arrays, objetos y atributos. Se agrupo en las siguientes columnas:

* Nombre: Este es el nombre del elemento.
* Entidad: Aquí esta la entidad padre a la que pertenece.
* Tipo de dato: Contiene el tipo de dato del elemento puede ser (Array, Objeto, Number o String).
* Descripción: esta es una breve descripción del elemento.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Entidad** | **Tipo de dato** | **Descripción** |
| Beneficiarios | Raíz | Array | Listado de los beneficiarios del programa Talento Digital |
| ID\_BENEFICIARIO | Beneficiarios | Objeto | Identificador único de la entidad beneficiario. |
| ciudad | Beneficiario | Number | Id de la ciudad de residencia del beneficiario. |
| convocatoria | Beneficiario | Number | Id de la convocatoria en la cual en beneficiario inicio con talento digital |
| correo | Beneficiario | String | Correo del beneficiario. |
| departamento | Beneficiario | Number | id del departamento de residencia del beneficiario. |
| nombre | Beneficiario | String | Nombre completo del beneficiario. |
| TipoIdentificacion | Beneficiario | Number | id del tipo de identificación del usuario |
| datosEntidadPublica | Beneficiario | Objeto | Contiene los datos de la entidad publica en la que el beneficiario va a ejecutar su proyecto de grado. |
| celular | datosEntidadPublica | String | Numero de celular de el coordinador responsable en la entidad publica. |
| ciudad | datosEntidadPublica | Number | Id de la ciudad de residencia del coordinador responsable en la entidad publica. |
| coordinadorResponsable | datosEntidadPublica | String | Nombre de el coordinador responsable en la entidad pública. |
| correo | datosEntidadPublica | String | Correo de el coordinador responsable en la entidad pública. |
| departamento | datosEntidadPublica | Number | Id del departamento de residencia del coordinador responsable en la entidad publica. |
| dependenciaResponsable | datosEntidadPublica | String | Dependencia de la entidad publica. |
| direccionContacto | datosEntidadPublica | String | Dirección de contacto de la entidad pública. |
| nombre | datosEntidadPublica | String | Nombre de la entidad publica. |
| telefono | datosEntidadPublica | String | Numero de teléfono de la entidad publica. |
| datosProyecto | Beneficiario | Objeto | Objeto que contiene la información del proyecto que el beneficiario desea realizar |
| avance | datosProyecto | Number | Numero entre 0 y 100 que indica el avance del proyecto. |
| cartaPresentacion | datosProyecto | String | Indica si necesita carta de presentación o no para su proyecto. |
| contactoEntidad | datosProyecto | String | Indica si tiene contacto con la entidad publica o no. |
| descripcion | datosProyecto | String | Breve descripción del proyecto a realizar por el beneficiario. |
| duracion | datosProyecto | Number | Duración en meses del proyecto. |
| metodologia | datosProyecto | String | Metodología a seguir en el proyecto. |
| nombre | datosProyecto | String | Nombre del proyecto. |
| objetivos | datosProyecto | String | Objetivos del proyecto. |
| observaciones | datosProyecto | String | Observaciones del proyecto. |
| relacion | datosProyecto | String | Relación que tiene el proyecto con la estrategia de gobierno en línea. |
| datosUniversidad | Beneficiario | Objeto | Objeto que contiene los datos de la universidad en la que esta estudiando el beneficiario. |
| ciudad | datosUniversidad | Number | Id de la ciudad en la que esta ubicada la universidad. |
| departamento | datosUniversidad | Number | Id del departamento en que esta ubicada la universidad. |
| nivelFormacion | datosUniversidad | Number | Id del nivel de formación del beneficiario. |
| nombre | datosUniversidad | String | Nombre de la universidad. |
| nombrePrograma | datosUniversidad | String | Nombre del programa que cursa el beneficiario. |
| semestre | datosUniversidad | Number | Semestre que esta cursando el beneficiario. |
| semestreInicioTD | datosUniversidad | Number | Semestre en el que el beneficiario inicio con Talento Digital. |
| configuracion | Raiz | Array | Contiene las configuraciones del sistema. |
| convocatorias | configuracion | Array | Contiene las convocatorias de talento digital. |
| ID\_CONVOCATORIA | convocatorias | Objeto | Identificador único de la entidad convocatoria. |
| descripcion | convocatoria | String | Descripción de la convocatoria. |
| id | convocatoria | Number | Id de la convocatoria |
| nombre | convocatoria | String | Nombre de la convocatoria |
| estados | configuracion | Array | Contiene los estados del sistema. |
| ID\_ESTADOS | estados | Objeto | Identificador único de la entidad estado. |
| descripcion | estado | String | descripción del estado. |
| id | estado | Number | identificador del estado. |
| nombre | estado | String | nombre del estado. |
| tiposIdentificacion | configuracion | Array | Contiene los tipos de identificación usados en el sistema. |
| ID\_TIPO\_IDENTIFICACION | tiposIdentificacion | Objeto | Identificador único de la entidad tipo de Identificación. |
| descripcion | tipoIdentificacion | String | Descripción del tipo de identificación. |
| id | tipoIdentificacion | Number | identificador del tipo de identificación. |
| nombre | tipoIdentificacion | String | nombre del tipo de identificación. |
| rolesUsuario | Raiz | Array | Contiene el rol de cada usuario. |
| ID\_ROL\_USUARIO | rolesUsuario | Objeto | Identificador único de la entidad rol Usuario |
| rol | rolUsuario | Number | id del rol. |
| uid | rolUsuario | Number | identificador del usuario. |
| usuarios | Raiz | Array | Contiene el listado de usuarios funcionarios del sistema. |
| ID\_USUARIO | usuarios | Objeto | Identificador unico de la entidad usuario. |
| correo | usuario | String | Correo del funcionario. |
| estado | usuario | Number | estado del funcionario. |
| nombre | usuario | String | Nombre del funcionario. |

