

**PROTOTIPO DE CREACIÓN DE FORMULARIOS PARA SISTEMATIZAR LA  
CAPTURA Y ALMACENAMIENTO DE DATOS DE FORMA EFICAZ Y  
CONFIABLE POR MEDIO DE UN ENTORNO WEB**

**JORGE ANTONIO GONZÁLEZ MORENO**

**OSCAR FERNANDO LARROTA JAIMES**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENERIAS FISICO-MECANICAS  
ESCUELA DE INGENERIA DE SISTEMAS E INFORMATICA  
BUCARAMANAGA**

**2013**

**PROTOTIPO DE CREACIÓN DE FORMULARIOS PARA SISTEMATIZAR LA  
CAPTURA Y ALMACENAMIENTO DE DATOS DE FORMA EFICAZ Y  
CONFIABLE POR MEDIO DE UN ENTORNO WEB**

**JORGE ANTONIO GONZÁLEZ MORENO**

**OSCAR FERNANDO LARROTA JAIMES**

**Proyecto para optar al título de ingeniero de Sistemas**

**Director**

**CARLOS FELIPE REYES CONTRERAS**

**Ingeniero de sistemas**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER**

**FACULTAD DE INGENERIAS FISICO-MECANICAS**

**ESCUELA DE INGENERIA DE SISTEMAS E INFORMATICA**

**BUCARAMANAGA**

**2013**

## DEDICATORIA

“A Dios por estar conmigo en cada momento y permitirme alcanzar un logro más en mi vida, bendiciéndome con amor, salud y su infinita bondad.

A mis padres que me dieron la vida y han estado conmigo en todo momento. Gracias por su amor y apoyo incondicional en todos los momentos de mi vida y quiero decirles que los amo.

A mis hermanos por ser un apoyo en mi vida y saber que siempre puedo contar con ellos, los quiero mucho.

A Danielito que llego para alegrar nuestras vidas.”

Jorge A. González Moreno.

“Primero que todo dedico este proyecto a Dios quien me ha fortalecido e iluminado para alcanzar esta meta tan importante.

A mis padres, mis hermanos y mi familia que siempre han sido un apoyo incondicional durante mi vida y siempre se han esforzado por darme lo mejor.

A mi hijo Oscar Alejandro quien me ha servido de inspiración para mantenerme enfocado en lo importante de la vida y por quien he luchado para lograr este sueño.

Por último a mis amigos que hicieron de este viaje más llevadero compartiendo conmigo momentos muy importantes.”

Oscar Fernando Larrota Jaimes.

## **AGRADECIMIENTOS**

A la universidad Industrial de Santander.

A la Escuela de Ingeniería de Sistemas e Informática.

Al profesor CARLOS FELIPE REYES CONTRERAS, director del proyecto, por la confianza brindada, por los consejos, ideas, conocimientos y paciencia que compartido con nosotros durante todo el proyecto.

A Nuestros docentes quienes se esforzaron por darnos una formación integral.

## TABLA DE CONTENIDO

1. INTRODUCCIÓN .....	15
2. JUSTIFICACIÓN .....	16
3. OBJETIVOS .....	17
3.1 OBJETIVO GENERAL .....	17
3.2 OBJETIVOS ESPECIFICOS.....	17
4. MARCO TEORICO O DE REFERENCIA .....	18
4.1 HTML.....	18
4.2 PHP.....	20
4.3 JAVASCRIPT .....	21
4.4 JQUERY.....	23
4.5 BASES DE DATOS.....	24
4.5.1 BASES DE DATOS ESTÁTICAS.....	25
4.5.2 BASES DE DATOS DINÁMICAS.....	25
4.5.3 SISTEMA DE GESTIÓN DE BASES DE DATOS DISTRIBUIDA (SGBD).....	25
4.6 MYSQL.....	26
4.7 HOJAS DE ESTILO EN CASCADA.....	26
4.8 SUBLIME TEXT 2 .....	27
4.9 TEXTO PLANO .....	27
5. METODOLOGIA .....	28
5.1 FASE ESPECIFICACIÓN INICIAL.....	30
5.2 FASE DESARROLLO DEL PRODUCTO .....	30
5.3 FASE IMPLEMENTACIÓN USO Y EVALUACIÓN.....	31
5.4 FASE RE-ESPECIFICACIÓN .....	31
6. DESARROLLO DEL PROTOTIPO WEB.....	32
6.1 DIAGRAMAS UML .....	32
6.1.1 DIAGRAMA DE CONTEXTO:.....	32
6.1.2 DIAGRAMA DE CASOS DE USO: .....	33
6.1.3 DIAGRAMA DE SECUENCIA.....	43
6.1.4 DIAGRAMA DE CLASES:.....	48
6.1.5 DIAGRAMA DE ACTIVIDADES:.....	52
6.2 ESTRUCTURA DE DATOS .....	55
7. INTERFACES DEL SISTEMA .....	56
8. CONCLUSIONES .....	64

9. RECOMENDACIONES ..... 65  
BIBLIOGRAFIA ..... 66

## LISTA DE TABLAS

Tabla 1. Usuario directo .....	35
Tabla 2. Usuario indirecto .....	35
Tabla 3. Caso de uso Registrar Usuario .....	36
Tabla 4. Caso de uso Ingresar al sistema.....	38
Tabla 5. Caso de uso Crear formulario .....	39
Tabla 6. Caso de uso Llenar formulario .....	40
Tabla 7. Caso de uso Ver información.....	41
Tabla 8. Diagrama de actividades.....	52

## LISTA DE FIGURAS

Figura 1. Modelo de desarrollo evolutivo .....	29
Figura 2. Diagrama de contexto.....	33
Figura 3. Diagrama de casos de uso. ....	34
Figura 4. Diagrama Secuencia: Registrar usuario .....	43
Figura 5. Diagrama Secuencia: Ingresar al sistema .....	44
Figura 6. Diagrama Secuencia: Crear Formulario.....	45
Figura 7. Diagrama Secuencia: Llenar Formulario.....	46
Figura 8. Diagrama Secuencia: Ver Información .....	47
Figura 9. Diagrama Relacional.....	50
Figura 10. Diagrama Registrar usuario .....	53
Figura 11. Diagrama crear formulario .....	53
Figura 12. Diagrama obtener información.....	54
Figura 13. Diagrama diligenciar formulario .....	54
Figura 14. Ventana de inicio .....	56
Figura 15. Ventana de registro.....	57
Figura 16. Ventana de ingreso.....	58
Figura 17. Ventana Contacto .....	59
Figura 18. Ventana crear formulario .....	60
Figura 19. Visualizar formulario .....	61
Figura 20. Administrador de formularios. ....	62
Figura 21. Visualizar información.....	63

## RESUMEN

### TITULO

PROTOTIPO DE CREACIÓN DE FORMULARIOS PARA SISTEMATIZAR LA CAPTURA Y ALMACENAMIENTO DE DATOS DE FORMA EFICAZ Y CONFIABLE POR MEDIO DE UN ENTORNO WEB \*.

### AUTORES

Jorge Antonio González Moreno

Oscar Fernando Larrota Jaimes \*\*

### PALABRAS CLAVES

Prototipo web, generador, archivo plano, captura de datos, formulario.

### DESCRIPCION

Este proyecto surgió porque la cobertura geográfica de una empresa ha pasado de ser una instalación física local, a diversos puntos descentralizados y aislados desde los cuáles se está generando información continuamente. Casos como la captura de pedidos, control de tiempos en campamentos de obra, control de seguimiento y evolución de obras; entre otros, son situaciones en las que en la actualidad aún se maneja la captura de datos en formatos de papel para luego ser transcritos.

Analizando esta problemática se presenta el desarrollo de un prototipo web (FormUIS) con el propósito de generar formularios, donde el usuario determina la estructura del formulario, como el número de campos, el tipo de campo y algunas características como el tipo de letra y el color del formulario. Posteriormente permite acceder a los formularios mediante un enlace directo con el fin que los usuarios den a conocerlos para su posterior diligenciamiento y así poder recolectar la información que desean.

Cada vez que diligencian un formulario el sistema guarda esa información en una base de datos, la cual solamente el usuario creador del formulario puede acceder a ella, así brindándole mayor seguridad y confiabilidad. También permite descargar dicha información en un archivo plano para que los usuarios luego puedan transportarla a sus sistemas sin tener que volver a digitar la información.

---

\*Trabajo de grado

\*\*Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería de sistemas e informática.

## **ABSTRACT**

### **TITLE**

PROTOTYPE CREATION OF FORMS FOR CAPTURE AND STORAGE SYSTEMATIC DATA EFFECTIVELY THROUGH RELIABLE WEB ENVIRONMENT \*.

### **AUTHORS**

Jorge Antonio González Moreno

Oscar Fernando Larrota Jaimes \*\*

### **KEYWORDS**

Web Prototype, generator, flat file, data capture, form.

### **DESCRIPTION**

This project arose because the geographical coverage of a company has grown from a local physical installation at various points decentralized and isolated from what information is being generated continuously. Cases such as order capture, time control labor camps, monitoring and control evolution works, among others, are situations in which today still handles data capture in paper form before being transcribed.

Analyzing this problem we present the development of a prototype web (FormUIS) in order to generate forms, where the user determines the structure of the form, as the number of fields, the field type and some features like font and the form color. Later allows access to forms a direct link to users to provide to know for later filling out so we can collect the information they want.

Each time you deliberately fill out a form the system saves this information in a database, which only the user who created the form can access it, and providing improved security and reliability. Also download this information in a flat file so that users can then transport it to their systems without having to type in the information.

---

\* Work degree.

\*\*Faculty of Physical-Mechanical Engineering, School of Engineering and computer systems.

## 1. INTRODUCCIÓN

FormUIS es un prototipo web realizado para facilitar la recolección de datos mediante el uso de formularios y así garantizar un buen manejo de la información.

La aplicación ofrece una herramienta para el diseño de formularios en la web sin la necesidad de tener conocimiento de algún lenguaje de programación. Al mismo tiempo permite sistematizar y optimizar la forma de recolectar los datos teniendo como principal objetivo la descentralización de la captura de información desde sitios distantes a la empresa, compañía o institución.

FormUIS además disminuye tiempo de proceso ya que evita transcribir datos recolectados el cual sería un costo asociado al manejo de papel.

## **2. JUSTIFICACIÓN**

Hoy en día la cobertura geográfica de una empresa ha pasado de ser una instalación física local, a diversos puntos descentralizados y aislados desde los cuáles se está generando información continuamente.

Recopilar esta información en formatos pre impresos para enviarla a un centro en donde se transcribe en un sistema de información puede resultar demorado, poco confiable y susceptible a errores.

Casos como la captura de pedidos, control de tiempos en campamentos de obra, control de seguimiento y evolución de obras; entre otros, son situaciones en las que en la actualidad aún se maneja la captura de datos en formatos de papel para luego ser transcritos.

Con el desarrollo de este proyecto se propone una alternativa que permita a las organizaciones definir la estructura de sus formatos, hacerlos disponibles desde la web para su diligenciamiento y captura y luego exportarlos a sus sistemas sin tener que volver a digitar la información.

Se busca agilidad, precisión y confiabilidad al momento de capturar datos en puntos distantes de la organización.

