Institución Universitaria Politécnico Grancolombiano.

Facultad de Ingeniería y Ciencias Básicas

Primera Maratón de ingeniería de Software.

Proyecto “NEMO”



Ingeniería de Sistemas.



Integrantes:

Carlos Charris

Diego Meza

Jesús David Ríos Moreno

Octubre de 2015

Contenido

[**1.** **PARTE 1: ORIENTADA AL CLIENTE.** 4](#_Toc432612496)

[**1.1.** **ESTUDIO DEL PROBLEMA Y PROPUESTA DE UNA SOLUCIÓN DE SOFTWARE.** 4](#_Toc432612497)

[**1.2.** **DESCRIPCIÓN DEL PROBLEMA.** 4](#_Toc432612498)

[**1.3.** **DESCRIPCIÓN DE LA SOLUCIÓN DE SOFTWARE.** 4](#_Toc432612499)

[**1.4.** **TECNOLOGÍAS UTILIZADAS.** 4](#_Toc432612500)

[**1.4.1.** **MONGODB.** 4](#_Toc432612501)

[**1.4.2.** **Express** 5](#_Toc432612502)

[**1.4.3.** **Angular JS** 5](#_Toc432612503)

[**1.4.4.** **NODEJS.** 6](#_Toc432612504)

[**1.4.5.** **GruntJS.** 6](#_Toc432612505)

[**1.4.6.** **KarmaJs.** 7](#_Toc432612506)

[**1.4.7.** **HotNodeJs.** 7](#_Toc432612507)

[**1.4.8.** **Materialize.** 7](#_Toc432612508)

[**1.4.9.** **Esquema de Funcionamiento del MEAN.** 8](#_Toc432612509)

[**DESCRIPCIÓN DETALLADA DEL SOFTWARE** 8](#_Toc432612510)

[**1.4.10.** **MOCAPS** 8](#_Toc432612511)

[**1.4.11.** **HISTORIA DE USUARIO** 8](#_Toc432612512)

[**1.4.12.** **DIAGRAMA UML** 8](#_Toc432612513)

[**1.** **FORMULACIÓN DEL PROBLEMA.** 10](#_Toc432612514)

[**2.** **ORIENTADA AL INGENIERO.** 10](#_Toc432612515)

[**3.** **DIAGRAMA DE FLUJO DE DATOS.** 10](#_Toc432612516)

[**4.** **ARQUITECTURA.** 10](#_Toc432612517)

[**5.** **DIAGRAMA DE INTERFAZ.** 10](#_Toc432612518)

[**6.** **CONCLUSIONES:** 11](#_Toc432612519)

[Bibliografía 12](#_Toc432612520)

Tabla De Figuras:

[Figura 1 Diagrama de clases. 4](#_Toc432605795)

[Figura 1 Diagrama de secuencia. 5](#_Toc432605796)

**PARTE 1: ORIENTADA AL CLIENTE.**

## **ESTUDIO DEL PROBLEMA Y PROPUESTA DE UNA SOLUCIÓN DE SOFTWARE.**

El planteamiento del proyecto consta en la construcción, diseño y realización de un Aplicación en la cual se dé solución al problema de objetos perdidos en el campus Universitario.

## **DESCRIPCIÓN DEL PROBLEMA.**

La pérdida de elementos personales en la Universidad Politécnico Grancolombiano es un problema que no se ha podido atacar de una manera exitosa. Además de la perdida de los objetos también se dificulta la forma de reportar y devolver objetos que el estudiante, profesor o empleado se ha encontrado.

## **DESCRIPCIÓN DE LA SOLUCIÓN DE SOFTWARE.**

Para la solución del problema en mención, proponemos la siguiente solución de software apoyándonos en los siguientes aspectos.

Descripción de las tecnologías a utilizar.

Descripción detallada en diagramas de usuario, diagramas UML, historias de usuarios y casos de uso.

## **TECNOLOGÍAS UTILIZADAS.**

En la aplicación “NEMO” se trabajó un stack MEAN, el cual está compuesto por tecnologías como MongoDB, ExpressJS, AngularJS y Node.JS. Con estas tecnologías se permite el uso de JavaScript en todos los niveles de la aplicación.

## **MONGODB.**



Sistema de base de datos NoSQL (Sistemas de bases de datos amigables para el desarrollo), orientada al uso de documentos y no en registros, estos documentos los almacena en BSON (formato de intercambio de datos). Este tipo de bases de datos no necesita seguir el esquema predefinido. MongoDB puede ejecutar los distintos comandos que utiliza JavaScript tales como variables, funciones entre otras. Vamos a utilizar mongooseJS, el package de Node que sirve para trabajar con MongoDB.

## **Express**



Express es una plataforma ligera para construir aplicaciones web usando NodeJS. Nos ayuda a organizar las aplicaciones web en el lado del servidor. En la web de ExpressJS, lo describen como “un framework de desarrollo de aplicaciones web minimalista y flexible para Node.js“.

Express esconde muchas funcionalidades internas de Node, lo que te permite sumergirte en el código de tu aplicación y conseguir tus objetivos de forma muy rápida. Es fácil de aprender y te deja cierta flexibilidad con su estructura.

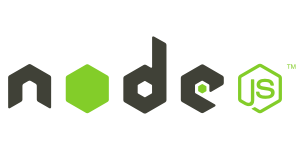
## **Angular JS**



Angular, creado por Google, es un framework de JavaScript MVC para el desarrollo front end de forma rápida y dinámica.

Permite extender el vocabulario de tu HTML con directivas y atributos para crear componentes dinámicos. Si alguna vez has hecho una página web dinámica sin Angula te habrás dado cuenta de ciertas complicaciones frecuentes, como el data binding, validación de formulario, manejador de eventos con DOM (Document Object Model) y otras muchas. Angular presenta una solución “todo-en-uno” a esos problemas.