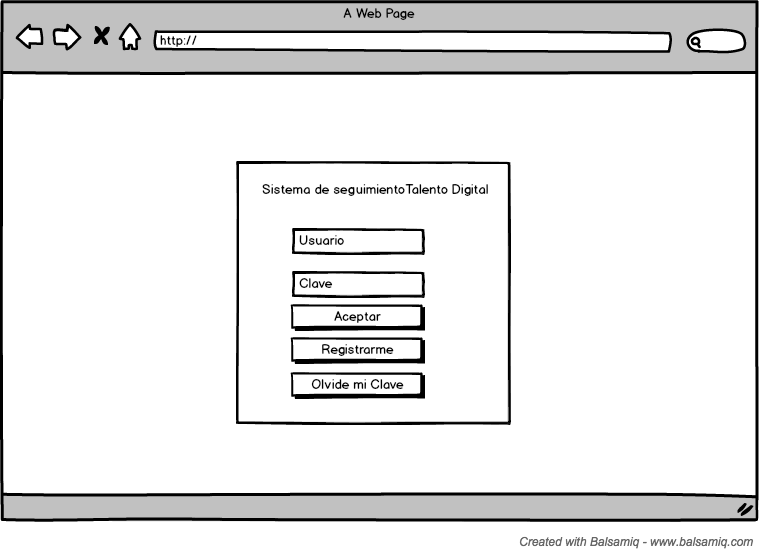
**Tabla Diccionario de datos.**

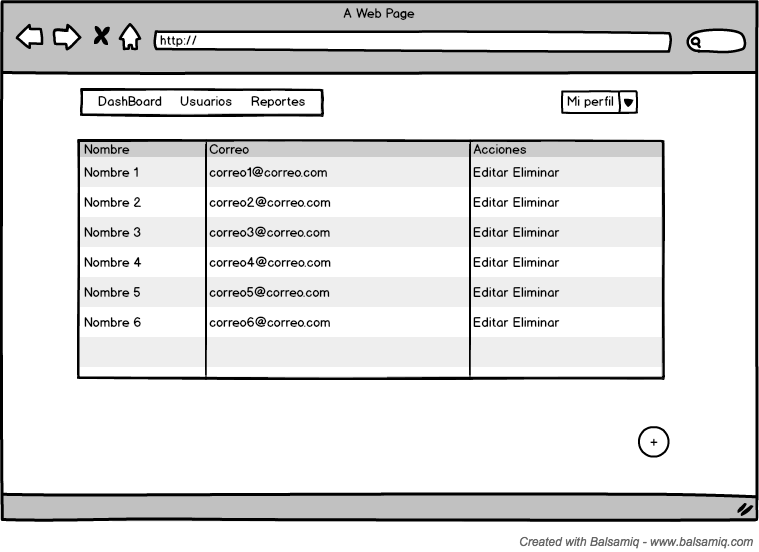
* + 1. **Interfaces de usuario Interfaces de usuario**

Como una buena practica se hace necesario elaborar unos prediseños de las interfaces de usuario antes de empezar con la etapa de desarrollo, esto con el propósito de dar una idea mas clara de lo que se quiere llegar con el sistema final, estas interfaces fueron evaluadas con los usuarios finales para tener una constante evaluación y llegar a construir un sistema optimo.