### **3. OBJETIVOS**

#### **3.1 OBJETIVO GENERAL**

Diseñar un prototipo web que permita la creación de formularios para mejorar la recolección de la información de una manera simple y confiable.

#### **3.2 OBJETIVOS ESPECIFICOS**

- Desarrollar un módulo web de fácil manejo en la que el usuario pueda diseñar sus formularios de manera sencilla y según sus necesidades.
- Crear un acceso web al formulario con el fin que el usuario pueda compartirlo y recolectar la información requerida.
- Desarrollar un módulo mediante el cual el formulario diseñado por el usuario, actúe como interfaz de captura de datos tantas veces lo requiera el usuario y la almacene para su posterior uso.
- Desarrollar un módulo en el cual el usuario pueda ingresar y visualizar la información recolectada mediante la utilización de sus formularios creados.
- Permitir al usuario descargar la información capturada por medio de un archivo plano para que sea fácilmente transportada a sus sistemas.

#### **4. MARCO TEORICO O DE REFERENCIA**

Un formulario es un documento impreso creado con el fin de obtener datos e información de una forma estructurada para luego procesarla y almacenarla. Los formularios están diseñados para registrar diferente información, la cual tendrá una determinada utilización según el área a la que pertenezca. También es considerado como una herramienta de trabajo ya que transporta información para simplificar y facilitar los procedimientos administrativos.

Por lo general un formulario consta de 3 partes, un extremo superior o cabeza que se utiliza para escribir el título y es la identificación del organismo, otro es el cuerpo del formulario que posee una distribución y una secuencia lógica de acuerdo con la información que se desea obtener, y la tercera es el extremo inferior y se utiliza para escribir datos de cierre o conclusión del formulario. También firmas o sellos para aprobar la información registrada en el cuerpo del mismo.

Los formularios nacen con el fin de llenar las necesidades que las oficinas o empresas demandan ya que facilitan el flujo de la información y el procesamiento de los datos, manteniendo uniformidad en los procedimientos, relacionando información y dando valor legal a determinadas transacciones.

##### **4.1 HTML**

Es un acrónimo de HyperText Markup Lenguaje (lenguaje de marcado de hipertexto), en la actualidad es el lenguaje predominante en la construcción de páginas web debido en gran medida al soporte por parte de los navegadores existentes que son quienes traducen el código HTML para mostrar como resultado las páginas web. Este lenguaje es usado principalmente para generar páginas

estáticas, consta de dos partes, el contenido que es el texto que se mostrara en ventana y las etiquetas y atributos que contienen una serie de atribuciones o información adicional acerca del texto. El HTML puede describir hasta un cierto punto la apariencia en que se presenta un documento web.

El lenguaje HTML es un lenguaje utilizado únicamente para dar estructura a una página web, es de un estándar reconocido en todo el mundo por todas las empresas relacionadas con el mundo de Internet, así una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.[<sup>1</sup>]

HTML5 es la quinta revisión del lenguaje donde especifica dos variantes de sintaxis para HTML, un clásico HTML (text/html) y una variante XHTML conocida como sintaxis XHTML5.

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos son técnicamente similares a las etiquetas <div> y <span>, pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web). También proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>.

Mejoras en el elemento <canvas>, capaz de renderizar en los navegadores más importantes (Mozilla, Chrome, Opera, Safari e IE) elementos 3D. [<sup>2</sup>]

---

<sup>1</sup> Html: <https://belenus.unirioja.es/~guprado/pagweb/carachtml.html>

<sup>2</sup> Html5: <http://es.wikipedia.org/wiki/HTML5>

## 4.2 PHP

Es un lenguaje de programación de código abierto diseñado para el desarrollo web dinámico que puede ser incrustado directamente dentro del código HTML, fue uno de los primeros lenguajes de uso general del lado del servidor, este toma el código php como entrada y ejecuta el intérprete de php que procesa el script solicitado que genera el contenido de manera dinámica creando páginas web como salida. Aunque fue diseñado para trabajar desde el lado del servidor hay que resaltar que puede ser usado como cualquier otro lenguaje de programación. Sus principales ventajas son:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada a una base de datos.
- El código fuente escrito en php invisible al navegador y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar el resultado HTML al navegador. Esto hace que la programación en php sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos llamados ext's o extensiones.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

- Tiene manejo de excepciones (desde PHP5).
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

### **Inconvenientes**

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado.
- Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de cache tanto en archivos como en memoria.
- Las variables al no ser tipadas dificulta a los diferentes IDEs para ofrecer asistencias para el tapeado del código. [<sup>3</sup>]

## **4.3 JAVASCRIPT**

Java Script es un lenguaje de programación interpretado, dialecto del estándar ECMA Script. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

---

<sup>3</sup> Php: <http://php.net/manual/es/index.php>

Las principales características son:

- **Imperativo y estructurado**

JavaScript soporta gran parte de la estructura de programación de C como sentencias if, bucles for, sentencias switch y hace distinción entre expresiones y sentencias. Una diferencia sintáctica con respecto a C es que en JavaScript los puntos y coma que finalizan una sentencia pueden ser omitidos.

- **Tipado dinámico**

Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable. Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una cadena. JavaScript soporta varias formas de comprobar el tipo de un objeto, incluyendo duck typing.

- **Objetual**

JavaScript está formado casi en su totalidad por objetos. Los objetos en JavaScript son arrays asociativos, mejorados con la inclusión de prototipos. Los nombres de las propiedades de los objetos son claves de tipo cadena. Las propiedades y sus valores pueden ser creados, cambiados o eliminados en tiempo de ejecución.

- **Prototipos**

JavaScript usa prototipos en vez de clases para el uso de herencia. Es posible llegar a emular muchas de las características que proporcionan las clases en lenguajes orientados a objetos tradicionales por medio de prototipos en JavaScript.

- **Entorno de ejecución**

JavaScript normalmente depende del entorno en el que se ejecute para ofrecer objetos y métodos por los que los scripts pueden interactuar con el mundo exterior. De hecho, depende del entorno para ser capaz de proporcionar la capacidad de incluir o importar scripts.

- **Funciones como métodos**

A diferencia de muchos lenguajes orientados a objetos, no hay distinción entre la definición de función y la definición de método. Más bien, la distinción se produce durante la llamada a la función; una función puede ser llamada como un método. Cuando una función es llamada como un método de un objeto, la palabra clave `this`, que es una variable local a la función, representa al objeto que invocó dicha función.<sup>[4]</sup>

#### **4.4 JQUERY**

jQuery es una biblioteca JavaScript rápido, pequeño y rico en funciones que permiten simplificar la manera de interactuar con los documentos HTML, el control de eventos, animaciones mucho más simples con un API fácil de usar que funciona a través de una multitud de navegadores.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera

---

<sup>4</sup> JavaScript: <http://es.wikipedia.org/wiki/JavaScript>

requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.<sup>[5]</sup>

## 4.5 BASES DE DATOS

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

---

<sup>5</sup> JQuery: <http://librojquery.com>

Los Sistemas de Gestión de Base de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

#### **4.5.1 Bases de datos estáticas**

Son bases de datos de solo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

#### **4.5.2 Bases de datos dinámicas**

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta.

#### **4.5.3 Sistema de Gestión de bases de datos distribuida (SGBD)**

La base de datos y el software SGBD pueden estar distribuidos en múltiples sitios conectados por una red. Hay de dos tipos:

- Distribuidos homogéneos: utilizan el mismo SGBD en múltiples sitios.
- Distribuidos heterogéneos: Da lugar a los SGBD federados o sistemas multibase de datos en los que los SGBD participantes tienen cierto grado

de autonomía local y tienen acceso a varias bases de datos autónomas preexistentes almacenados en los SGBD, muchos de estos emplean una arquitectura cliente-servidor.[<sup>6</sup>]

## **4.6 MYSQL**

Mysql es un sistema de gestión de bases de datos relacional, multiplataforma y multiusuario, posiblemente uno de los más utilizados en el desarrollo web. Se distingue por su facilidad de uso, bajo costo en requerimientos para la elaboración de bases de datos y por su velocidad y seguridad al realizar las operaciones. Estas características y muchas otras además de que funciona bajo la licencia GNU GPL la han convertido en una opción muy atractiva para los desarrolladores web. De esta manera MYSQL se ha convertido en uno de los gestores de bases de datos con más alto rendimiento y apropiados para la web.

## **4.7 HOJAS DE ESTILO EN CASCADA (CSS)**

Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas.

CSS es un lenguaje de estilo que define la presentación de los documentos HTML. CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas.

---

<sup>6</sup> Base de datos: [http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos)

Entre los beneficios concretos de CSS encontramos:

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (ventana, impresión, etc.).
- Numerosas técnicas avanzadas y sofisticadas.<sup>[7]</sup>

## **4.8 SUBLIME TEXT 2**

Sublime Text 2 es un editor de código al estilo TextMate, Kate o Redcar, su interfaz es limpia e intuitiva y soporta el uso de Snippets, Plugins y sistemas de construcción de código (Build Systems). Sublime Text 2 aún conserva un modo de edición tipo vi llamado “Vintage mode”.

Sublime Text 2 está escrito en C++ y Python, incluye una implementación de CPython2.6 embebida así como una consola que no es sino un intérprete de Python desde el cual podemos ejecutar comandos y realizar tareas de retrospcción y hacks múltiples.

## **4.9 TEXTO PLANO**

Los archivos de texto plano (en inglés plain text) son aquellos que están compuestos únicamente por texto sin formato, sólo caracteres. Estos caracteres se pueden codificar de distintos modos dependiendo de la lengua usada. También se le conoce como archivos de texto llano, o texto simple, por carecer de información destinada a generar formatos (negritas, subrayado, cursivas, tamaño, etc.) y tipos de letra (por ejemplo, Arial, Times, Courier, etc.). El término texto plano proviene de una traducción literal del término inglés plain text, término que en lengua castellana significa texto simple o texto sencillo.

---

<sup>7</sup> Css: <http://es.wikipedia.org/wiki/Css>

Las aplicaciones destinadas a la escritura y modificación de archivos de texto se llaman editores de texto. Los archivos de texto plano son aquellos que no tienen ningún tipo de formato como tipo de letra, que corresponde a una información específica definida, donde los campos que hacen parte del archivo están separados por un carácter, especialmente usados la coma (,) los (;) y el PIPE (|), donde cada línea corresponde a un registro y que generalmente se usa para transportar información de manera liviana, especialmente para el cargue de datos en tablas de una base de datos. [8]

## 5. METODOLOGIA

Para la implementación del prototipo se plantea una metodología de desarrollo evolutivo gracias a la propiedad del método, en el cual mediante un análisis cuidadoso de los requerimientos del sistema se seleccionan los que son bien comprendidos, con los cuales se hace el desarrollo inicial del prototipo que luego de exponerse se va refinando de acuerdo a la retroalimentación de los usuarios a los desarrolladores, esto con lleva a la especificación de nuevos requerimientos y una segunda versión del producto es desarrollada y desplegada. Este proceso se repite indefinidamente.

Los riesgos asociados con el desarrollo software son elevados, una forma de reducir estos riesgos es empezar a construir una parte del sistema, reservando algunos aspectos para más adelante, por tal motivo el desarrollo evolutivo es una buena alternativa en la cual en el proceso de construcción se van incrementando subconjuntos de requerimientos del sistema.