## **NODEJS.**



Node.js está basado en el motor V8 de Javascript de Google Chrome y juega su papel en el lado del servidor dentro de nuestra aplicación con MEAN. Este motor está diseñado para correr en un navegador y ejecutar el código de Javascript de una forma extremadamente rápida. ¿Qué significa esto? En un stack LAMP tienes tu servidor web (Apache, Nginx, etc) corriendo con un lenguaje de script del lado del servidor (PHP, Perl, Python) para crear un sitio web dinámico. El código del lado del servidor se usa para crear el entorno de la aplicación extrayendo datos de la base de datos (MYSQL) y después es interpretado por el servidor para producir la página web. Cuando se hace la petición de una nueva conexión, Apache crea un nuevo thread (hilo) o proceso para controlar esta petición, haciendo que sea multitarea (multihilo). A menudo tendrás un número de procesos hijos inactivos en espera para ser asignados a una nueva petición. Si configuras el servidor para tener a solo a 50 procesos hijos inactivos y entran 100 peticiones, algunos usuarios pueden pasar por un tiempo de espera de conexión hasta que varios de esos procesos se liberen.

## **GruntJS.**



Grunt.js es una librería JavaScript que nos permite configurar tareas automáticas y así ahorrarnos tiempo en nuestro desarrollo y despliegue de aplicaciones webs.

Con un simple fichero JS que llamaremos Gruntfile, indicamos las tareas que queremos automatizar con un simple comando y las escribimos en él en formato JSON.

## **KarmaJs.**



Karma se encarga de ejecutar los test de Javascript según se vayan construyendo, de tal forma que ante cualquier fallo el desarrollador se dará cuenta de inmediato.

## **HotNodeJs.**

HotNode se encarga de reiniciar el servidor cada vez que se hagan cambios en los documentos, esto con el fin de no estar bajando y subiendo el servicio manualmente.

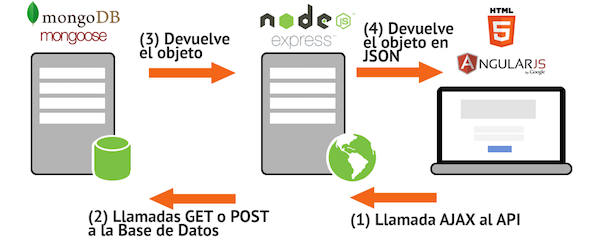
## **Materialize.**



Es un lenguaje de diseño que combina los principios clásicos del diseño combinado con las nuevas tecnologías y diseños innovadores.