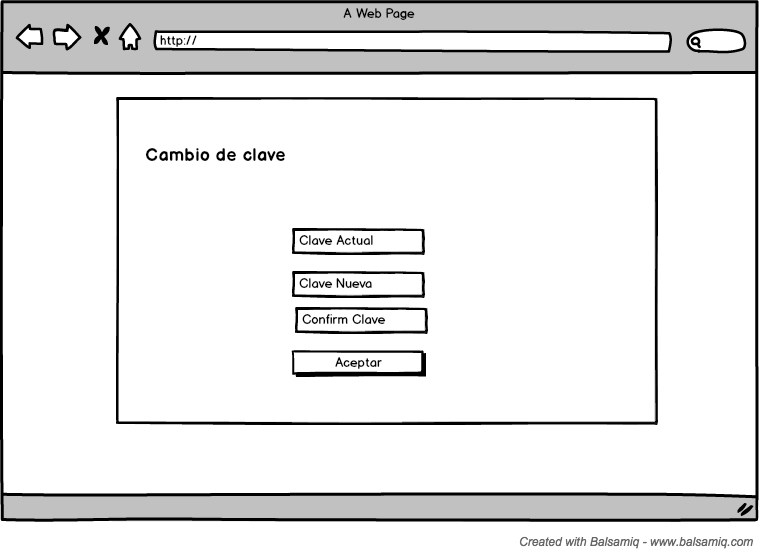
Los prediseños de las interfaces de usuario fueron realizados con la herramienta de software online llamada Balsamiq. Esta plataforma de diseño de interfaces o wireframes, cuenta con una gran variedad de elementos prediseñados, gracias a la interacción que se tiene con el software se pueden diseñar interfaces de usuario.

De lo anterior, a continuación se muestran las interfaces que se diseñaron para la inicio de sesión de los usuarios, lista de usuarios, reportes, cambiar clave del sistema:

  
Prediseño login

  
Prediseño listado de usuarios.

  
Prediseño pagina de reportes.



Prediseño pagina cambio de clave del sistema.

* 1. Desarrollo del sistema de información

En esta tercera fase del proyecto, se debía continuar con la elaboración de cada uno de los módulos propuestos según la especificación de requerimientos de la sección 5.1.2 del presente documento, utilizando el framework Angular.js para el frontend y firebase para toda la persistencia de datos y manejo de usuarios.

La primera etapa a superar el la de aprendizaje ya que se debía investigar como funciona las herramientas. La principal fuente de conocimiento fueron las paginas oficiales tanto de Angular.js como la de firebase las cuales cuentan con una amplia documentación y ejemplos para apoyar a los desarrolladores.

La segunda etapa fue crear una aplicación en firebase esta estaría ubicada en la pagina web de este framework de manera online, habilitando los servicios que estaban disponibles como almacenamiento de datos, hosting, gestión de usuarios y correos.

La tercera etapa fue buscar un repositorio para tanto el código fuente como la documentación, elijiendose GitHub ya que es un repositorio distribuido de gran acojida en la actualidad para trabajar proyectos opensource.

Por ultimo, se empezó el desarrollo de las funciones: Gestion de usuarios, reportes, consulta de beneficiarios, cambio de clave, cambio de correo, olvido de clave y configuración:

En el desarrollo se usarion librerías hechas por terceros como apoyo al desarrollo ya que proveeian una amplia gama de funciones y recursos para acelerar el proceso. En el listado que se muestra a continuación aparece la reacion de las tecnologías que se usaron.

Librería: Twitter Bootstrap   
Versión: 3.3.5   
Página de descarga: http://getbootstrap.com/   
Descripción: Permite crear páginas web autoajustables.

Librería: Bootstrap Material Desing  
Versión: 0.3.0  
Página de descarga: http://fezvrasta.github.io/bootstrap-material-design  
Descripción: Tema para boostrap con Material Desing.

Librería: Jquery   
Versión: 2.1.4  
Página de descarga: https://jquery.com/   
Descripción: Permite agregar dinamismo a las páginas web.

Librería: Angular.js  
Versión: 1.4.5  
Página de descarga: https://angularjs.org/   
Descripción: Framework fronted javascript.

Librería: Angular Google Chart  
Versión: 0.0.11  
Página de descarga: https://github.com/angular-google-chart/angular-google-chart  
Descripción: Permite agregar reportes alimentados con JSON.

Librería: Angular Local Storage  
Versión: 0.2.2  
Página de descarga: https://github.com/grevory/angular-local-storage  
Descripción: Permite acceder al local storage del browser

Librería: Angular Fire  
Versión: 1.1.2  
Página de descarga: https://github.com/firebase/angularfire/  
Descripción: Permite comunicar Angular con Firebase.

* + 1. Proceso de desarrollo

En la siguiente sección se van a describir cada uno de los pasos que se llevaron a cabo para ejecutar el desarrollo.