Los principales beneficios del desarrollo evolutivo son:

---

<sup>8</sup> Texto plano: [http://es.wikipedia.org/wiki/Archivo\\_de\\_texto](http://es.wikipedia.org/wiki/Archivo_de_texto)

- Construir un sistema pequeño es mucho más seguro que construir un sistema grande.
- Al desarrollar algunas funcionalidades es más sencillo determinar si los requerimientos planeados para los siguientes niveles son correctos.
- Si se realiza un error importante sólo la última iteración necesita ser descartada.
- Minimizando el tiempo de desarrollo del sistema decrecen las probabilidades que esos requerimientos de usuarios puedan cambiar durante el desarrollo.
- Los errores de desarrollo cometidos en una evolución pueden ser arreglados antes del comienzo de la próxima evolución.

Figura 1. Modelo de desarrollo evolutivo



Fuente: Desarrollo evolutivo. En línea.

## **5.1 FASE ESPECIFICACIÓN INICIAL**

Definición del problema y especificación inicial en base a los requerimientos definidos.

En esta fase inicial describimos el prototipo web partiendo del objetivo general del proyecto e identificamos los primeros requerimientos y funcionalidades, con ello logramos definir las siguientes características del prototipo:

- Permitir el ingreso al sistema mediante usuario y contraseña.
- Presentar a los usuarios una herramienta dinámica de fácil comprensión con las diferentes opciones para el diseño del formulario.
- Implementar una opción en la cual se crea y publica el formulario.
- Permitir al usuario un acceso a la información recolectada.

## **5.2 FASE DESARROLLO DEL PRODUCTO**

Desarrollo del software en base a un proceso con énfasis en la rapidez de la liberación. En esta fase de desarrollo se define el modelo, la estructura de los archivos y el diseño de las interfaces que interaccionan con el usuario, para esta implementación se realizó:

- Diseño de la interfaz inicial.
- Diseño de la interfaz de ingreso al sistema.
- Diseño de la interfaz de creación del formulario.
- Diseño de la interfaz de visualización del formulario.
- Diseño de la interfaz de acceso a la información recolectada.
- Diseño de la base de datos para la integridad del prototipo.

Para la realización de la interfaz inicial, de ingreso al sistema, de visualización del formulario, de acceso a la información recolectada se utilizaron los lenguajes de programación HTML y PHP.

Para la interfaz de creación de formularios se utilizó JQUERY para brindarle un estilo dinámico y fácil al usuario.

Para la realización de base de datos utilizamos MySQL.

### **5.3 FASE IMPLEMENTACIÓN USO Y EVALUACIÓN**

Implantación y uso del software en ambiente de explotación, monitoreo de los nuevos requerimientos.

En el transcurso de esta fase se implementa la codificación y el desarrollo del prototipo web y se utiliza la última versión creada del prototipo para verificar su correcto uso, su desempeño, su almacenamiento y concordancia de los datos.

### **5.4 FASE RE-ESPECIFICACIÓN**

Re-definición del problema en base a los nuevos requerimientos.

El desarrollo de software con una metodología evolutiva requiere un riguroso cuidado en la manipulación de documentos de las diferentes versiones del software creadas, cada paso debe ser registrado y los cambios deben ser realizados de una manera controlada.

En esta fase se realiza un análisis global del prototipo para verificación de los requerimientos e implementación en caso que surjan nuevos requerimientos.

## **6 DESARROLLO DEL PROTOTIPO WEB**

### **6.1 DIAGRAMAS UML**

Desde el inicio y tomando como base la aplicación web se utilizó UML como modelado del prototipo y se escogieron los siguientes diagramas:

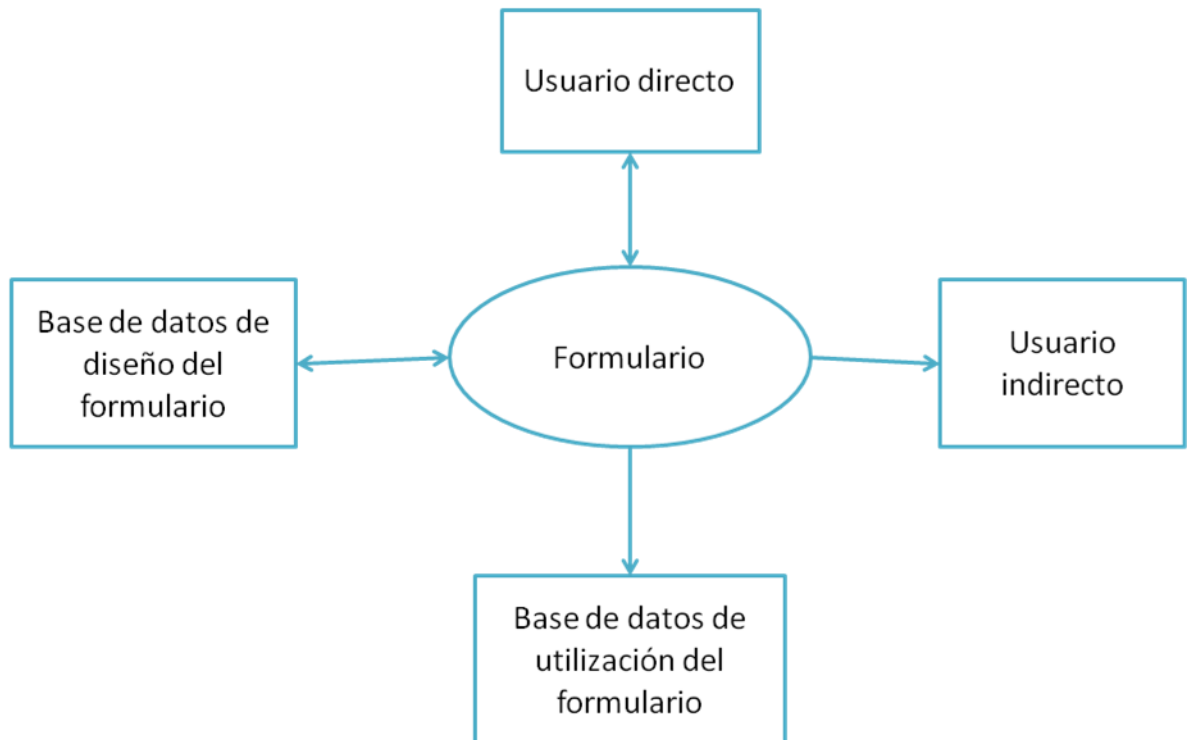
- Diagrama de Contexto
- Diagrama de casos de uso
- Diagrama de secuencia
- Diagrama de Clases

#### **6.1.1 Diagrama de contexto:**

Utilizamos este diagrama ya que nos proporciona la representación de los límites del sistema y su entorno identificando lo que hace y no parte de nuestro sistema.

Los actores del sistema en este caso los definimos como usuario directo e indirecto, siendo el usuario directo aquel que se registra en el sistema y accede a él con el fin de crear un formulario para su posterior utilización con las respectivas opciones, mientras que el usuario indirecto es aquel usuario que solo accede a la ubicación de un formulario para diligenciarlo y no tiene más interacción con el sistema si no solo enviar los datos ingresados al formulario.

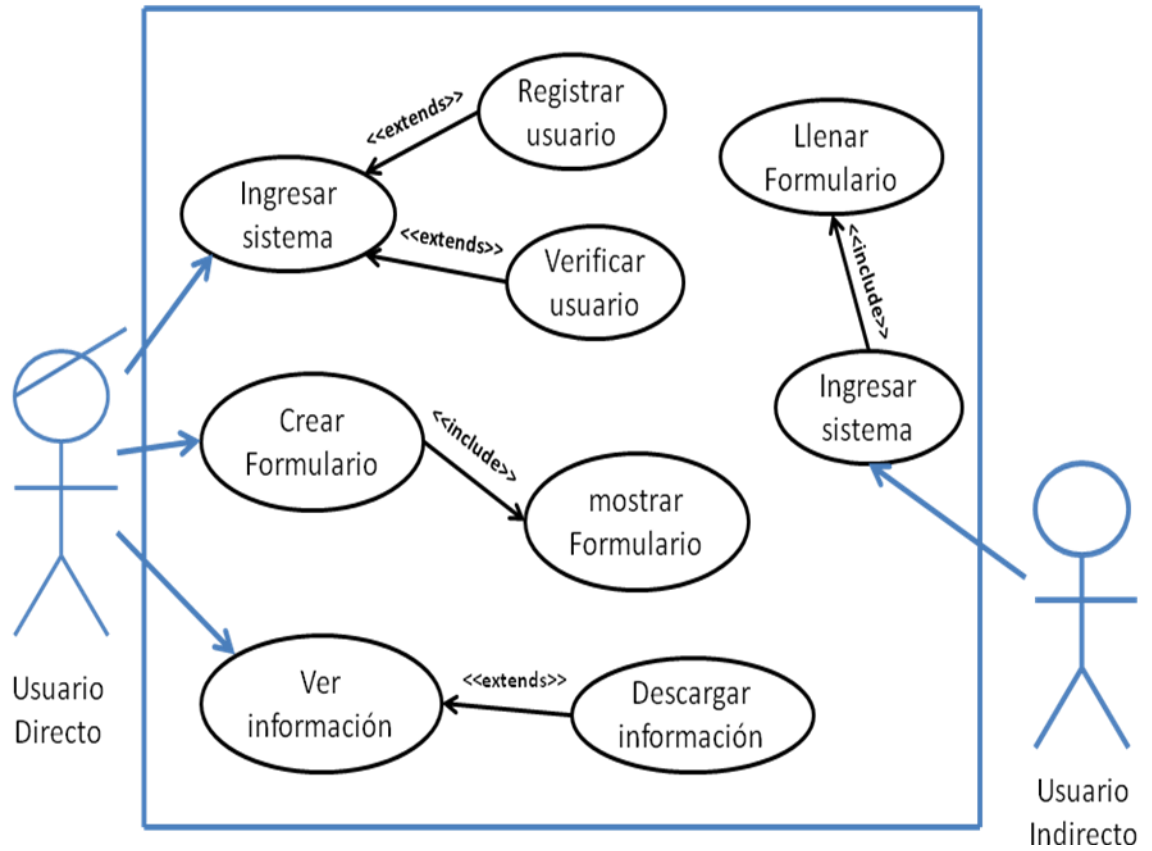
Figura 2. Diagrama de contexto



### 6.1.2 Diagrama de casos de uso:

Utilizamos este diagrama porque nos ayuda a representar la funcionalidad del sistema donde describe los actores con los cuales interactúa el sistema, el mismo sistema, los casos de uso o servicios que ejecutara el sistema y la relación entre estos elementos.

Figura 3. Diagrama de casos de uso.



➤ **Especificaciones caso de uso**

Tabla 1. Usuario directo

<b>ACTOR</b>	
<b>Actor</b>	Usuario Directo
<b>Casos de Uso</b>	Crear formulario, ver información
<b>Tipo</b>	Iniciador, primario
<b>Descripción</b>	Este actor es la persona que diseña y crea un formulario para obtener determinada información.

Tabla 2. Usuario indirecto

<b>ACTOR</b>	
<b>Actor</b>	Usuario Indirecto
<b>Casos de Uso</b>	Llenar formulario
<b>Tipo</b>	Primario
<b>Descripción</b>	Este actor es la persona que accede a un formulario creado para ingresar la información requerida.

