Analisis y Diseño de un Sistema de Informacion para Glasswing Nicaragua



**Brayan Alexander Gaitán Montenegro**

**ANÁLISIS Y DISEÑO DE UN SISTEMA DE INFORMACIÓN PARA GLASSWING INTERNACIONAL EN NICARAGUA**

Managua, Nicaragua

Ing. Ing. Rene ManSell

Managua, Nicaragua

noviembre de 2016

Monografía para optar al título de

**INGENIERO EN SISTEMAS**

Dedicatorias

Agradecimientos

Índice

Índice de tablas y figuras



1. Introducción

Con el avance de la tecnología las organizaciones se ven cada vez más en la necesidad de adaptarse a las nuevas tecnologías que vienen apareciendo, para así mantenerse en el mercado o en competencia; esto no solo aplica a organismos con fines de lucro, sino a organizaciones no gubernamentales y que trabajan en la ejecución de proyectos sociales. Ambos tipos de entidades requieren aplicar estrategias gerenciales para poder administrar mejor sus recursos y ejecutar sus planes que les permitan alcanzar el éxito y la satisfacción que desean.

La administración de la información es elemento clave para el funcionamiento de las organizaciones ya que, en base a la cantidad de información, lo estratificada que esta esté, y otras condiciones ayudan a la gerencia en la toma de decisiones, así mismo a tener un mejor manejo de los recursos.

Las ONGs se han visto en la necesidad de implementar software para el manejo de información. Existen ONG’s que desconocen las ventajas competitivas que brindan estas herramientas. Como toda empresa en red, las ONGs no son la excepción para quedarse al margen de la utilización de las herramientas tecnológicas, como es el caso particular del uso de una herramienta de administración de sistemas, ya que generan mucha información en base a sus actividades y personas involucradas en tales. A través del procesamiento de esta información es posible que este tipo de instituciones puedan fijar directrices estratégicas al fin de optimizar recursos y planear actividades para sus beneficiarios.

Comúnmente, Las ONGs como Glasswing trabajan con donaciones de empresas que desean desarrollar proyectos de responsabilidad empresarial, o simplemente desean realizar obras de caridad, como toda ONG, tiene que responder de los gastos que se realizan, logro de los proyectos, personas involucradas, beneficiados, indicadores, objetivos cometidos, entre otras cosas a dichas empresas. Glasswing requiere de un sistema de Información que le permita manejar datos de manera centralizada, y a la vez le permita, realizar control estadístico y seguimiento de proyectos.

Por razones como estas me eh interesado en desarrollar un Análisis y Diseño de un sistema de información para la gestión, administración y control estadístico de los datos en los proyectos para apoyar a Glasswing en la toma de decisiones gerenciales en sus actividades, actual ONG, de la que fui parte como voluntario, y cuyos dirigentes locales de la institución me mostraron gran apoyo e interés sobre el tema.

En este documento, se exponen los principales conceptos relacionados al tema, además se describen las herramientas a utilizar para la obtención y procesamiento de resultados. Este estudio se divide de la siguiente manera, Introducción, Objetivos, Marco Teórico, Planteamiento del Problema, Diseño metodológico, Resultados (desglosado por cada objetivo específico), Conclusiones y Recomendaciones; por último, se añaden las referencias bibliográficas, citas, glosario y anexos.

1. Objetivos
2. Objetivo General

Diseñar una herramienta tecnológica de carácter web, para el manejo de información administrativa y de control operativo de los proyectos, así como controles estadísticos presentados en una dinámica de reportaría, que permita a Glasswing la administración efectiva, y oportuna de la información y datos para optimizar controles y estadísticas de los diferentes proyectos.

1. Objetivos Específicos
2. Realizar un análisis de requerimientos y requisitos para determinar el desarrollo de un Software de un Sistema de información para Glasswing Nicaragua.
3. Presentar un informe de Factibilidad y de Viabilidad para una futura decisión por parte de Glasswing, de Desarrollar e Implementar el Sistema completamente.
4. Diseñar el sistema y procesos de entrada - salida de datos y de procesos negocios en UML y BPMN con el propósito de brindar una documentación para el desarrollo e implementación en la organización de un sistema de información.
5. Elaborar el modelo de las bases de datos relacionales que se acomoden a lo recolectado durante el análisis de requisitos y requerimientos, que permitirá manejar la información centralizada y un control seguro de usuarios.
6. Elaborar y Diseñar los prototipos de la interfaz visual de una herramienta que permita a Glasswing monitorear y evaluar los proyectos ejecutados, además de poder, en base a los requisitos y requerimientos que la gerencia de Glasswing brinde.
7. Marco Teórico

Durante esta capitulo se busca definir y detallar todos los conceptos involucrados durante el desarrollo de este proyecto y que proporcionen base a la parte metodológica de la presente monografía.

Este marco teórico llevara la siguiente estructura:

1. Arquitectura de Software

2. Ingeniería en Software

3. Análisis de requerimientos y requisitos

4. Técnicas utilizadas en la Ingeniería de requerimientos

5. Ciclo de vida de un proyecto de software

6. Factibilidad y Viabilidad

7. Lenguaje Unificado de Modelado- UML

8. Notación de modelado de Procesos de negocios- BPMN

9. Programación Orientado a Objetos -POO

10. Análisis y Diseño Orientado a Objetos -ADOO

1. Arquitectura de Software
   1. Sistema de Información

Un sistema de información técnicamente es un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización (1).

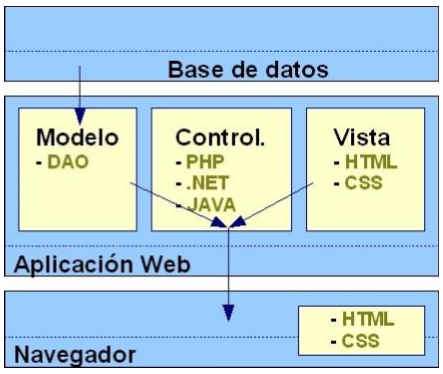
Existen varios tipos de Sistemas de Información, el seleccionado a diseñar es un sistema del tipo MIS.

* + 1. Sistemas de Información para la Administración o Gerenciales (MIS - Management Information Systems)

Se llama MIS, al conjunto organizado de personas, procedimientos, software, bases de datos y dispositivos para suministrar la información rutinaria a administradores y tomadores de decisiones (2).

Los MIS proporcionan informes periódicos para planeación, control y toma de decisiones gerenciales. Cumplen tres funciones principales: recopilar datos, almacenarlos y procesarlos, transmitir información.

Debido a las necesidades de Glasswing, se seleccionó este tipo de sistema de información. Glasswing desea un sistema que almacene datos, y que genere reportes por periodos de tiempo para poder ejecutar acciones en base a los resultados obtenidos.

* 1. Diseño Web
     1. MVC

Según el artículo de la Universidad de Alicante, Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

* El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
* La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
* El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (3).
  + 1. PHP

Php (*PHP Pre Hypertext -processor*), es un lenguaje de programación de código abierto, utilizado para la creación de aplicaciones y desarrollo Web dinámico (Petición --> Procesado y preparación --> Respuesta), el cual es posible ser incrustado en HTML. Fue creado en 1995 por Rasmus Lerdorf; la actual versión del lenguaje se encuentra en PHP 7.0.

* + - 1. CakePHP

CakePHP es un framework creado en código Php, para el desarrollo de aplicaciones web, pero el cual fue creado en los conceptos de R*uby on Rails,* se inició a desarrollar en 2005, y actualmente se encuentra en la versión 3.0 y es distribuido bajo la licencia MIT.

Este framework se seleccionó para el desarrollo del Sistema, debido a la flexibilidad de desarrollo que brinda, además de que se encuentra en constante desarrollo, por una comunidad Open Source y por su base en ser un framework basado en MVC.

* + 1. Servidor Local
       1. WAPP

Acrónimo de Windows + Apache + PostgreSQL, es una herramienta OpenSource, creado por el grupo Bitnami. WAPP provee de un ambiente de desarrollo para trabajar aplicaciones web desarrolladas con PHP y PostgreSQL.

* + - 1. XAMPP

Considerada una herramienta de software libre, para generar un servidor local, el cual contiene principal mente módulos de servidor web Apache, gestor de base de datos MySql/MariaDB, y lenguajes de script Perl y PHP. Esta herramienta nos será de utilidad para la ejecución y desarrollo del sistema de manera local. El Acrónimo significa X (Cualquier Sistema) + Apache+ Mysql+ PHP+ Perl.

* + - 1. FileZilla

Es una herramienta Cliente FTP (*File Transference Protocol*) de Software Libre y código abierto, utilizado para la transferencia de archivos multimedia. Esta herramienta será utilizada para sincronizar los archivos del proyecto con el servidor o hosting web.

* 1. Licenciamiento
     1. GNU-GPL

Según la página oficial de GNU “Llamada comúnmente GPL de GNU, se usa para la mayoría de los programas de GNU y para más de la mitad de los paquetes de software libre. La última es la versión 3”, asegura que el software está disponible de manera libre, en otras palabras, que pueda ser distribuido libremente, copiado, distribuido, vendido o modificado por cualquier usuario siempre y cuando se mantenga el mismo tipo de licencia (4). Bajo este tipo de licencia se desarrollaría el Sistema de Información de Glasswing Nicaragua.

* + 1. Creative Commons (CC)

El significado que esta licencia es que “algunos derechos están reservados”, esta licencia es utilizada para distribuir Documentos con carácter educativo y de manera libre. Sería la licencia utilizada para la documentación producida relacionada a los diagramas y prototipos; basado en la información de niveles de licencias obtenida del articulo *Licencias,* de *Creative Commons*-España (5). Bajo las opciones brindadas por el selector de licencia de la página, se determinó la **de Atribución-NoComercial-SinDerivar 4.0 Internacional.**

* 1. Control de Versiones

El control de versiones es un sistema que guarda o registra los cambios producidos a archivos durante su periodo de desarrollo o evolución.

* + 1. Git

Es un software de control de versiones que fue creado por el mismo que creo Linux, Linus Tolvards, basado en conceptos de confiabilidad y eficiencia, para el mantenimiento de proyectos informáticos, cuando existe un gran número de archivos fuentes.

* + 1. Github

Github, es una plataforma de control de versiones, de tipo de desarrollo colaborativo de software, que permite alojar proyectos con el tipo de control de versiones Git; a la vez brinda herramientas para trabajar en equipo el desarrollo del proyecto.

* + 1. 9cloud

Es una herramienta IDE (*Integrated Development Environment*) en línea, bajo la distribución de licencia GNU- GPL v3, que soporta actualmente numerosos lenguajes de programación, permite cargar proyectos contenido en ciertos repositorios (sincronizar) como Github, así mismo faculta a los desarrolladores trabajar, codificar o programar y realizar pruebas de proyectos de software en espacios de trabajos ya pre-configurados. Esta herramienta será muy útil a la hora de corregir errores y poder probar el sistema antes de ponerlo en el espacio de dominio.

* 1. Base de datos SQL
     1. ¿Qué es un gestor de base de datos?

Los sistemas gestores de bases de datos (SGBD) o *Database Management System* (DBMS), son un conjunto de programas interrelacionados (Software), especifico cuyo objetivo es servir de interfaz que permita el almacenamiento, modificación, extracción y manipulación de información en una Base de datos por parte de un usuario, desde un lenguaje de manipulación de datos y de un lenguaje de consulta.

* + 1. Base de datos SQL

Son aquellos tipos de base de datos que utilizan el modelo Relacional (tablas), que poseen la característica de que cada tabla es compuesta por un conjunto de campos (columnas) y registros (filas), la relación entre tablas padre e hijos se da por medio de llaves foráneas y primarias, además que no pueden existir dos tablas con el mismo nombre ni registro. Para el manejo y creación de los diagramas de la base de datos se utilizó Navicat Premiun 11.2.10.

* + - 1. MariaDB

Es un gestor de base de datos SQL basado en Mysql 5.1, por lo cual mantiene compatibilidad con base de datos Mysql, es desarrollada como software libre, bajo la licencia GPL. Su actual versión es la 10.0. Es el Gestor de base de datos a ser utilizado para el Sistema a Crearse, debido a su compatibilidad con el lenguaje PHP y por lo tanto con el Framework CakePHP.

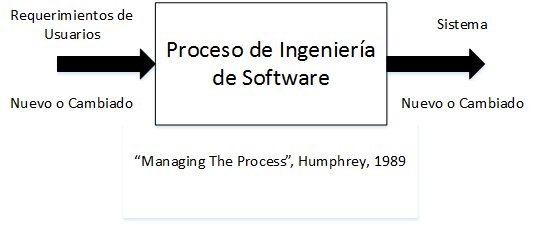
* + - 1. PostgreSQL

En un ORDBMS cuya versión actual es la 9.6.1, basado en el proyecto POSTGRES, de la universidad de Berkeley, es un gestor de base de datos OpenSource, a la altura de muchos gestores comerciales, con grandes características que lo hacen un gestor fiable, escalable seguro, robusto y potente, pero referente a velocidad de respuesta, puede existir un hincapié, debido a que puede lucir deficiente referente a base de datos pequeñas, pero mantiene esa misma velocidad para DB que son mucho más grande. Es desarrollado bajo la licencia PostgreSQL, la cual es similar a la MIT y la BSD.

1. Ingeniería en Software

El software hoy en día juega un papel en todas las empresas, el software es el encargado de manejar el producto más importante para todas las organizaciones, como es la “*información*”. La ingeniería de software es la rama disciplinaria de la informática que crea, desarrolla y mantiene las aplicaciones de software usando tecnológicas, y un conjunto de herramientas, técnicas, prácticas y métodos de las ciencias de la computación.

En el glosario de la IEEE (Stad. 610.12-1990), Ingeniería en software se define como “*la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software”*. Por otro lado, SEI, da este concepto:” *Esa forma de ingeniería que aplica los principios de la informática y las matemáticas para conseguir soluciones rentables a problemas de software*”.

* 1. Proceso de Ingeniería en software

Al “conjunto de etapas parciamente ordenas con la intención de lograr un objetivo, en este caso, un software de calidad”, se le conoce como proceso de ingeniería en software. Por otro lado, el proceso de desarrollo o ciclo de desarrollo “es el proceso, o ciclo, en que los requerimientos del usuario son traducidos en diseño, el cual es implementado en código, que es probado documentado, y certificado para su uso final”.

1. Análisis de Requerimientos-requisitos

Al realizar la toma de requisitos y requerimientos, es necesario entender que, al hablar requisitos, referimos a las condiciones, funcionabilidades, características y restricciones que el software tiene que cumplir para su correcto funcionamiento; por otro lado, requerimientos son las necesidades y deseos pedidos por las personas y clientes a quienes se les desarrollara el Software o sistema.

Se define como análisis de requisitos o de requerimientos al conjunto de técnicas y procedimientos que nos permitan conocer los elementos básicos para la planificación de un proyecto de desarrollo de software. El análisis de requerimientos busca capturar y detallar de manera específica los requerimientos de funcionabilidad y de calidad del servicio del producto que se va a desarrollar, desde el punto de vista del cliente y los usuarios que utilizaran dicho sistema; en cambio en el análisis de requisitos buscamos describir, analizar y verificar los requisitos que el software debe tener. A través de la descripción que hace J.M Drake en Programación: Análisis de requisitos (Stander,2008), formulamos nuestro proceso de tomar de requisitos y requerimientos:

1. Técnicas utilizadas en la recolección de requisitos y requerimientos
   1. Entrevistas y Cuestionarios

Bajo el estilo de una conversación la técnica de **entrevista** busca recolectar información de los clientes con una comunicación intrapersonal.

El otro aspecto a mencionar es el uso del **cuestionario,** permite a los recolectores de requerimientos reunir mucha información a partir de un grupo grande de personas. Para obtener mejores resultados, los encuestados debe ser seleccionados en base a los conocimientos y experiencia necesaria que sirva de ayuda para el avance del proyecto. En anexos encontramos las preguntas realizadas al equipo de Glasswing.

* 1. Prototipos

En el caso de querer mostrar una visualización del proyecto al cliente, para evitar volver atrás en cuestiones de diseño de interfaces, existen varias herramientas de comunicaciones estas y los desarrolladores y programadores.

* + 1. Wireframe

Los Wireframes son básicamente el esqueleto de la interfaz de un software, pagina o aplicación web. Describen detalles como posición de botones, vistas y cantidad de funciones que estas podrían tener o que pueden ser modificadas. No muestran ningún diseño. La finalidad de un wireframe es mostrar al cliente un diseño o boceto rápido y facilitar la comunicación entre cliente y desarrollador, permitiendo realizar cambios si son necesarios, previniendo que estos “cambios” se lleguen a realizar ya con el software desarrollado o en etapa de desarrollo. Esta herramienta de bosquejo es cuando el diseño este algo crudo. La herramienta Utilizada es Balsamiq Mockups 3.5.1.

* + 1. Mockups

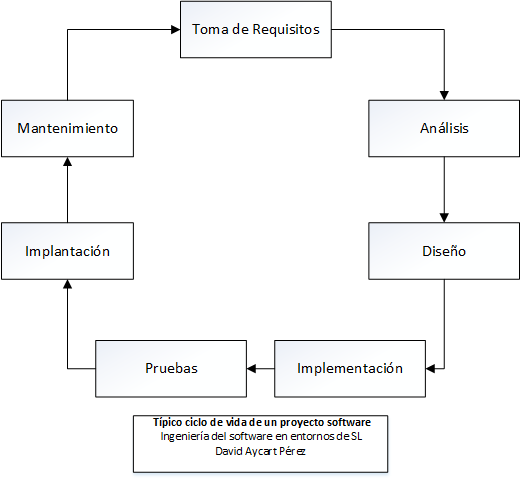
Esta herramienta de bosquejo se utiliza cuando se tiene un diseño visual más refinado. Los mockups son diseños visuales más detallados de las interfaces de un software, pagina o aplicación web, por lo tanto, más fieles (un nivel medio) a las tareas que el producto final debe ejecutar (6). La herramienta utilizada es Justinmind Prototyper 7.5.0.

* 1. Casos de Uso

Cuando hablamos de los casos de uso, hablamos de un recurso grafico de UML que muestra las cadenas de interacciones entre los actores del negocio y el sistema que se desea crear, en el cual satisfacemos gráficamente una descripción de la funcionalidad y las características de calidad de servicio de sistema

1. Ciclo de vida de un Proyecto de Software

Al conjuntode fases que pasa un proyecto de software desde su concepción es conocido como Ciclo de Vida software, según Aycart en Ingeniería del software en entornos de SL2007 y abarca o incluyen las actividades o fases ahora mencionadas:

* Toma de requisitos y requerimientos
* Análisis
* Diseño
* Desarrollo
* Pruebas (Validación, aseguramiento, de la calidad)
* Instalación (implantación)
* Uso
* Mantenimiento
* Obsolescencia

De los pasos anteriormente expuestos, se buscarán ejecutar los tres primeros, y el desarrollo de un módulo del software. Existen distintos modelos de para su ejecución, lineal, cascada, evolutivo, espiral, concurrente, entre otros. El seleccionado para el desarrollo de este trabajo es el Modelo de cascada.

1. Factibilidad y Viabilidad
   1. Estudio de Factibilidad

Permite la recopilación de datos relevantes para continuar o no un proyecto, si es bueno o malo; y con qué condiciones se cumplen o deben contarse para que sea exitoso y seguir en el estudio, desarrollo e implementación del mismo.

Un estudio de factibilidad abarca diversas áreas: técnicas, financiera, operacional, geográfica, tiempo, recursos, legal y política. En el presente documento abarcaremos solo 4, descritas a continuación:

* 1. Estudio de Viabilidad

La realización de un análisis de viabilidad intenta predecir en base a un objetivo, el eventual éxito de un proyecto, no importa en qué campo se desea implementar, sea en lo educativo o en lo tecnológico, a nivel gubernamental o corporativo. Este es ejecutado luego de demostrar que un proyecto es primeramente factible y muestra como el proyecto pretende beneficiar a los procesos del negocio. El estudio de viabilidad debe mostrar si en verdad vale la pena continuar con el levantado de requerimientos y el resto del proceso de desarrollo.

En el estudio de viabilidad es necesario evaluar cuatro partes (Técnica, económica, operacional, y legal), de las cuales abordaremos:

La información del estudio de viabilidad se puede recopilar durante entrevistas, esta consiste en brindar suficiente cantidad de datos para que la gerencia tenga suficientes elementos para proceder a un siguiente paso. Al fin y al cabo, el estudio de Viabilidad debe dar una respuesta a varias interrogantes: ¿Contribuye el sistema a desarrollarse con los objetivos de la empresa?, ¿se pude desarrollar el sistema dentro del tiempo y presupuesto con que cuenta la empresa? ¿Es posible unificar o poner a trabajar el sistema a desarrollar con los sistemas actuales con que cuenta la empresa? (7).

1. Lenguaje de modelado Unificado-UML

UML, de *Unified Modeling Lenguage* (Lenguaje Unificado de modelado), Es el lenguaje utilizado para la modelación de sistemas de software, nacido en la década de los noventa, más específicamente en 1997. UML cubre la necesidad como lenguaje grafico para visualizar, especificar, construir y documentar planos de software/sistema, así como sucede en otras áreas como construcción, mecánica o en la industria aeroespacial.

UML tiene un objetivo principal “Proporcionar a los arquitectos en sistemas, ingenieros de software y desarrolladores de software, herramientas para el análisis, diseño e implementación de sistemas basados en software, así como para el modelado de negocios y procesos similares.” (8). Para el diseño de los diagramas UML se utilizará las herramientas WhiteStartUML 5.8.1.0, Ms Visio 2016 e IBM Rational Rose 2007.

* 1. Clasificación de los diagramas-UML

En UML 2.0 los tipos de diagramas están agrupados en tres categorías: Estructurales (6), de Comportamiento (3), y de interacción (4).

|  |  |
| --- | --- |
| Las estructuras de diagramas | Diagrama de clase, de objeto, de componentes, de estructura compuesta, de paquetes, de despliegue/ implementación. |
| Los diagramas de Comportamiento | Diagramas de casos de uso, de actividad, de estado. |
| Los diagramas de interacción | Diagrama de secuencia, de colaboración, de tiempo, y de interacción. Estos se derivan del grupo anterior. |

* 1. Elementos estructurales de UML

Representan cosas conceptuales o materiales, partes estáticas.

|  |  |  |
| --- | --- | --- |
|  | Herramienta | Representación |
| 1 | **Clases**: es la descripción de un objeto que posee atributos, operaciones, relaciones y semántica. |  |
| 2 | **Interfaz:** colección de operaciones de un servicio de una clase o componente determinado. Describen el comportamiento visible externamente del objeto, mas no describe la implementación. |  |
| 3 | **Colaboración**: una interacción, y es un conjunto de roles y otros elementos, representan una implementación de patrones en forma de sistema. |  |
| 4 | **Caso de uso:** la descripción del conjunto de acciones que un usuario con el sistema. |  |
| 5 | **Clase activa:** una clase con uno o más procesos en hilos de ejecución, por lo cual pueden dar lugar a actividades de control. |  |
| 6 | **Componentes**: parte física y reemplazable de un sistema. |  |
| 7 | **Nodos:** representa un recurso físico computacional que existe en un tiempo de ejecución, casi siempre disponen de algo de memoria, y usualmente con capacidad de procesamiento. Es representado por un cubo. (Desarrollo Orientado a objetos con UML- Raúl Alarcón, 2000). |  |

* 1. Elementos de Comportamiento

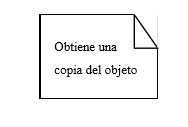
Son los elementos dinámicos de UML, que cumplen con el papel de expresión en tiempo y espacio.

|  |  |  |
| --- | --- | --- |
|  |  | Representación |
| 1 | **Interacción**: comprende un intercambio de mensajes o acciones entre un conjunto de objetos, para lograr un objetivo específico. |  |
| 2 | **Máquinas de estados**: En este comportamiento se especifica el orden de estados en que deben pasar objetos o interacciones durante se dé respuesta a estos eventos. |  |

* 1. Elementos de Agrupación

Son encargados de la manera organizativa de los modelos UML.

**Paquetes:** su existencia es presente durante el desarrollo, es puramente conceptual.

* 1. Elementos de anotación 

Son los encargados de guardar alguna anotación explicativa en UML, normalmente comentarios que describen algo importante.

**Nota**: muestra anotaciones, restricciones y comentarios.

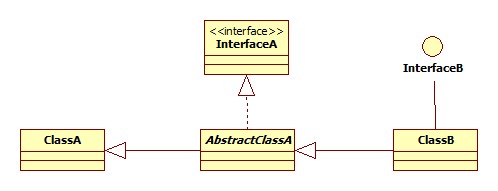
* 1. Relaciones

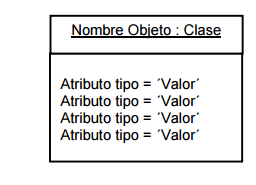
Existen 4 tipos en UML.

|  |  |  |
| --- | --- | --- |
|  |  | Representación |
| 1 | **Dependencia**: Muestra que un elemento depende de otro. Se representa por una línea discontinua. |  |
| 2 | **Asociación**: Muestra que dos elementos están asociados, pero que el primero requiere de cierta información del segundo elemento. |  |
| 3 | **Generalización**: aplica en si propiedades de Programación orientada a objetos (en especial al ser usadas en diagramas de clase), como es la herencia. |  |
| 4 | **Realización**: Alarcón lo define de esta manera, una relación semántica entre clasificadores, donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. |  |

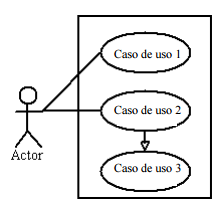
* 1. Diagramas

La finalidad de los diagramas es mostrar distintos puntos de vista o perspectivas del sistema, describen lo que los clientes desean que el sistema haga, y la forma en que los desarrolladores pueden entenderlo y graficarlo, sin decir el cómo se hará.

1. **De Clases:** son diagramas estáticos del sistema, que comúnmente describen interfaces, colaboraciones, además de las relaciones entre las clases existentes en el sistema, los atributos y operaciones o métodos de cada uno, similar al de entidad-relación de base de datos.



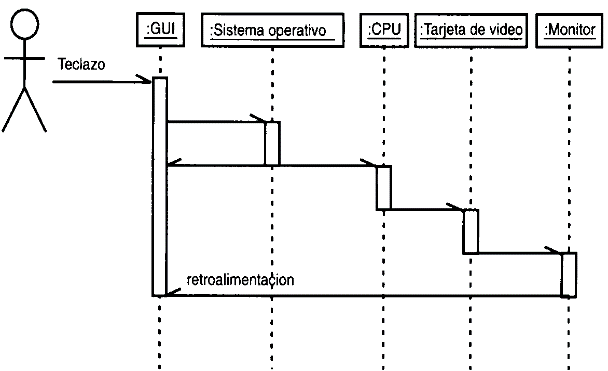
1. **De objetos**. Describen la estructura estática de un sistema en un momento peculiar. Muestran un conjunto de objetos y sus relaciones, ejemplifican diagramas de clases complejos, demostrando como se verían instancias actuales y sus relaciones, demostrando así la precisión de los diagramas de Clases.