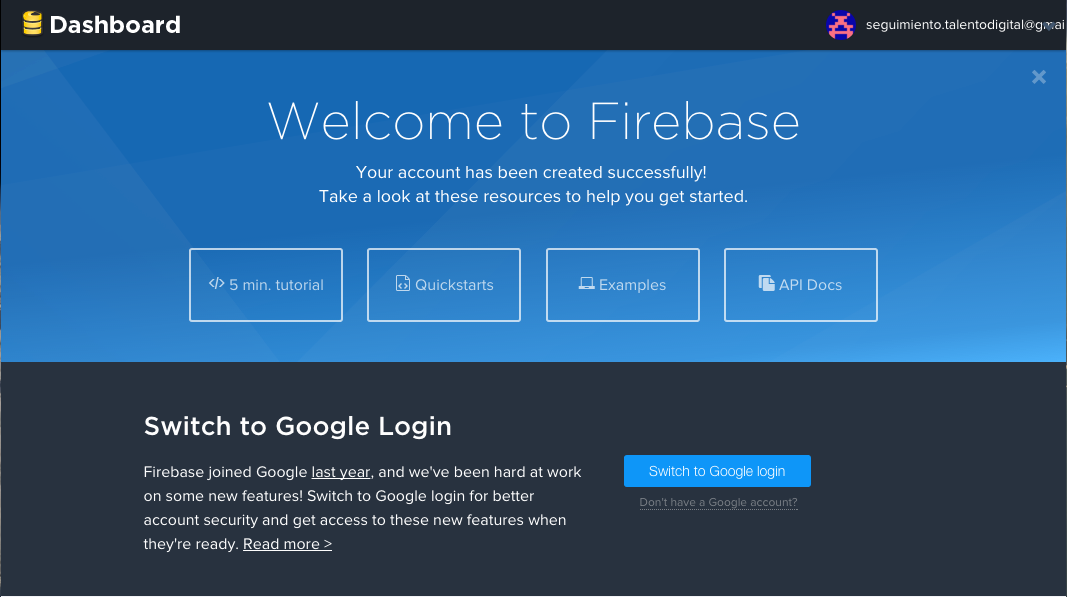
* Creación de la aplicación en firebase.

Para crear la aplicación se debe crear una cuenta en firebase en el siguiente enlace <https://www.firebase.com/login/>, esta aplicación fue creada con los siguientes datos:

Usuario: seguimiento.talentodigital@gmail.com

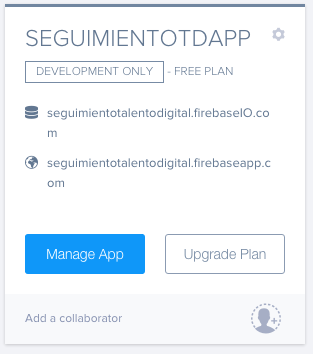
Clave: talentodigital2015\*

Una vez creada la cuenta en firebase se tiene algo como se muestra en la siguiente figura.

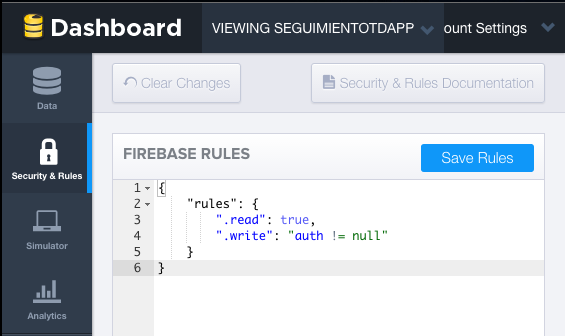


Pantalla inicial firebase.

Paso siguiente es crear la aplicación llenando el formulario con el nombre de la aplicación únicamente y dar click en el botón “CREATE NEW APP” y se tendrá algo como se muestra en la siguiente imagen.

  
Imagen – Aplicación creada en firebase

Se configuran las reglas de seguridad según lo que sugiere la documentación <https://www.firebase.com/docs/security/quickstart.html>, para que la aplicación solo pueda ser escrita por usuarios autorizados, la configuración que se utilizo esta en la siguiente imagen:

  
Imagen Reglas de seguridad firebase

Una vez terminados estos pasos ya se tiene lista la aplicación en firebase para ser conectada con AngularFire.

* Creación de la estructura Angular

Para organizar el proyecto se tomo la decisión de no usar ninguna plantilla, ya que existen varias que ayudan a crear esta estructura de manera automática pero muchas veces generan código innecesario que pone pesada la aplicación, para este proyecto se uso la siguiente estructura.

Nombre Carpeta: app  
Ubicación: /app  
Descripcion: En esta carpeta esta toda la aplicación de Angular.

Nombre Carpeta: bower\_components  
Ubicación: /app/bower\_components  
Descripcion: En esta carpeta están ubicadas cada una de las librerías hechas por terceros y que se administran por medio del manejador de paquetes bower.

Nombre Carpeta: css   
Ubicación: /app/css  
Descripcion: En esta carpeta están ubicadas cada una de las hojas de estilo y temas del proyecto.

Nombre Carpeta: img   
Ubicación: /app/img  
Descripcion: En esta carpeta están ubicadas cada una de las imágenes del proyecto.