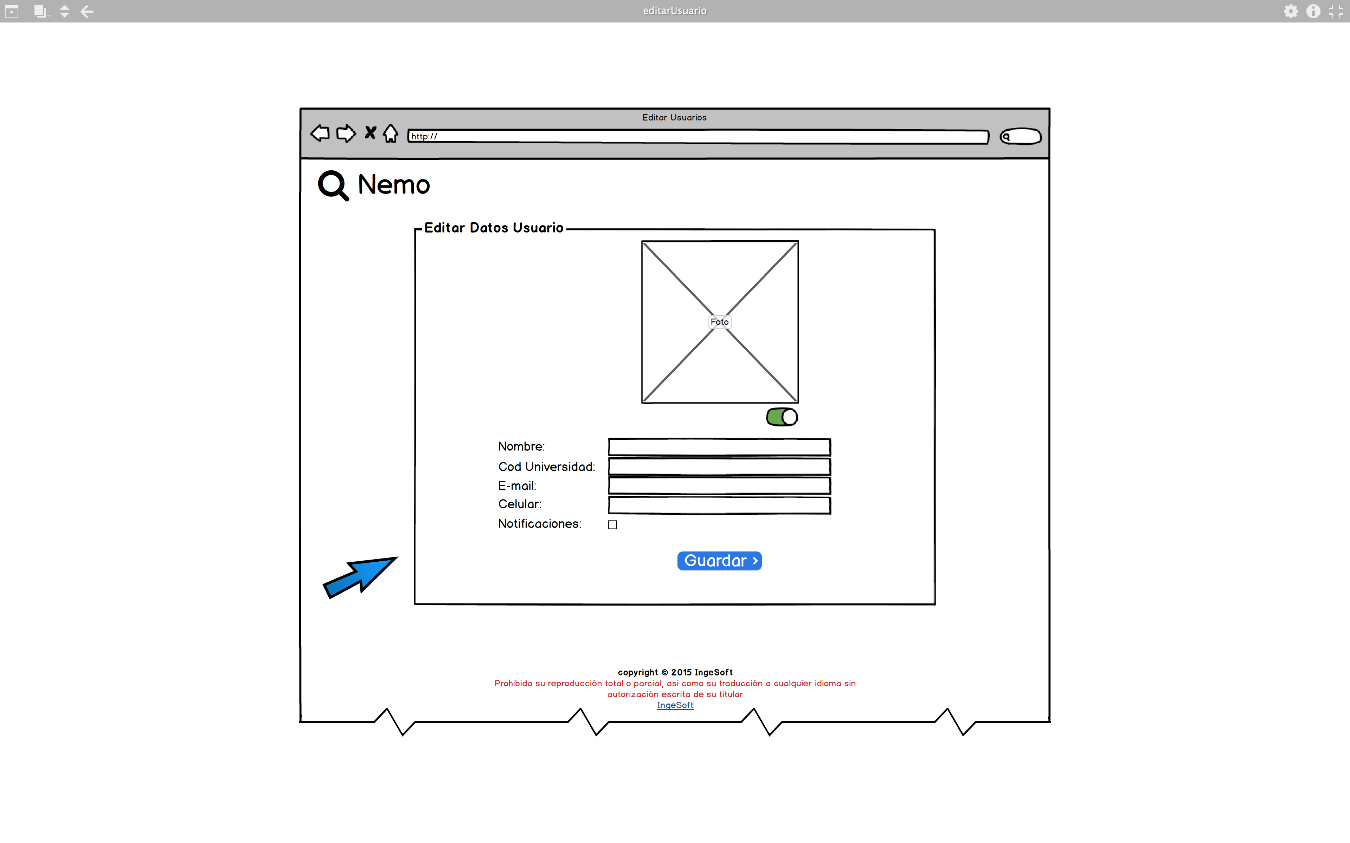
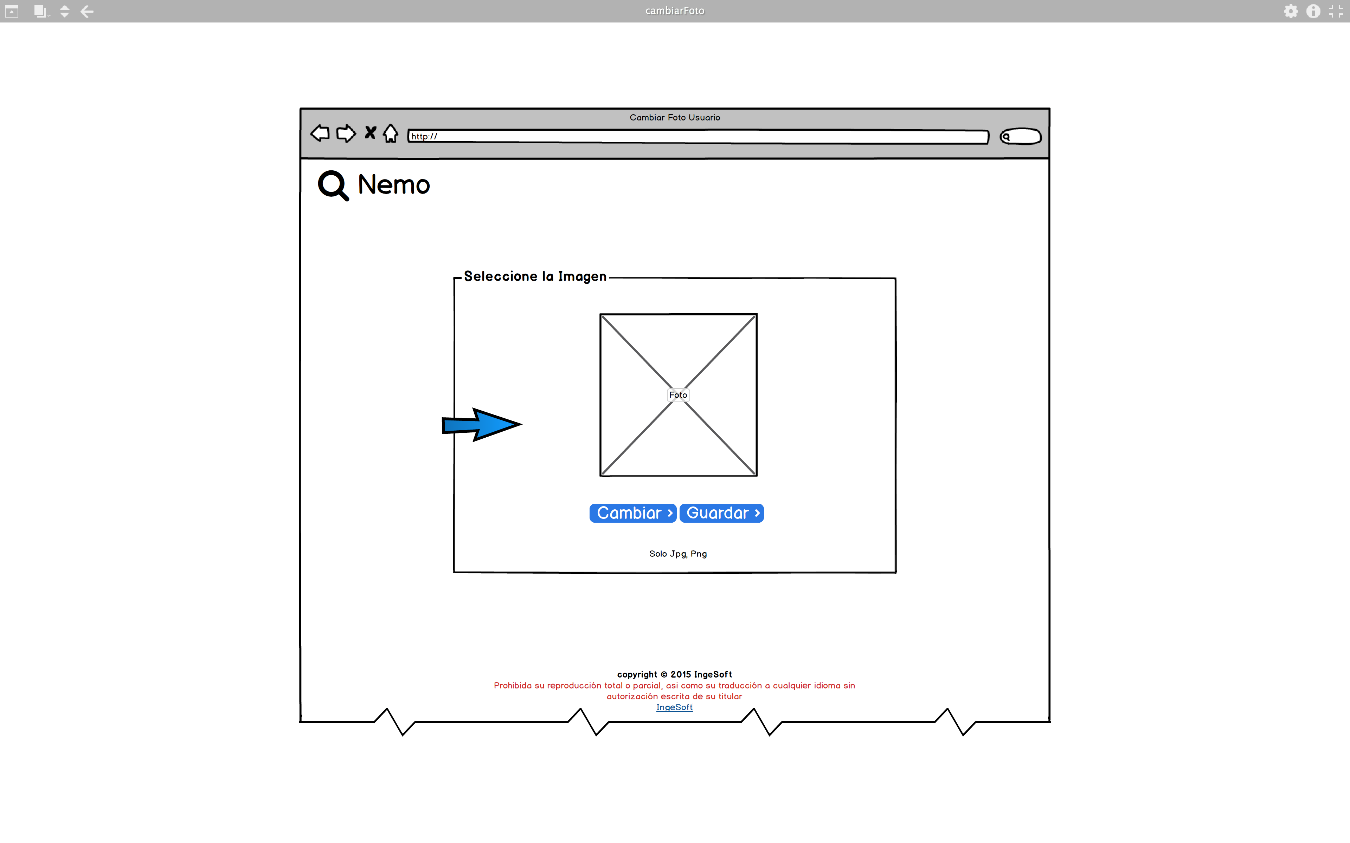
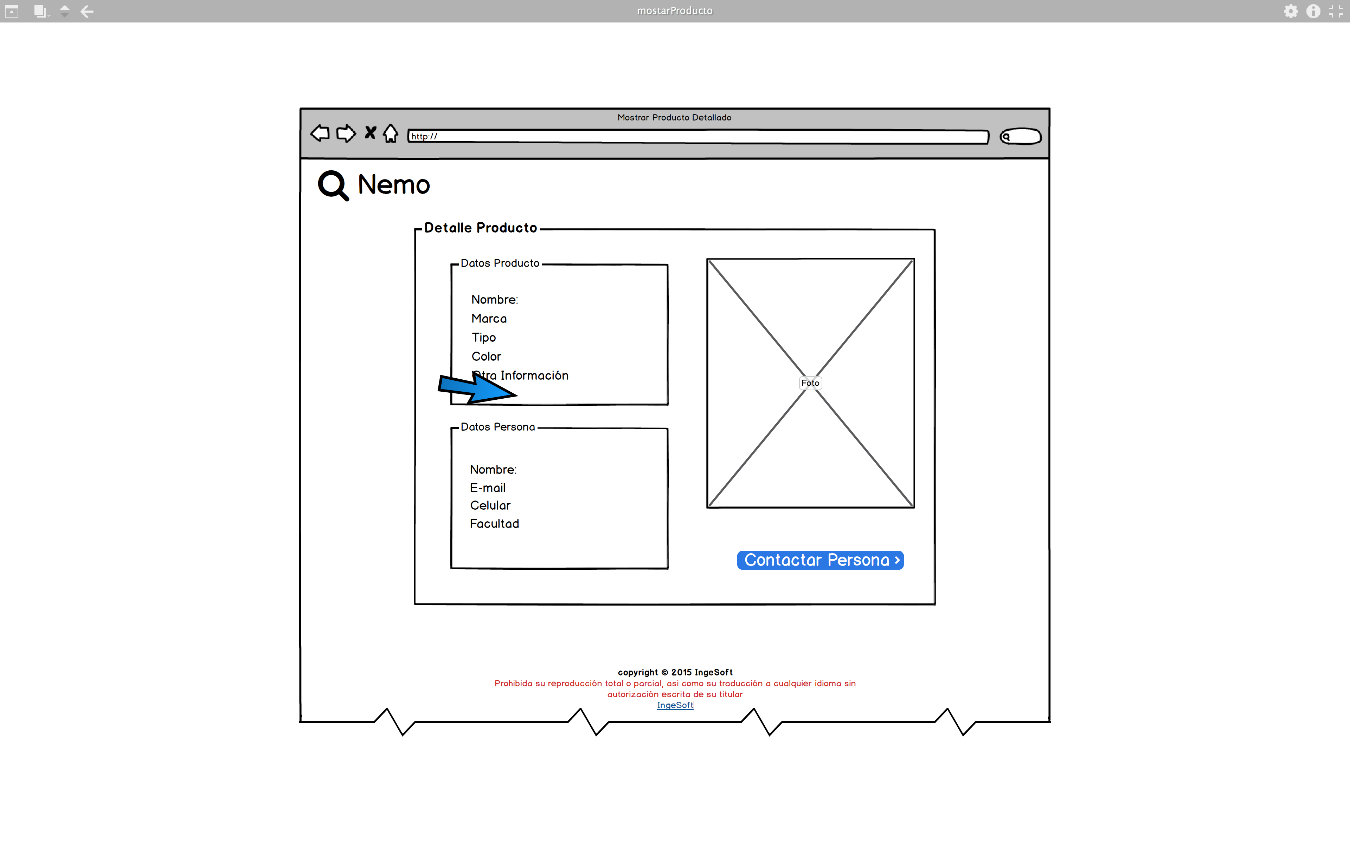
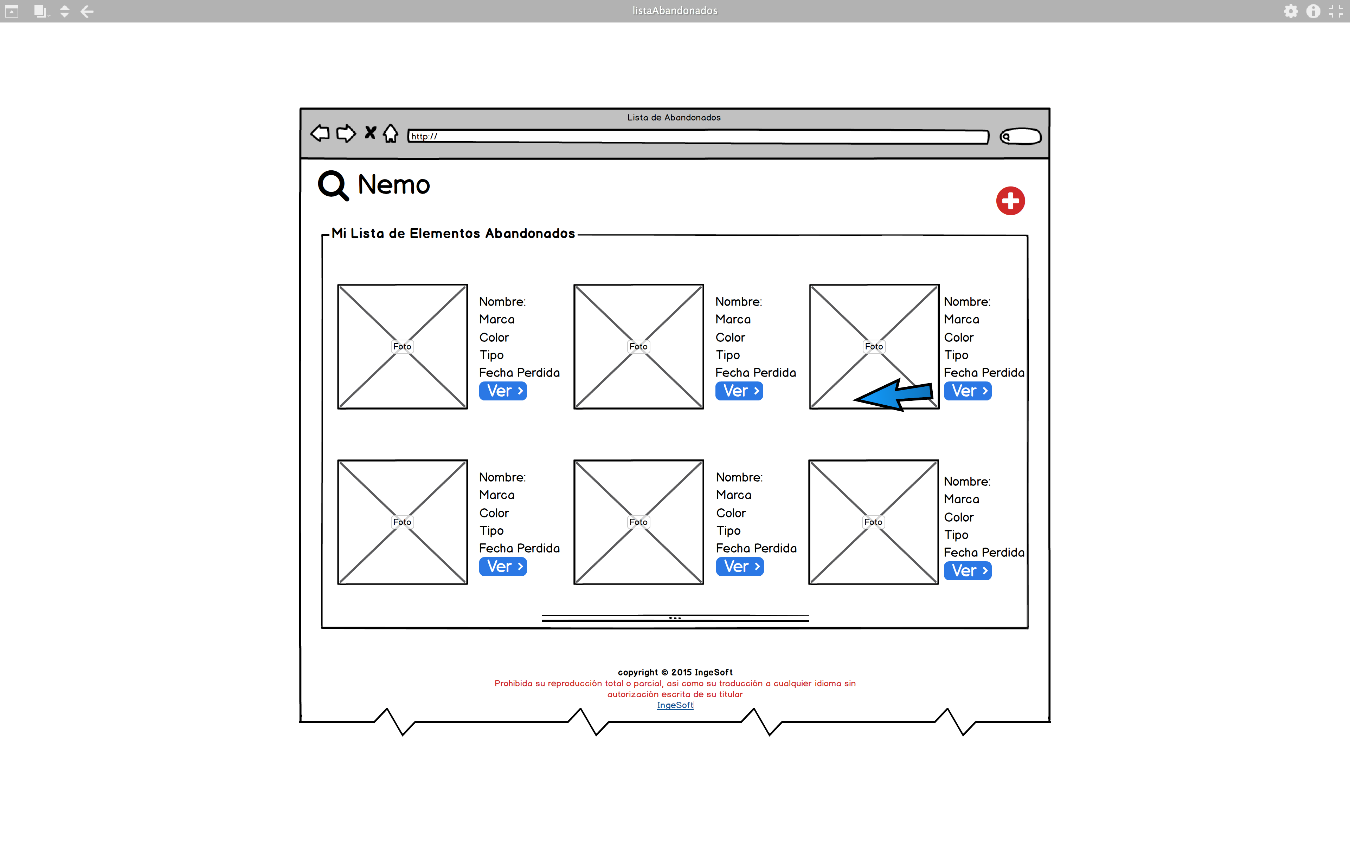