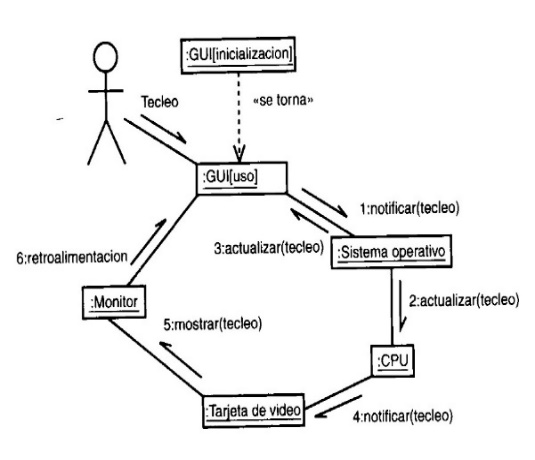


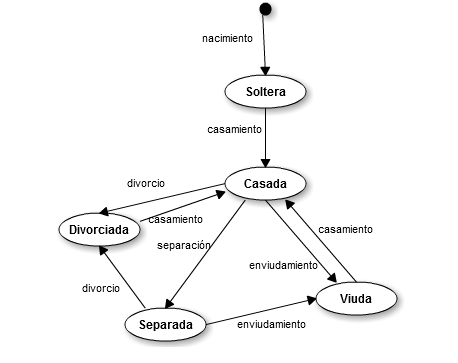
1. **De casos de Usos**: Muestran el conjunto de casos de uso (servicios y/o funciones) y actores participantes del sistema y sus relaciones.

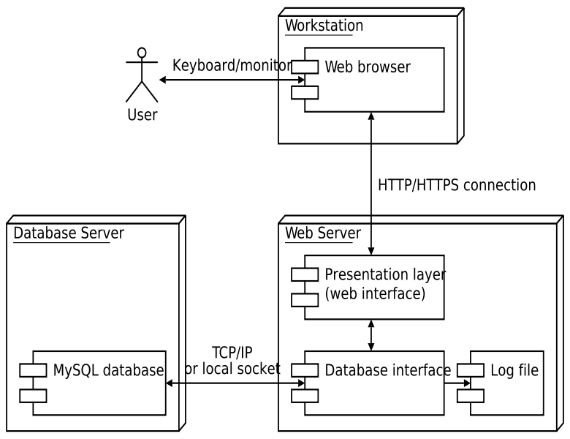
Los casos de uso describen las acciones del sistema desde el punto de vista del usuario. A través de ellos se modela la funcionabilidad del sistema.

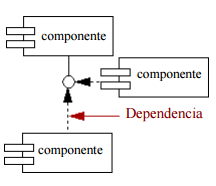
1. **De Secuencia & colaboración**: Son un conjunto de diagramas de interacción, a la vez cubren la parte dinámica del sistema.



Los diagramas de *secuencia* muestran la interacción entre objetos con base a tiempos (de manera temporal) y orden de las secuencias (que va primero, que va después…).

Los diagramas de *colaboración* muestran la organización estructural de los objetos que envían y reciben los mensajes (interacciones). Representan una combinación entre diagramas de clases, casos de uso, y de secuencias, describiendo comportamiento tanto de la estructura estática como de la dinámica.

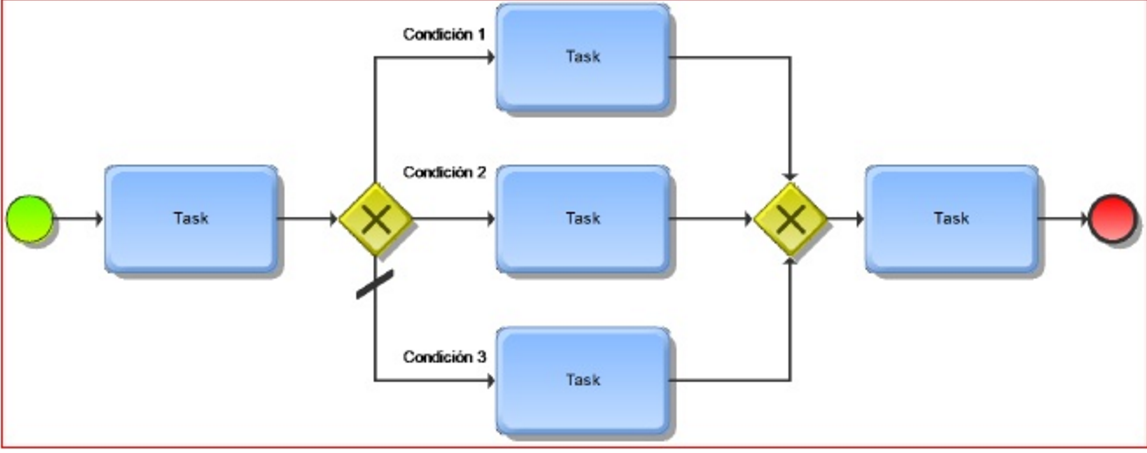
1. **De estados:** exponen una máquina de estados compuesta por estados, transiciones, eventos y actividades; típicamente son el complemento de una clase.
2. **De actividades:** Se centran en mostrar el flujo de actividades dentro del sistema, cubren la parte dinámica de un sistema, mediante el modelado del flujo ocurrente de manera secuencia de actividad en actividad.
3. **De Despliegue/ distribución:** Muestran la arquitectura de hardware y el software del sistema y la configuración de los nodos (que contienen uno o más componentes). Puede representar a los equipos, dispositivos, interconexiones y el software por cada máquina.



1. **De componentes:** describen la organización y dependencias de los componentes físicos de un sistema, los cuales suelen componerse por una o más clases, interfaces o colaboraciones, por lo cual tienen mucha relación con los diagramas de clase.

1. Notación de Modelado de procesos de Negocio -BPMN

BPMN, de *Business Process Modeling Notation* (en español Notación de modelado de procesos de negocio), Es un estándar y notación Internacional grafico que describe de manera lógica, procesos de negocios y servicios web. Este estándar buscar agrupar la planificación y gestión del flujo de trabajo, así como el modelado y la arquitectura, proporcionando a los actores del proceso, un lenguaje común para la especificación de forma clara, completa y eficiente el detalle de un proceso. La actual versión (2.0) es avalada por la OMG. Se utilizará la herramienta llamada *BonitaBPMN* para la creación de los diagramas.



En el manual de BPMN 2.0 encontramos que “BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente (…) BPD es un diagrama diseñado para representar gráficamente la secuencia de todas las actividades que ocurren durante un proceso, basado en la técnica de “*Flow Chart*”, incluye además toda la información que se considera necesaria para el análisis. BPD es un diagrama diseñado para ser usado por los analistas, quienes diseñan, controlan y gestionan procesos.” (9). Para la diagramación de procesos BPMN utilizaremos el software Bonita BPMN.

* 1. Niveles de Modelado de Proceso

En el modelado de BPMN, se distinguen 3 niveles:

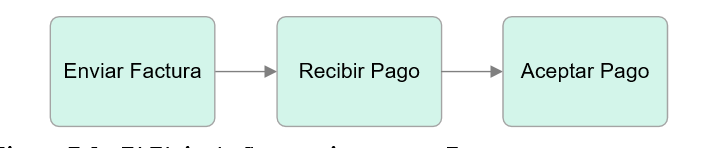
* 1. Elementos de BPMN

Se encuentran clasificados en 4 categorías:

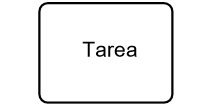
* + 1. Objetos de flujo

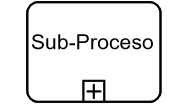
Son los principales elementos que definen los comportamientos de los procesos. Existen 3 objetos de flujo:

* **Eventos:** una acción o algo que debe ocurrir durante el curso de un proceso, este afecta al flujo del proceso, comúnmente posee una causa y un resultado. Encontramos 3 tipos de eventos: De Inicio, Intermedios, De Fin.



* **Actividades**: representan el trabajo o labor ejecutada dentro de un proceso de negocio, estas pueden ser y a la vez no compuestas o atómicas. Hallamos dos tipos de actividades tareas y subprocesos.



* **Tarea**: una actividad atómica, es incluida dentro de un proceso. Las tareas pueden ser: simples, manuales, automáticas, de usuario, recibido, servicio y envió.
* **Subproceso:** conjunto de actividades incluidas dentro de un proceso. Pueden ser: embebidas, reusables y transaccionales.
* **Compuertas**: llamadas también *Gateway* estos elementos del modelado son usados para controlar divergencia y convergencia del flujo. Existen varios tipos de compuertas: Exclusiva, Basada en eventos, Paralela, Inclusiva y Compleja.
  + 1. Objetos de Conexión

Estos elementos son utilizados para conectar objetos del flujo dentro de un proceso, y definen el orden de ejecución de las actividades. Existen tres tipos de conectores.

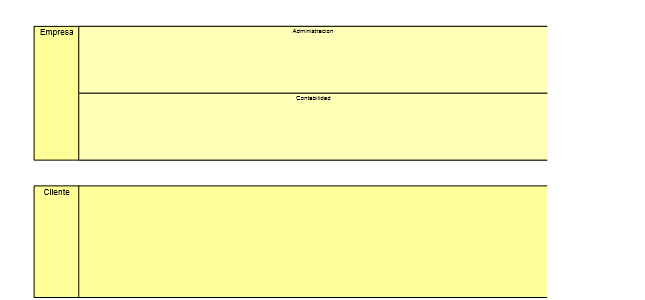
**Líneas de Secuencia:** Muestran el orden de los eventos y decisiones que se realizan dentro del proceso.

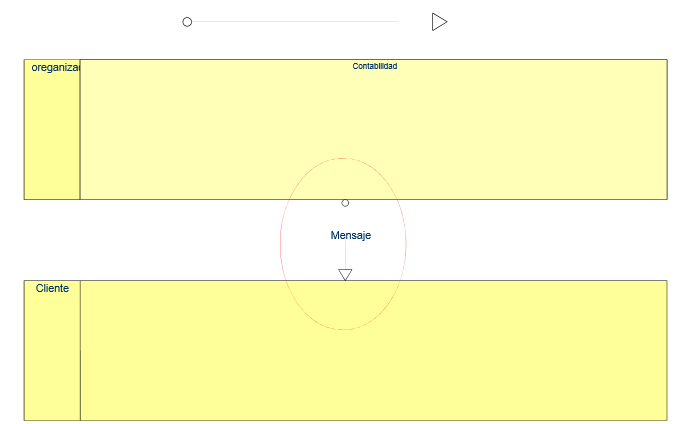
**Líneas de Mensaje:** Indican flujo de mensaje entre diferentes objetos a los artefactos de los procesos.

Asociación: Asocia diferentes objetos con otros

* + 1. Canales

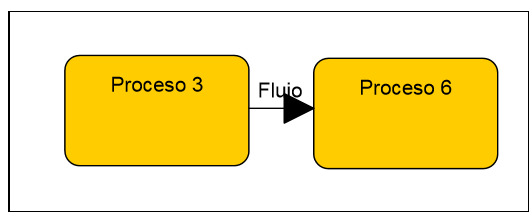
Estos elementos son los medios utilizados para organizar actividades del flujo en diferentes categorías separadas visualmente: áreas funcionales, roles o responsabilidades.

 **Pools**: Los actores externos con los cuales interactúa un proceso.



**Lanes**: Representan participantes dentro del proceso.

* + 1. Artefactos

Estos objetos gráficos muestran información extra sobre elementos dentro del proceso sin afectar el flujo de información. Existen 3 tipos: Objetos de datos, Grupos y Anotaciones.

1. Programación Orientada a Objetos -POO

Es el estilo de programación, o técnica orientada a existencia de objetos, para desarrollar soluciones computaciones, los cuales posee propiedades, métodos y atributos, además de otras propiedades; el objeto de la POO es reutilizar código en vez de volver a crear o escribir.

En el artículo Fundamentos de la POO, se señala que POO:

“La POO es un paradigma de la programación de computadores; esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.

La POO se basa en el modelo objeto donde el elemento principal es el objeto, el cual es una unidad que contiene todas sus características y comportamientos en sí misma, lo cual lo hace como un todo independiente pero que se interrelaciona con objetos de su misma clase o de otras clases, como sucede en el mundo real” (8). Existen tres conceptos fundamentales dentro de la POO.

* 1. Objetos

Personas, lugar o cosas relevantes para la creación de un sistema, ejemplo pueden ser usuarios, clientes, artículos, pedidos, etc.; también pueden ser interfaces de un software.

* 1. Clases

Es la forma de agrupar y representar objetos, describiendo atributos (propiedad de un objeto) y métodos involucrados.

* 1. Herencia

Se refiere acerca de la creación de clases a partir de otras, donde las hijas reciben características o atributos de la clase padre.

1. Análisis y Diseño Orientado a Objetos -ADOO
   1. Análisis

En el caso de Análisis Orientado a objetos (AOO) se desarrollan modelos que describan el software a ser desarrollado en base a los requisitos definidos por el cliente y los usuarios, abarcando así el funcionamiento y comportamiento. En AOO se definen clases, atributos, operaciones y relaciones de comportamientos asociados a los problemas a resolver.

* 1. Diseño

En el caso de Diseño orientado a objetos (DOO), se da el diseño de la solución a los problemas propuestos en ADOO para la creación del software, produciendo un diseño que interconecte objetos de datos y operaciones de procesamiento. Para Desarrollar ADOO se trabaja de la mano con UML.

1. Planteamiento del Problema
2. Antecedentes

Hoy en día, las ONGs en el país, tienen a la mano sistemas de información sencillos que permiten administrar la información, reduciendo gastos de papelería y tiempo de acceso a los datos. Pero aun existiendo estos sistemas, se puede decir que no cumplen con las funciones correctas y específicas para cumplir con las necesidades de las organizaciones (seguridad, complejidad, eficiencia, rapidez, eficacia), lo cual conlleva tal vez a una reducción real sobre gastos y tiempo.

Glasswing Internacional, fundada en El Salvador; es una ONG que tiene actividad en el país desde el año 2015, trabajando con colegios y escuelas primarias, en base a donativos de empresas que desean realizar responsabilidad social empresarial, obras de caridad, o que simplemente desean realizar alguna actividad para apoyar a la niñez nicaragüense.

Actualmente Glasswing procesa el flujo de información a través de distintas aplicaciones ofimáticas, como son los datos de los personales (coordinadores, voluntarios, beneficiados, patrocinadores, etc.), en tablas de hojas de cálculo, y se generan informes sencillos a través de formularios de Excel, los cuales posteriormente son unificados por el personal, conllevando tiempo al procesar los datos.

Glasswing no cuenta con un sistema propio que centralice el procesamiento de datos e información de la empresa. Glasswing utiliza herramientas como el paquete de ofimática para el procesamiento de archivos, en el caso de Excel y Google Sheets para el procesamiento de hojas de cálculo, también los archivos se almacenan en Google Drive al momento de compartir o mantener alojados en un sitio. Este el motivo por cual se decidió realizar la presente herramienta de administración, que es compuesto por un Análisis de requerimientos, factibilidad, viabilidad y Diseño de Sistema y de procesos de negocios para un software que centralice la entrada y salida de datos de la organización, para un posterior desarrollo e implementación.

1. Problema

Glasswing como organización sin fines de lucro trabaja a través de donaciones y dinero invertido por empresas para realizar proyectos de responsabilidad empresarial, por tanto, maneja variada cantidad de información, la cual necesita tratar para entregar informes estadísticos sobre avances y resultados finales de cada proyecto. Actualmente Glasswing carece de un SI formal y propio para el procesamiento de datos, por lo cual le toca recurrir al uso de otras herramientas ofimáticas (Word, Excel, etc.), esto conlleva a gastos innecesarios sobre tiempo y utilería, ineficiencia al momento de generar reportes para las empresas socias que patrocinan los proyectos, o al momento de la toma de decisiones gerenciales de la ONG.

1. Justificación

Como **Justificación práctica,** es diseñar una herramienta tecnológica que facilite y automatice procesos de administración y gestión de información de la ONG, como apoyo a la toma de decisiones gerenciales, y que ayude a reducir gastos operativos y de utilería, como consecuencias ahorro en papelería y tiempo.

Por otra parte, la **justificación académica**, se tiene como misión el dar uso a las diferentes metodologías, y herramientas tecnológicas propias de la ingeniería en software para la toma de requisitos, requerimientos y procesos para la realización de Análisis y Diseño de un software.

1. Hipótesis

Por medio de la herramienta a diseñar, la organización llamada "Glasswing" podrá tener mejor control sobre los reportes e informes estadísticos a partir de información administrativa de dicha organización. Además, esta herramienta permitirá a Glasswing, tener una administración efectiva de la información y datos, optimizando controles y estadísticas de los diferentes proyectos ejecutados.

1. Alcances

Al finalizar dicho proyecto, se obtendrán los siguientes resultados:

* Documento de Análisis de requerimientos y requisitos del Sistema Informático a diseñar.
* Documento de Factibilidad del Sistema.
* Documento de Viabilidad del desarrollo del Sistema.
* Documentación del sistema que abarque diagramas UML del sistema y los procesos de entrada y salida de datos; diagramas BPMN sobre los procesos de los principales procesos de negocios involucrados en el diseño del sistema; y diagrama y diccionario de cada una del base de datos a ser utilizadas.
* Documentación de los Prototipos del sistema, como son ejemplo de bocetos de ventanas del sistema, y un prototipo que englobe los principales módulos del sistema.

1. Limitaciones

* El diseño del sistema será carácter Web,
* No se realizará desarrollo ni Implementación.
* El diseño puede no ser perfecto debido al surgimiento de nuevas necesidades (requerimientos y requisitos).

1. Diseño Metodológico
2. Metodología

La metodología presente en esta investigación es de carácter cualitativa, debido a que se aborda el estudio de diferentes sucesos complejos de informática, también no se especifica un orden o método de procedimiento de recolección de datos específica, y tampoco las variables utilizadas son susceptibles a medición.

También en base a la recolección de datos no es posible realizar análisis estadístico.

1. Tipo de Diseño

El tipo de diseño de investigación realizada es no experimental, debido a que las variables no requieren de manipulación directa, y se da en un ambiente controlado.

1. Tipo de Investigación

El desarrollo de esta investigación es de carácter descriptiva-correlacional, primeramente, descriptiva, se describen situaciones y eventos a automatizar en una solución informática para organización llamada Glasswing, involucrando actores y sus propiedades

Desde el punto de vista correlacional, se cumple con el papel de interrelacionar variables sin control experimental, y el investigador se limita a observar y recolectar información sin intervenir en la situación actual de la organización.

1. Universo, Población y Muestra

Según El Directorio **ONG de Nicaragua** (10) del año 2015, Nicaragua contaba aproximadamente con 150 organismos y redes no gubernamentales registradas en este directorio, que vendrían a conformar el Universo del presente trabajo monográfico.

La población tomada de este universo es conformada por una sola organización, que sería Glasswing.

La muestra es conformada por los procesos de control y administración de la organización.

El método de tipo de muestreo para esta monografía, es no probabilística, sino por conveniencia, debido a que el Software a diseñar será única y adecuada para esta organización seleccionada.

1. Fuentes de Información

|  |  |
| --- | --- |
| **Fuentes Primarias** | * Coordinadores de Proyectos de Glasswing * Director de Proyectos de Glasswing en Nicaragua * Tutor |
| **Fuentes Secundarias** | * Libros especializados * Monográficas Relacionadas * Trabajos documentales relacionados. * Artículos de Páginas web |

1. Técnicas de recolección de datos

Para la realización de recolección de datos se realizarán entrevistas y cuestionarios al personal de Glasswing y director de proyectos, así obtener los requisitos, requerimientos y comentarios necesarios para el diseño del sistema, también se realizará una investigación documental de los datos que se obtengan.

La investigación documental es alimentada teóricamente por fuentes de información sobre los temas UML, BPMN, Sistemas de información, ingeniería en software, Arquitectura de software, Programación y análisis y diseño orientado a objetos, base de datos, entre otros.

1. Técnicas de procesamiento y análisis de Información

Durante el procesamiento de datos clasificaremos los elementos más relevantes de la información recopilada durante las entrevistas realizadas, y de la observación sobre los procedimientos de procesamiento de datos actuales de Glasswing, lo recopilado es adaptado y adecuado al contexto del estudio de esta monografía.

Como se refirió anteriormente que la metodología de esta documentación monográfica es de carácter cualitativo, por medio del estudio se realizara una puesta de mejora a los procesos actuales de Glasswing, como son el estudio de factibilidad y viabilidad, y el diseño de la herramienta tecnológica.

1. Operalizacion de Variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | Definición | Indicador | Valor | Escala | Instrumento- Fuente |
| **Diseño de un Sistema de Información** |  |  |  |  |  |
| **Automatización de Procesos y actividades** |  |  |  |  |  |
| **Requisitos y Requerimientos del Sistema** |  |  |  |  |  |
| **Organización no Gubernamental** |  |  |  |  |  |

1. Marco Lógico

|  |  |
| --- | --- |
| Tema | Ingeniería de Software |
| Titulo | **Análisis y Diseño de un sistema de Información para Glasswing Internacional en Nicaragua** |
| Objetivo General: | Diseñar una herramienta tecnológica de carácter web, para el manejo de información administrativa y de control operativo de los proyectos, así como controles estadísticos presentados en una dinámica de reportaría, que permita a Glasswing la administración efectiva, y oportuna de la información y datos para optimizar controles y estadísticas de los diferentes proyectos. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Objetivos Específicos | Herramientas o Métodos a Utilizar | Marco Teórico | Actividades | Indicadores | Hitos | Recursos |
| Realizar un análisis de requerimientos y requisitos para determinar el desarrollo de un Software de un Sistema de información para Glasswing Nicaragua. | Entrevistas  Cuestionarios  Ingeniería de requerimientos  Análisis de documentos y procedimientos realizados.  Procesadores de datos (Excel- Word, etc.) | Técnicas utilizadas en la ingeniería de requerimientos para obtención de requerimientos y requisitos  ¿Qué es una entrevista?  Como crear un Cuestionario para la toma de requerimientos y requisitos | - Realización de entrevistas.  - Revisión de los requisitos y requerimientos del software con los futuros usuarios del sistema.  - Revisión del sistema actual que poseen para manejo de información con le previo permiso | - Tipos de requerimientos  - Tipos de requisitos.  - Debilidades que se poseen al momento de administrar y procesar la información.  -Lista de las necesidades a ser implementadas o desarrolladas en el sistema | - Documento descriptivo de los requerimientos y requisitos que el sistema debe cumplir. | - Entrevistas  - Cuestionarios  - Factor económico para movilizarse a la organización.  - Factor tiempo disponible de los responsables de la ONG.  - Personal de la Oficina de Glasswing.  - Hojas de Cálculo para manejo de información de Glasswing. |
| Presentar un informe de Factibilidad y de Viabilidad para una futura decisión por parte de Glasswing, de Desarrollar e Implementar el Sistema completamente. | Entrevistas y Cuestionarios  Procesadores de datos (Excel- Word, Project, etc.) | Componentes de un estudio de factibilidad  Componentes de un estudio de viabilidad | - Identificar herramientas necesarias para la creación del sistema.  - Determinar y recopilar los costos del desarrollo del proyecto.  - Identificación de beneficios del sistema a la empresa.  - Redacción de un informe de factibilidad y otro de viabilidad del Proyecto. | - Herramientas a ser adquiridas  - Tipos de costos  - Tipos de beneficios | - Documentación en la cual se presente un estudio de factibilidad y de viabilidad para el Sistema a ser diseñado | - Computadoras  - Teléfono  - Internet  - Libros  - Tutor |
| Diseñar el sistema y procesos de entrada - salida de datos y de procesos negocios en UML y BPMN con el propósito de brindar una documentación para el desarrollo e implementación en la organización de un sistema de información. | UML  BPMN  Herramientas de diseño de procesos de sistemas y procesos de negocios (IBM Rational Rose, Bonita BPMN, Visio 2016, WhiteStarUML) | Utilización de Diagramas UML para modelación de procesos de negocio  Arquitectura de un Software  El Proceso de la ingeniería de Software. | - Realización de los diferentes diagramas para el análisis del Sistema en UML 2.0  - Realización de Diagramas de Procesos de Negocio con BPMN 2.0  - Creación de diagrama de arquitectura. | - Limitaciones y alcances del sistema  -Tipos de Diagramas desarrollados | - Documento en el cual se presenten los diagramas del Sistema a crearse y de los procesos de negocios involucrados. | -Información recopilada de los requerimientos y requisitos del sistema para la ONG  - Manuales UML & BPMN  - Recursos Web acerca de diagramas UML & BPMN  - Personal de la Oficina para confirmación de procesos.  - Internet  - Tutor  - Herramientas de diseño UML & BPMN |
| Elaborar el modelo de las bases de datos relacionales que se acomoden a lo recolectado durante el análisis de requisitos y requerimientos, que permitirá manejar la información centralizada y un control seguro de usuarios. | UML  BPMN  Navicat Premiun, MariaDB, PostgreSQL,  Procesadores de datos (Excel- Word, Project, etc.) | Utilización de Diagramas BPMP para modelación de procesos de negocio.  Elementos UML para el diseño de procesos de sistema  Definición del sistema y la funcionabilidad  Teoría de base de datos | - elaboración de un diseño de la base de datos en base a los diagramas UML y BPMN creados.  - Elaboración de diagrama de base de datos  - Creación de un prototipo de base de datos  - Elaboración del diccionario de datos | - Diagramas del diseño de las bases de datos  - Elementos del diccionario de datos | - Documentación del diseño del sistema | - Requisitos y requerimientos detectados durante las entrevistas  - Herramientas de desarrollo UML / BPMN  - Herramientas para diseño de base de datos  -Tutor  - Internet |
| Elaborar y Diseñar los prototipos de la interfaz visual de una herramienta que permita a Glasswing monitorear y evaluar los proyectos ejecutados, además de poder, en base a los requisitos y requerimientos que la gerencia de Glasswing brinde. | Wireframes  Mockups  Procesadores de datos (Excel- Word, Project, etc.)  Software de diseño de prototipos y bocetos (Justinmind Prototyper, Balsamiq Mockups) | Creación de Prototipos de Software | - Creación de Wireframes del Sistema  - Creación de Mockups del sistema | Wireframes y Mockups creados  Diagrama de Mapeo de los principales módulos del Sistema | - Documento de los Wireframes y mockups creados | - Herramientas para la creación de Prototipos de software.  - Internet  - Computadora  - Manuales de desarrollo de software  - Tutor  - Equipo de la oficina de Glasswing y futuros usuarios a usar. |

1. Resultados
2. Realizar un análisis de requerimientos y requisitos para determinar el desarrollo de un Software de un Sistema de información para Glasswing Nicaragua.
   1. Identificación de Requisitos del sistema

Se pueden dividir en dos categorías: Funcionales (Afectan directamente al sistema) y No funcionales (Afectan indirectamente al sistema).