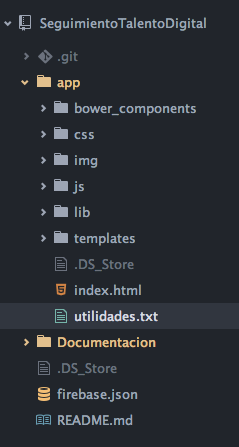
Nombre Carpeta: js   
Ubicación: /app/js  
Descripcion: En esta carpeta están ubicadas cada una de las clases javascript.

Nombre Carpeta: lib  
Ubicación: /app/lib  
Descripcion: En esta carpeta están ubicadas cada una de las librerías javascript.

Nombre Carpeta: templates  
Ubicación: /app/templates  
Descripcion: En esta carpeta están ubicadas cada una las plantillas de html.

Nombre Carpeta: Documentacion  
Ubicación: /Documentacion  
Descripcion: En esta carpeta están ubicada toda la documentación del proyecto.

La estructura completa se muestra en la siguiente imagen:

  
Estructura de carpetas Angular.js

* + 1. Pantallas finales

Una vez terminado el desarrollo y cumpliendo cada uno de los requerimientos solicitados se tuvieron como resultado cada una de las siguientes pantallas:

* Pantalla de login: en esta pantalla cada uno de los usuarios puede iniciar sesión, recuperar su contraseña y los beneficiarios pueden registrarse en el sistema.

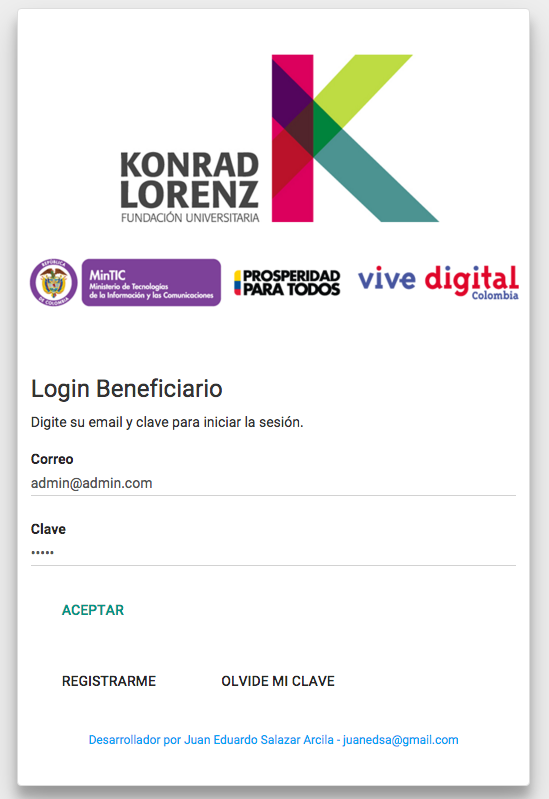


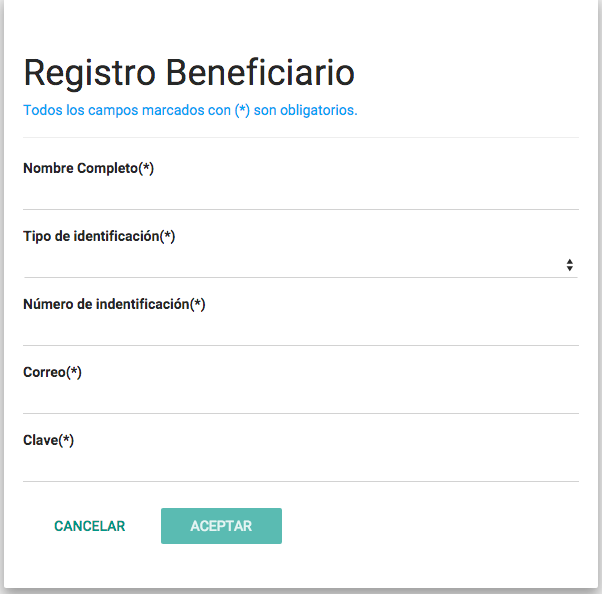
Imagen – Pantalla de login.

* Olvido de clave: Esta pantalla permite a los usuarios recuperar su clave ingresando la cuenta de correo con la que se registraron en el sistema, el sistema genera una nueva clave y se la envía por medio de un correo:



Imagen – Pantalla olvido de clave.

* Registro de un beneficiario: Esta pantalla permite que un beneficiario se registre en el sistema:

  
Imagen – Pantalla Registro de un beneficiario.

* Dashboard del beneficiario: en esta pantalla el beneficiario encuenta los link para actualizar su información.