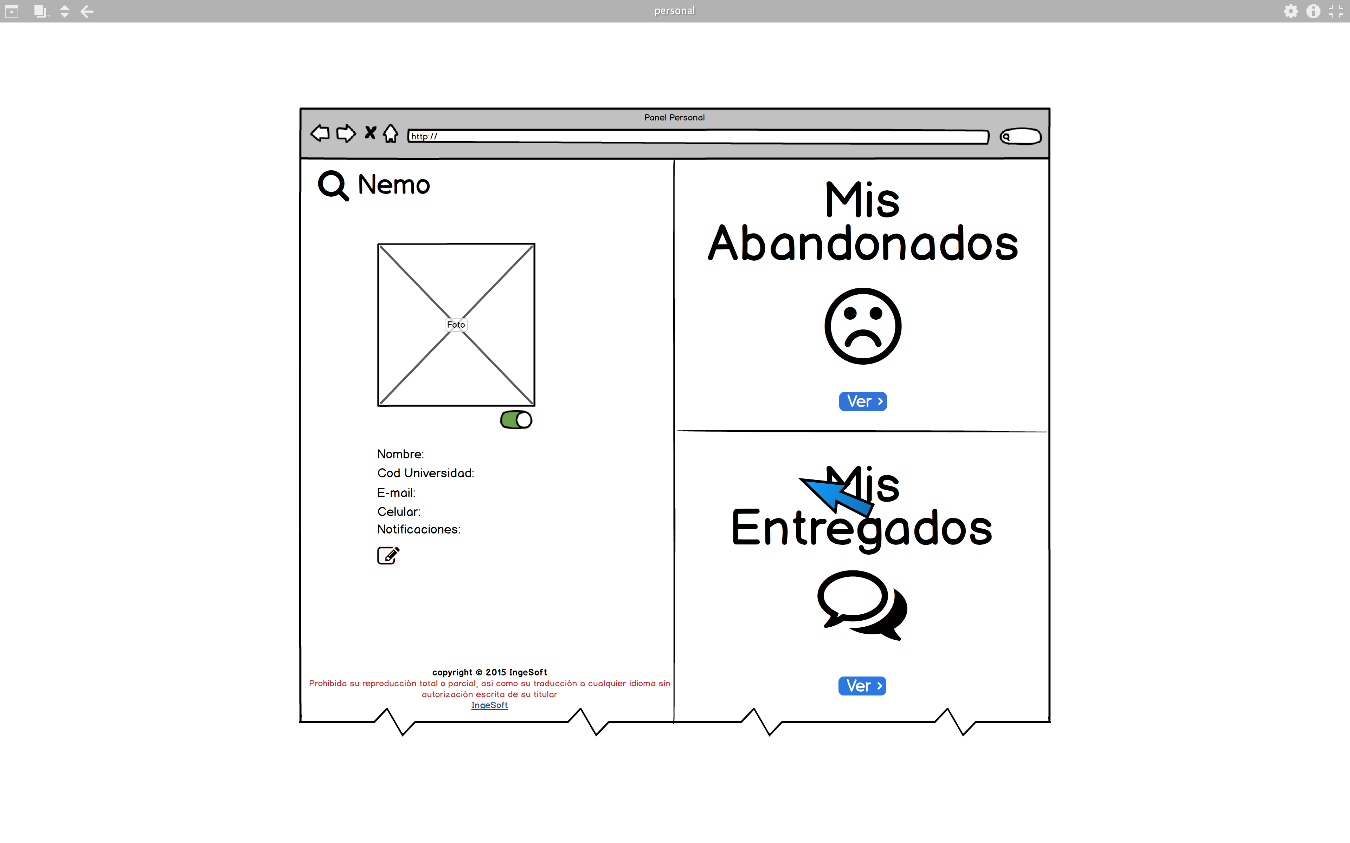
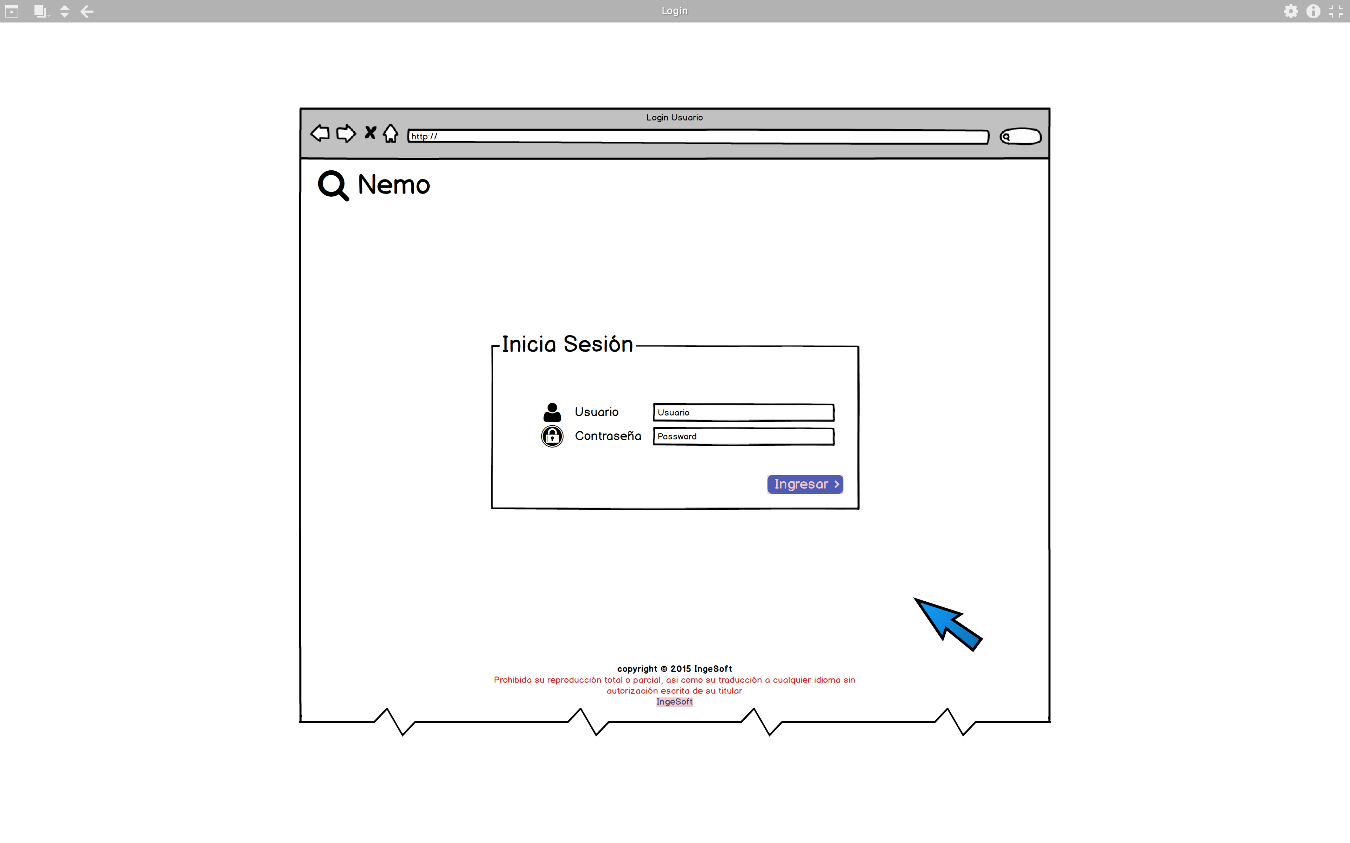
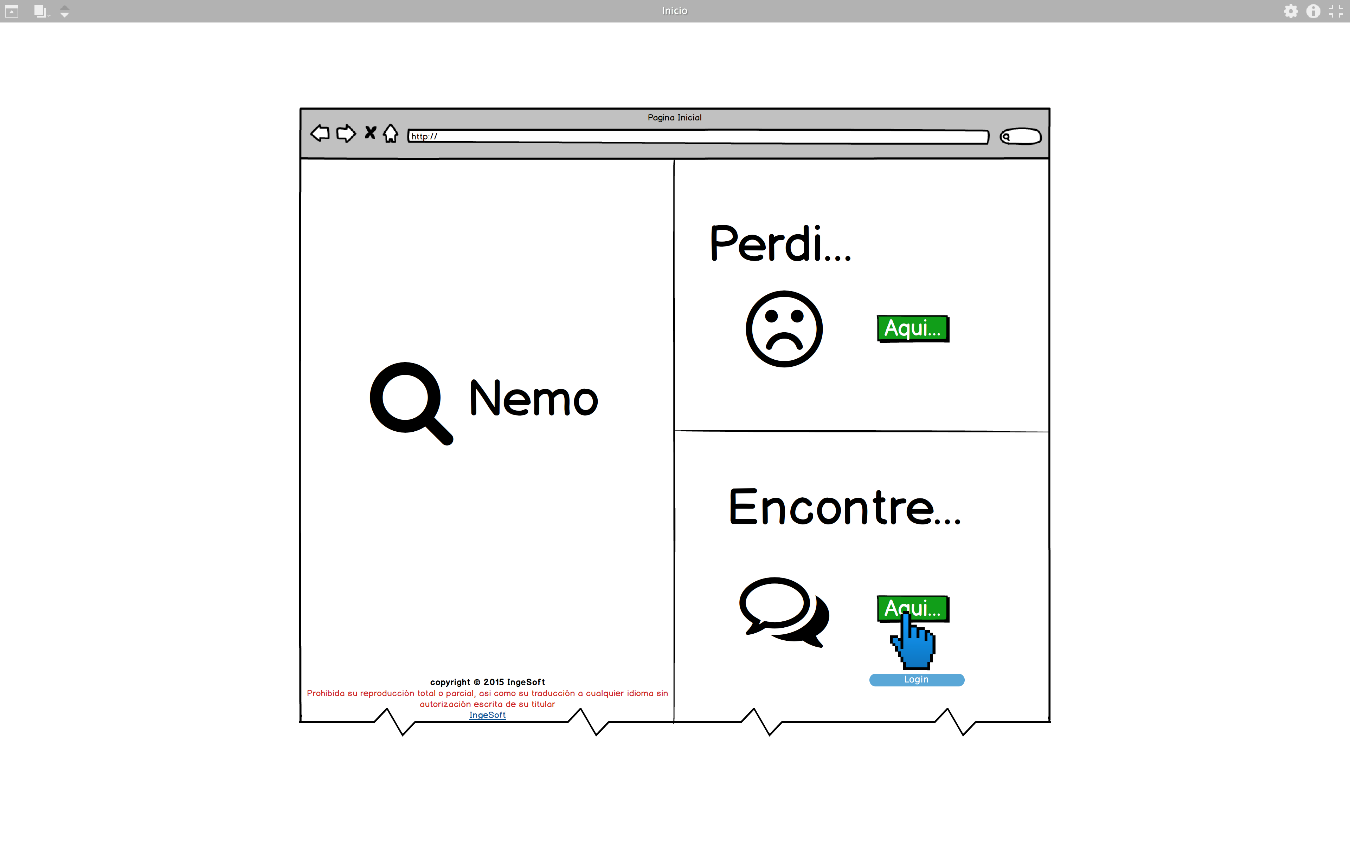
Material Design es creado y mantenido por Google y actualmente está siendo aplicado a sus productos tanto en la Web como en su sistema operativo móvil Android, específicamente en su versión 5.0 lollipop.