* + 1. Requisitos Funcionales
* El sistema será capaz de administrar (registrar y modificar) información de los colegios, profesores y estudiantes beneficiados, a la vez de las empresas patrocinadoras y de los voluntarios y/o pasantes que trabajen con Glasswing.
* El sistema poseerá un módulo para el manejo y creación de usuarios, sus permisos, restricciones y privilegios.
* El sistema almacenara información sobre los proyectos, cantidad de clubs extracurriculares y otras actividades que la organización realiza, esto refiriendo a apertura de nuevos clubes o proyectos, o cierre de los que se están ejecutando (Las tareas mencionas como parte de este requisito, podrán ser ejecutadas por los coordinadores y ciertas partes también podrá intervenir el Supervisor).
* El sistema deberá guardar o administrar información correspondiente a la ejecución de las campañas (proyectos, actividades, clubs, etc.) o planes de la organización. Además, deberá referir si es donación monetaria (cantidad, tipo de moneda, y donante) o donación de material (cantidad, costos, donante).
* El sistema controlara o gestionara posición que ocupen los colaboradores de la organización (trabajadores, pasantes o voluntarios).
* El sistema permitirá a los usuarios buscar y consultar información de la base de datos, sobre lo que se desea siempre y cuando el término exista.
* El sistema generara a través de la información almacenada estadísticas y reportes con las variables definidas por los usuarios que poseen el permiso. Ya sea, por ejemplo, acerca de la destinación de recursos, actividad de donantes, voluntarios/ pasantes; o algún otro criterio o variable.
* El sistema aceptara el acceso de una sola sesión por usuario al mismo tiempo.
* El sistema genera log o historial de todas las actividades que los usuarios realicen, por motivos de cualquier auditoria.
* El sistema permitirá llevar control de asistencia de los beneficiados y los voluntarios en las actividades y clubes que se realizan.
* El sistema poseerá la capacidad de visualizar la relación entre clubes y estudiantes, o estudiantes proyectos, clubes y voluntarios, empresas proyectos, así de otras variables relacionadas.
* El sistema genera un backup cada cierto periodo de tiempo, el cual será programado por solo un nivel de usuario.
* El sistema generara automáticamente reportes estadísticos cada vez que un proyecto se cierra.
* El sistema generara reportes a través de encuestas que se realizan a los beneficiados al finalizar cada proyecto, club o actividad programada.
* Los coordinadores serán capaces de registrar incidencias que ocurran a lo largo del calendario de cada proyecto, club o actividad.
* Los coordinadores serán capaces de generar informes contables, por ejemplo, presupuestos o facturas.
* El sistema será capaz de generar presupuestos a partir de indicaciones del usuario, en este caso los coordinadores para gastos de actividades de proyectos o clubes.
  + 1. Requisitos no Funcionales
* El software será desarrollado bajo licencia GPL.
* El sistema solo permitirá 3 intentos de inicio de sesión, luego de eso se necesario un tiempo de 5 minutos para entrar.
* El ingreso deberá ser restringido por medio de contraseñas y usuarios definidos.
* El sistema se desarrollará bajo la arquitectura M.V.C.
* El sistema se desarrollará en PHP, usando el framework CakePHP 3.0 o superior.
* La base de datos debe ser en lenguaje SQL y gestionada con MariaDB.
* Habrá una base de datos basada en PostgreSQL, la cual será usada para gestionar la información de usuarios para acceso del sistema.
* El software deberá cumplir el concepto de portabilidad, debe visualizarse y funcionar en cualquier navegador web en PC o dispositivo Móvil.
* El sistema debe estar alojado en un sitio web.
* Toda funcionabilidad del sistema debe responder al usuario en menos de 5 segundos.
* La base de datos debe estar en Línea.
* El sistema no debe tardar más de cinco segundos en procesar búsquedas para que se muestren resultados, tras este tiempo el sistema debe mostrar ya lo que ha encontrado.
* La Interfaz del sistema debe ser clara, sencilla y amigable con el usuario.
* Todo cambio hecho en cualquier registro de las tablas de la base de datos deber actualizada en menos de dos segundos, para el uso por parte de otro usuario.
* Todas las comunicaciones externas del sistema deben estar encriptados usando el algoritmo **RSA.**
* El sistema debe ser desarrollado bajo las buenas prácticas de CakePHP.
* El sistema debe ser posible manejar con solamente dos horas de capacitación al personal.
* El desarrollo de los diseños del sistema debe ser desarrollados con diagramas UML
* En vez de los diagramas de actividades de UML, se utilizarán para la descripción de los procesos de negocios diagramas BPMN.
* Sera necesario mínimo 4 wireframe y 5 mockups del sistema.
* El código del sistema debe ser contar con los comentarios necesarios para posibles modificaciones o correcciones de fallo para mejorar su funcionamiento.
* El Supervisor o superusuario debe ser quien pueda aprobar generación de presupuestos.
* El sistema solo podrá poseer un Supervisor, quien funcionará como superusuario.
* El sistema informara al Supervisor, a través de correo electrónico los cambios que se realicen en el sistema, últimos accesos al sistema, así mismo quienes están conectados entre ciertas horas.
  1. Módulos requeridos

A continuación, se describen los diferentes módulos que deberán estar incluidos en el sistema.

* + 1. Control Interno
  + **Módulo de gestión de usuarios**: control de acceso al Sistema, donde solo existen 3 niveles de usuarios Coordinadores, Director de proyectos (quien cumple el papel de Supervisor), y espectador.
  + **Módulo de Control de usuarios:** Los usuarios pueden gestionar cambios a sus cuentas.
    1. Escuelas
  + **Módulo de Colegios Beneficiados:** es el modulo que contiene las escuelas beneficiadas de los proyectos que la organización lleva a cabo, en este módulo el usuario puede añadir, modificar y deshabilitar registros.
  + **Módulo de Profesores Beneficiados:** Este módulo contiene las tareas relacionadas a la administración de profesores de las escuelas beneficiadas de los proyectos y clubes; Se registran también control, asistencia y seguimiento sobre las actividades en ellos están integrados.
  + **Módulo de Estudiantes Beneficiados:** Este módulo contiene las tareas relacionadas a la administración de estudiantes participantes de las escuelas beneficiadas de los proyectos y clubes; Se registran también control, asistencia y seguimiento sobre las actividades en ellos están integrados.
    1. Proyectos
  + **Módulo de Proyectos Ejecutados:** Este módulo mantiene la información y datos de los proyectos que Glasswing ejecuta. También sobre el seguimiento de las actividades, castos materiales didácticos; de la misma forma en este módulo el usuario puede añadir, modificar y deshabilitar registros.
  + **Módulos de clubs ejecutados:** Es este módulo se almacena los datos relacionados a los diferentes clubs desarrollados en los proyectos que Glasswing ejecuta en las escuelas beneficiadas.
  + **Módulo de Empresas Socias**: Se almacena la información referente a las empresas que patrocinan los proyectos ejecutados por Glasswing. En este módulo el usuario puede añadir, modificar y deshabilitar registros.
    1. Personal
  + **Módulo de Trabajadores:** el modulo que administra la información de los trabajadores oficiales de la organización, sus horas que han ejecutado, tareas, proyectos involucrados, etc.
  + **Módulo de Voluntarios:** el modulo que gestionara información referente a voluntarios, sus horas que han ejecutado, tareas, proyectos involucrados, etc.
  + **Módulo de Pasantes:** el modulo que controlara información referente a pasantes de la organización, sus horas que han ejecutado, tareas, proyectos involucrados, etc.
    1. Herramientas
  + **Modulo Buscador:** este módulo es utilizado para la búsqueda de términos generales o específicos dentro de todas las tablas de la base de datos.
  + **Módulo de Reportes Estadísticos:** El modulo encargado de la generación de reportes y estadísticas en base a las variables que el usuario defina, y sobre los datos almacenados en las tablas de la base de datos. en \*.pdf o \*xsl, de los datos requeridos por los usuarios del sistema.
  + **Respaldo de base de datos:** en este módulo el usuario puede generar respaldos de la base de datos, de manera manual o dejar programada.
  + **Modulo para Crear Presupuestos:**  en este módulo se crea un formulario para la realización de un presupuesto destinado a cierta actividad, proyecto o club.
  1. Entregables

A cierto punto los entregables son discutibles, y se pueden dividir en dos grupos

* **Entregas del Análisis y Diseño:** Documento de recolección de requerimientos, factibilidad y viabilidad del Proyecto, , 6 Wireframes que ejemplifiquen el sistema, Conjunto de diagramas BPMN acerca de los procesos de negocio del sistema, Diagramas UML para el desarrollo del sistema, 5 mockups del sistema, Repositorio en Github, Ambiente de desarrollo en 9cloud, Diagrama de base de datos del sistema y diccionario de las variables de la base de datos, y por ultimo entregables de todo lo anterior en un CD.
* **Entregables del Desarrollo e Implementación:** Repositorio de Github, ambiente de desarrollo en 9cloud, Archivo Zip de los componentes de sistema y la base de datos, script de base de datos, Software corriendo en el hosting, Manual de Uso del sistema, CD conteniendo el software, manual de uso, y documentación relacionada al sistema.
  1. Alcance del Sistema

Como parte de logística el sistema es capaz de administrar información de Empresas colaboradoras, Proyectos y Clubes que se ejecutan, Personal (Colaboradores, Voluntarios y Pasantes), Colegios beneficiados, así mismo de los beneficiados que refiere a Profesores y Estudiantes de estos colegios.

El sistema también será capaz de generar informes estadísticos y reportes dependiendo la necesidad del usuario, estos podrían ser, reportes de ejecución de proyectos, de actividades de los clubes, reportes de voluntarios o pasantes, de uno o varios estudiantes o profesores beneficiados, realización de toma de asistencia o seguimiento, generación de reportes a partir de encuestas realizadas, entre otros. El sistema será capaz de exportar estos informes, estadísticas y reportes en archivos de hojas de cálculo (\*.xsl) y pdf.

El sistema poseerá un módulo de gestión de usuarios, en donde se pueden crear perfiles y asignar permisos sobre las funciones del sistema.

Como procesos de auditoria, el sistema realizara control de historial o log del sistema, control de errores, modificaciones en registro y backup de información del BD. A medidas de seguridad el Sistema debe ser capaz de realizar validación de acceso de usuarios, encriptación de archivos del sistema y aceptar modificaciones solo de un usuario con permisos previos en dado caso ser el Superusuario.

1. Presentar un informe de Factibilidad y de Viabilidad para una futura decisión por parte de Glasswing, de Desarrollar e Implementar el Sistema completamente. 
   1. Factibilidad

En este inciso se reúne los elementos necesarios para demostrar si es proyecto es factible o no de realizar. Se describe detalles referentes a personal, software y hardware a utilizar (alternativas y costos), quienes son el cliente y los usuarios, además de los riesgos que se pueden presentar sus y posibles soluciones.

* + 1. Cliente

El sistema será diseñado para atender los requisitos y requerimientos de la ONG Glasswing en Nicaragua.

* + 1. Usuarios

Se definen tres tipos de usuarios para el uso del sistema con los siguientes permisos:

|  |  |  |
| --- | --- | --- |
| Tipo de Usuario | Corresponde a | Principales Privilegios |
| Coordinadores | Coordinadores de Proyecto | * Visualizar todo * Administrar módulos, excepto los de control y gestión de usuarios. * Agregar Nuevos registros * Deshabilitar registros. * Creación de Reportes |
| Supervisor | Director de proyectos | * Visualizar todo * Modificar ciertos registros de ciertos módulos |
| Espectador | Agente de la Filial Central en el Salvador | * Solo Visualizar * Acceso bajo autorización a generar ciertos reportes. |

* + 1. Alternativas de Software

Para la realización del sistema se necesitan especificar ciertas dependencias necesarias como son: un motor de base de datos, el lenguaje de programación que se utilizara para el desarrollo del sistema, así mismo como una selección de framework M.VC, por ultimo definir otras herramientas útiles al momento de desarrollo, pruebas y uso final. Estos puntos son abordados en este bloque.

* + - 1. Base de Datos

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo | Nombre | Costo | Licencia |
| Open Source | María DB | Gratis | GPL |
| PostgreSQL | Gratis | Licencia PostgreSQL |
| MySql community | Gratis | GPL |
| Oracle Express | Gratis | Apache |
| De pago | Oracle DB | $ 165 por 25 licencias al año  $ 825 por 25 licencias  perpetuas | Proprietary OTN Standard License |
| IBM DB2 | U$$ 530-22800 |  |
| MS SQL Server | Standard $931 | Microsoft EULA |
| MySql Enterprise | U$$ 600-5000 | GPL/Comercial |

Se hizo un análisis comparativo entre los distintos gestores de base de datos posibles a utilizar, se compararon entre gestores de pago y gratuitos, para presentar ante un presupuesto o no por parte de Glasswing. Los gestores seleccionados se escogieron bajo las condiciones de ser compatibles con los lenguajes de programación propuestos en el siguiente inciso.

En base a las necesidades de Glasswing y presupuesto disponible, se decidió que no es necesario por el momento usar o invertir en un gestor de base de datos de pago, por lo cual se seleccionaron 3 posibles soluciones gratuitas: *MySql Community*, *MariaDB* y *PostgreSQL*; basadas en sus beneficios y características, para después en el inciso de “Alternativa seleccionada” definir cuál utilizar.

* + - 1. Lenguajes de Programación

Bajo las condiciones de ser *open source*, se propusieron una lista de lenguajes que sean compatibles con los gestores de base de datos posibles a utilizar. En la siguiente tabla se enlistan los lenguajes de desarrollo web gratuitos propuestos, y opciones de posibles IDE’s (una gratuita y una de pago) para la programación.

|  |  |  |
| --- | --- | --- |
| Lenguaje | IDE Open Source | IDE De pago |
| PHP | Netbeans | phpDesigner |
| Python | Pycharm community | Pycharm Professional |
| Ruby | Aptana | Ruby mine |
| Perl | Padre | UltraEdit |

Al tener seleccionadas los posibles gestores de base datos, y revisar las opciones de lenguajes de programación propuestos, se seleccionaron tres de estos, de los cuales existan opciones de frameworks M.V.C que respondan a las necesidades de requisitos y requerimientos recolectados. Los lenguajes analizados y propuestos son: *PHP*, *Python* & *Ruby*.

* + - 1. Frameworks

Luego de haber escogido los posibles lenguajes a utilizar, se analizaron los diferentes frameworks webs existentes, se seleccionaron dos de cada uno, se hizo una comparativa, para escoger una sola opción por lenguaje, ya los cuales se le tomo el ranking de popularidad los cuales se obtuvieron de la página Hotframeworks (11) , basándose en el “Overall Score”. (En anexos “Comparativa de Frameworks”).

|  |  |  |  |
| --- | --- | --- | --- |
| Framework | Versión Actual | Licencia | Costo |
| PHP |
| CakePHP | 3.2.8 | MIT | Gratis |
| Symfony | 3.1.0 RC-1 | MIT | Gratis |
| Python |
| Django | 1.9.4 | BSD License | Gratis |
| Web2py | 2.14.6 | GNU Lesser GPL v3.0 | Gratis |
| Ruby |
| Rails | 5.0.0.1 | MIT | Gratis |
| Sinatra | 1.2.6 | MIT |  |

Al final la selección de las 3 alternativas de lenguaje es: *CakePHP* (Ranking #79), *Django* (Ranking #91) y *Rails* (Ranking #96), estos son los frameworks propuestos para desarrollar el sistema, Documentación amplia existente, y soportan las alternativas de base de datos Propuestas.

* + - 1. Otros Software

Se necesita adquirir en otro software para realizar pruebas u otras tareas, acá se enlistan los posibles a utilizar:

|  |  |
| --- | --- |
| Programa | Costo |
| Diagramas UML | |
| WhiteStartUML | - |
| Diagramas BPMN | |
| BonitaBPMN | - |
| Wireframes & Mockups | |
| Justmind | - |
| IDE para desarrollo | |
| Netbeans | - |
| Servidor Local para Pruebas | |
| XAMPP | - |
| WAPP | - |
| Software FTP |  |
| File Zilla Client | - |
| Diagramas de base de datos | |
| MySQL Workbench | - |
| Navegador Web (recomendados) | |
| Chrome (mínimo versión 35) | - |
| Firefox (mínimo versión 34) | - |

Se busca gastar lo menos posible en adquisición de software, también que el sistema pueda ser accedido no solo por computadoras, sino a través de otros dispositivos móviles (Tablets, Smarthphones, etc…).

* + 1. Selección de Alternativa Final de Software

Como alternativa Final se decidió utilizar el Framework **CakePHP**, con el gestor de base de datos **MariaDB,** para la gestión de los datos, pero se seleccionó utilizar una base de datos **PostgreSQL** para el control de acceso de usuarios, como medida de seguridad. Entre las razones que hicieron la toma de esta decisión:

**CakePHP** posee las características como framework MVC, la cual ofrece la facilidad, flexibilidad y agilidad para el desarrollo de una aplicación web; CakePHP cuenta con una gran comunidad de activos y un equipo de desarrollo, que agregan valor al proyecto, además gran cantidad y variedad de plugins, que le aumentan funcionabilidad, además de contar con muy buena documentación, lo cual permite a los desarrolladores a contratar, aprender rápido acerca del framework; al almacenarlo en línea no requiere de tanta configuración local, y el hosting a usar es compatible con el lenguaje de páginas PHP; otra de las razones es el tipo de licencia con que es distribuido, y que no se requiere el pago de licencias para IDE, ya que se utilizara Netbeans, el cual es gratuito y soporta el framework a usar.

**MariaDB**, es el gestor a ser utilizado para el manejo de la información del sistema, por ser un fork de Mysql, comparte muchas características con este gestor, lo cual lo hace compatible con el framework que se ha seleccionado a utilizar, además que integra nuevas características no presentes en Mysql; por otro lado, es un gestor con una amplia comunidad de apoyo y desarrollo; es compatible con bases de datos mysql, así mismo la base de datos creado con MariaDB es compatible con el hosting a usar; yes de licencia abierta, por lo cual no se necesita pagar por usar este producto, es suficiente para las tareas de almacenamiento que la organización necesita.

En el caso de **PostgreSQL,** se decide utilizar esta base de datos para controlar el acceso al Sistema (Control de usuarios), debido a la robustez, seguridad y garantía de integridad de datos del este gestor, es el más conveniente, además que no es necesario paga, este se mantendría en un servidor externo, por cualquier intento de intrusión.

Por otro lado, no se seleccionó otras opciones gratuitas como MS SQL Express o Oracle DB Express, por las limitaciones que estas presentan en especial con la conexión al hosting o servicio web que se usara, y los costos que implican en un equipo de desarrollo.

|  |  |  |
| --- | --- | --- |
|  | | Costo |
| Lenguaje y Framework | PHP – CakePHP 3 | - |
| Base de datos | MariaDB | ­- |
| IDE | Netbeans |  |
| Total, en dólares | | $0 |

* + 1. Alojamiento Web

Glasswing cuenta con un alojamiento Web, por lo tanto, no será necesario realizar una inversión en este aspecto.

El actual servicio web es compatible con el gestor de base de datos y el lenguaje del framework a utilizar, además de ser compatible con JavaScript también y Php.

* + 1. Hardware para desarrollo

Actualmente Glasswing cuenta con máquinas disponibles para realizar las tareas de desarrollo, siendo las características de estas, suficientes para la programación y las pruebas locales del sistema antes de hacer uso del Software en línea.

|  |  |
| --- | --- |
| Ítem | Descripción |
| Sistema Operativo | Windows 7 ProSp1 |
| Tarjeta Madre | ASROCK |
| Disco Duro | 500 Gb Sata |
| Memoria RAM | 12 Gb DDR3 |
| Tarjeta de Red inalámbrica | - |
| Tarjeta de Red ethernet | Broadcom |
| Procesador | Quad Core 2.22Ghz |
| Monitor | Hp Compac LE1711 |
| Marca | Hp Compac |

Debido a que el sistema estará alojado en un Hosting web, no es necesario invertir en máquinas para acceso local, ya que el sistema estará en línea, y podrá ser accedido siempre y cuando el dispositivo a usar para conectarse al Sistema, cuente con conexión a internet.

* + 1. Recursos Humanos

Se considera que el proyecto de desarrollo puede ser realizado en tres meses. Para el desarrollo del proyecto se considera necesario lo siguiente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cantidad | Personal | Horas | Salario Mensual | Totales |
| 2 | Programador | 8 | $1000 | $6000 |
| 1 | Analista | 8 | $1000 | $3000 |
| 1 | Tester | 8 | $600 | $1800 |

En el caso del uso del sistema, existirán dos categorías, y solo podrán acceder al sistema 7 usuarios (Coordinadores de proyecto (3), Director de proyectos (1) quien cumple el papel de Supervisor, y un espectador). Estos requieren ser capacitados en el uso del sistema.

* + 1. Costo Total de Software y Personal

Se enlistan todas las herramientas a utilizar en el desarrollo, además se añaden el gasto total en personal.

|  |  |
| --- | --- |
| Herramienta/ personal | Costo |
| Netbeans | - |
| WhiteStartUML | - |
| Bonita BPMN | - |
| CakePHP | - |
| MariaDB | - |
| Mysql Workbench Community | - |
| Programadores (2x3 meses) | $6000 |
| Analista (1x3meses) | $3000 |
| Tester (1x3meses) | $1800 |
| Total | **$10800** |

* 1. Viabilidad

Al terminar el estudio de factibilidad, analizamos que beneficios y alcance traería para la organización Glasswing, análisis de riesgos existentes, relaciones referentes a los costes y posibilidades de la empresa.

* + 1. Beneficios del Sistema
* Facilidad de almacenar información de manera rápida y segura
* Centralizar la información de voluntarios, coordinadores, escuelas beneficiadas, beneficiarios y patrocinadores en un sistema creado para las necesidades de la institución.
* Llevar control de los proyectos y actividades que la organización realiza
* Poseer un Sistema que pueda procesar informes estadísticos con las variables disponibles de la base de datos para toma de decisiones por parte de la gerencia.
* Facilidad de realizar informes para enviar o compartir con el resto de filiales de la organización.
* Reducción de tiempo para procesar informes que se producirán según el periodo de tiempo que se requiera.
* Facilidad de acceso al sistema desde cualquier parte, por ser un sistema en línea
* Permitirá respaldar la base de datos, impidiendo perdida de esta.
* Análisis Costo -Beneficio
  + 1. Viabilidad Económica

El desarrollo del sistema se llevará al cabo de tres meses, por el conjunto de 2 programadores en Php, un analista y un tester. Se propone a incluir como propuesta en el presupuesto del año 2017, asumiendo que los socios deseen ampliar las escuelas beneficiadas.

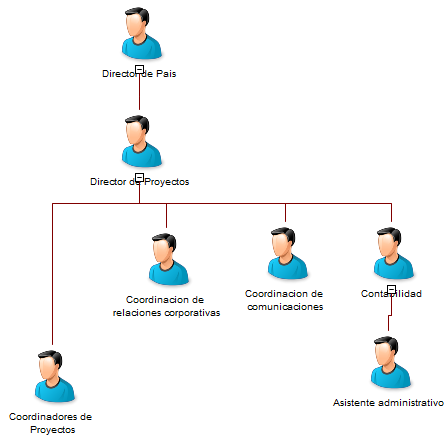
En el caso de la adquisición de software para el desarrollo, no se necesitará obtener o pagar licencias acerca de software privativo. Por otro lado, tampoco será necesario invertir en equipo nuevo, con los computadores que actualmente la ONG tiene disponible sería suficiente para realizar desarrollo y pruebas.

* + 1. Viabilidad técnica

Aplicando el uso de las tecnologías actuales permiten el desarrollo de la metodología de desarrollo de cascada para obtener resultados referentes a la creación del software; así mismo los softwares descritos para el desarrollo no presentan problemas al ejecutarse en el entorno de desarrollo.

* + 1. Viabilidad operacional

Se tendría que realizar un pequeño manual para educar a los usuarios en el uso del sistema.

Glasswing presente la estructura mostrada en el organigrama siguiente:

El punto más relevante de tocar en el organigrama de Glasswing, sería el de los coordinadores, actualmente se cuentan con tres proyectos, cada uno con su propio coordinador o persona responsable, quienes serían los mayores beneficiados del sistema, pues son quienes más lo utilizarían.

* + 1. Mercado de Glasswing Nicaragua

El actual mercado de Glasswing, al ser una organización sin fines de lucro es el conjunto de beneficiados de los proyectos que la fundación realiza a través de las donaciones de empresas dispuestas a realizar caridad o RSE.

* + 1. Viabilidad conceptual

Creamos un Análisis FODA acerca de la implementación del sistema:

* + 1. Análisis de Riesgo

Durante el transcurso del proyecto se prevén la aparición de uno o más riesgos, en especial durante dos etapas (desarrollo e implementación), acá se detallan estas y una posible solución.