Tabla 3. Caso de uso Registrar Usuario

<b>CASOS DE USO</b>		
<b>Caso de Uso</b>	<b>Registrar Usuario</b>	
<b>Actor(es)</b>	Usuario nuevo	
<b>Propósito</b>	Registrarse para poder ingresar a las funciones que brinda el sistema.	
<b>Descripción</b>	El sistema brinda la opción para que nuevos usuarios puedan ingresar y utilizar el generador de formularios, para ello cuenta con un registro donde un nuevo usuario debe ingresar sus datos personales como nombre, apellidos, teléfono, sexo, fecha de nacimiento, email y proporcionar una contraseña para un mejor ingreso al sistema.	
<b>Precondición(es)</b>	Ninguna.	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<ol style="list-style-type: none"> <li>1. Un futuro usuario desea crear un formulario por primera vez.</li> <li>2. Dicho usuario llena un formulario el cual tendrá los datos principales para información y acceso a su cuenta.</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema valida y</li> </ol>

	<p>4. Si la notificación fue satisfactoria el nuevo usuario ya puede ingresar al sistema, en caso contrario deberá rectificar los datos ingresados.</p>	<p>almacena la información suministrada por el nuevo usuario y envía una notificación haciéndole saber al nuevo usuario si el registro fue o no satisfactorio.</p>
<b>Sub-flujos</b>	<p>Si el nuevo usuario se quiere registrar con un email ya existente en la base de datos el sistema no lo permitirá y le notificara con un mensaje de error.</p>	
<b>Poscondición(es)</b>	<p>El sistema debe permitirle al nuevo usuario ingresar al sistema si el registro se realizo satisfactoriamente.</p>	

Tabla 4. Caso de uso Ingresar al sistema

<b>CASOS DE USO</b>		
<b>Caso de Uso</b>	<b>Ingresar al sistema</b>	
<b>Actor(es)</b>	Usuario Directo	
<b>Propósito</b>	Acceder al sistema para crear un formulario, para consultar y descargar información recolectada mediante el uso de un formulario.	
<b>Descripción</b>	El usuario accede al sistema con su email y contraseña y el sistema valida dicha información para permitir o no el ingreso.	
<b>Precondición(es)</b>	El usuario debe estar registrado.	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<ol style="list-style-type: none"> <li>1. El caso de uso inicia cuando el usuario desea crear, consultar información acerca de un formulario.</li> <li>2. El usuario digita su email y contraseña (password).</li> <li>4. El usuario accede al</li> </ol>	<ol style="list-style-type: none"> <li>3. El sistema valida el email y la contraseña para dar acceso o no al sistema.</li> </ol>

	sistema para su uso.	
<b>Sub-flujos</b>	<p>El usuario debe estar registrado de lo contrario el sistema le muestra un mensaje de error.</p> <p>El formulario tiene que estar creado para poder acceder a él.</p>	
<b>Poscondición(es)</b>	El sistema debe permitirle al usuario realizar sus respectivas funciones.	

Tabla 5. Caso de uso Crear formulario

<b>CASOS DE USO</b>		
<b>Caso de Uso</b>	<b>Crear formulario</b>	
<b>Actor(es)</b>	Usuario Directo	
<b>Propósito</b>	Permitir al usuario crear un formulario.	
<b>Descripción</b>	El usuario crea un formulario de acuerdo a sus necesidades para mostrarlo y recolectar la información deseada.	
<b>Precondición(es)</b>	El usuario ya ingreso al sistema con su nombre de usuario y contraseña.	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El usuario crea con las diferentes opciones y a su medida el formulario.</p>	<p>2. El sistema almacena la</p>

	3. El usuario da a conocer el formulario creado.	estructura del formulario diseñado por el usuario.
<b>Sub-flujos</b>	Ninguno	
<b>Poscondición(es)</b>	El sistema debe permitir dejarlo acceder al formulario creado.	

Tabla 6. Caso de uso Llenar formulario

<b>CASOS DE USO</b>	
<b>Caso de Uso</b>	<b>Llenar Formulario</b>
<b>Actor(es)</b>	Usuario Indirecto
<b>Propósito</b>	Diligenciar un formulario.
<b>Descripción</b>	Permitirle al usuario acceder al formulario sin la necesidad que este registrado en el sistema para suministrar y enviar los datos que ingresa al formulario.
<b>Precondición(es)</b>	El formulario debe estar creado y disponible para la recolección de datos.

<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	<p>1. El usuario accede a la ubicación del formulario.</p> <p>3. El usuario diligencia el formulario y envía la información.</p>	<p>2. El sistema muestra el formulario.</p> <p>4. El sistema almacena la información suministrada.</p>
<b>Sub-flujos</b>	Ninguno	
<b>Poscondición(es)</b>	Ninguna.	

Tabla 7. Caso de uso Ver información

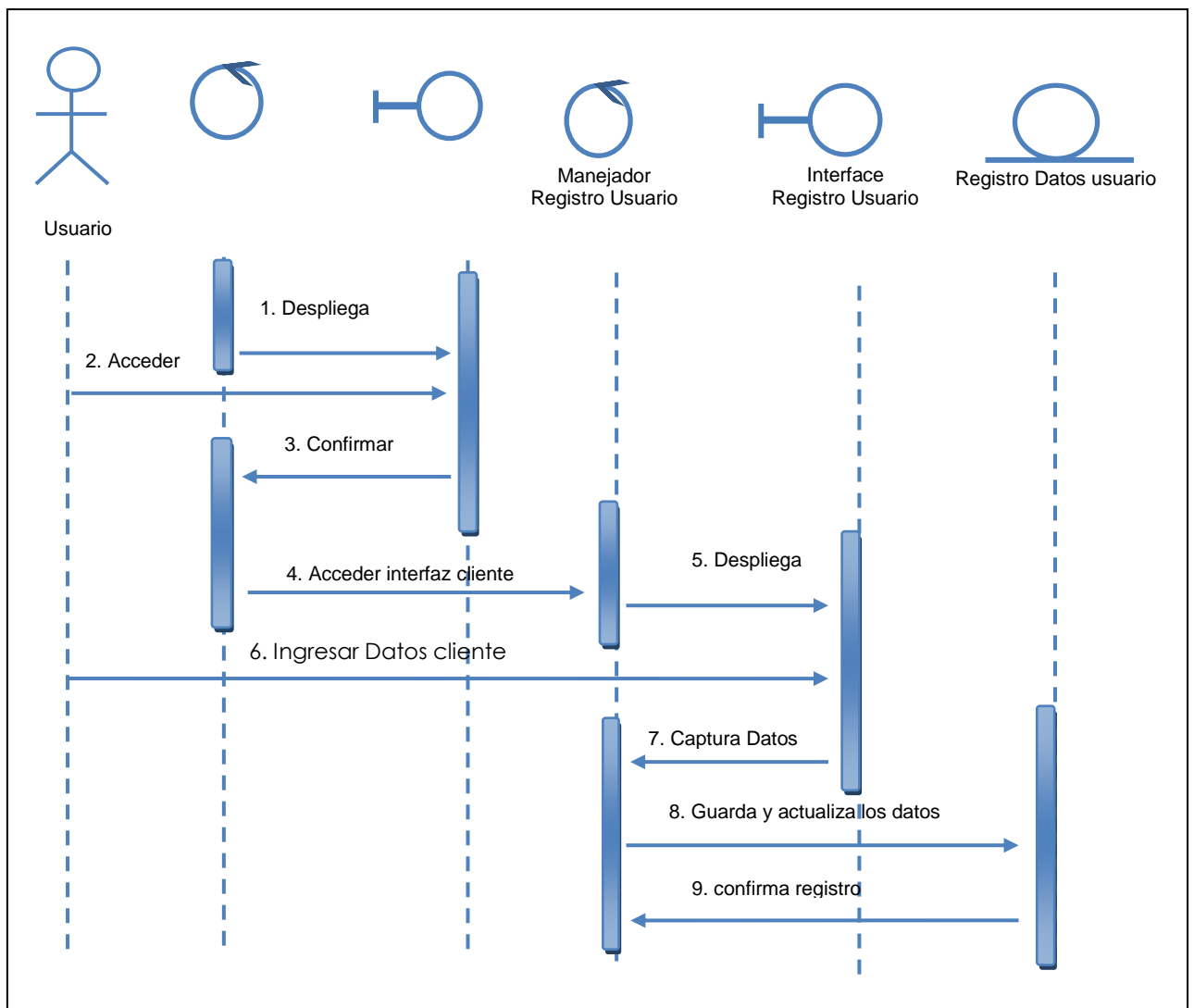
<b>CASOS DE USO</b>	
<b>Caso de Uso</b>	<b>Ver información</b>
<b>Actor(es)</b>	Usuario Directo
<b>Propósito</b>	Acceder a la información.
<b>Descripción</b>	Permitirle al usuario obtener la información recolectada

	mediante el uso del formulario creado.	
<b>Precondición(es)</b>	El usuario dio a conocer el formulario y este fue respectivamente llenado.	
<b>Flujo Principal</b>	<b>Acciones de Actor(es)</b>	<b>Respuestas del Sistema</b>
	1. El usuario accede a la información recolectada mediante su formulario.	2. El sistema muestra la información almacenada cada vez que utilizaron el formulario.
<b>Sub-flujos</b>	Ninguno	
<b>Poscondición(es)</b>	El sistema debe permitir descargar la información para su posterior utilización.	