## **Esquema de Funcionamiento del MEAN.**



## **DESCRIPCIÓN DETALLADA DEL SOFTWARE**

## **MOCKUP**



## **HISTORIA DE USUARIO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Tema** | **Como Un…** | **Necesito** | **Así podré** | **Notas** | **Prioridad** | **Status** |
| 1 | cuenta | Usuario | Crear una cuenta | Ingresar a la plataforma |  | Alta | 85% |
| 8 | cuenta | Usuario | Iniciar sesión | Ingresar a la aplicación |  | Alta | 20% |
| 4 | cuenta | Usuario | Editar mis Datos | Modificar o Actualizar mis Datos |  | Alta | 85% |
| 2 | Objetos | Usuario | Crear un Objeto Perdido | Informar Objetos encontrados |  | Alta | 85% |
| 7 | Objetos | Usuario | Editar un Objeto Perdido | Modificar o Actualizar Objetos creados |  | Alta | 85% |
| 11 | cuenta | Usuario | Validar la Creación de cuentas y/o Usuarios | Asegurarme que no hay duplicidad de Usuarios |  | Alta | 20% |
| 12 | Objetos | Usuario | Validar la Creación de Objetos | Asegurarme que el Usuario esta Logeado en la aplicación |  | Alta | 20% |
| 3 | Objetos | Usuario | Lista de productos | Consultar todos los Objetos Perdidos |  | Media | 65% |
| 9 | Objetos | Usuario | Eliminar un Objeto Perdido | Borrar Objetos que se crearon por error |  | Media | 85% |
| 6 | correo | Usuario | Contactar otro usuario | Realizar acercamiento para entregar o recibir los Objetos perdidos |  | Media | 10% |
| 10 | correo | Usuario | Notificaciones | Recibir correo de las personas que desean contactarme |  | Media | 10% |
| 5 | cuenta | Usuario | Editar imagen Perfil | Modificar o Actualizar la foto de mi perfil |  | Media | 85% |
| 13 | cuenta | Usuario | Bloquear usuarios tramposos | Mejorar el ambiente de la comunidad |  | Media | 5% |