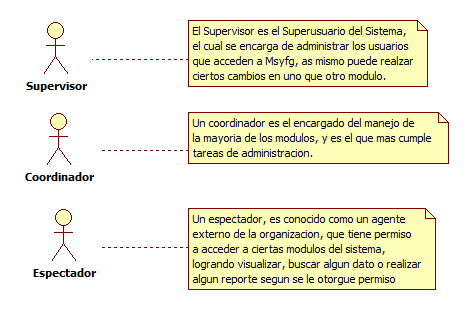
* + 1. Riesgos durante el desarrollo

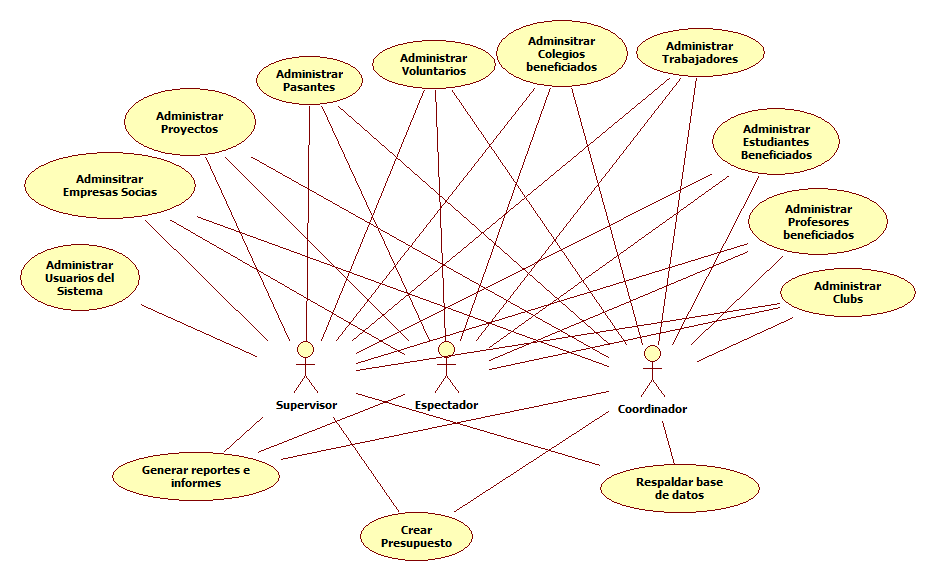
|  |  |  |
| --- | --- | --- |
|  | Riegos | Soluciones |
| 1 | Uno de los programadores renuncie. | Basándonos en que contamos con el personal necesario podemos contar con que el desarrollo se continúe aun con un programador, mientras se contrate otro. |
| 2 | Problemas de perdida de datos archivos de desarrollo, o que los equipos utilizados presenten daños. | Se contará con respaldos en nubes (Dropbox, por ejemplo), además se aplicará control de versiones con la herramienta de Github y 9cloud para continuar el desarrollo en línea si es necesario y pruebas pertinentes. |
| 3 | Errores durante el desarrollo del código, que no cumplen con las funciones requeridas | Se realizarán constantes pruebas durante, la finalización de cada módulo, y que sean los diseños lo más fieles a lo descrito por los prototipos |
| 4 | Atrasos y fallos con la estrategia de desarrollo utilizada | Se realizará un análisis de la situación, acerca del porcentaje recuperable del proyecto, seguido se evaluará lo que se posee al momento, y se tomaran nuevas medidas para continuar el proyecto. |
| 5 | Malas prácticas de desarrollo | Se buscará ir documentando todo lo desarrollado en código, buscando aplicar las buenas prácticas recomendadas por la comunidad del framework. |

* + 1. Riesgos durante la ejecución del sistema

|  |  |  |
| --- | --- | --- |
|  | Riegos | Soluciones |
| 1 | El sistema no trabaja de manera optima | Se realizarán modificaciones o correcciones necesarias, siempre y cuando sean dentro de los limites acordados, en otras palabras, sea responsabilidad de los desarrolladores, al mismo tiempo se actualizara la documentación. |
| 2 | Resistencia al cambio por parte del usuario | Se darán capacitaciones para el uso del sistema, y documentación (Documento de desarrollo y Manual), además de soporte técnico por tiempo determinado. |
| 3 | Caída el sistema del sistema en línea | Esto dependerá de la empresa que este prestando el servicio de hospedaje. |
| 4 | Mal utilización del sistema, eliminación de algún archivo, o daño de la base de datos | Dependerá de la capacitación del personal y de quien administre el sistema. |

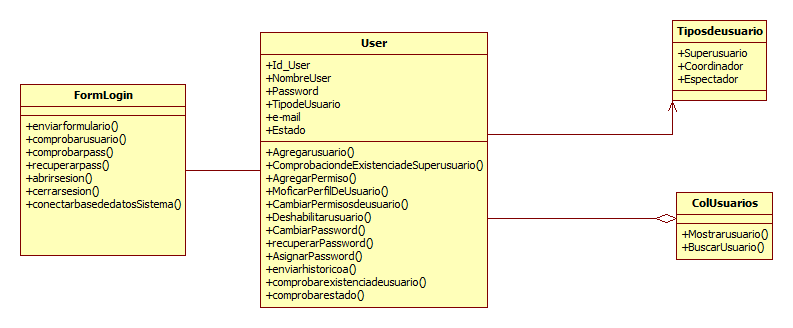
1. Diseñar el sistema y procesos de entrada - salida de datos y de procesos negocios en UML y BPMN con el propósito de brindar una documentación para el desarrollo e implementación en la organización de un sistema de información.
   1. Diagramas UML

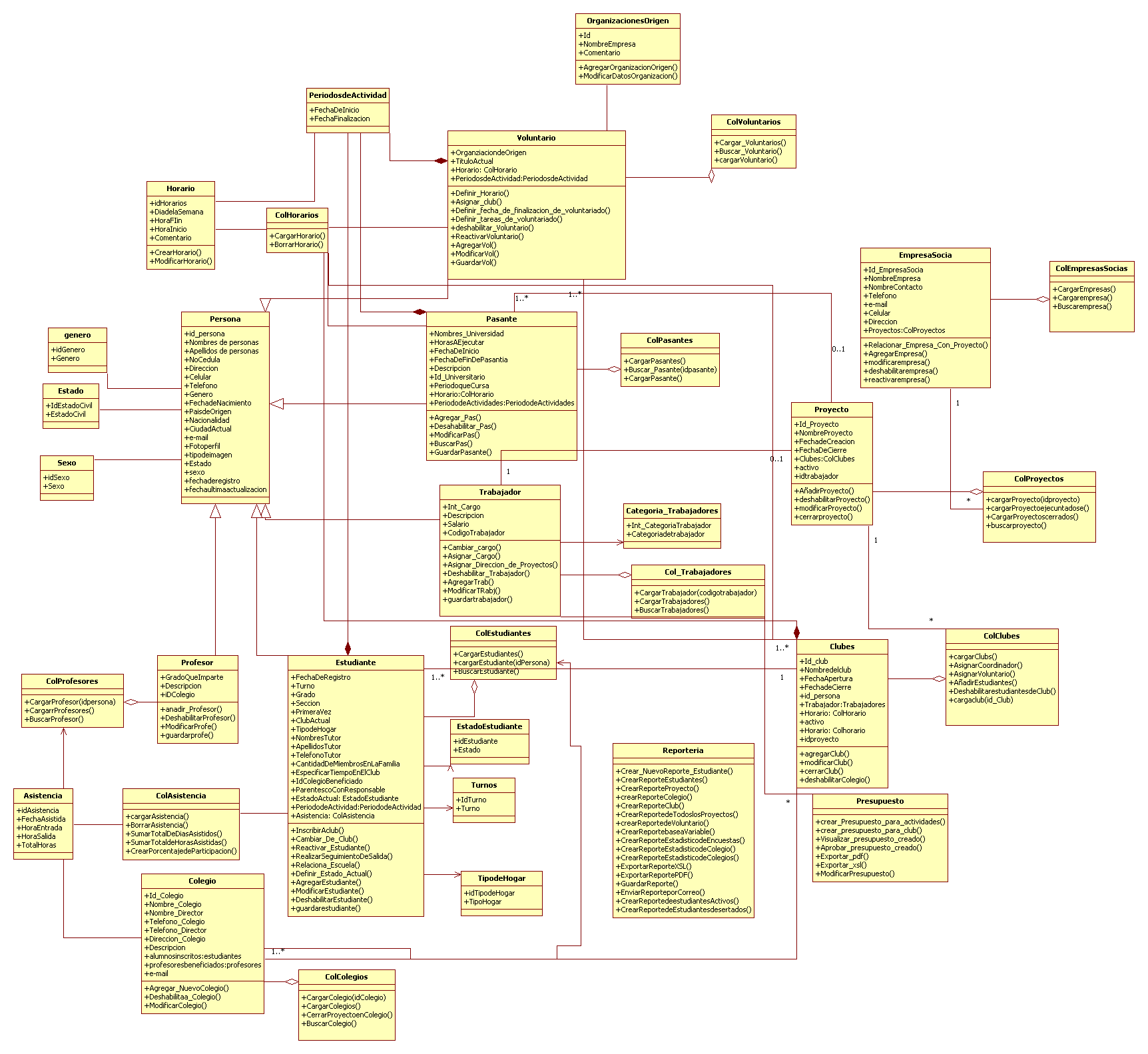
Diagramas de Actores del Sistema

Diagramas de Casos de Uso

Diagramas de Clases

Diagrama de Clases de Gestión y Control de Usuarios



Diagrama de Clases de Manejo de Datos

Diagramas de Objetos

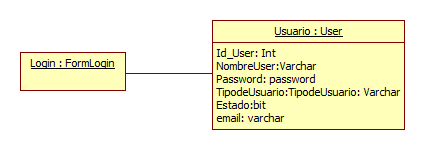
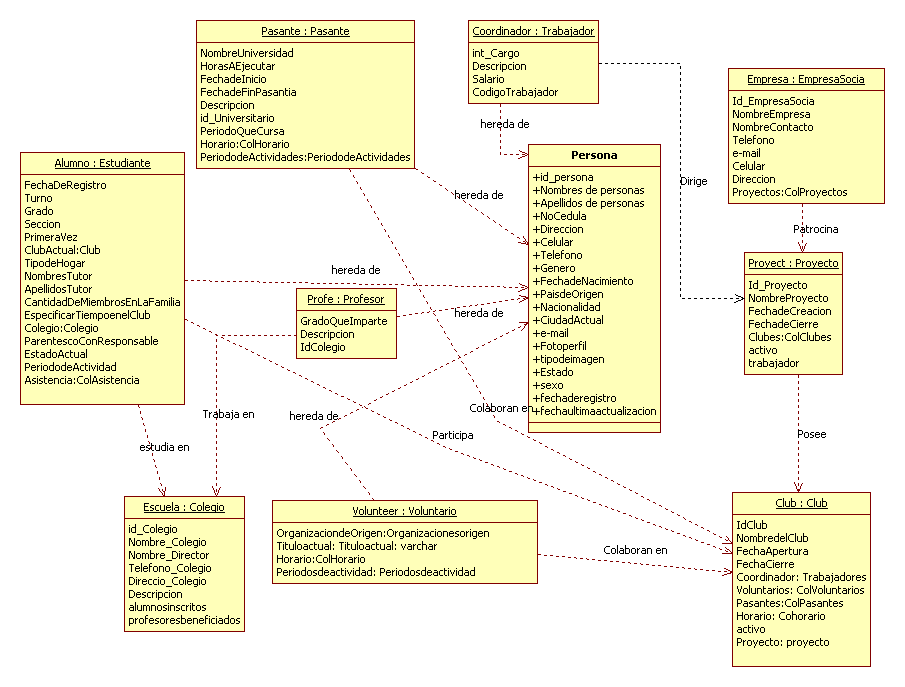
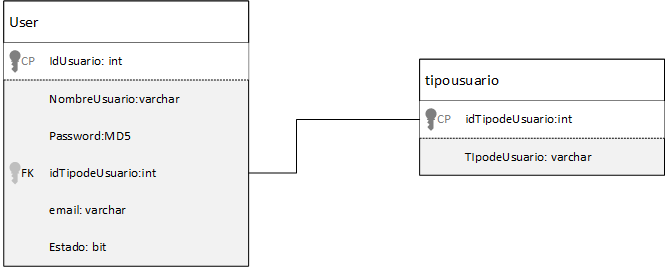
Diagrama de objetos de Gestión y Control de Usuarios

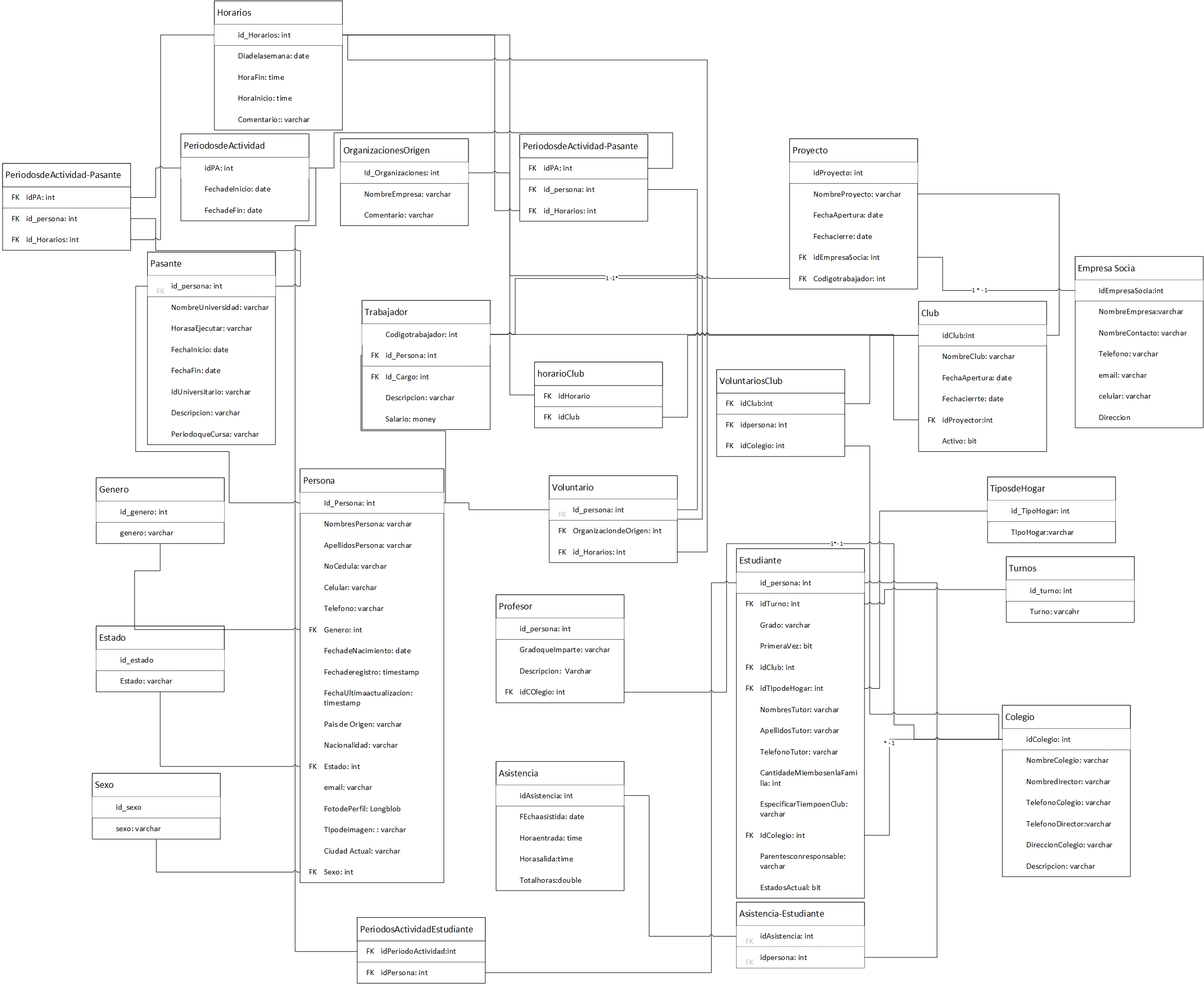
Diagrama de Objetos de Manejo de Datos



Diagramas de Mapeo

Diagrama de Mapeo de Gestión y Control de Usuarios



Diagrama de Mapeo de Manejo de Datos

Diagramas de Estados

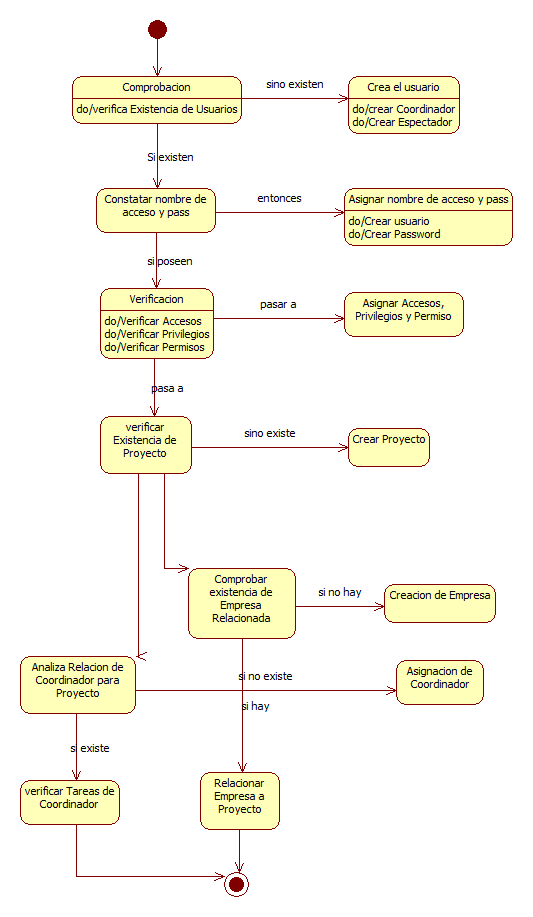
 Diagrama de Estados de Supervisor

Diagrama de Estados de Coordinador

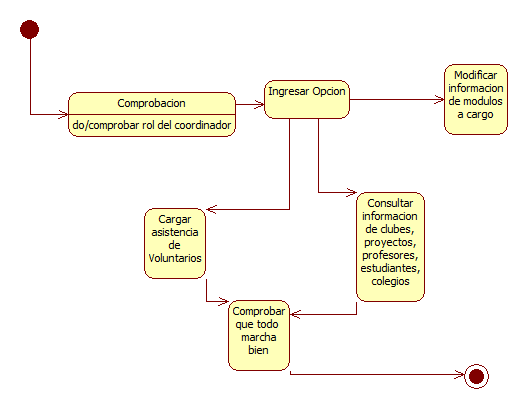


Diagrama de Estados de Espectador

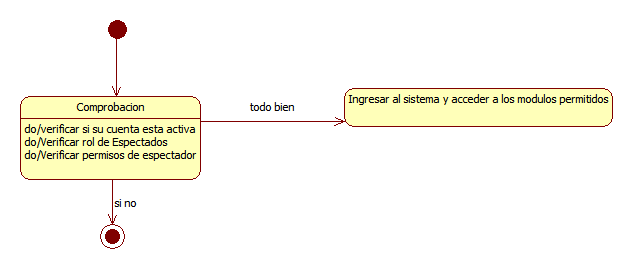
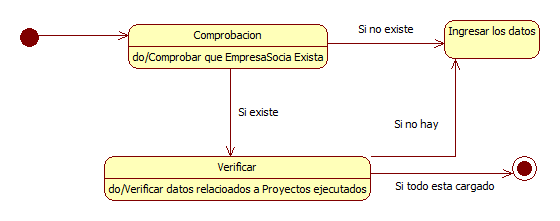


Diagrama de Estados de Empresa Socia



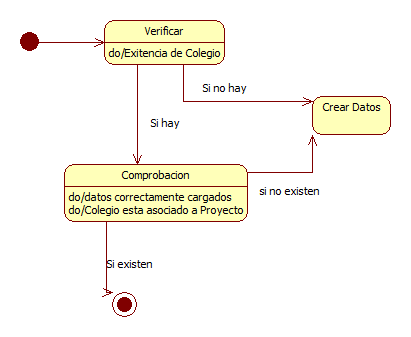
Diagrama de Estados de Colegio Beneficiado

Diagrama de Estados de Proyecto

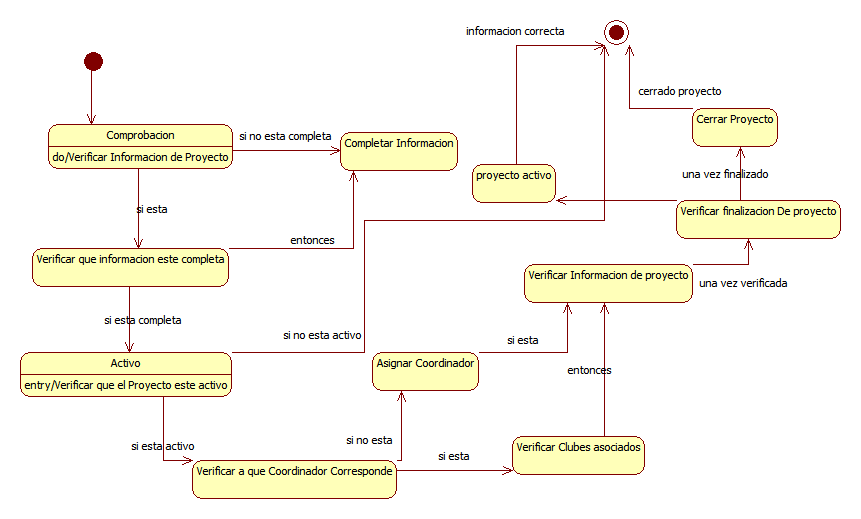


Diagrama de Estados de Club

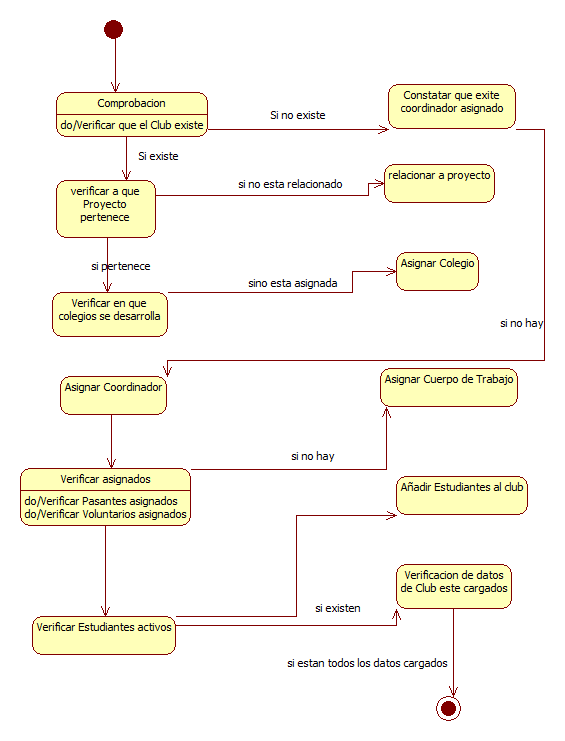


Diagrama de Estados de Estudiante Beneficiado

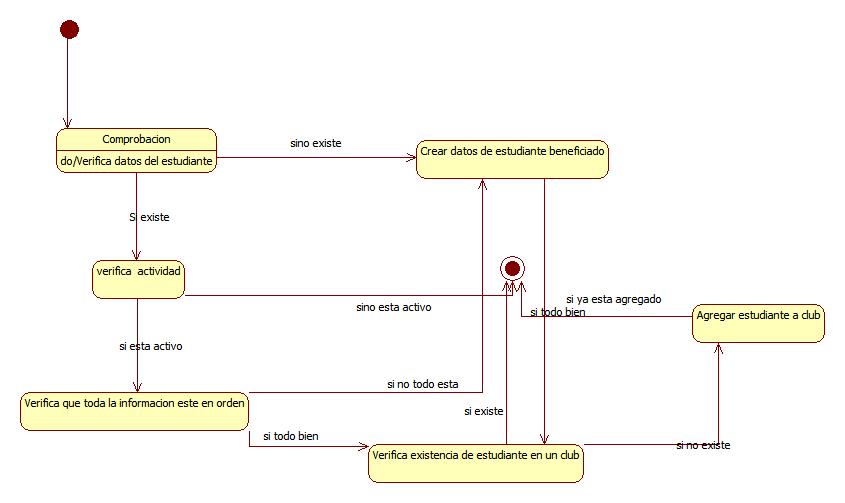


Diagrama de Estados de Profesor beneficiado

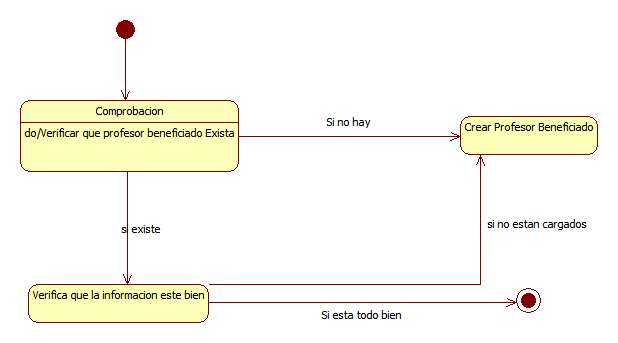


Diagrama de Estados de Colaborador de Oficina

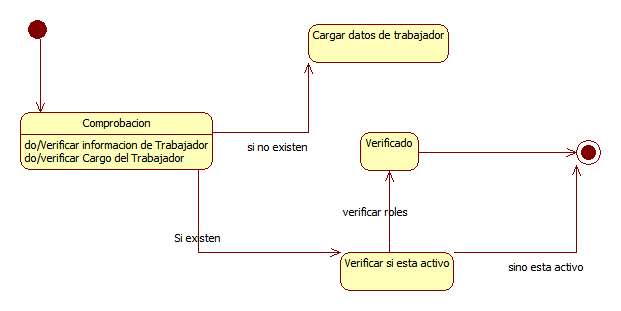


Diagrama de Estados de Voluntario

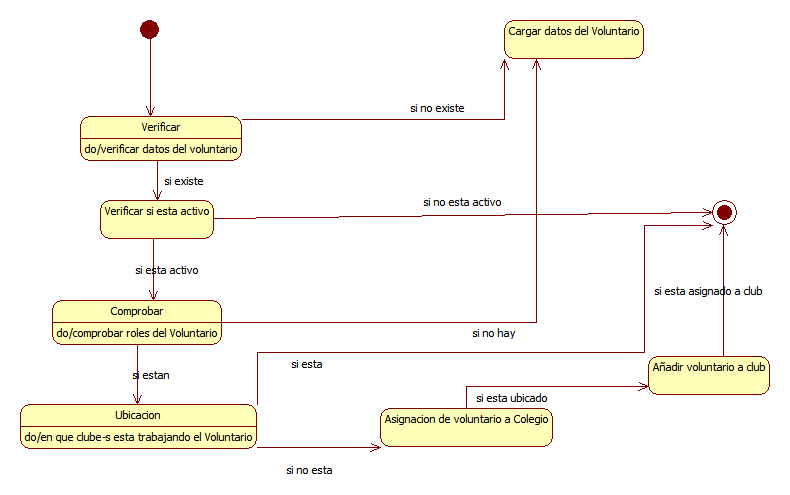
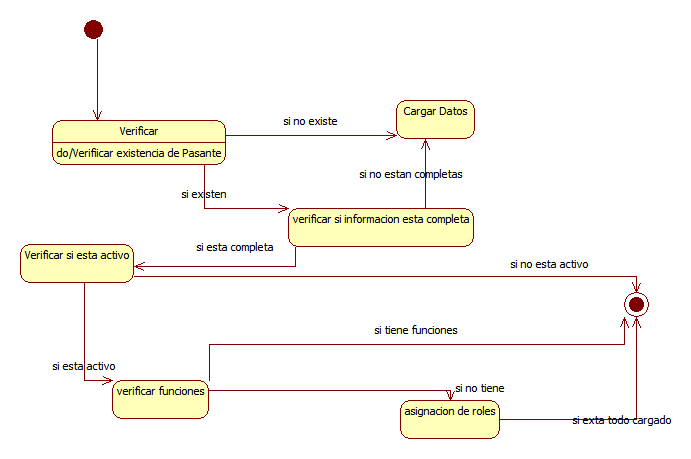
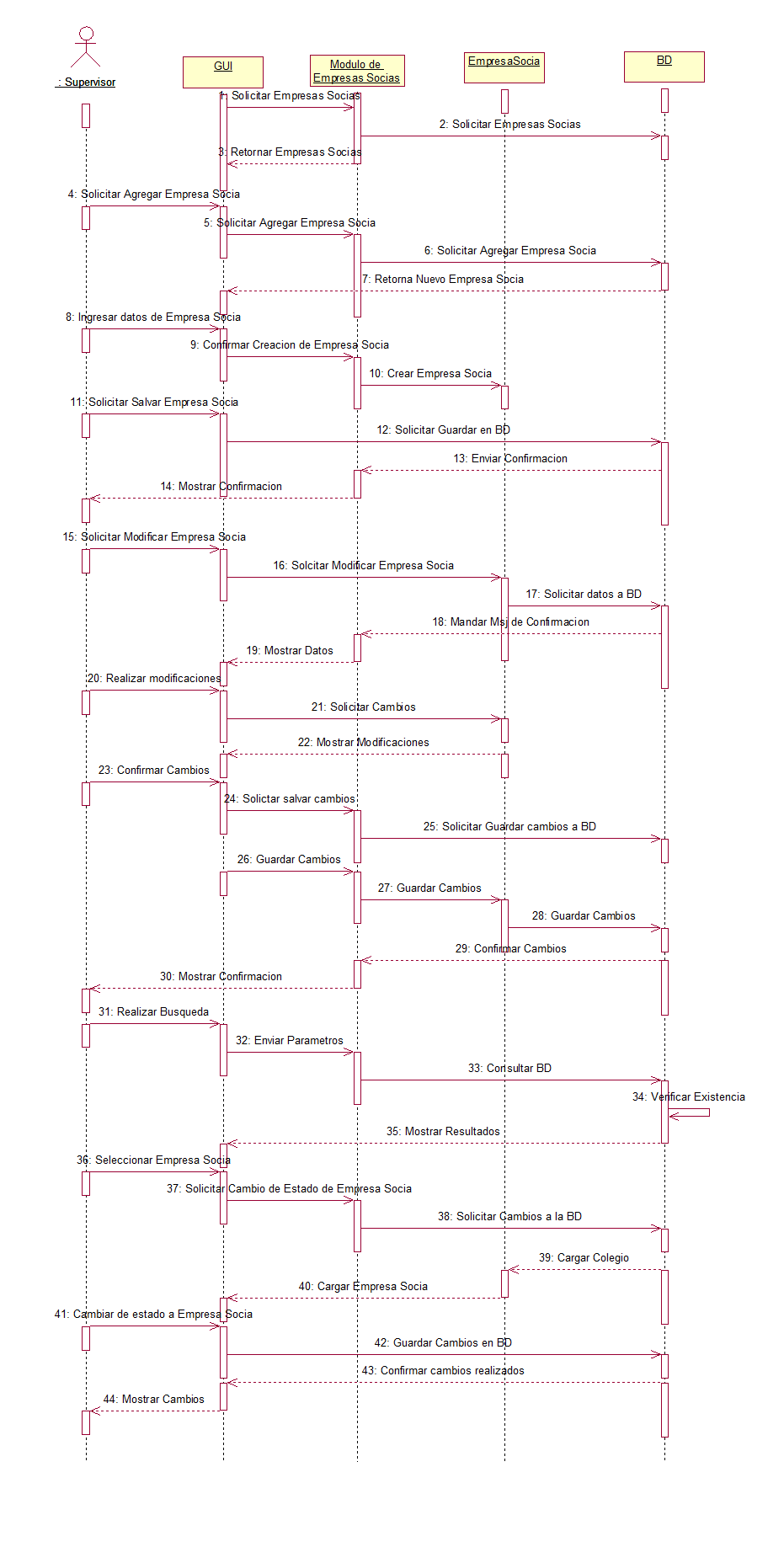