### 6.1.3 Diagrama de secuencia

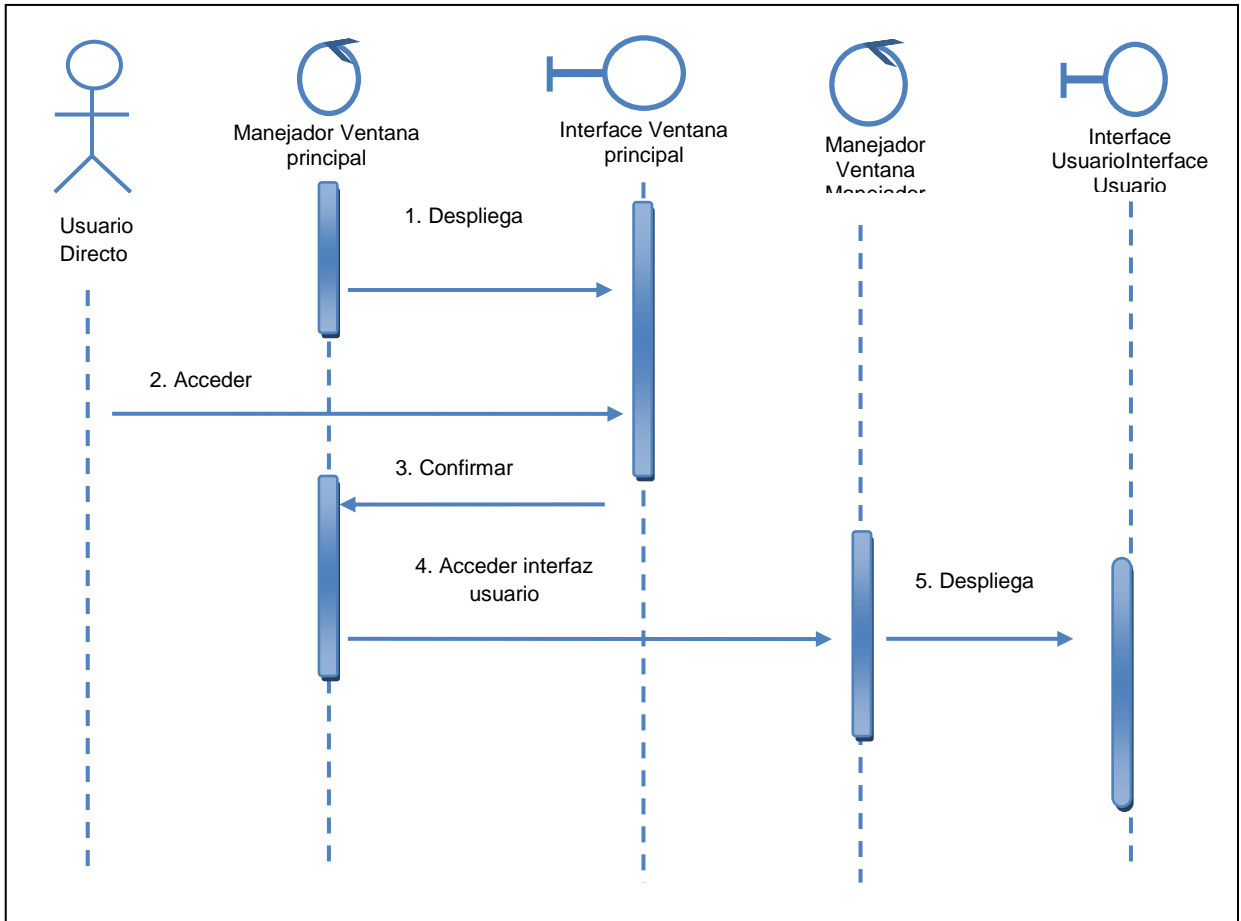
#### ➤ Registrar Usuario

Figura 4. Diagrama Secuencia: Registrar usuario



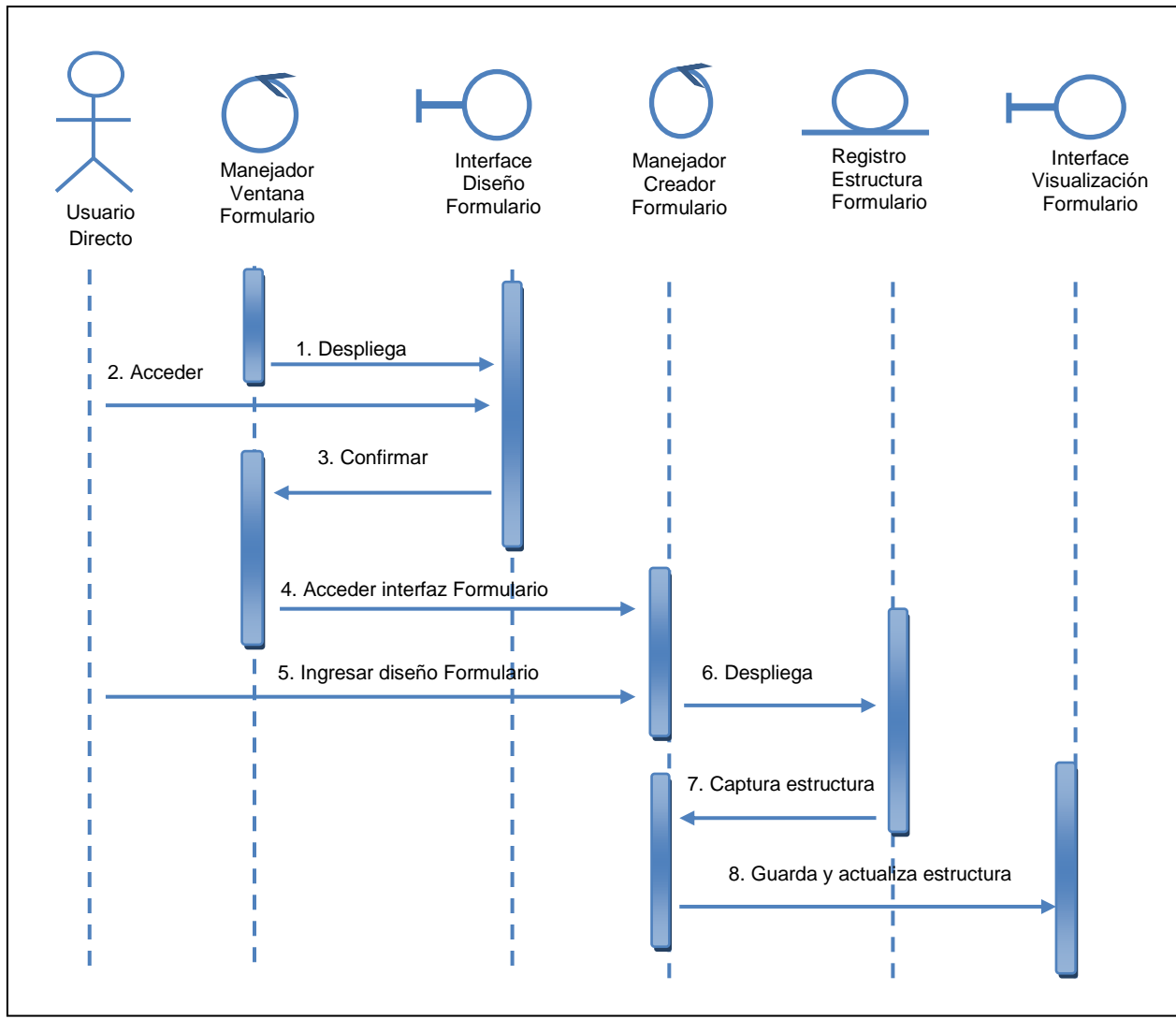
➤ Ingresar al sistema Usuario Directo

Figura 5. Diagrama Secuencia: Ingresar al sistema



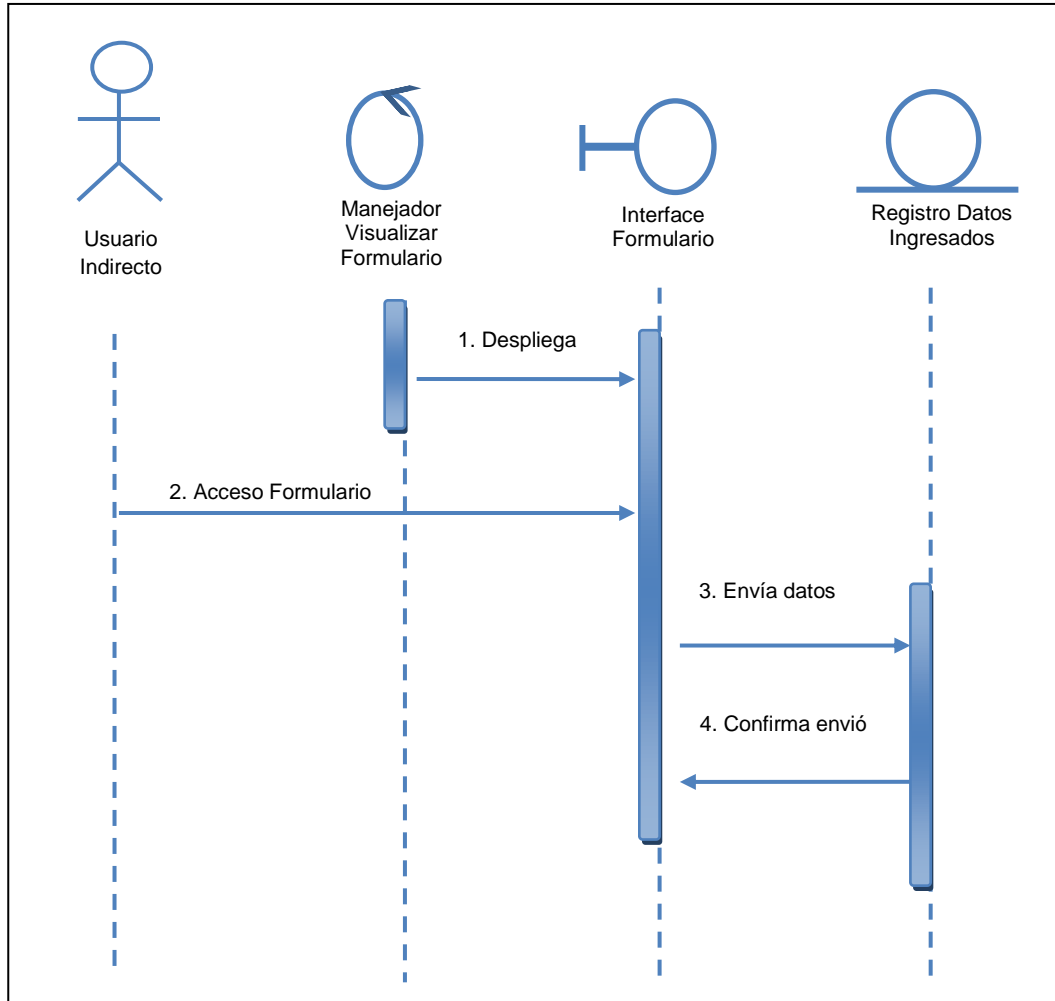
➤ **Crear Formulario**

Figura 6. Diagrama Secuencia: Crear Formulario



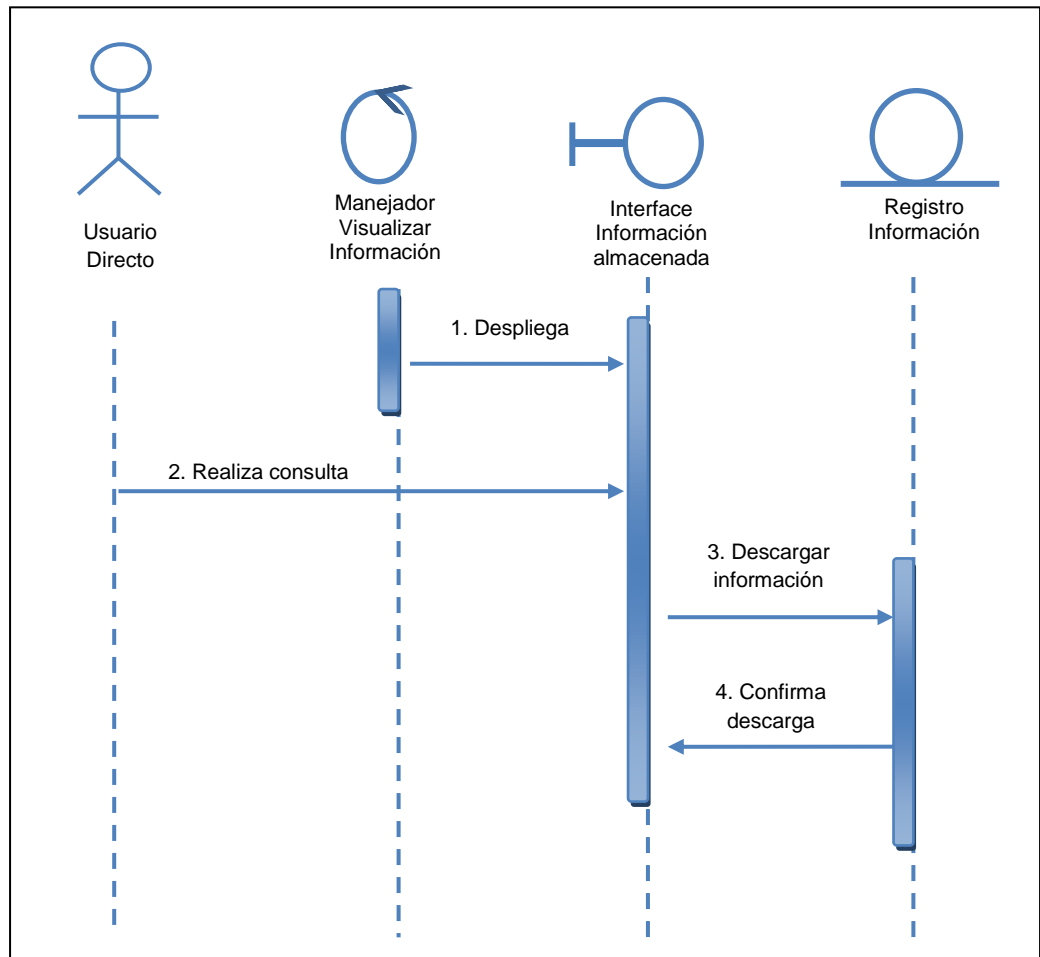
➤ **Llenar Formulario**

Figura 7. Diagrama Secuencia: Llenar Formulario



➤ **Ver información**

Figura 8. Diagrama Secuencia: Ver Información



#### 6.1.4 Diagrama de clases:

Utilizamos este tipo de diagrama estático ya que describe la estructura del sistema con sus respectivas clases, atributos y la relación entre ellos. Además porque estos diagramas son muy utilizados durante el proceso de análisis y diseño del sistema ya que se crea el diseño conceptual de la información que se manejará en el sistema. También se hace la descripción del sistema y se subrayan los sustantivos o lo que representara una clase y se obtiene una lista de clases candidatas.

##### ➤ Descripción del Sistema Subrayado

FORMUIS es un prototipo web que brinda control y manejo en el área de la creación y diseño de formularios sin que el usuario tenga o posea conocimientos en lenguajes de programación. Posee un acceso por medio de usuario y contraseña para la creación de formularios, acceso a la información obtenida y un ingreso directo sin usuario y contraseña para acceder a un determinado formulario para su posterior diligenciamiento de manera fácil y rápida.

El prototipo proporciona confiabilidad de la información ya que la almacena en una base de datos y permite conocerla y extraerla con el fin que el usuario pueda utilizarla en otro sistema que la requiera.

La ventana inicial del prototipo nos da unas determinadas opciones para registrarse o ingresar si ya se encuentra registrado, para ver un demo cómo funciona y una opción de contacto para alguna sugerencia o inquietud que posea.

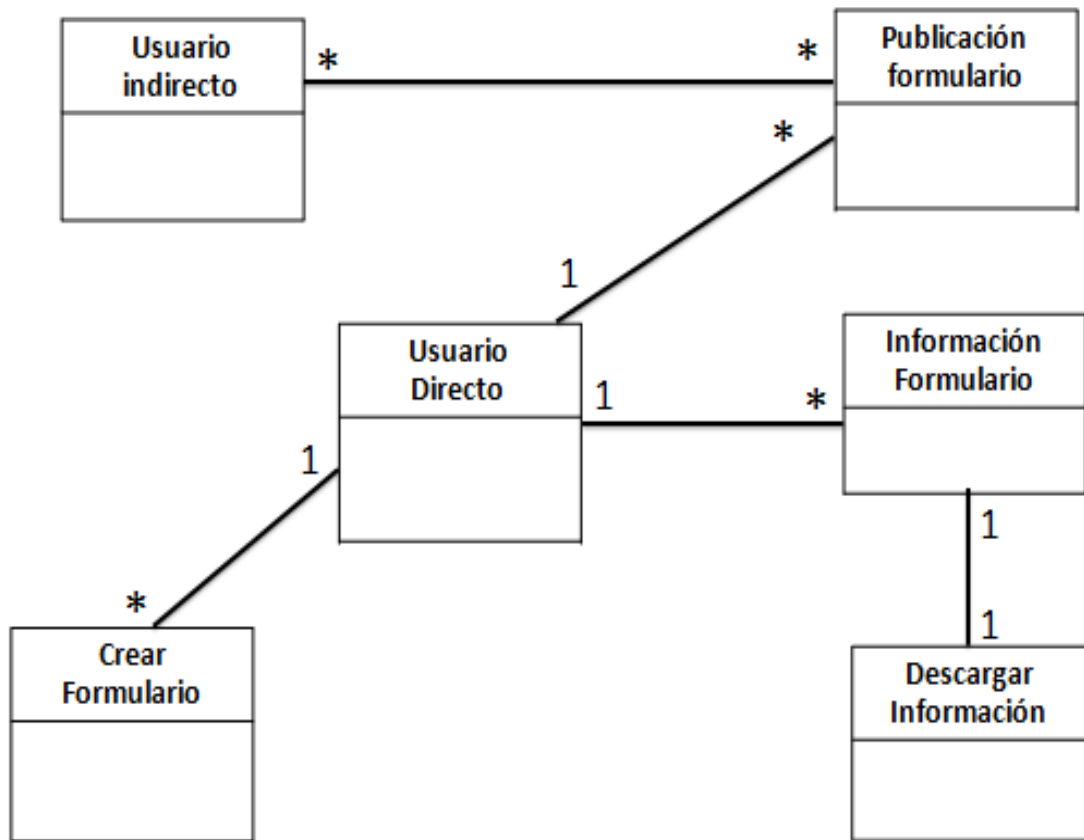
Una vez que ingresa al sistema se muestra una interfaz donde está el perfil, la opción de crear un formulario nuevo, de ver uno ya creado, acceder a la información recolectada y la opción de salir del sistema.