## **DIAGRAMA UML**

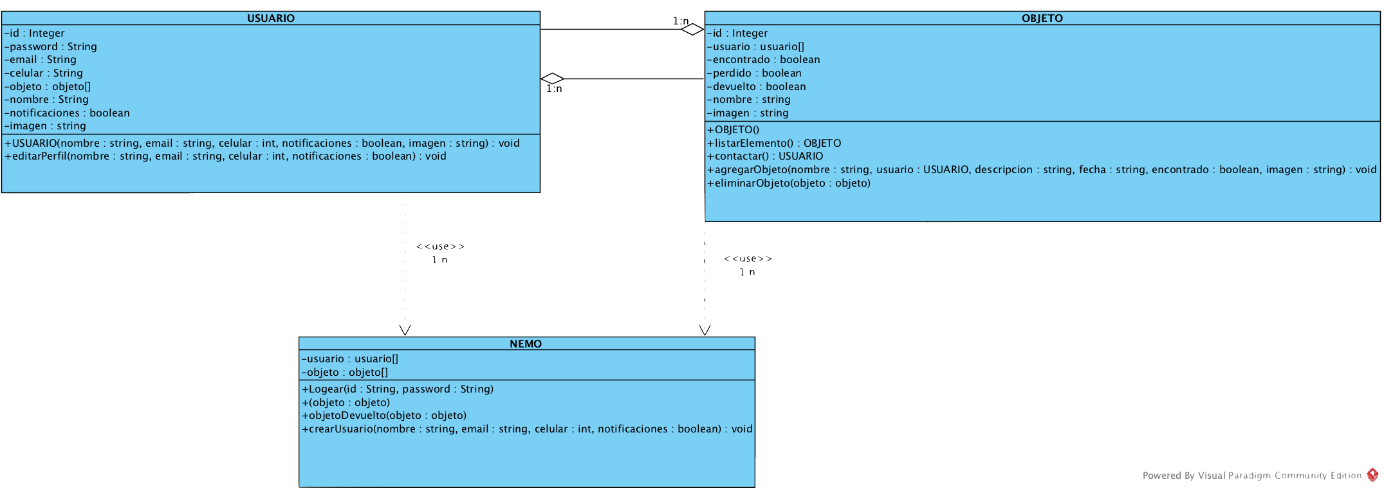


Figura 1 Diagrama de clases.