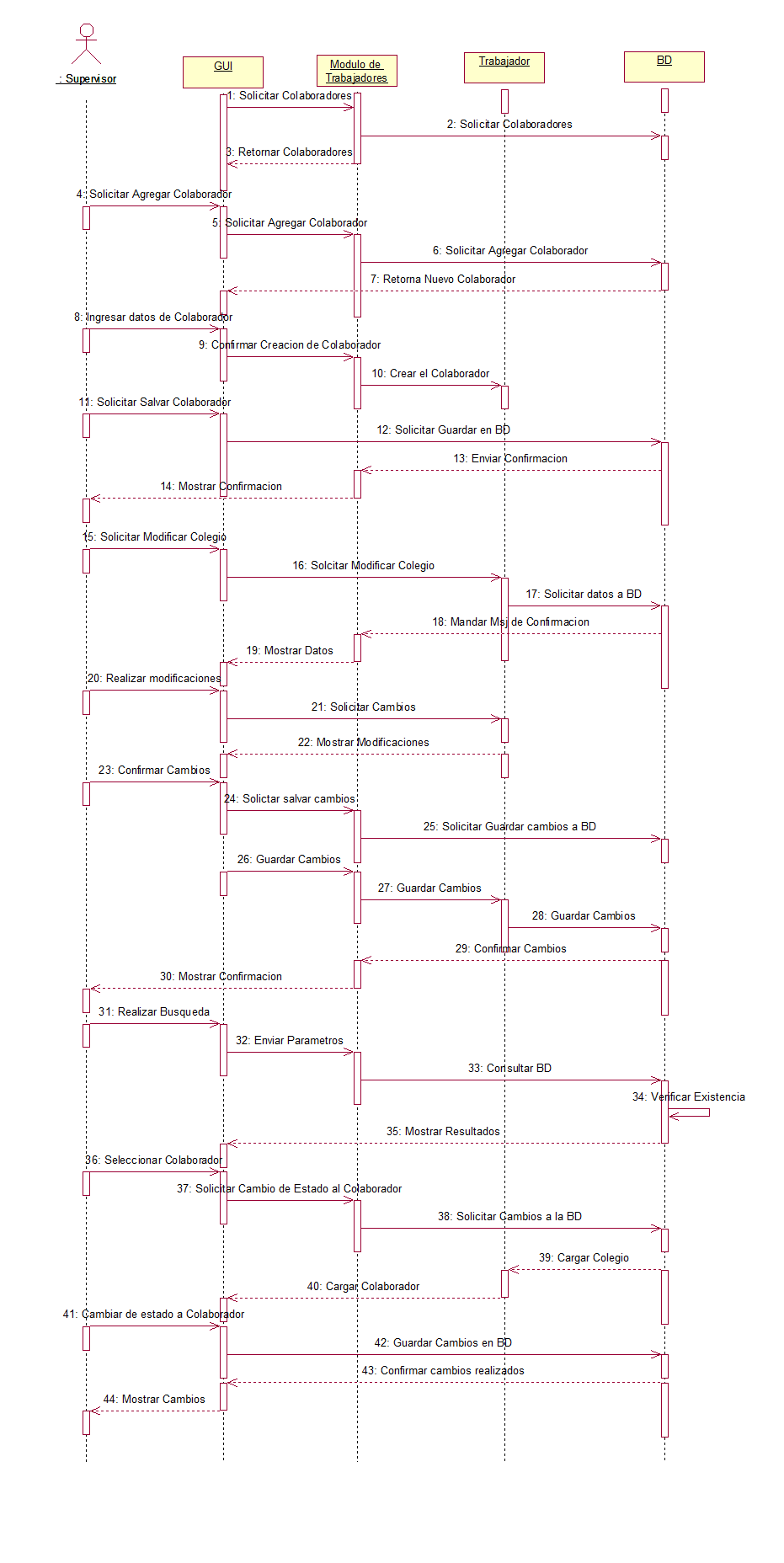
Diagrama de Estados de Pasantes

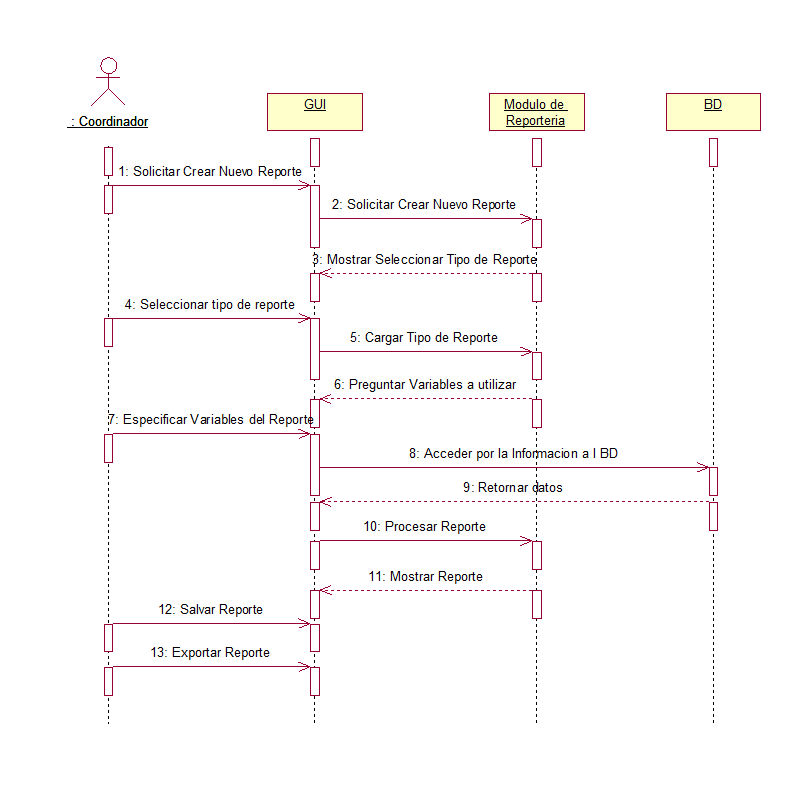


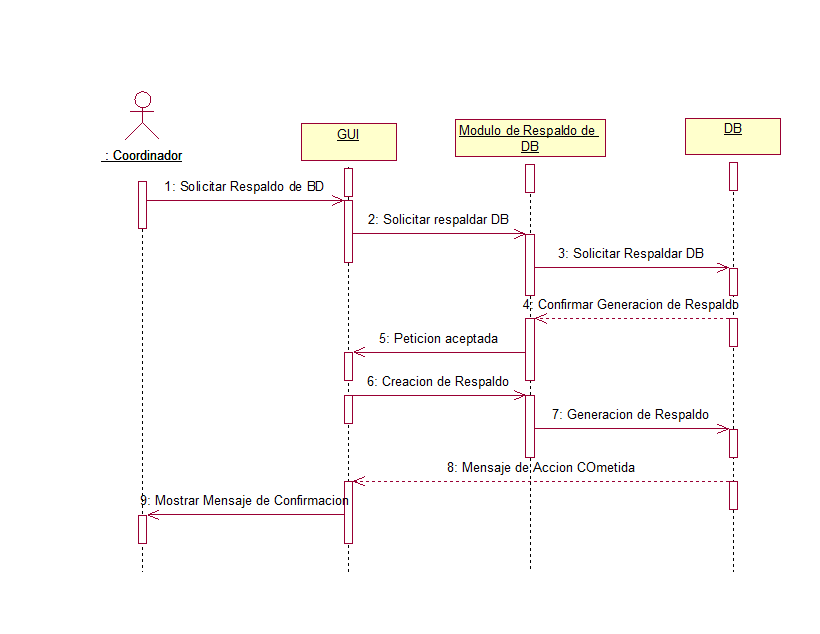
Diagramas de Secuencia

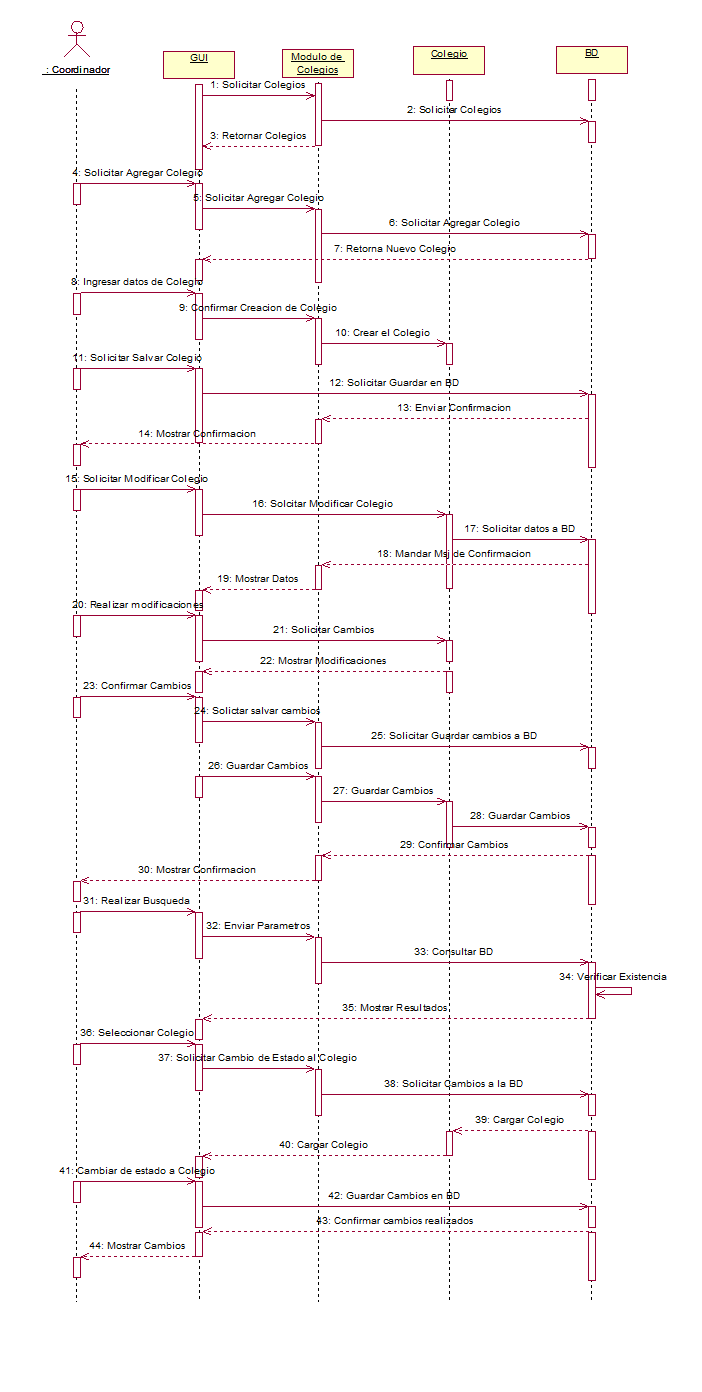
Diagrama de Secuencia de Supervisor y Administración de Usuarios

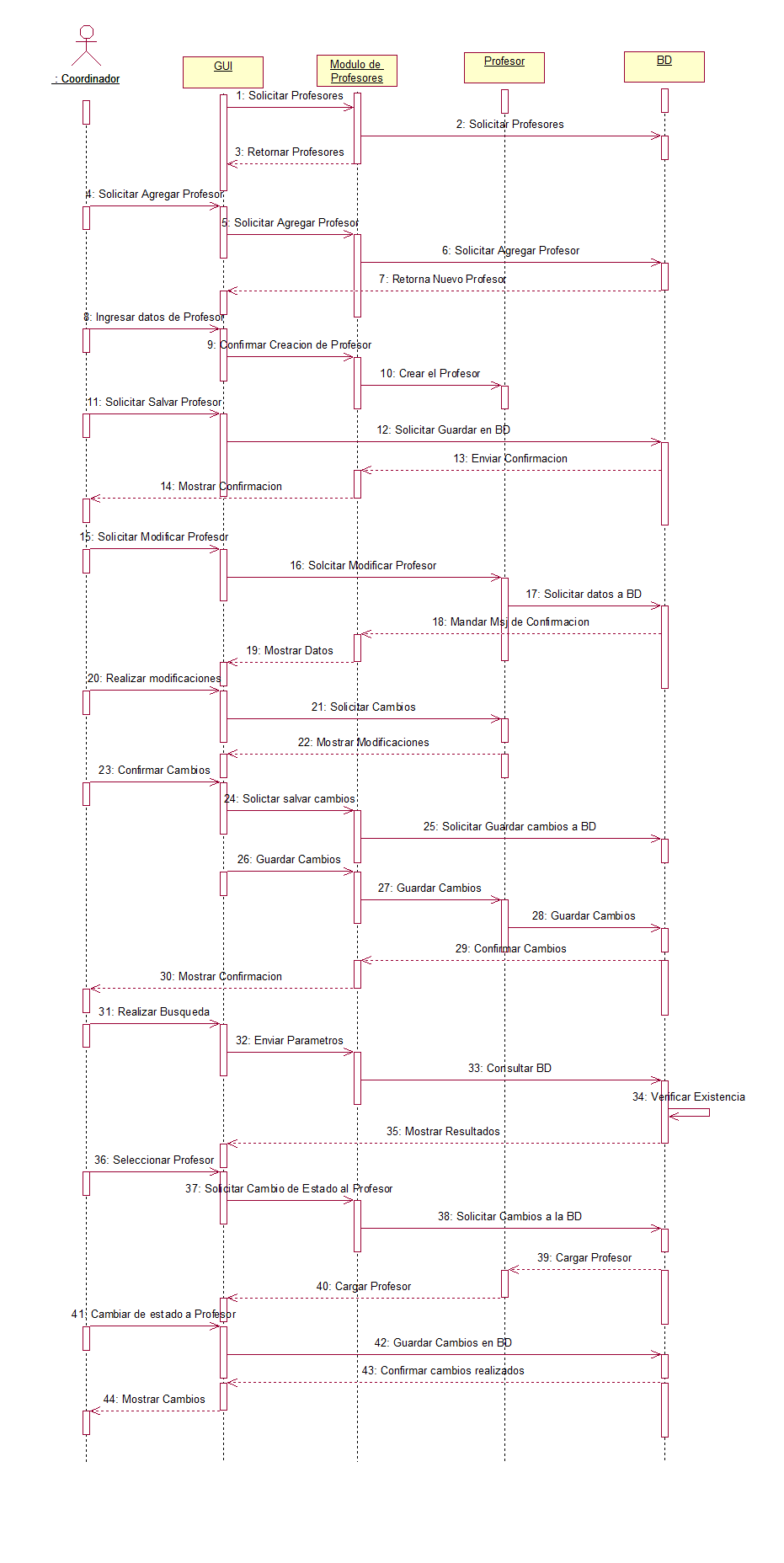
Diagrama de Secuencias de Supervisor y Empresas Socias

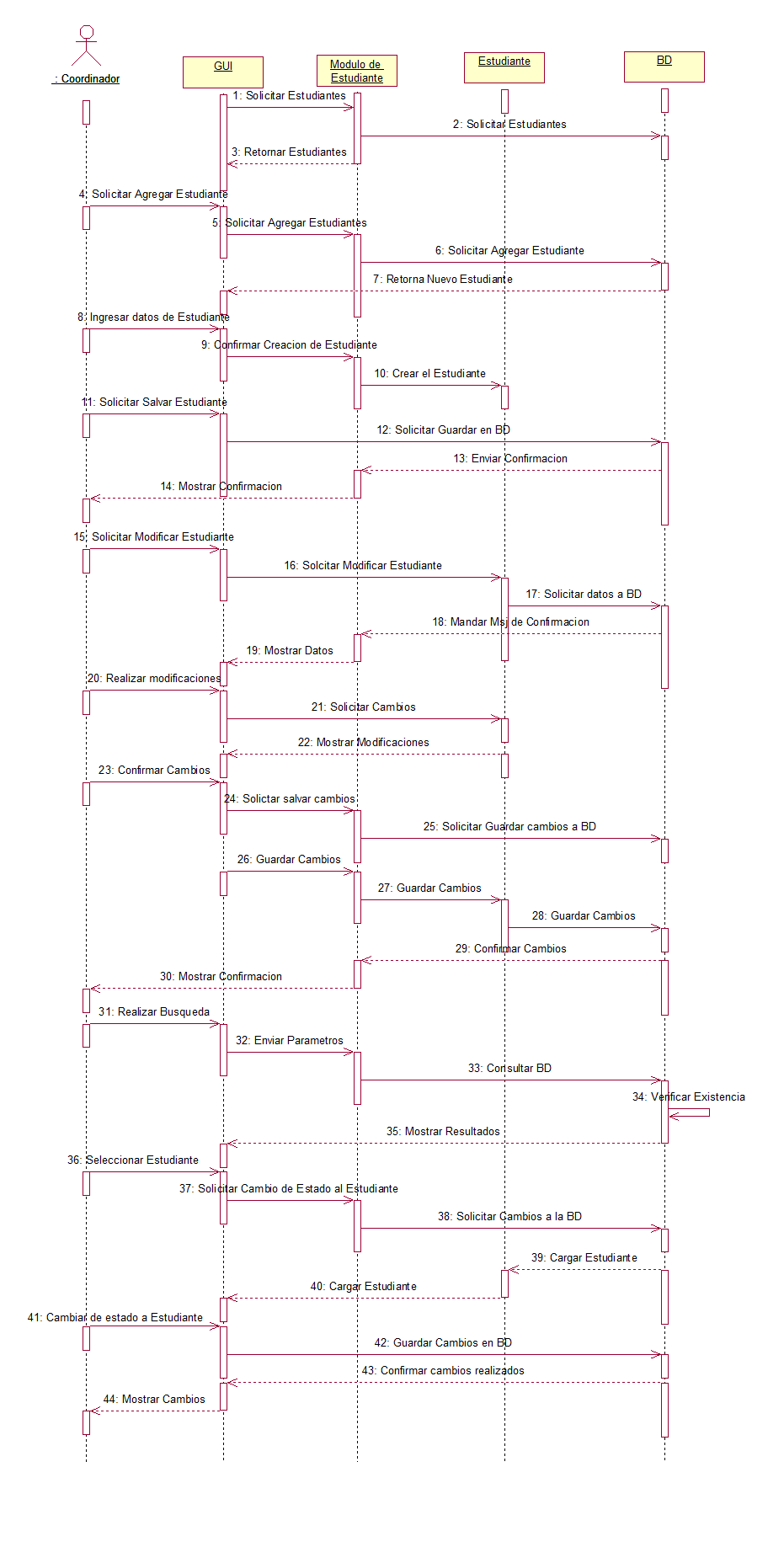
Diagrama de Secuencias de Supervisor y Modulo de Colaboradores

Diagrama de Secuencias de Coordinador y Modulo de Reporteria

Diagrama de Secuencias de Coordinador y Modulo de Respaldo del BD.

Diagrama de Secuencias de Coordinador y Modulo de Colegios beneficiados

Diagrama de Secuencias de Coordinador y Modulo de Profesores beneficiados

Diagrama de Secuencias de Coordinador y Modulo de Estudiantes beneficiados

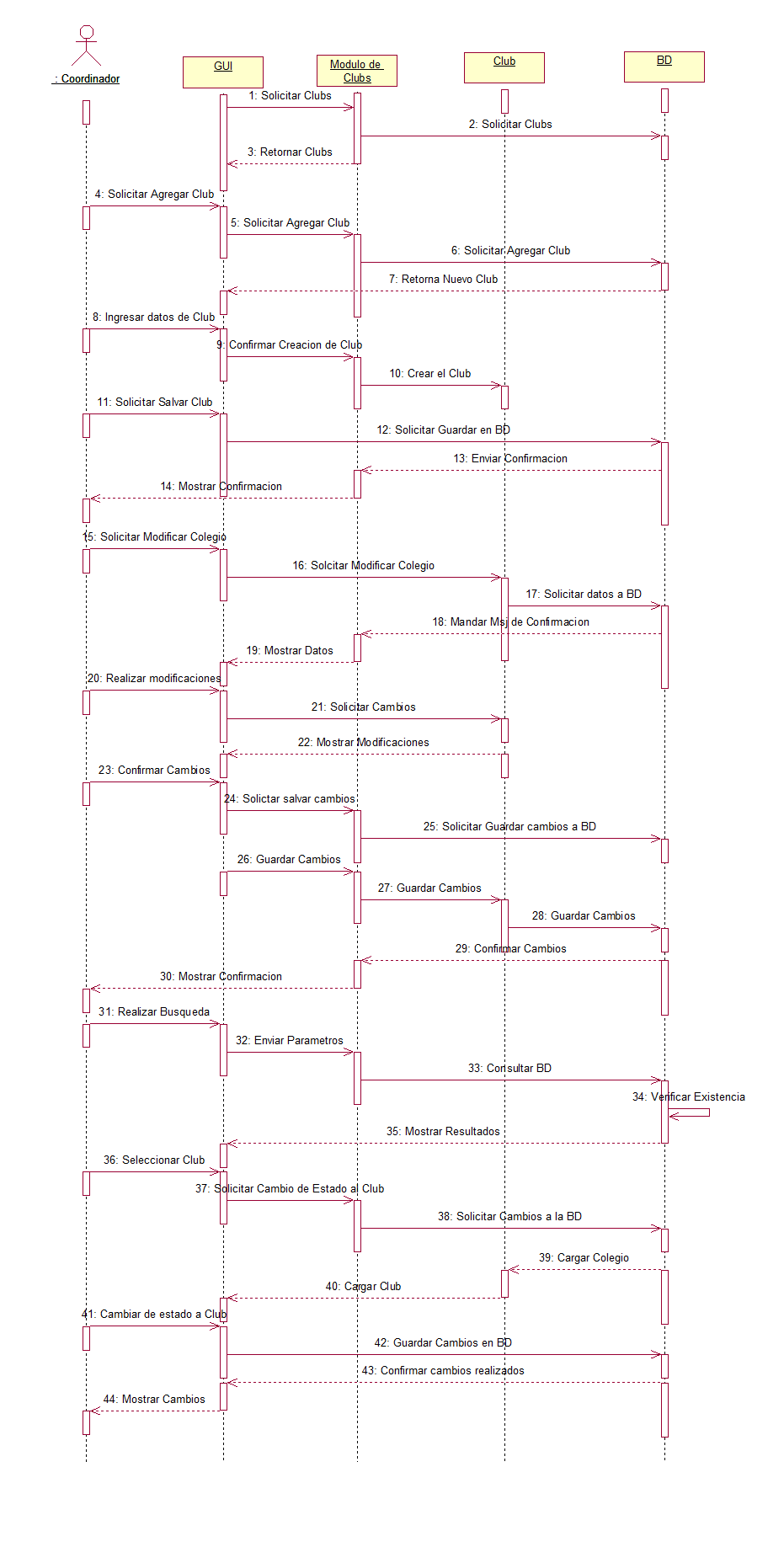
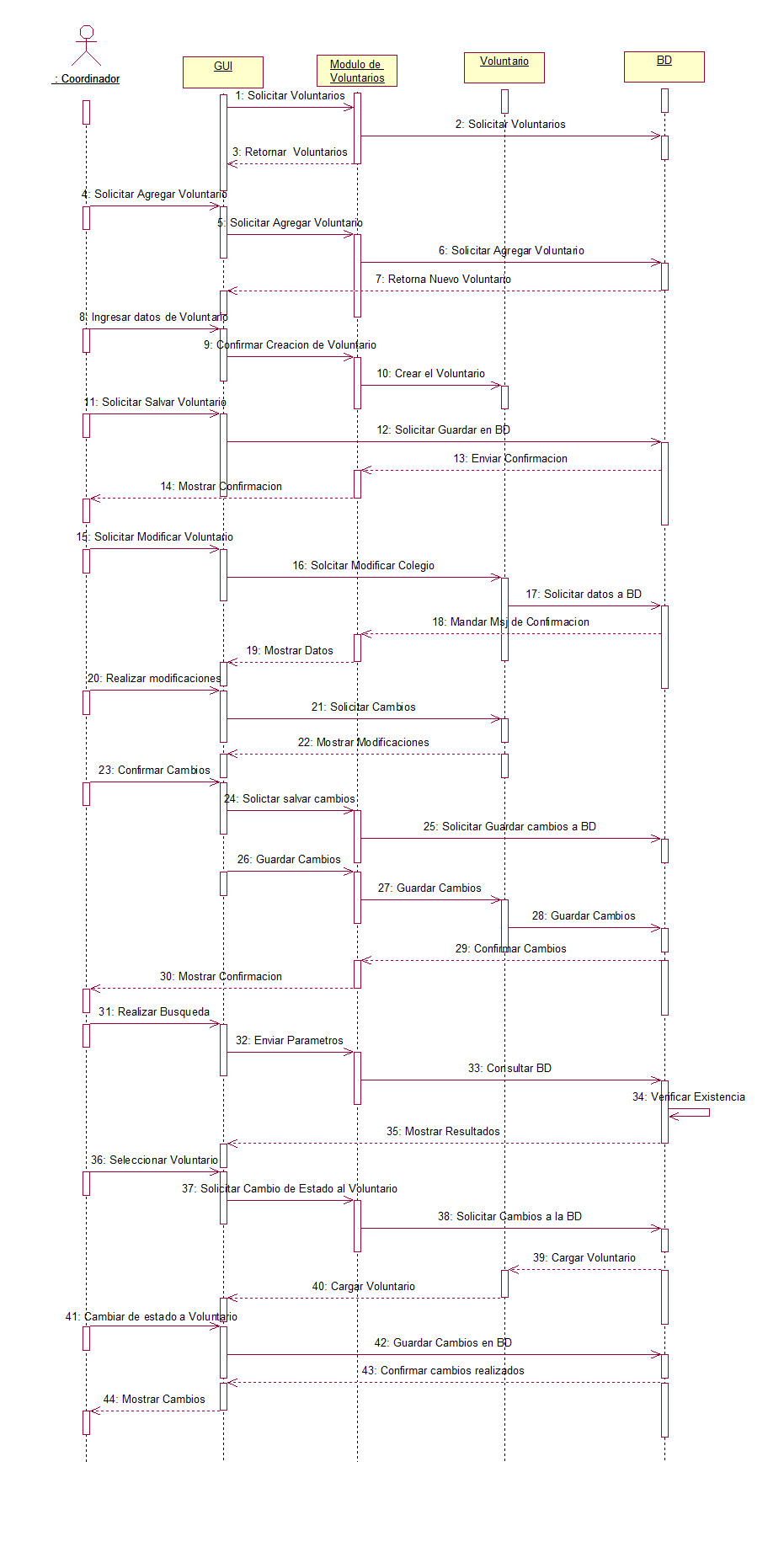
Diagrama de Secuencias de Coordinador y Modulo de Clubs