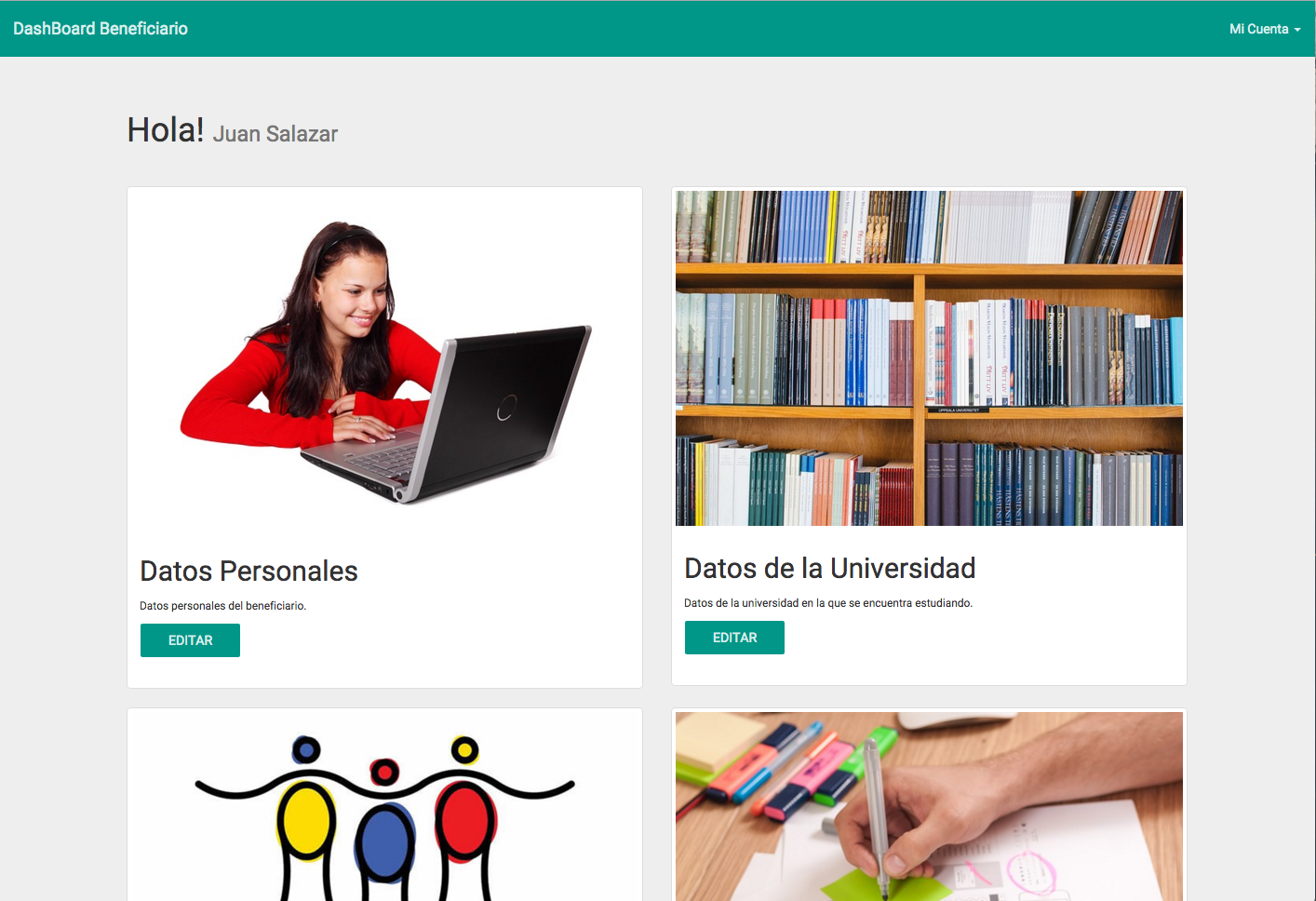
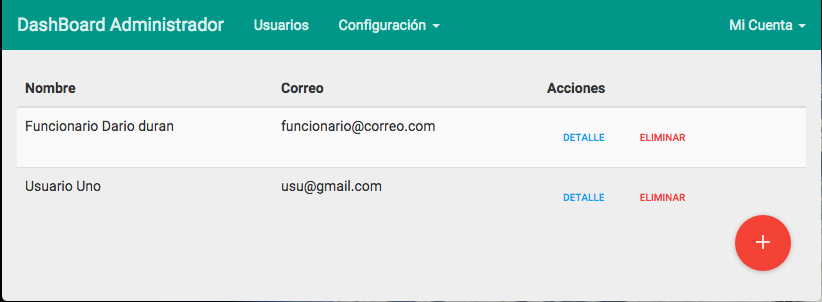


Imagen – Pantalla dashboard del beneficiario.

* Cambio de clave: en esta pantalla los usuarios pueden cambiar su clave.

  
Imagen – Pantalla cambio de clave.

* Pantalla listado de usuarios: en esta pantalla el administrador puede ver el listado de usuarios.