## **DIAGRAMA DE SECUENCIA**

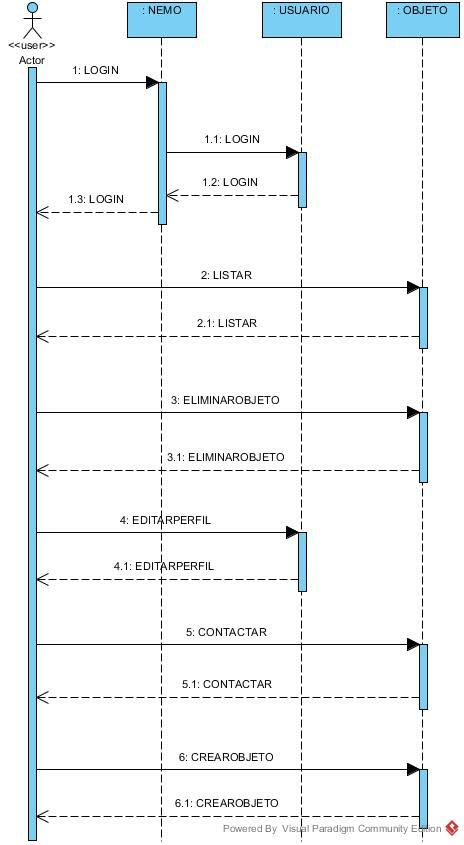
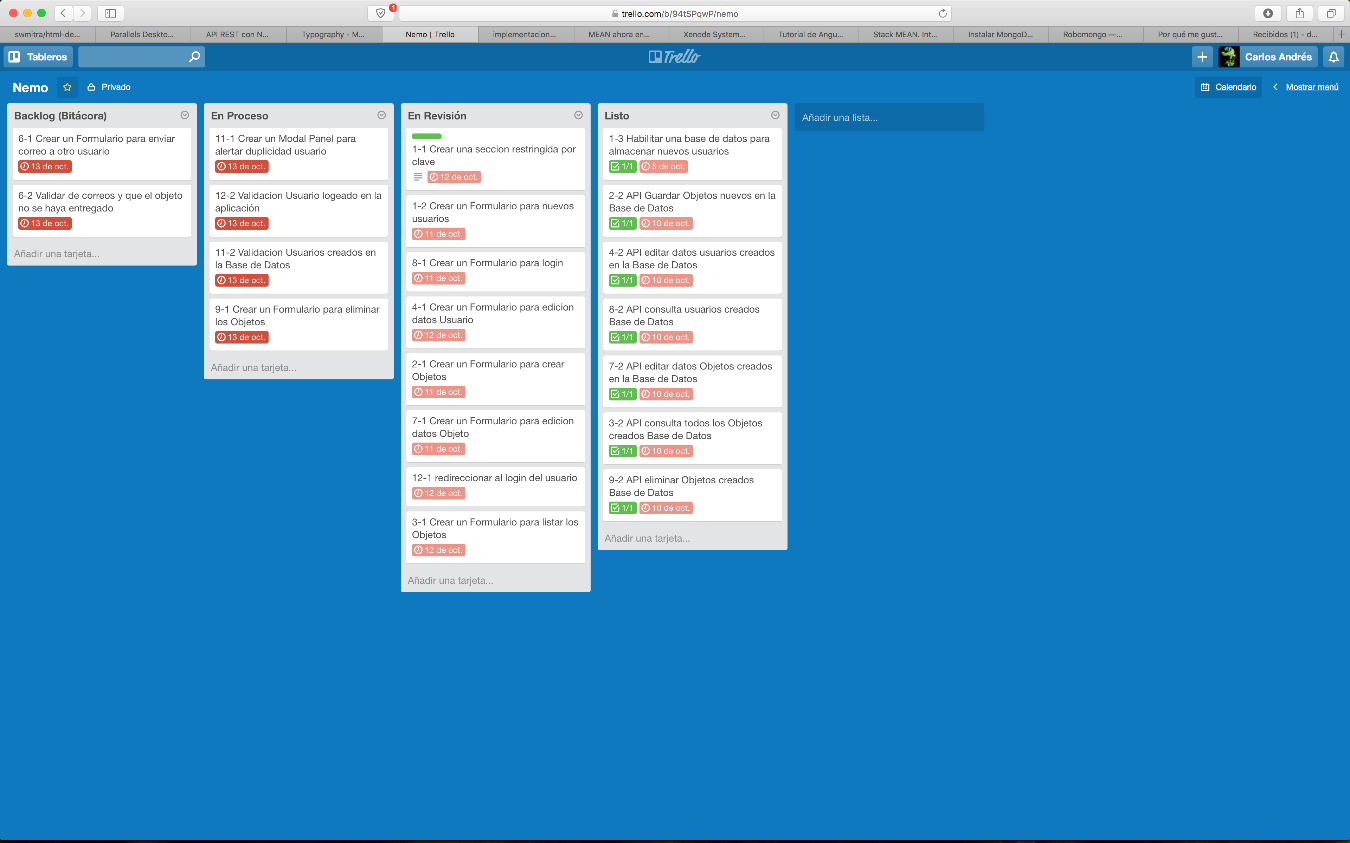
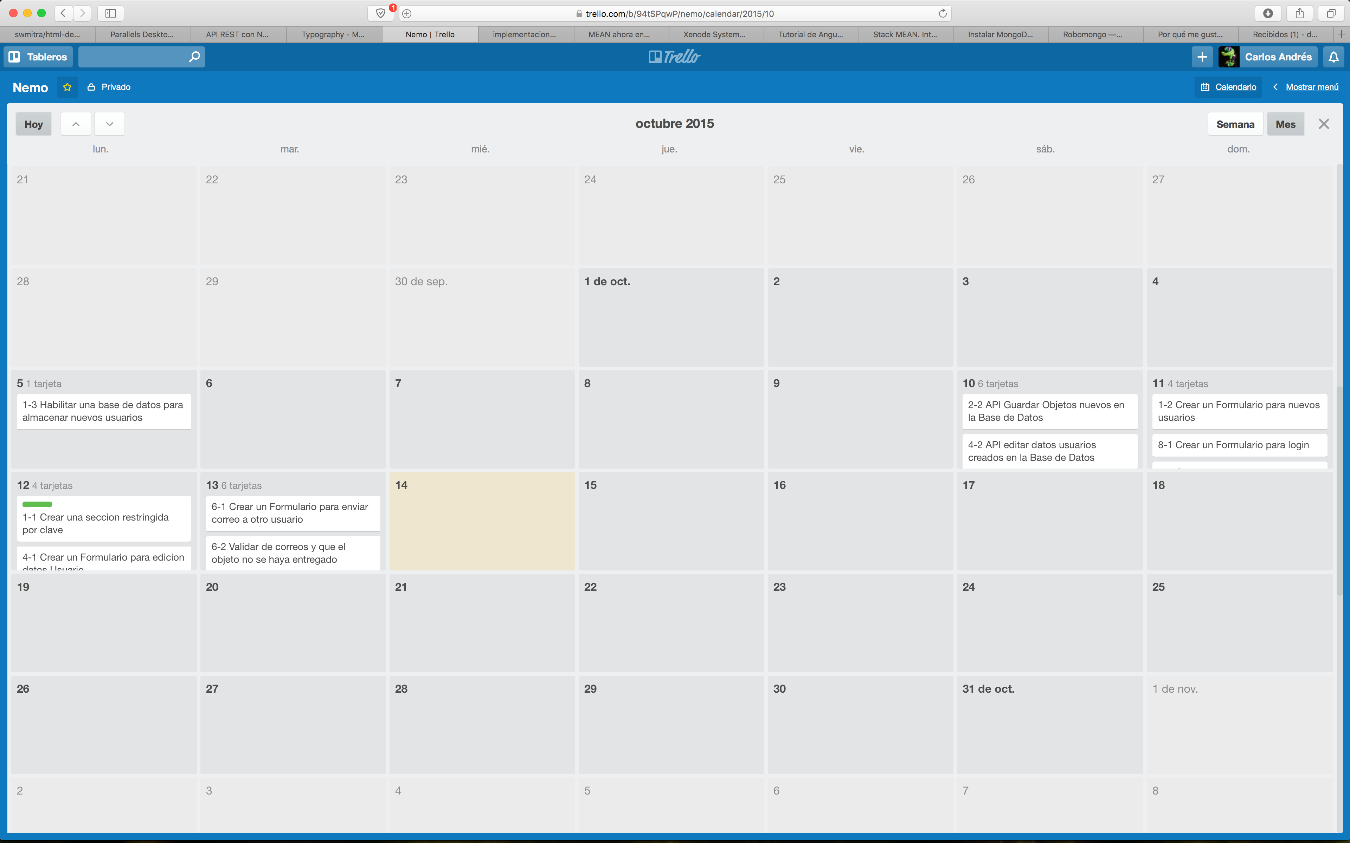


Figura 2 Diagrama de secuencia.

## **SPRINT 1**



## **Burndown Chart**

# 