Diagrama de Secuencias de Coordinador y Modulo de Proyectos

Diagrama de Secuencias de Coordinador y Modulo de Voluntarios

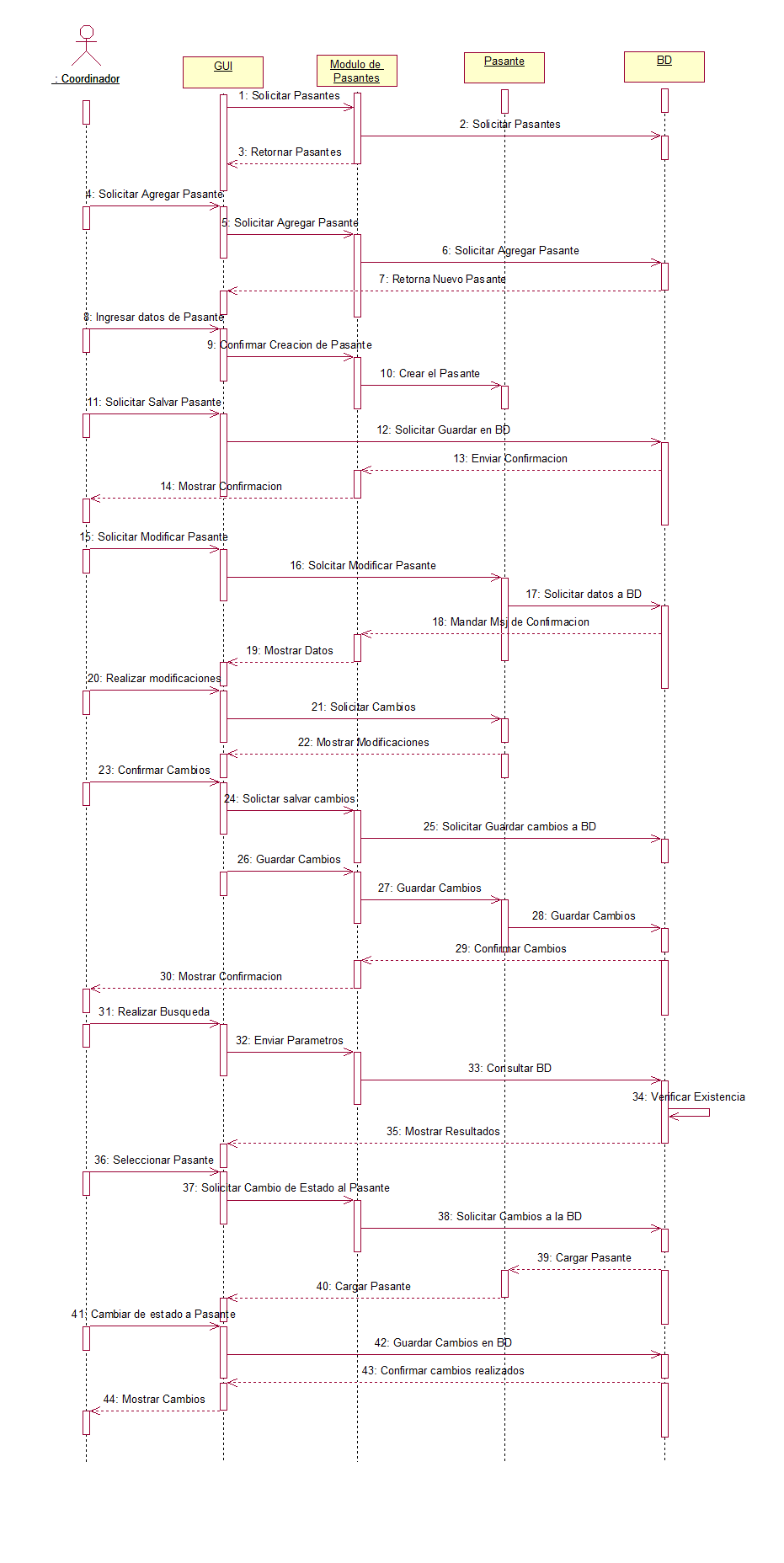
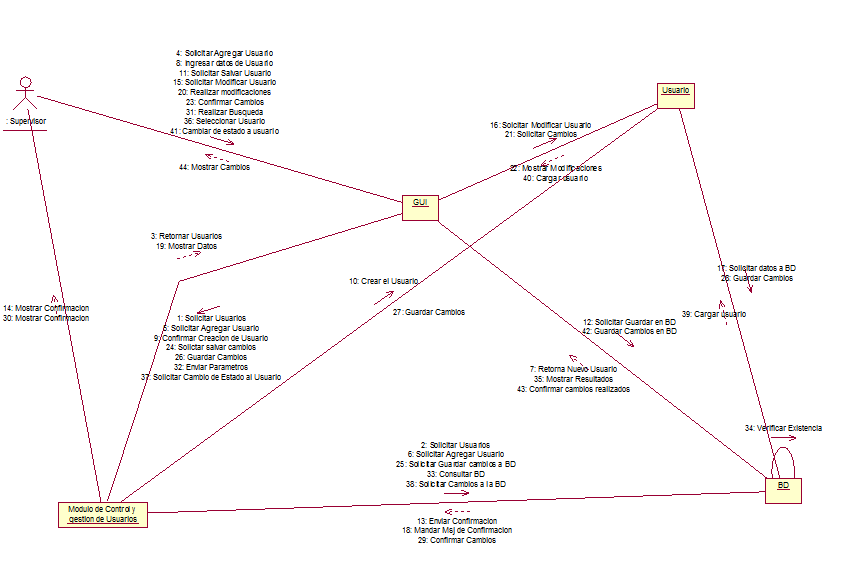
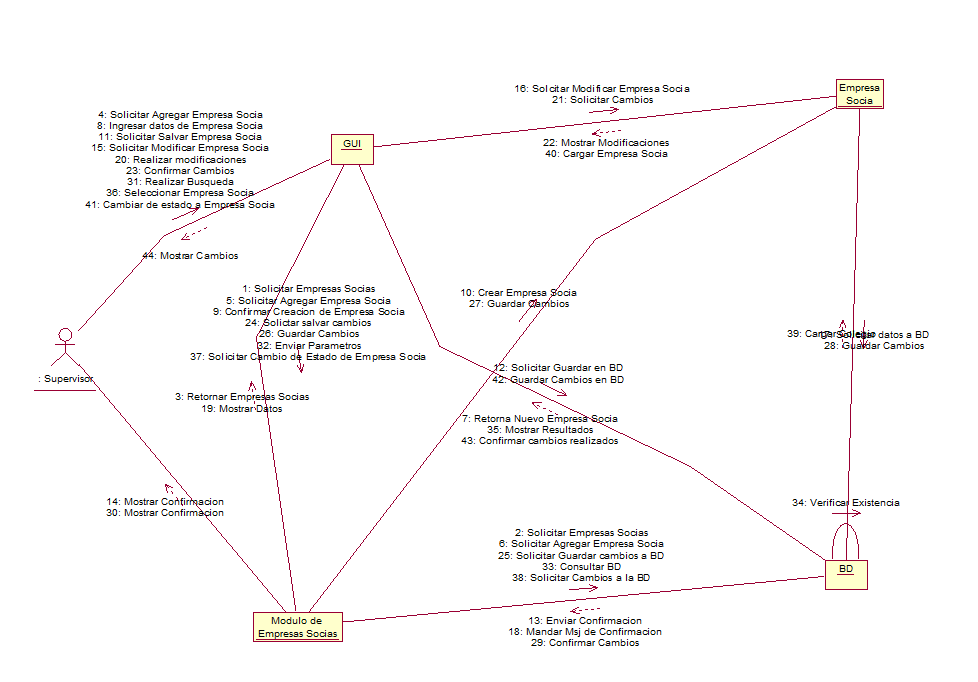
Diagrama de Secuencias de Coordinador y Modulo de Pasantes

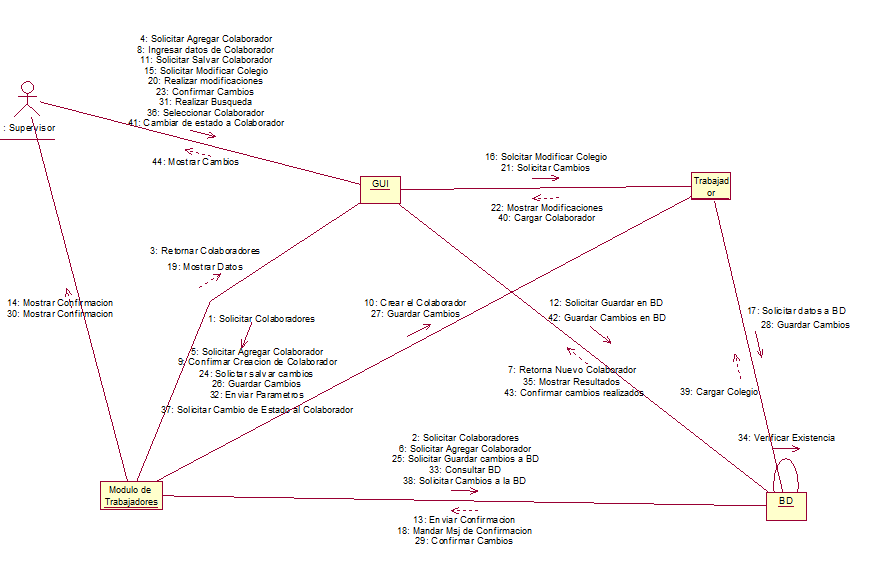
Diagrama de Secuencia General

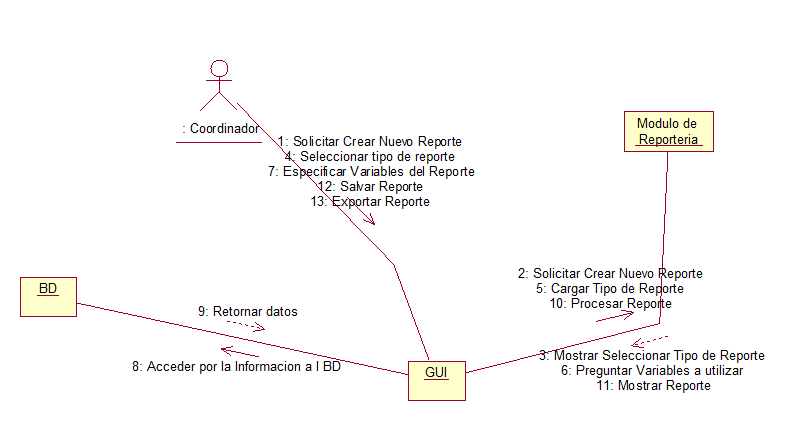
Diagramas de Colaboración (Comunicación)

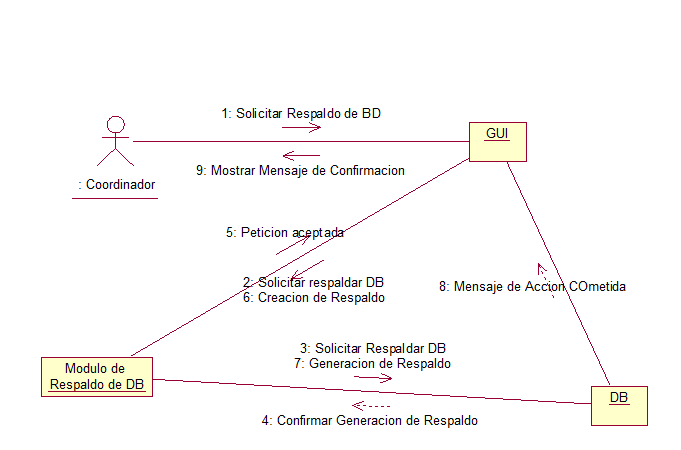
Diagrama de Colaboración de Supervisor y Administración de Usuarios

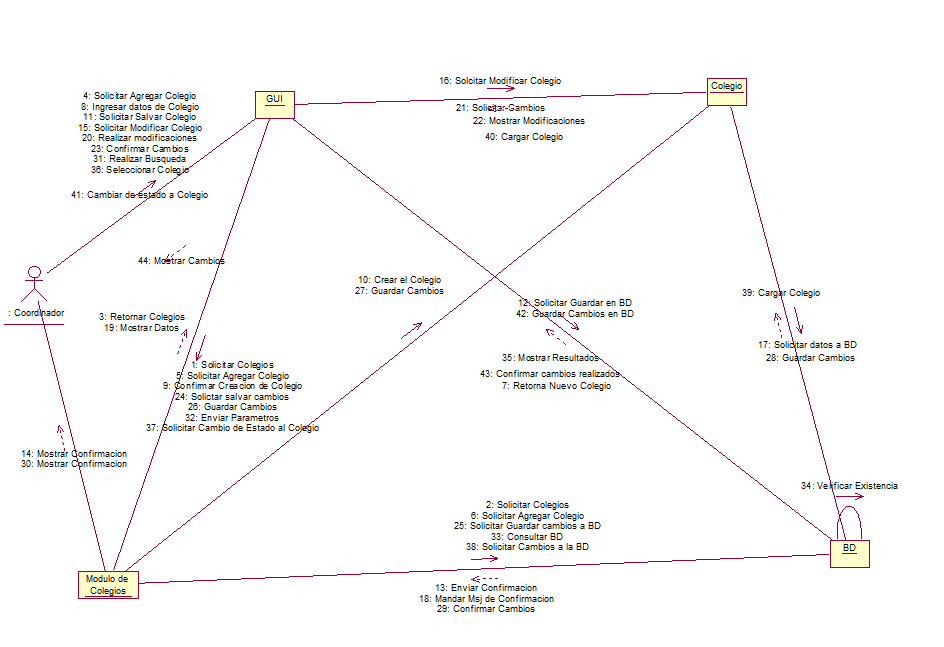


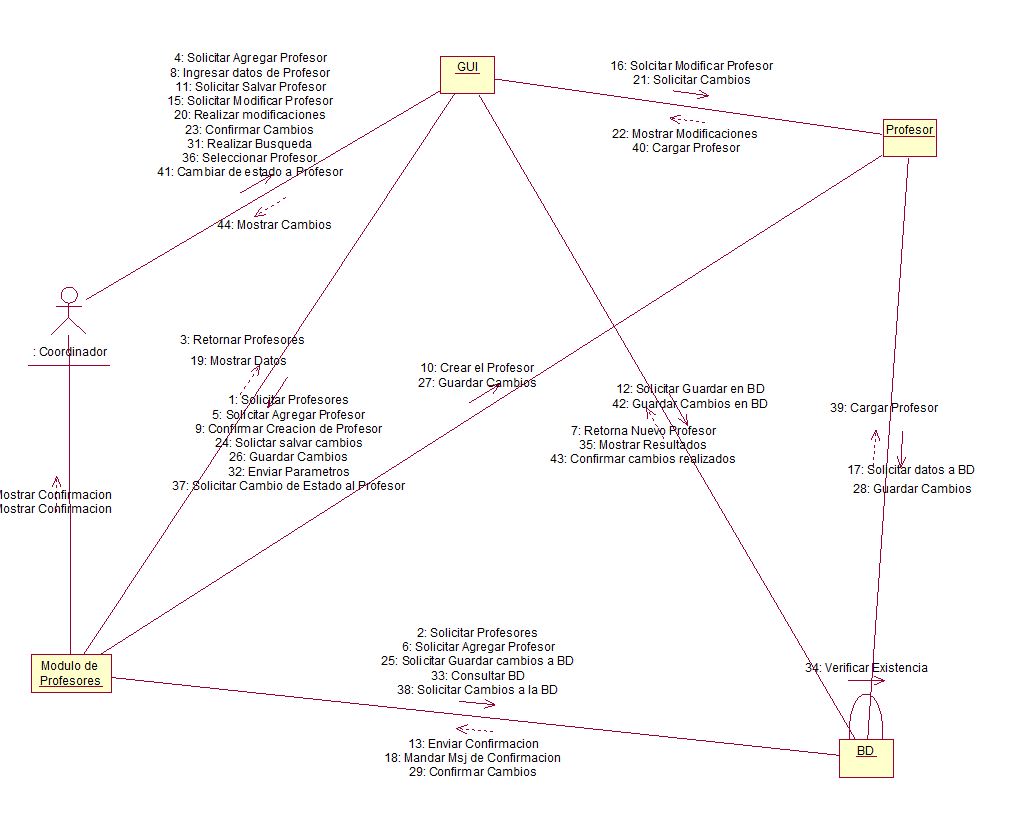
Diagrama de Colaboración de Supervisor y Empresas Socias

Diagrama de Colaboración de Supervisor y Modulo de Colaboradores

Diagrama de Colaboración de Coordinador y Modulo de Reporteria

Diagrama de Colaboración de Coordinador y Modulo de Respaldo del BD.

Diagrama de Colaboración de Coordinador y Modulo de Colegios beneficiados

Diagrama de Colaboración de Coordinador y Modulo de Profesores beneficiados

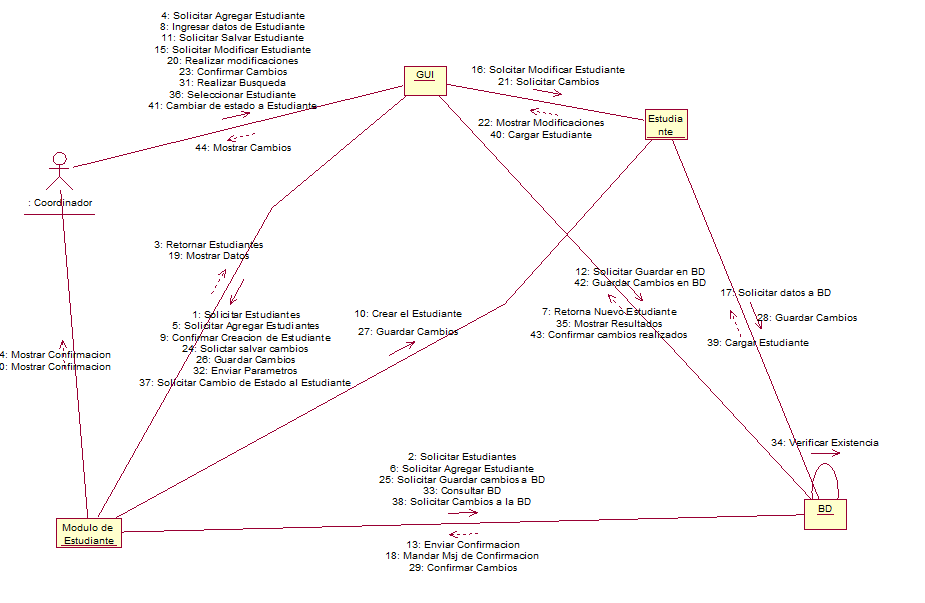
Diagrama de Colaboración de Coordinador y Modulo de Estudiantes beneficiados

Diagrama de Colaboración de Coordinador y Modulo de Clubs



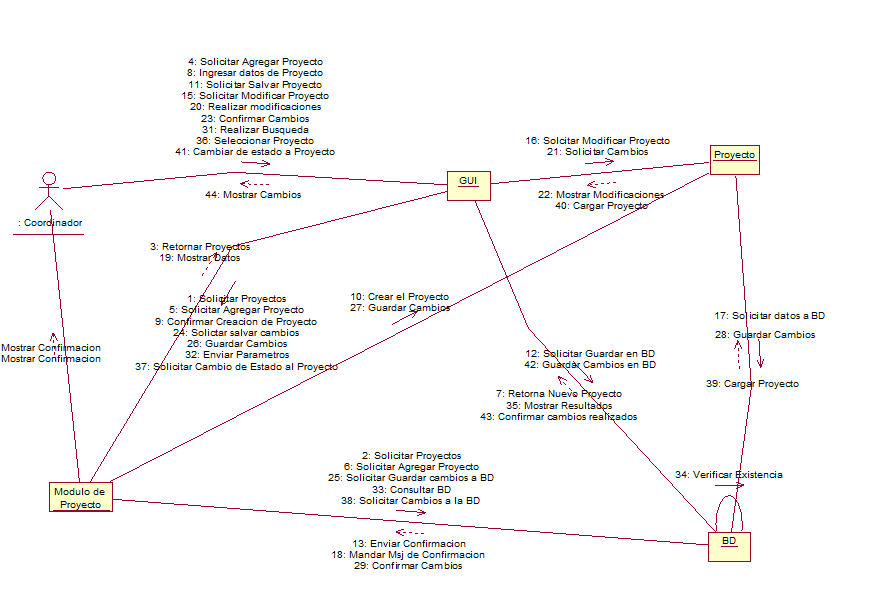
Diagrama de Colaboración de Coordinador y Modulo de Proyectos

Diagrama de Colaboración de Coordinador y Modulo de Voluntarios

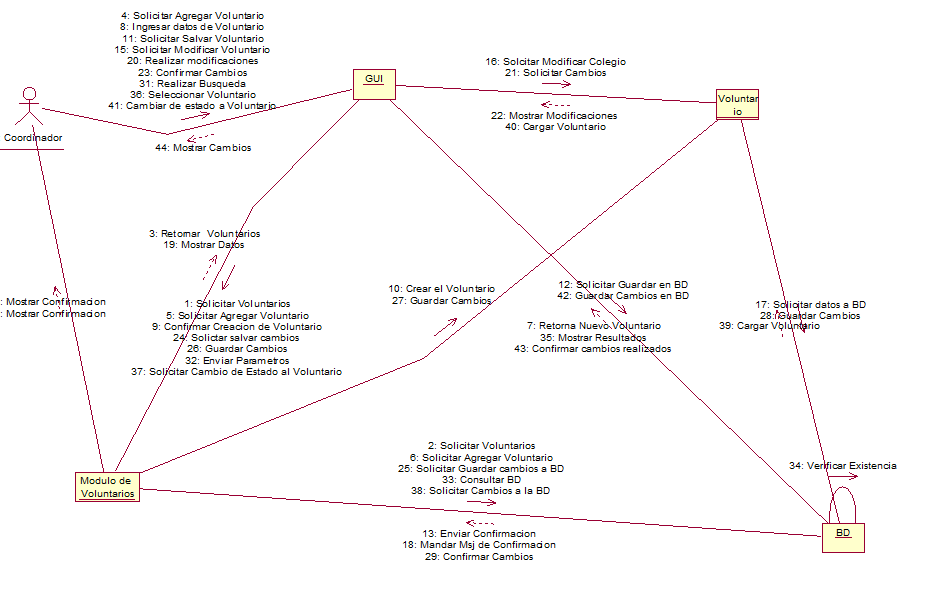
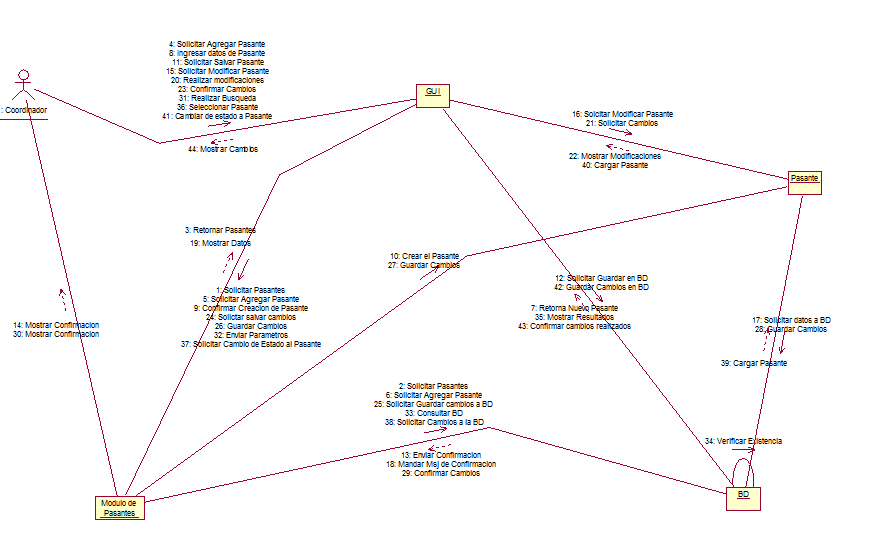
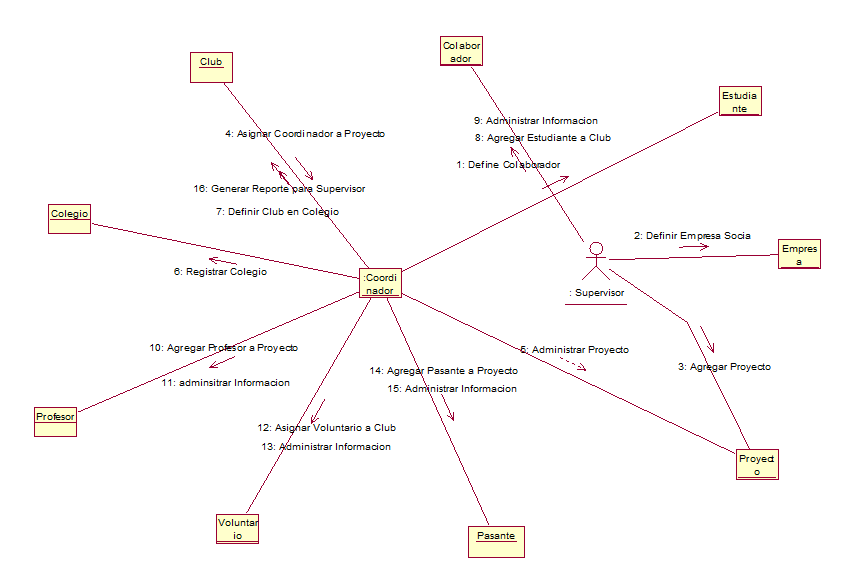
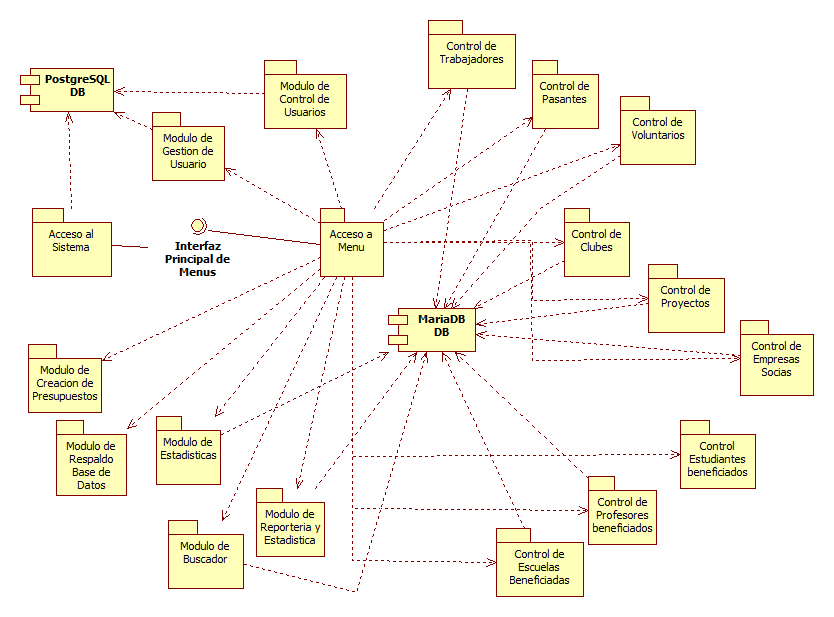