  
Imagen – Pantalla listado de usuarios.

* Pantalla listado de estados: en esta pantalla como en las otras de configuración solo es posible modificar o agregar nuevos.

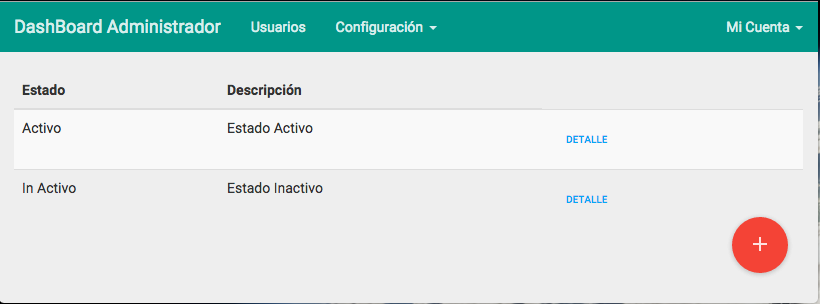


Imagen – Pantalla listado de estados

* Pantalla listado de beneficiarios: en este listado el funcionario puede ver los beneficiarios que esta registrados en este momento.

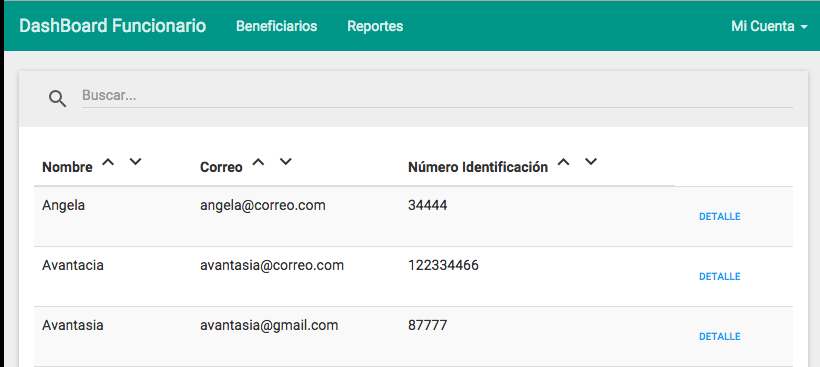
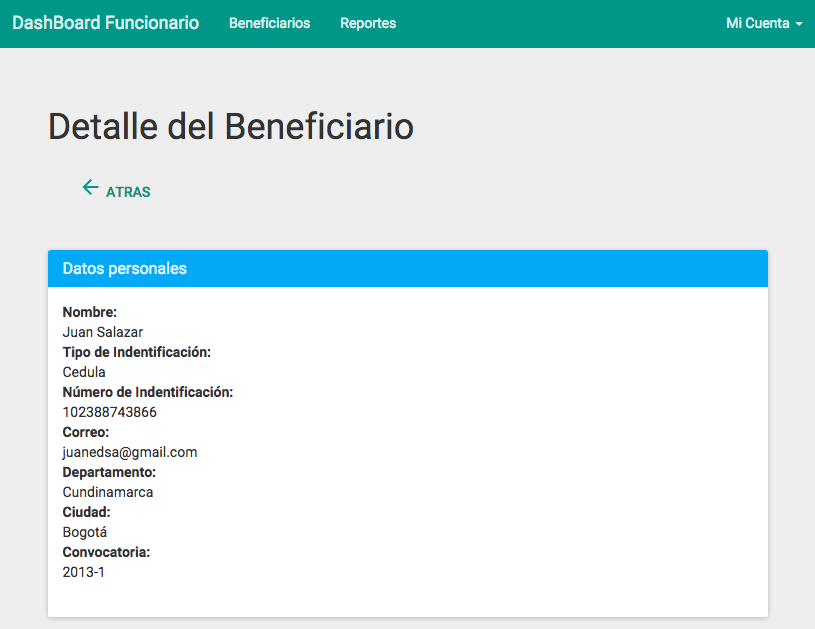


Imagen – listado de beneficiarios.

* Pantalla detalle de un beneficiario: En esta pantalla el funcionario puede ver el detalle de un beneficiario seleccionado en la pantalla anterior.

  
Imagen – Detalle beneficiario.

* Pantalla de reportes: en esta pantalla el funcionario puede ver reportes como Niveles de formación de los beneficiarios, porcentaje de beneficiarios con proyectos, cantidad de beneficiarios por ciudad, cantidad de beneficiarios por convocatoria.