➤ **Lista de conceptos involucrados en el sistema o lista de clases candidatas.**

- Prototipo
- Área
- Creación
- Diseño
- Formularios
- Usuario
- Programación
- Confiabilidad
- Información
- Base de datos
- Ingresar
- Demo
- Crear
- Sistema
- Acceder

➤ Diagrama de clases Relacional

Figura 9. Diagrama Relacional



➤ **Diccionario de clases**





Tabla diccionario de clases

<b>DICCIONARIO DE CLASES</b>	
<b>Clase</b>	<b>Descripción</b>
<b>Usuario Directo</b>	Representa la persona registrada en el sistema y tiene acceso a las diferentes opciones del prototipo web.
<b>Usuario Indirecto</b>	Representan las personas que acceden al formulario creado con el fin de llenarlo y enviar los datos suministrados.
<b>Formulario</b>	<ul style="list-style-type: none"><li>• <b>Crear:</b> Hace referencia al módulo donde el usuario diseña un formulario de acuerdo a su necesidad.</li><li>• <b>Publicar:</b> Es el módulo mediante el cual el formulario diseñado por el usuario directo actúa como interfaz para la captura de datos</li><li>• <b>Acceso información:</b> Es el modulo en el cual se muestran los datos capturados atreves de formulario publicado.</li><li>• <b>Descargar:</b> Es la opción que permite extraer la información recolectada mediante el uso del formulario para una determinada acción que requiere el usuario.</li></ul>

### 6.1.5 Diagrama de actividades:

Un diagrama de actividades muestra un flujo de acción, son nodos que ejecutan un proceso, generalmente son secuenciales y representan los resultados de dichas acciones. Sirven para capturar las acciones internas de un proceso, la especificación de un caso de uso y mostrar flujos entre procesos.<sup>9</sup>

Tabla 8. Diagrama de actividades

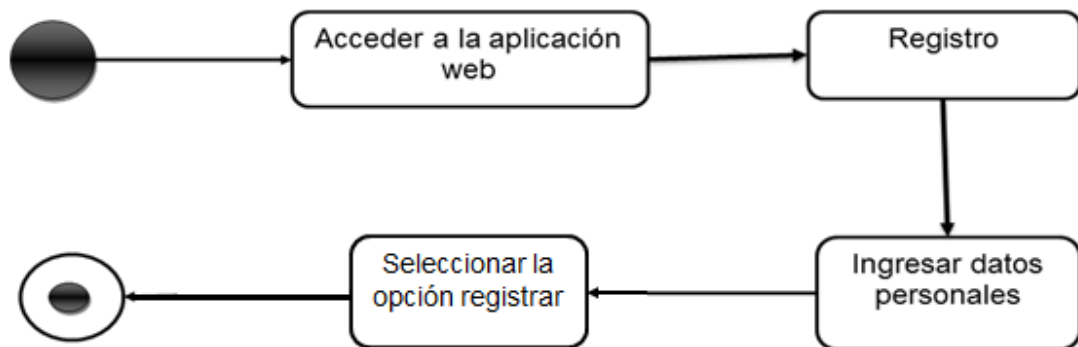
Símbolo	Nombre	Descripción
	Nodo inicial	Muestra el punto de partida del flujo de acciones.
	Acción	Representa una actividad o acción. El nombre generalmente empieza con un verbo.
	Flujo o transición	Muestra el orden de ejecución de las actividades.
	Nodo final	El final de todos los flujos de acciones en el diagrama.

Fuente: Diagrama de actividades. En línea.

<sup>9</sup> Diagrama de actividades: <http://www.slideshare.net/camiloan40/diagrama-de-actividades-uml>