Diagrama de Colaboración de Coordinador y Modulo de Pasantes

Diagrama de Colaboración General

Diagramas de Componentes

Diagramas de Distribución (Despliegue)

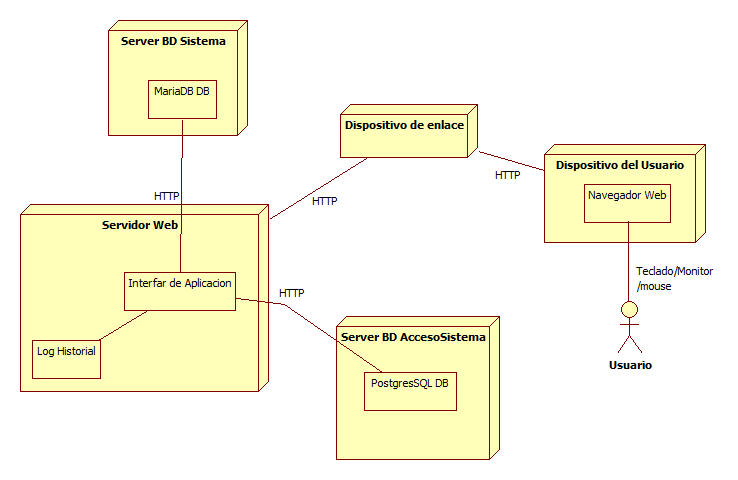


Diagrama de Paquetes

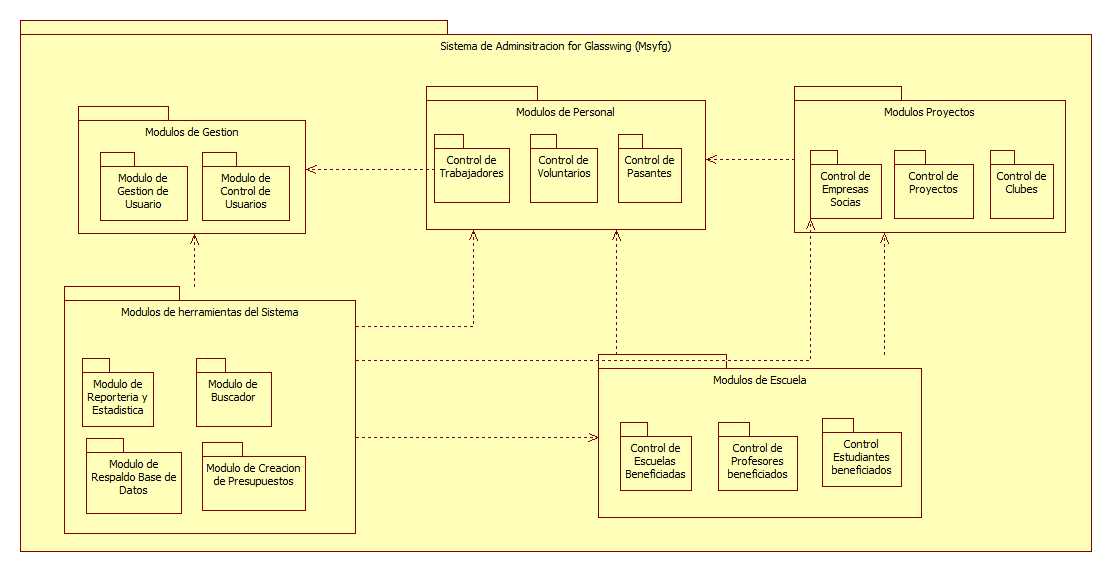
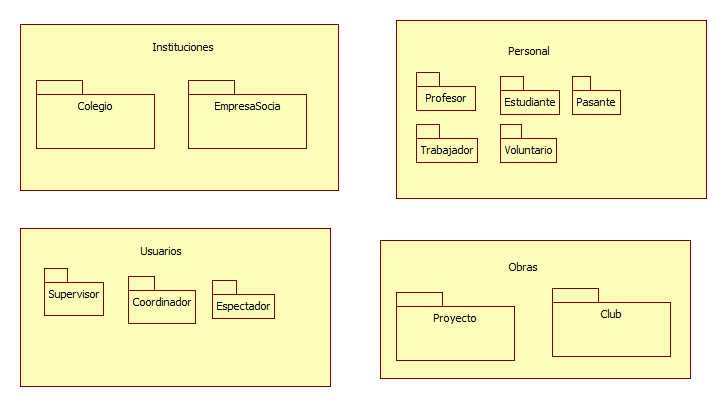
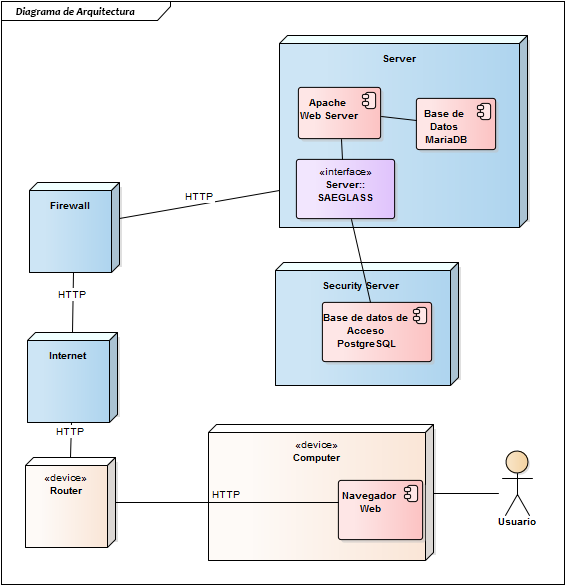
 Diagrama de Paquetes por módulos

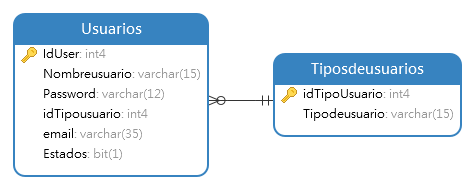
Diagrama de Paquetes del Sistema

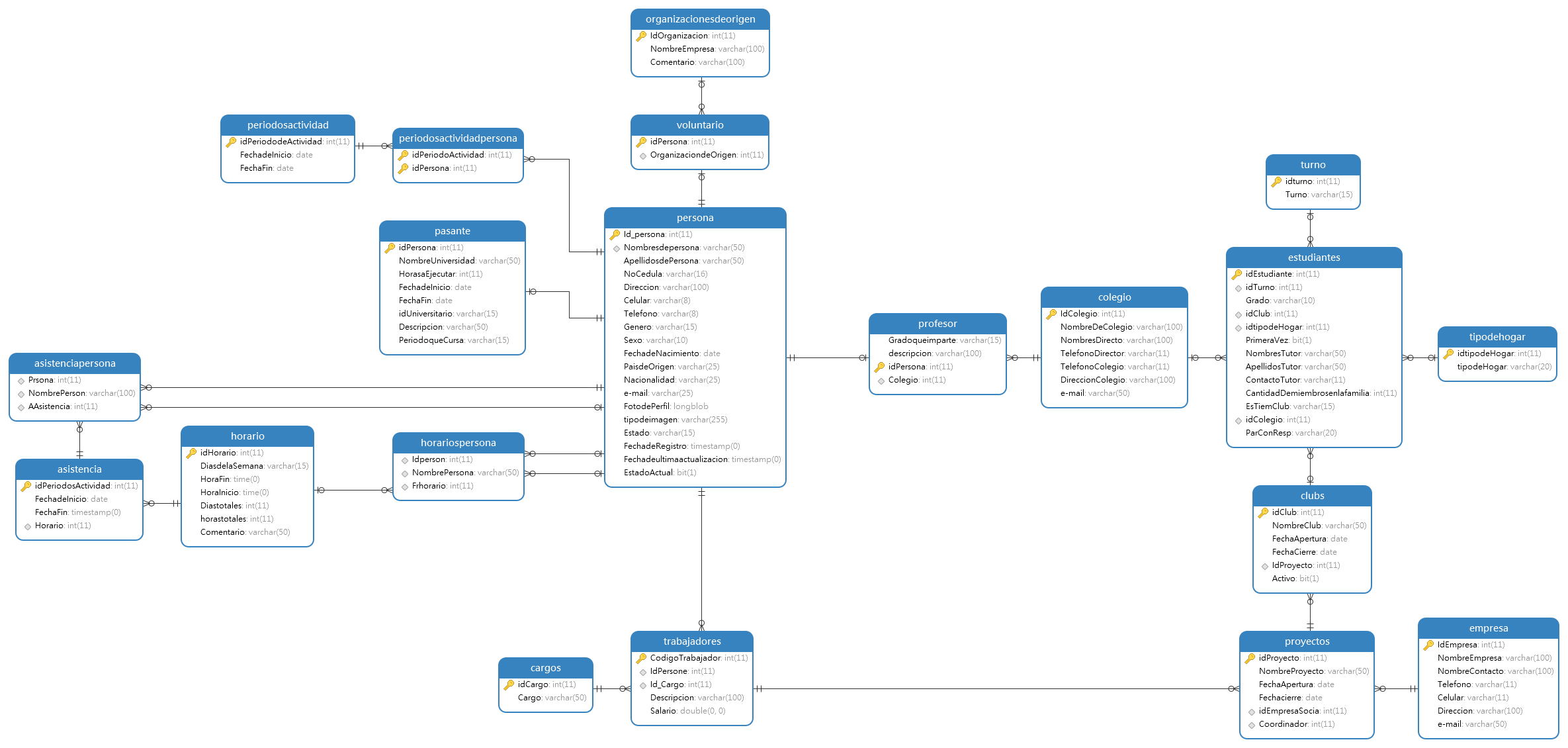
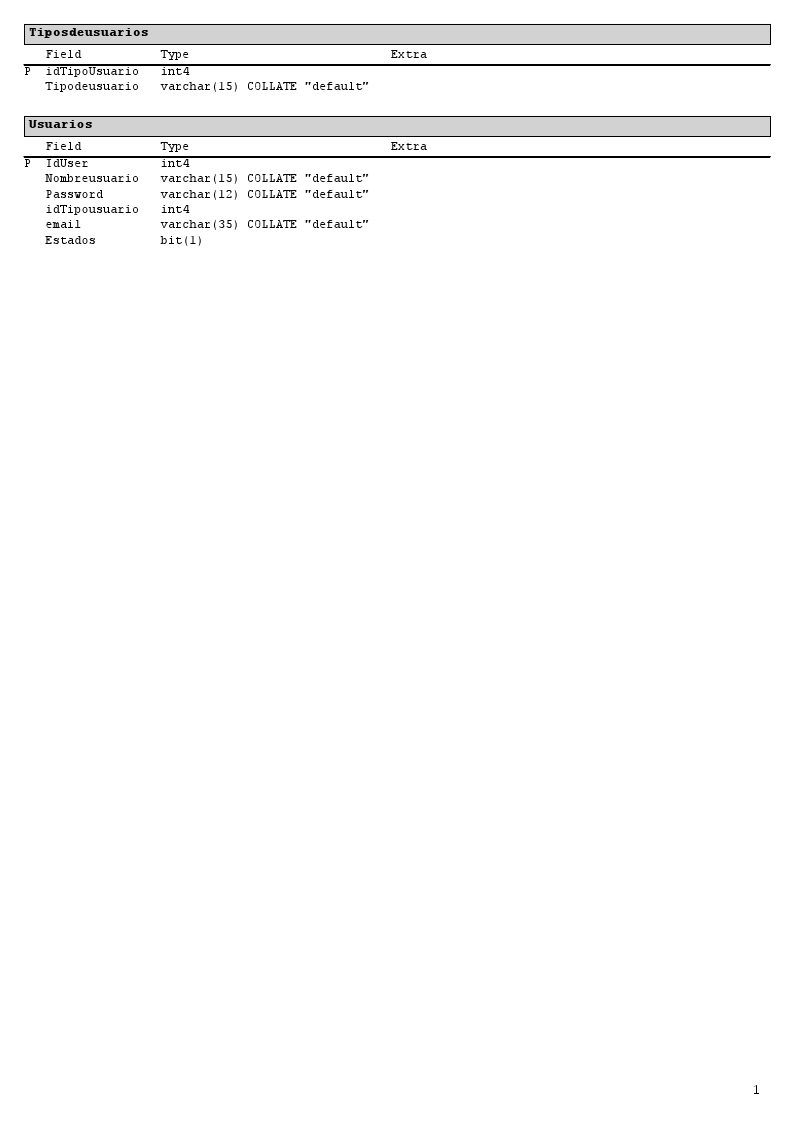
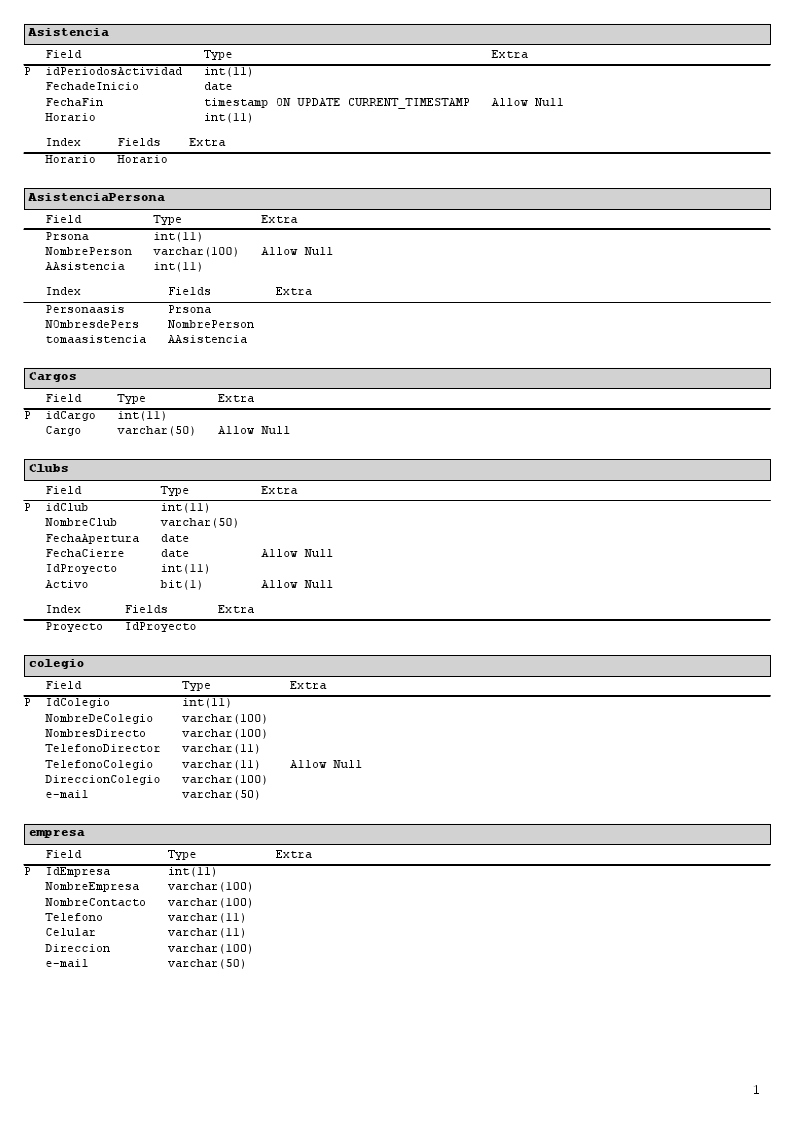


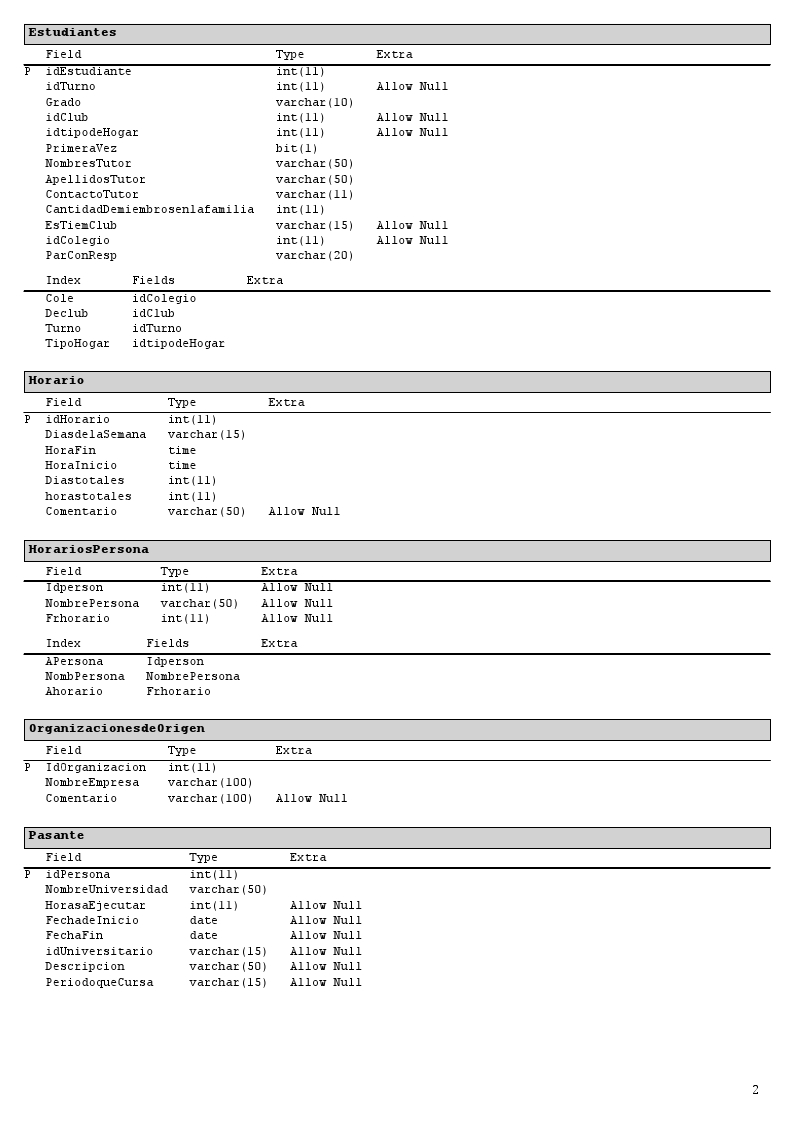
* 1. Diagramas BPMN
  2. Diagrama de Arquitectura



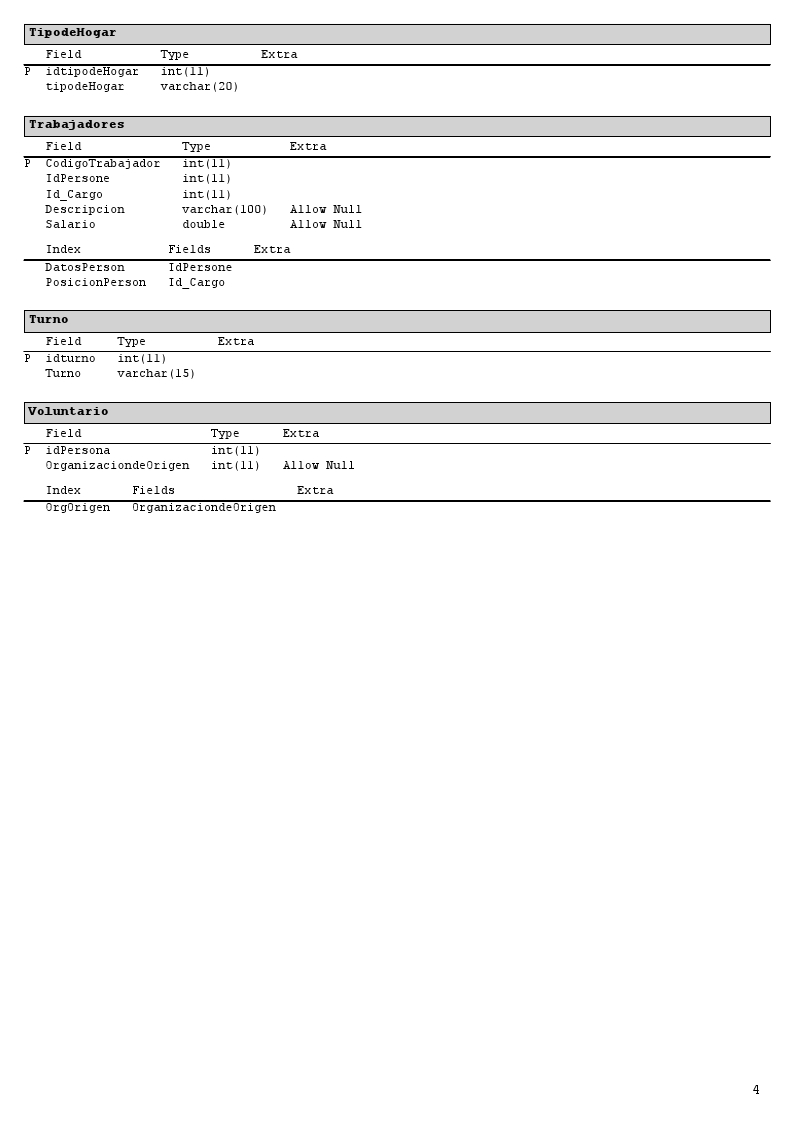
1. Elaborar el modelo de las bases de datos relacionales que se acomoden a lo recolectado durante el análisis de requisitos y requerimientos, que permitirá manejar la información centralizada y un control seguro de usuarios.
   1. Diagramas de Base de datos
      1. Diagrama de Base de datos de Acceso de Usuarios del Sistema



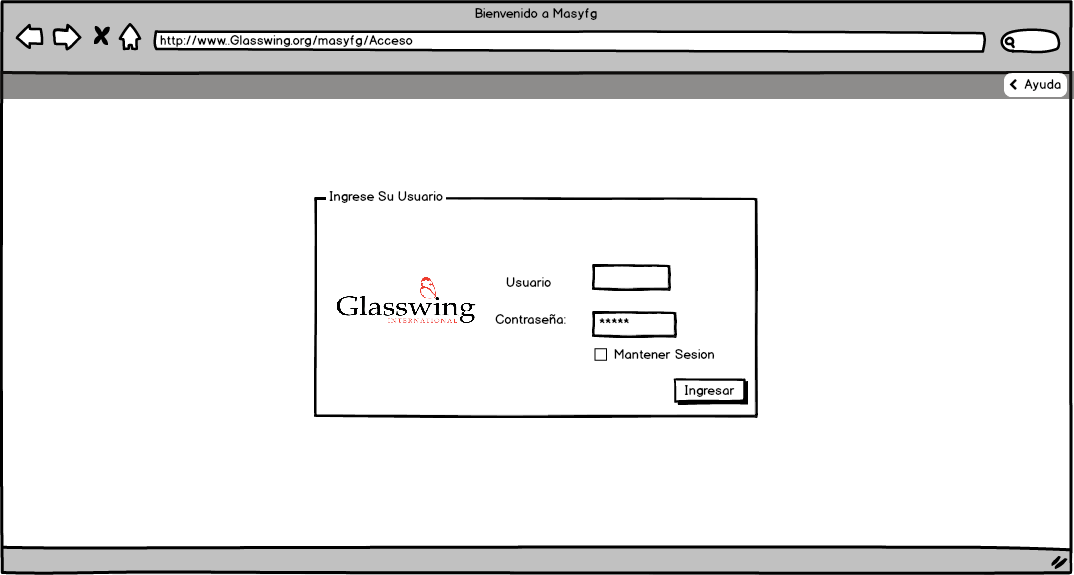
* + 1. Diagrama de Base de datos del Sistema
  1. Diccionario de datos
     1. Diccionario de BD de Accesos
     2. Diccionario de BD de Información