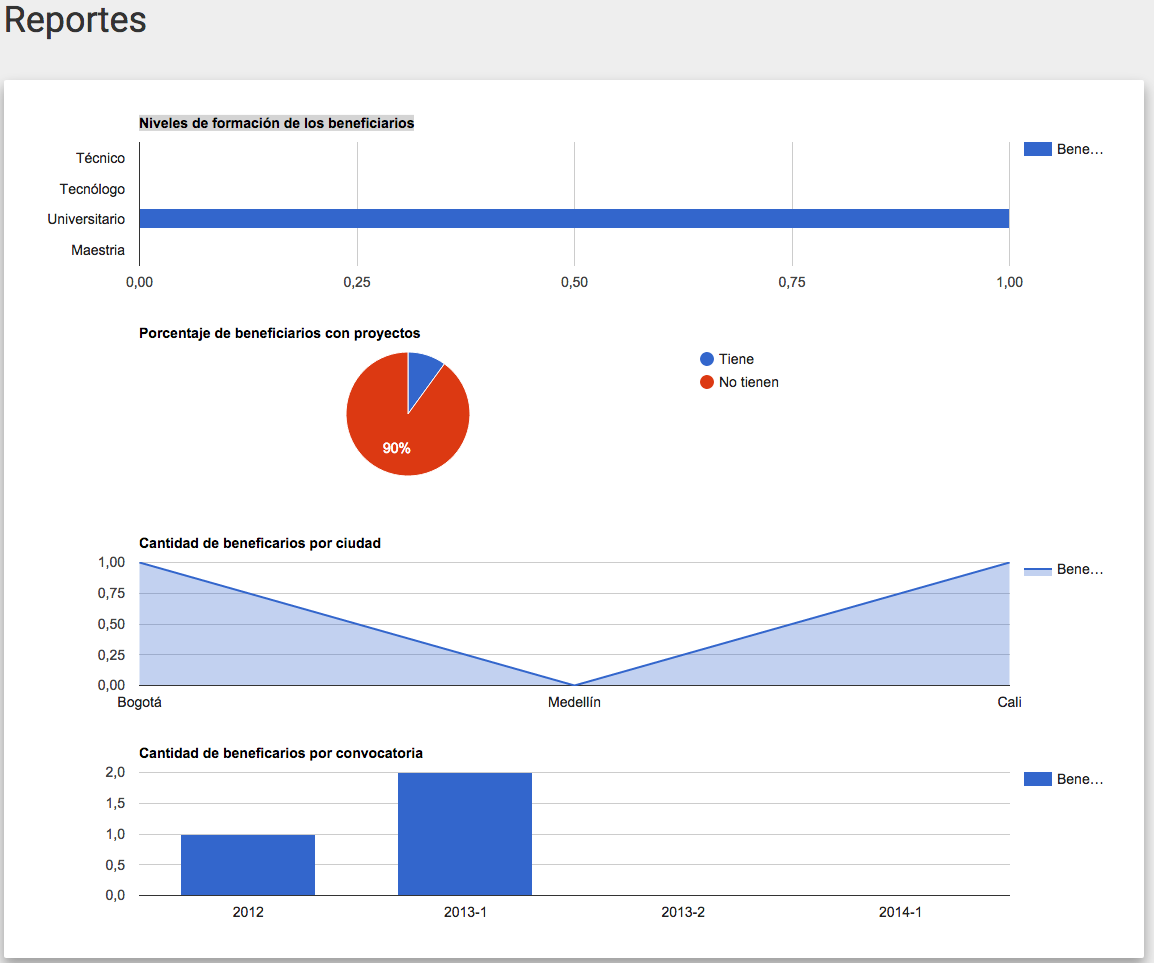


Imagen – Pantalla reportes.

BIBLIOGRAFÍA

* + Conde, jesus, video tutorial php-Mysql Dvd español,2005, España: intercambios virtuales, 1.03 GB.
  + Gilfillan,Ian,MySQL, España: Anaya multimedia, 841 paginas.
  + Aguilar , luis joyanes, programación orientada a objetos, 2da edición, España: magraw-hill, 163 paginas
  + Alarcon, raul, diseño orientado a objetos con uml, España: grupo eidos,117 paginas
  + Batiz, juan de dios, desarrollo orientado a objetos con UML, universidad de castilla- La mancha, 38 paginas
  + Kendall, Kenneth y kendall, julie, análisis y diseño de sistemas de información, 3ra edición, USA: prentice-hall,923 paginas
  + León, Gonzalo serrano, ingeniería de sistemas de software, 1ªedicion, España: isdefe, 211 paginas
  + Pressman, roger, ingeniería de software: un enfoque práctico, 6ª edición, mexico: McGraw-Hill, 900 paginas

INFOGRAFIA

* **Angular JS**  
  Documentación Publica oficial  
  <https://drive.google.com/drive/folders/0BxgtL8yFJbacQmpCc1NMV3d5dnM>  
  Fecha de consulta: 1 Julio de 2015.

**ANEXOS**

* **ANEXO A.**

1. Definición tomada desde <http://talentodigital.mintic.gov.co/625/w3-channel.html> 10 de mayo de 2015 [↑](#footnote-ref-1)