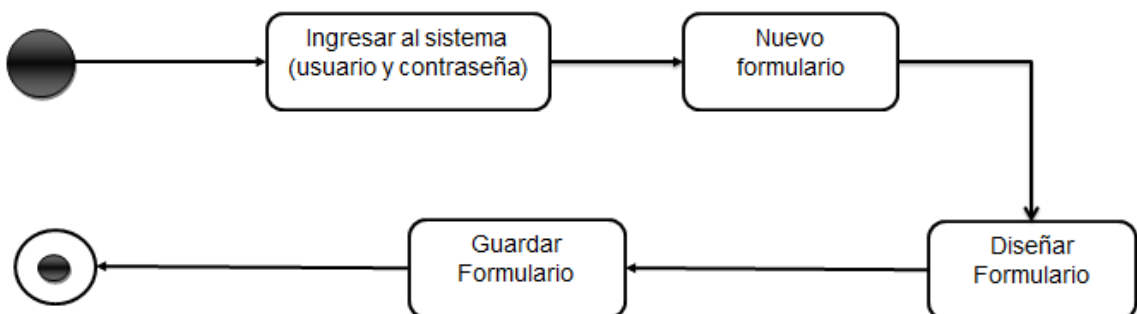
➤ **Diagrama registrar usuario**

Figura 10. Diagrama Registrar usuario



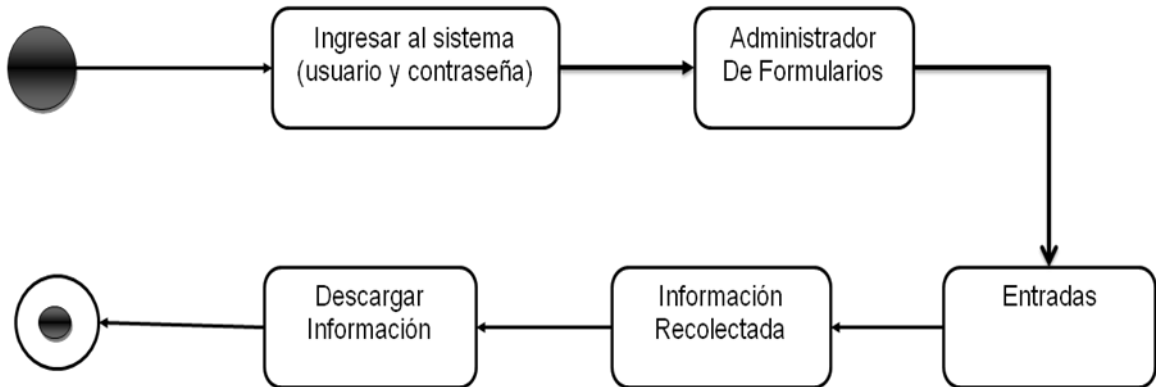
➤ **Diagrama crear formulario**

Figura 11. Diagrama crear formulario



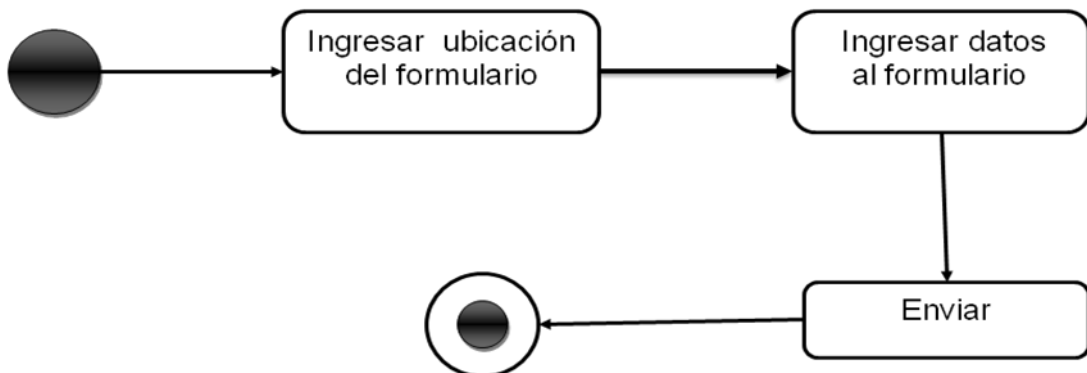
➤ **Diagrama obtener información**

Figura 12. Diagrama obtener información



➤ **Diagrama diligenciar formulario**

Figura 13. Diagrama diligenciar formulario



## 6.2 ESTRUCTURA DE DATOS

FormUIS se soporta en una base de datos para generar un formulario, en la cual se insertan las siguientes características:

- Almacenamiento del usuario.
- Nombre del formulario.
- Identificador del formulario (id).
- Método de envío.
- Estilos del formulario.

A su vez almacena las características de los campos del formulario escogidos por el usuario. Los cuales son:

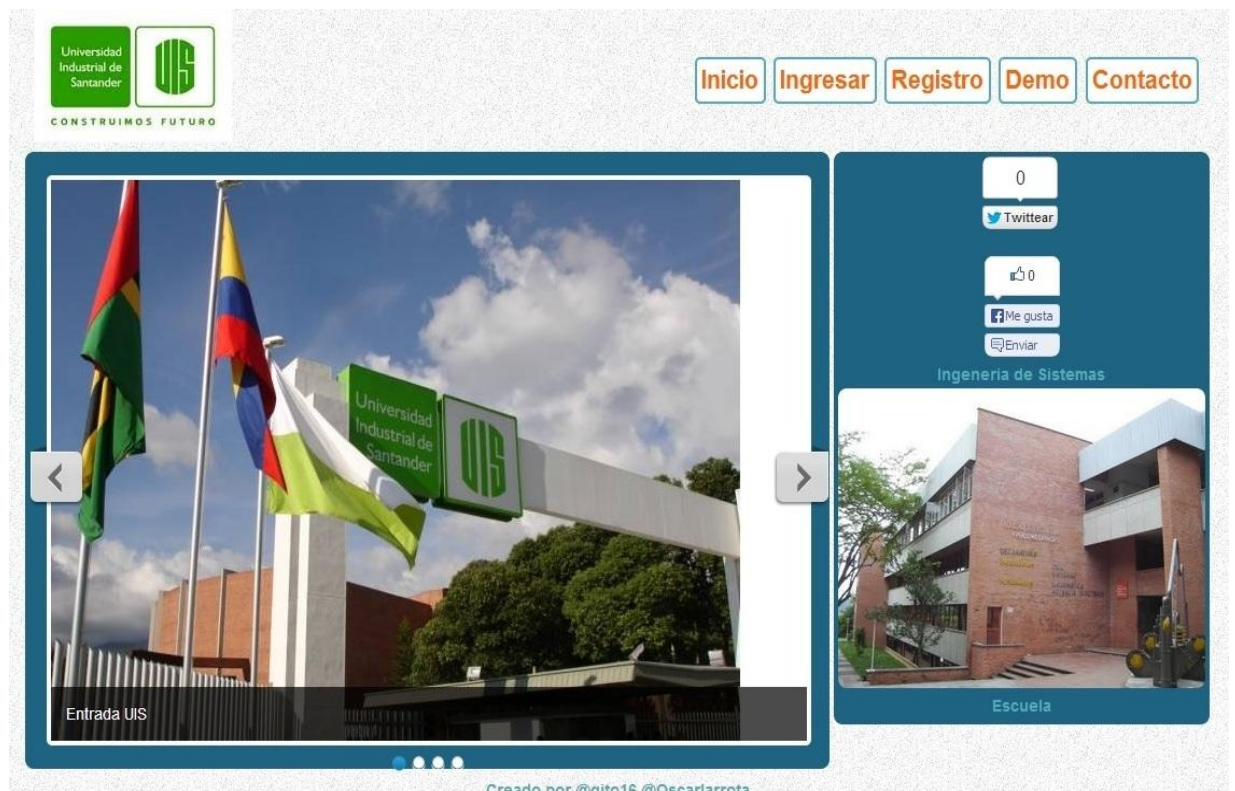
- **Tipo de campo:** El cual se usa para identificar la forma en la que se capturan los datos.
- **Nombre del campo:** Se utiliza para recolectar la información que se ingresa dentro del campo.
- **Identificador del campo:** Para el formulario utilizamos dos identificadores el primero para identificar a que formulario pertenece y el segundo es el identificador único dentro del formulario.
- **Index:** Se utiliza para identificar la posición del campo dentro del formulario.
- **Label:** Este componente se utiliza para desplegar textos o mensajes estáticos ingresados por el usuario.
- **Descripción:** Este componente se utiliza para brindar una orientación.
- **Estilos del campo:** Se utilizan para darle apariencia al campo.

## 7. INTERFACES DEL SISTEMA

### Ventana Inicio

La ventana de inicio está conformada por una serie de botones o vínculos como registro, ingresar, contacto en los cuales el usuario podrá navegar entre ellos y así utilizar las diferentes opciones establecidas. También cuenta con una serie de imágenes de la Universidad Industrial de Santander ya que quisimos exaltar el campus universitario, algunas de estas imágenes son un vínculo que permite acceder a la página web de la universidad y de nuestra escuela Ingeniera de sistemas. Por último decidimos colocar unas opciones de redes sociales.

Figura 14. Ventana de inicio



## Ventana Registro

La ventana de registro consta de un formulario donde se solicitan algunos datos personales para el usuario a registrarse como son el nombre, apellidos, email, una contraseña para el futuro ingreso, teléfono, sexo y su fecha de nacimiento, luego de haber diligenciado el formulario el usuario da clic en registrar y el sistema realiza una validación para dejar registrar o no al usuario, una vez satisfactorio el registro ya podrá ingresar al sistema.

Figura 15. Ventana de registro

Universidad Industrial de Santander  
CONSTRUIMOS FUTURO

[Inicio](#) [Ingresar](#) [Registro](#) [Demo](#) [Contacto](#)

### Regístrate

#### Datos Personales

Nombre:

Apellidos:

Email:

Contraseña:

Teléfono:

Sexo:  Masculino  Femenino

Fecha de nacimiento:

Creado por @gito16 @Oscarlarrota

## Ventana Ingresar

La ventana ingresar consta de un formulario donde se solicita el email y la contraseña, una vez el usuario ingresa los datos y da clic en ingresar el sistema valida que la información suministrada sea correcta, si es así le permite acceder al sistema.

Figura 16. Ventana de ingreso

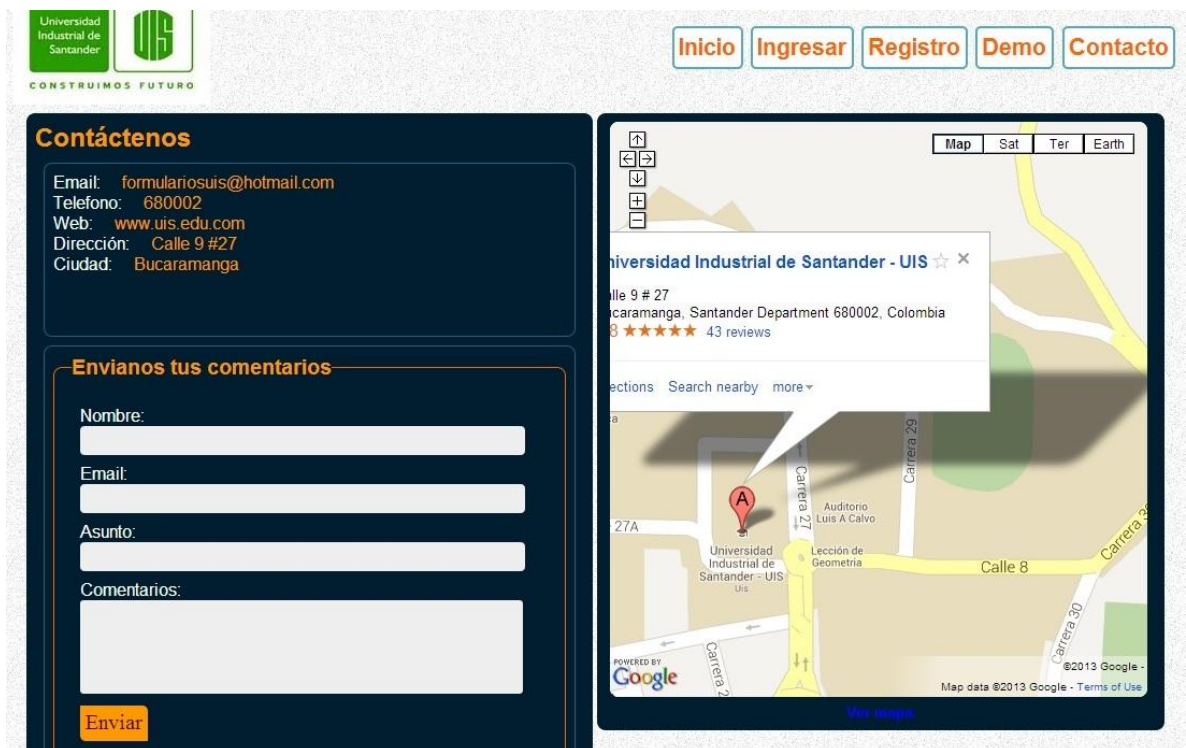


The image shows a web interface for the Universidad Industrial de Santander. In the top left corner, there is a logo with the text "Universidad Industrial de Santander" and "CONSTRUIMOS FUTURO". In the top right corner, there are five navigation buttons: "Inicio", "Ingresar", "Registro", "Demo", and "Contacto". The central focus is a dark blue login form titled "Ingresar" with the subtitle "Escribe tus datos". The form contains two input fields: "Email:" with the placeholder "Escribe tu email" and "Contraseña:" with the placeholder "Escribe tu clave". Below the fields is an orange "Ingresar" button. At the bottom of the page, there is a small text credit: "Creado por @gito16 @Oscariarota".

## Ventana Contacto

La ventana contacto muestra cierta información de la de universidad y en un mapa donde se encuentra ubicada y da la opción de enviar un mensaje mediante un formulario donde se debe ingresar el nombre, el email, el asunto y el comentario de la razón por el cual decide escribir.

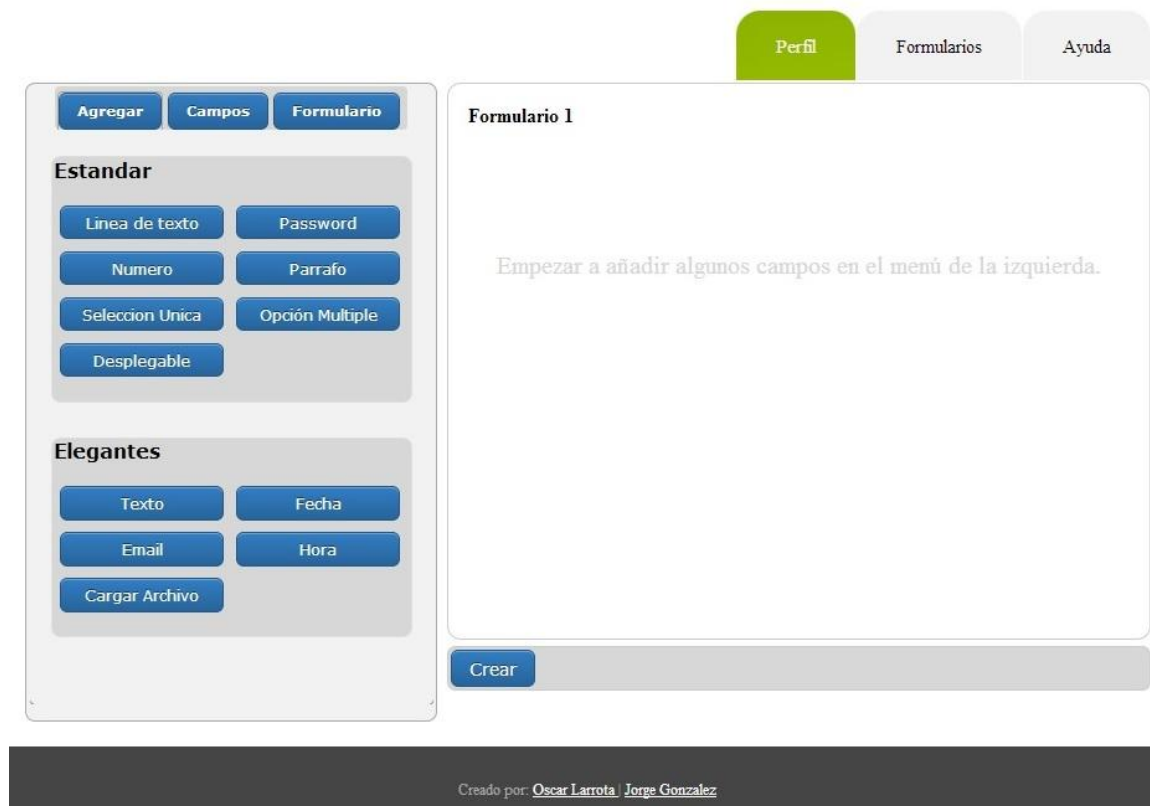
Figura 17. Ventana Contacto



## Ventana Crear Formulario

La ventana crear formulario consta de una serie de botones ubicados en la parte izquierda de la ventana los cuales brindan de forma dinámica las diferentes opciones necesarias para la elaboración de un formulario, el cual se estará visualizando en la parte derecha cada vez que el usuario inserta o modifica una opción o campo de su formulario. Una vez diseñado el formulario por el usuario da clic en crear y este será almacenado para su posterior uso.

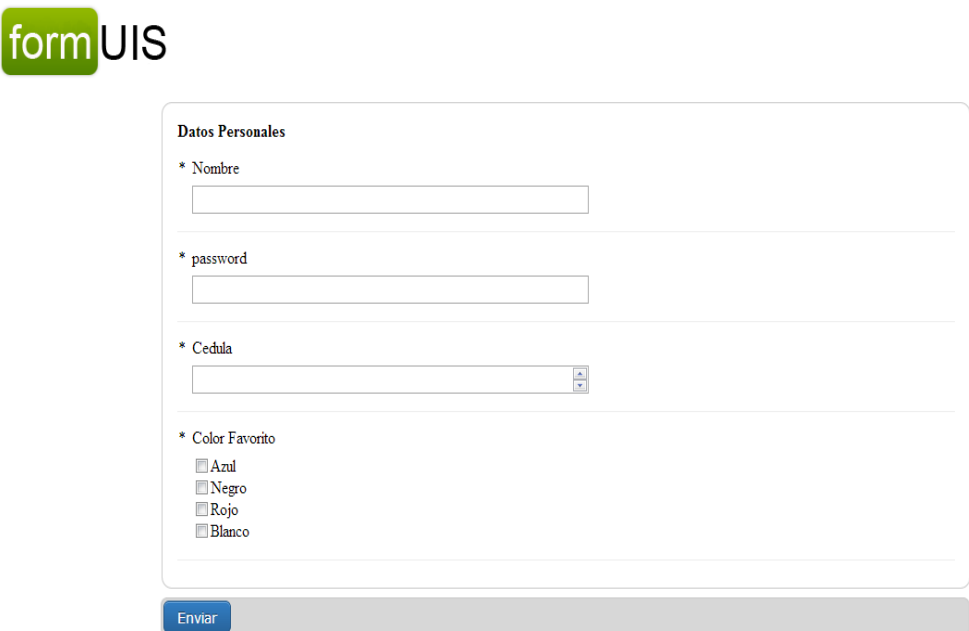
Figura 18. Ventana crear formulario



## Ventana visualizar formulario

La ventana visualizar formulario muestra un formulario creado por el usuario con el fin de recolectar la información deseada.

Figura 19. Visualizar formulario



The image shows a web interface for 'formUIS'. The logo 'formUIS' is displayed in the top left, with 'form' in a green box and 'UIS' in black. Below the logo is a form titled 'Datos Personales'. The form contains four sections, each with a required field marked with an asterisk (\*):

- \* Nombre**: A text input field.
- \* password**: A text input field.
- \* Cedula**: A text input field with a dropdown arrow on the right side.
- \* Color Favorito**: A list of radio buttons with the following options:
  - Azul
  - Negro
  - Rojo
  - Blanco

At the bottom of the form is a blue button labeled 'Enviar'.

## Ventana Administrador de Formularios

La ventana de administración de formularios muestra todos los formularios creados por un usuario y las acciones permitidas para cada uno de ellos.

Figura 20. Administrador de formularios.

formUIS

Perfil Formularios Ayuda

Administrador de formularios

Nuevo Formulario

Mis Formularios			
finca raiz	Ver	Entradas	Eliminar
Formulario	Ver	Entradas	Eliminar
animales	Ver	Entradas	Eliminar
universidad	Ver	Entradas	Eliminar
Datos Personales	Ver	Entradas	Eliminar

Creado por: Oscar Larrota | Jorge Gonzalez

## Ventana visualización de la información

La ventana de visualización de la información muestra los datos que los usuarios ingresaron en el formulario.

**Figura 21.** Visualizar información



The screenshot displays the 'formUIS' application interface. At the top left is the logo 'formUIS'. On the right, there are navigation tabs for 'Perfil', 'Formularios', and 'Ayuda'. The main content area is titled 'Datos del formulario' and contains a table with the following data:

universidad	
carrera	uis
codigo	919299
clave	123456
carrera	udes
codigo	2051703
clave	123456

Below the table is a blue 'Descargar' button. At the bottom of the application window, it says 'Creado por: Oscar Larrota | Jorge Gonzalez'. The browser's download bar at the very bottom shows a file named 'universidad (1).bt' and a link to 'Mostrar todas las descargas...'.

## 8. CONCLUSIONES

- La sistematización y descentralización de la captura de datos optimiza el proceso del manejo de la información.
- Al tener acceso a la información desde cualquier parte en tiempo real permite tener un mayor control y aprovechamiento de la misma.
- El diseño de una interfaz dinámica y de fácil interacción permite al usuario mayor destreza al momento de utilizar la aplicación, y así aprovechar todas las características que brinda el software.
- La integridad y confiabilidad de un sistema depende del manejo que se le da a la información durante el proceso de almacenamiento y extracción en la base de datos.

## 9. RECOMENDACIONES

- La escuela ingeniería de sistemas debe seguir fomentando y fortalecimiento el proceso de formación en programación en diversos lenguajes, con el fin que los estudiantes pierdan la apatía que tienen por programar.
- A medida que sea probada la efectividad del proyecto se podrá realizar una integración con algunas dependencias de la universidad y así mejorar la forma de recolección de datos.
- Continuar con el desarrollo de otras versiones del software y la implementación de nuevos campos que no estén contenidos en el prototipo.

## BIBLIOGRAFIA

- [8] Archivo de texto. Fecha de consulta, junio de 2013. Disponible en internet:  
[http://es.wikipedia.org/wiki/Archivo\\_de\\_texto](http://es.wikipedia.org/wiki/Archivo_de_texto)
- [6] Base de datos. Fecha de consulta, junio de 2013. Disponible en internet:  
[http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos)
- [7] Css. Fecha de consulta, junio de 2013. Disponible en internet:  
<http://es.wikipedia.org/wiki/Css>
- [9] Diagrama de actividades. Fecha de consulta, junio de 2013. Disponible en internet: <http://www.slideshare.net/camiloan40/diagrama-de-actividades-uml>
- [1] Html. Fecha de consulta, Junio del 2013. Disponible en internet:  
<https://belenus.unirioja.es/~guprado/pagweb/carachtml.html>
- [2] Html5. Fecha de consulta, junio de 2013. Disponible en internet:  
<http://es.wikipedia.org/wiki/HTML5>
- [4] JavaScript. Fecha de consulta, junio de 2013. Disponible en internet:  
<http://es.wikipedia.org/wiki/JavaScript>
- [5] JQuery. Fecha de consulta, junio de 2013. Disponible en internet:  
<http://librojquery.com/> [3] Php. Fecha de consulta, junio de 2013. Disponible en internet: <http://php.net/manual/es/index.php>