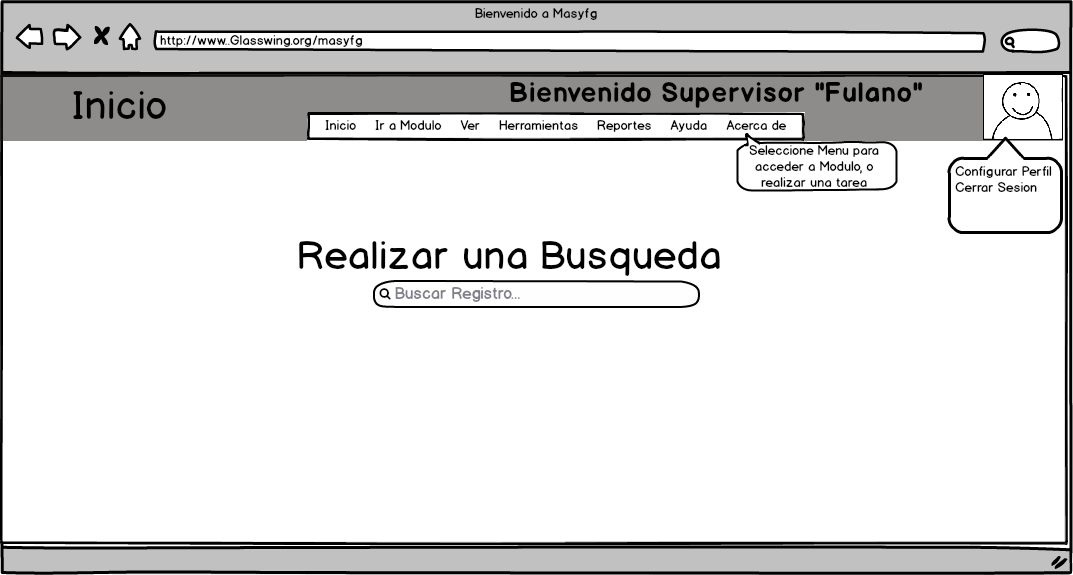


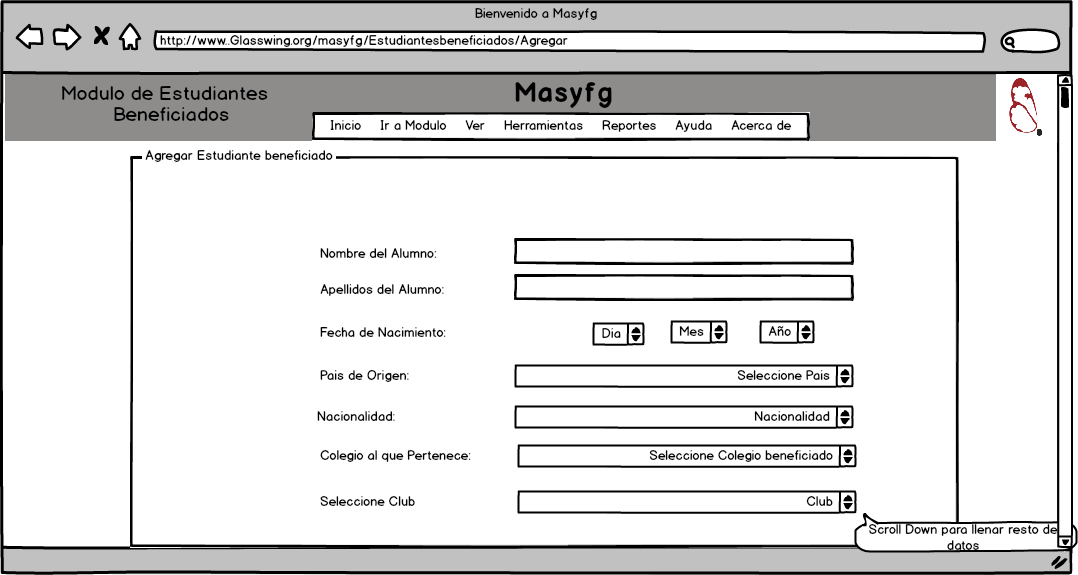


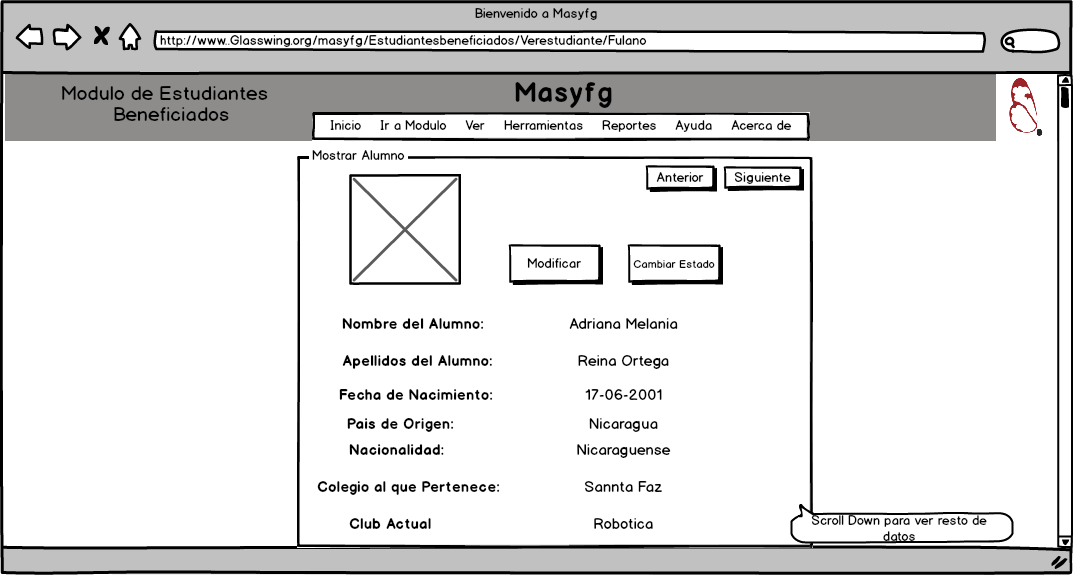


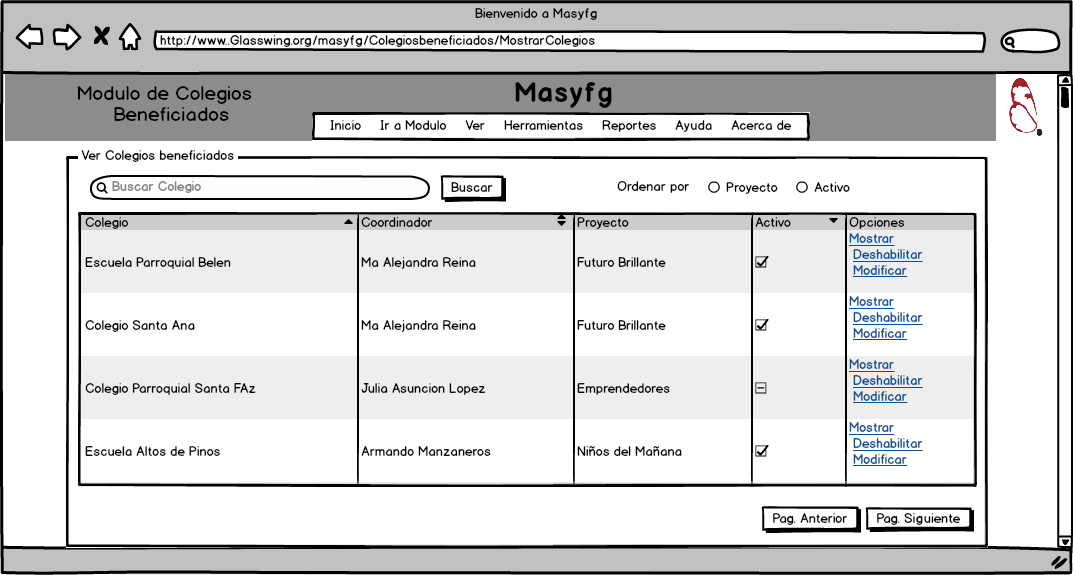
1. Elaborar y Diseñar los prototipos de la interfaz visual de una herramienta que permita a Glasswing monitorear y evaluar los proyectos ejecutados, además de poder, en base a los requisitos y requerimientos que la gerencia de Glasswing brinde.
   1. Prototipos
      1. Vistas de Wireframes (Bocetos)

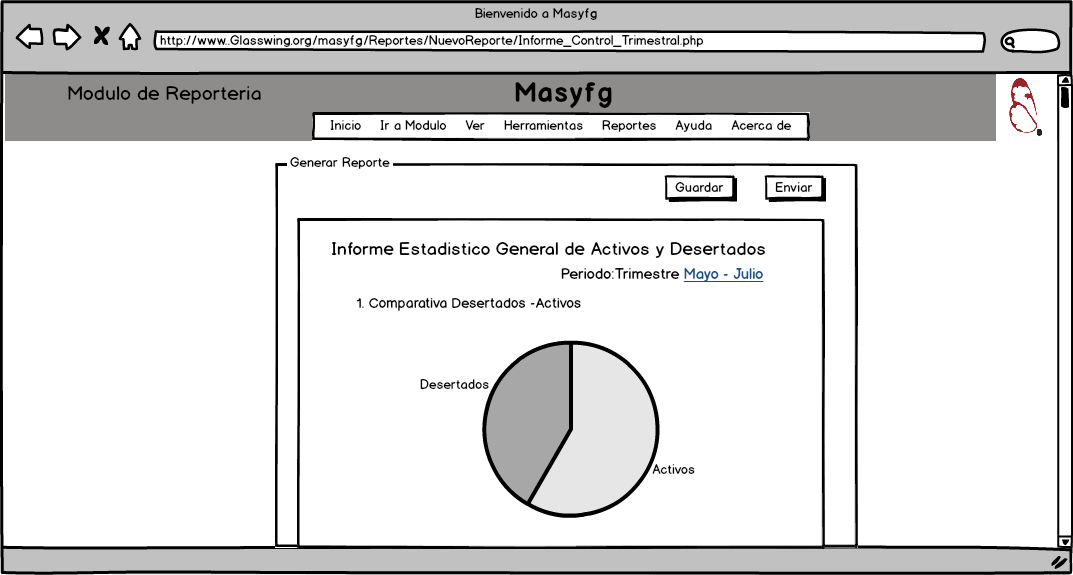
Boceto de Pantalla de Acceso

Boceto de Pantalla de Inicio del Sistema

Boceto de Pantalla de Agregar Nuevo Estudiante

Boceto de Pantalla de Mostrar Estudiante Beneficiado

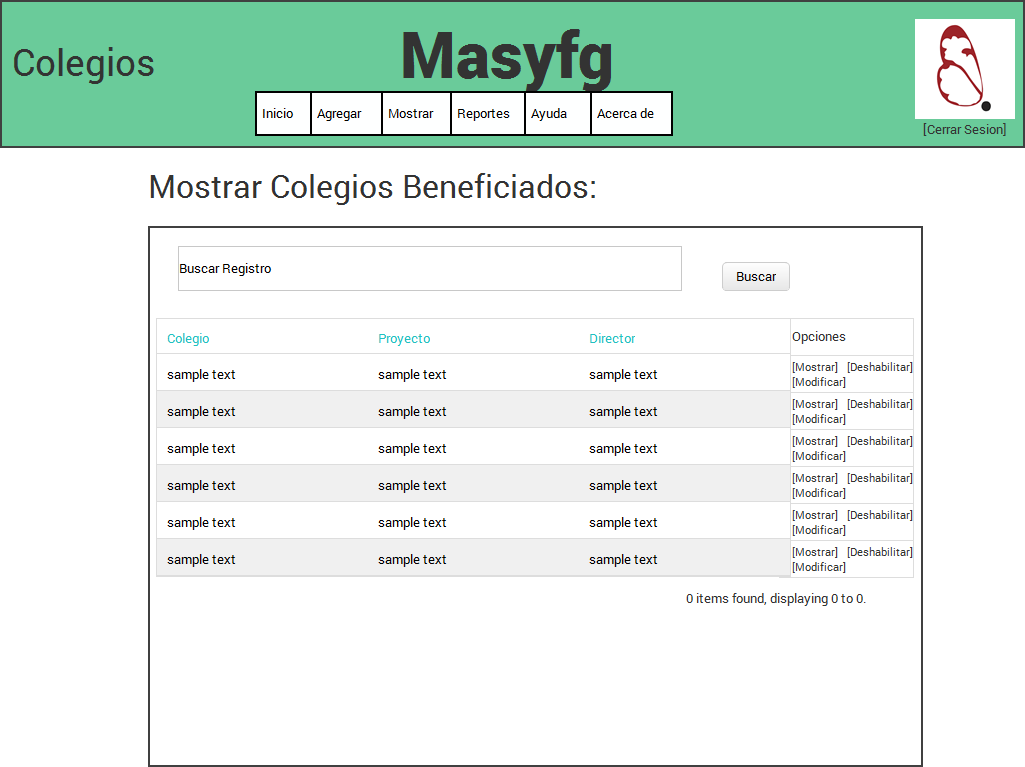
Boceto de Pantalla de Mostrar Colegios Beneficiados

Boceto de Pantalla de Generar Reporte

* + 1. Vistas de Mockups (Prototipos de Pantallas del Sistema)

Mockup de Inicio

Mockups de Colegios Beneficiados

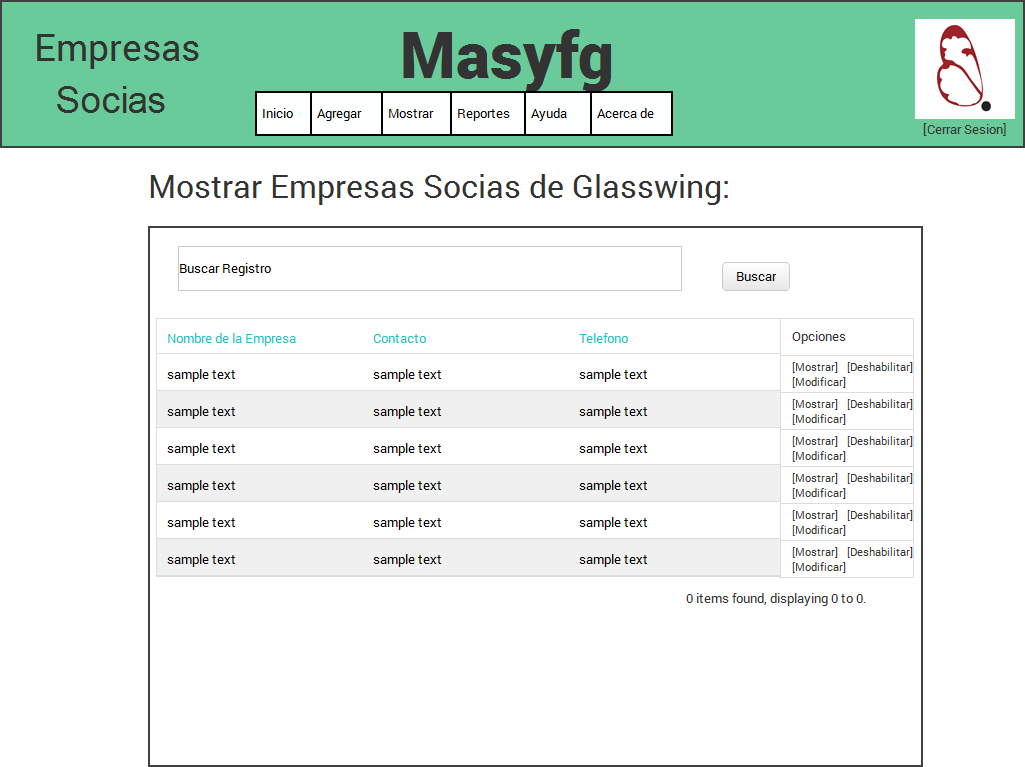
Vista de Pantalla Principal de Colegios Beneficiados

Vista de Pantalla de Agregar Colegio Beneficiado

Vista de Pantalla de Mostrar Colegio Beneficiado



Mockups de Empresas Socias

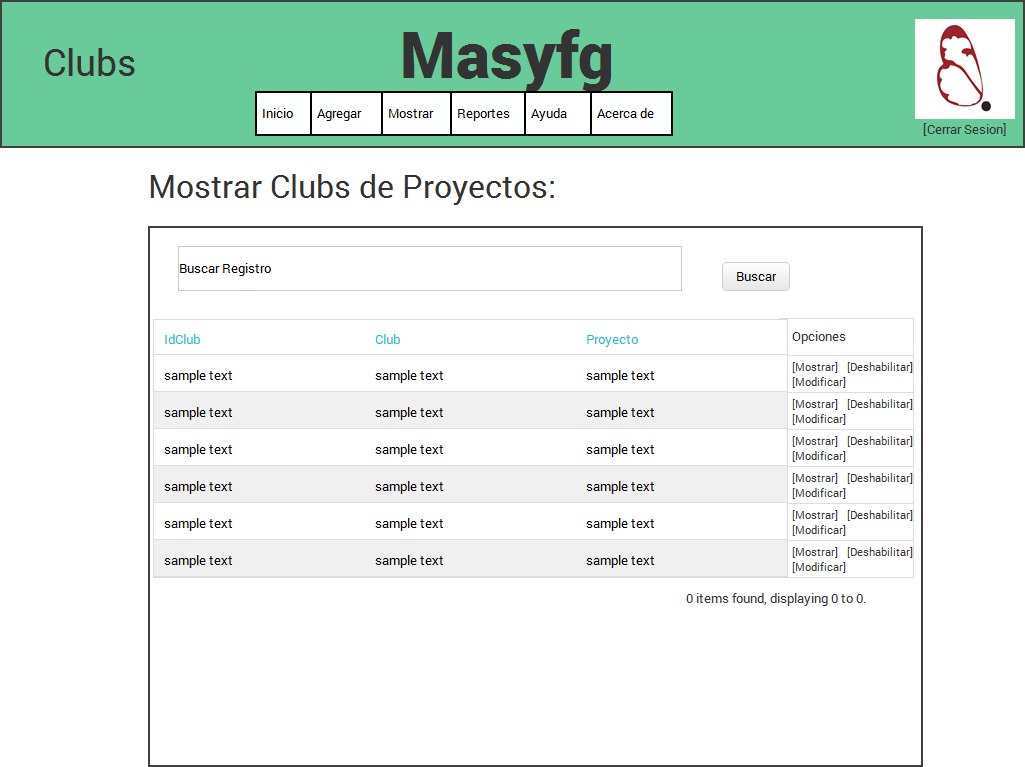
Vista de Pantalla Principal de Empresas Socias

Vista de Pantalla de Agregar Empresa Socia

Vista de Pantalla de Mostrar Empresa Socia



Mockups de Clubs

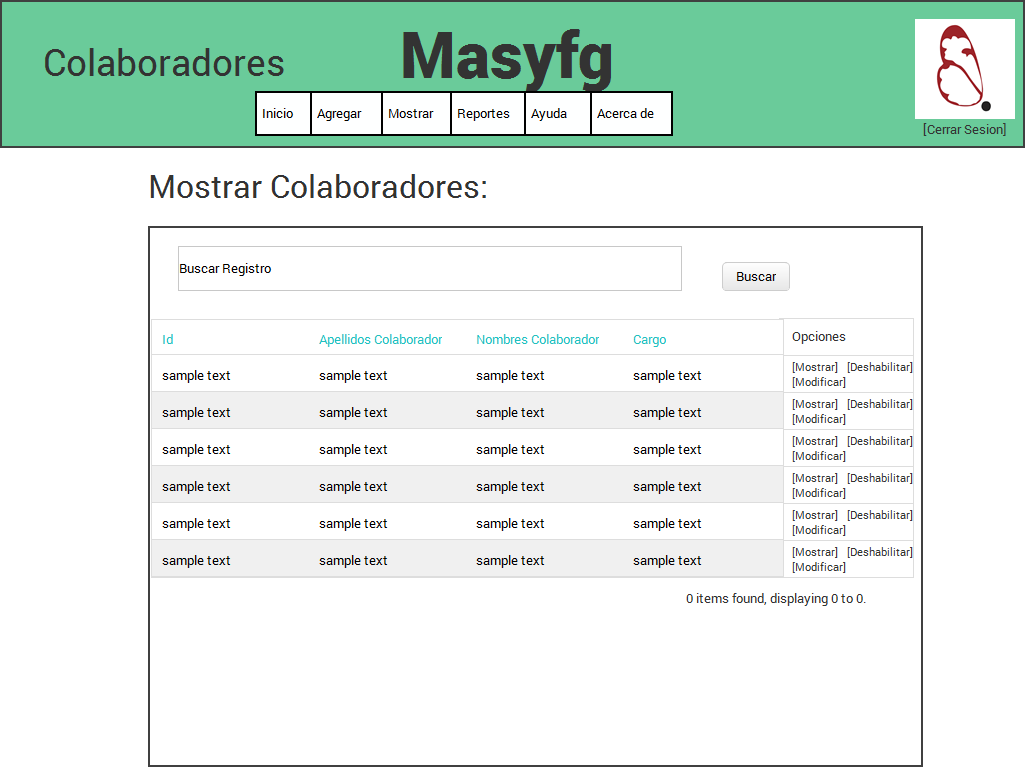
Vista de Pantalla Principal de Clubs

Vista de Pantalla de Agregar Club

Vista de Pantalla de Mostrar Club



Mockups de Colaboradores

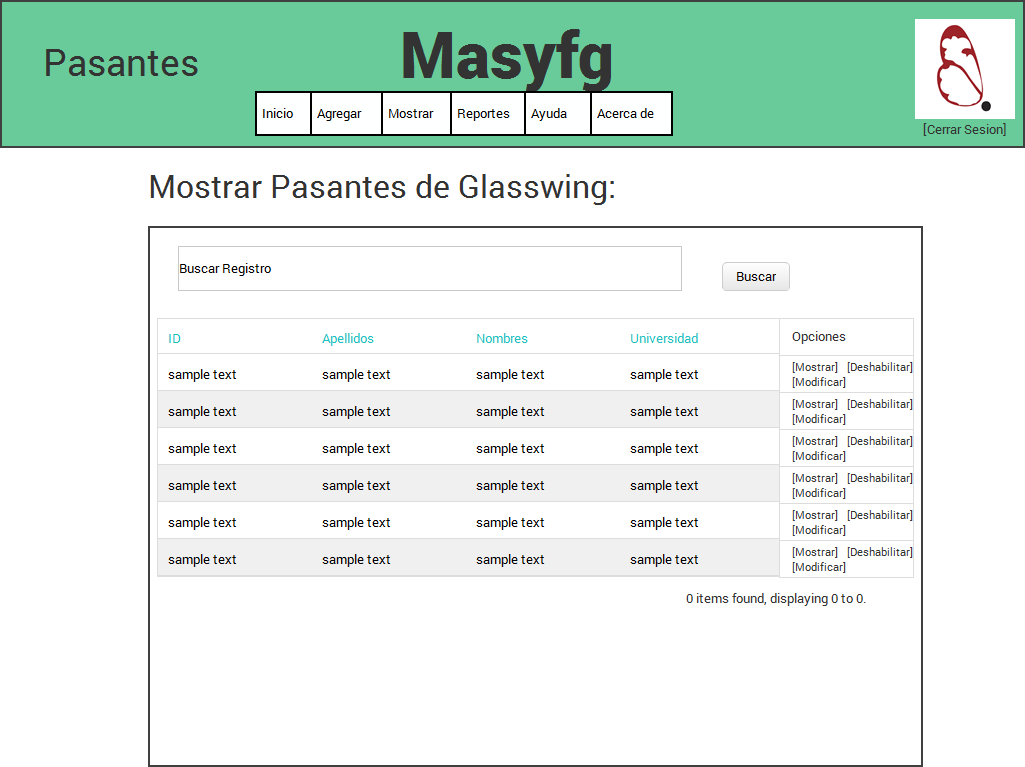
Vista de Pantalla Principal de Colaboradores

Vista de Pantalla de Agregar Colaborador

Vista de Pantalla de Mostrar Colaborador



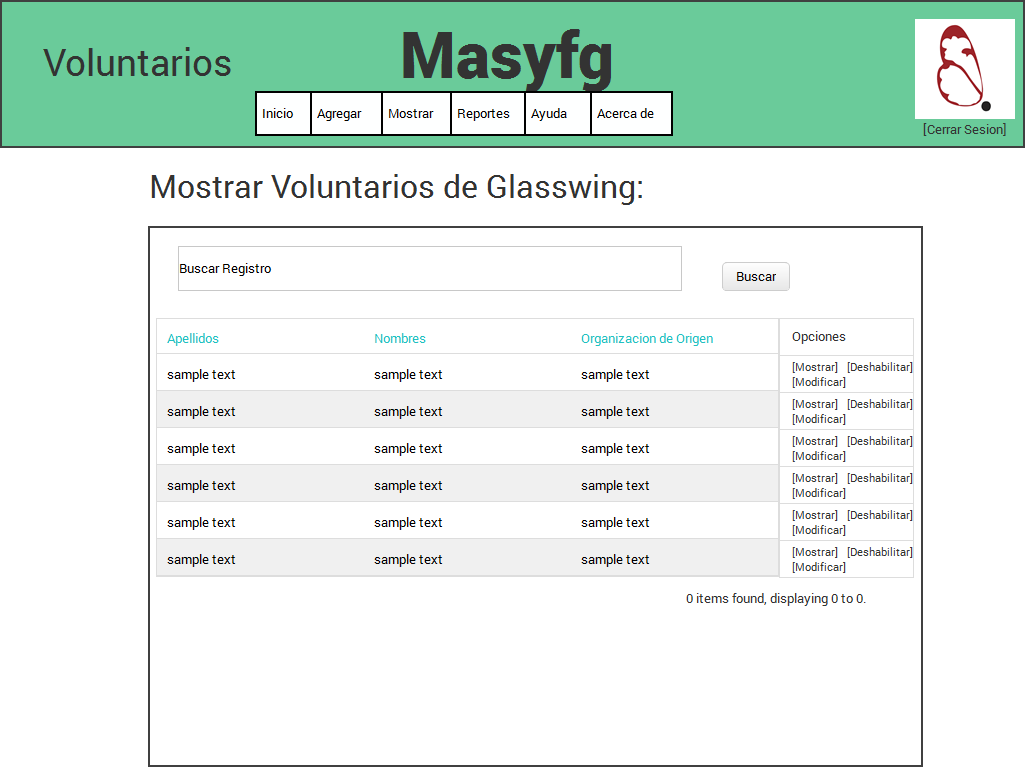
Mockups de Pasantes

Vista de Pantalla Principal de Pasantes

Vista de Pantalla de Agregar Pasante

Vista de Pantalla de Mostrar Pasante

Mockups de Voluntarios

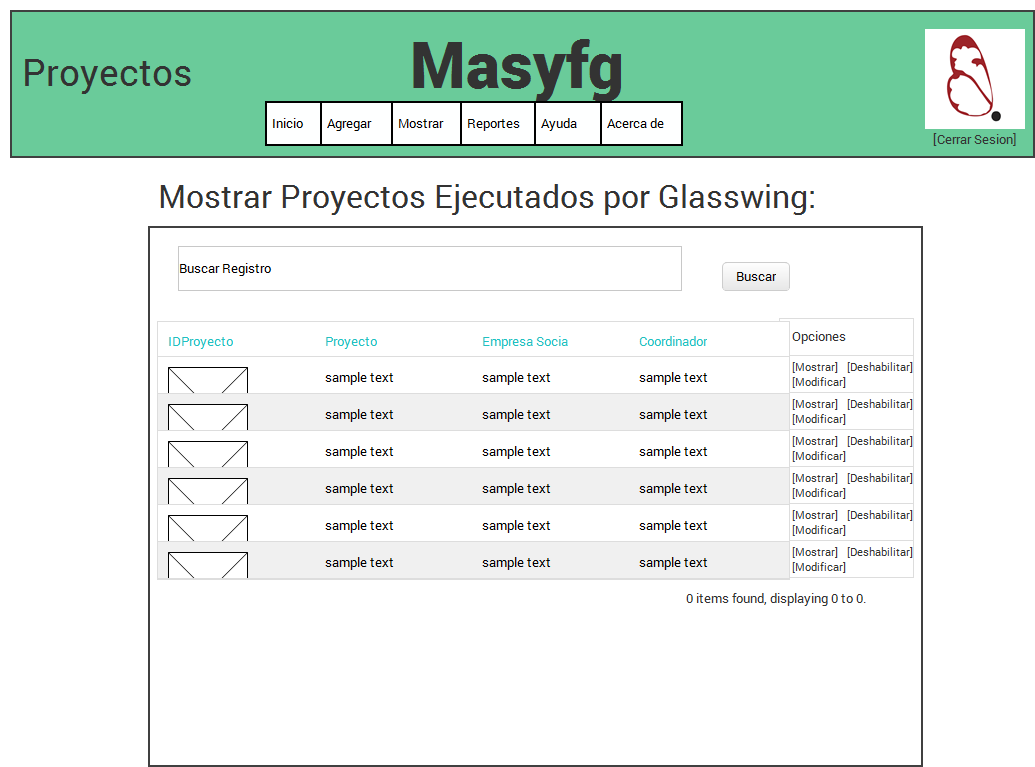
Vista de Pantalla Principal de Voluntarios

Vista de Pantalla de Agregar Voluntario



Vista de Pantalla de Mostrar Voluntario

Mockups de Proyectos

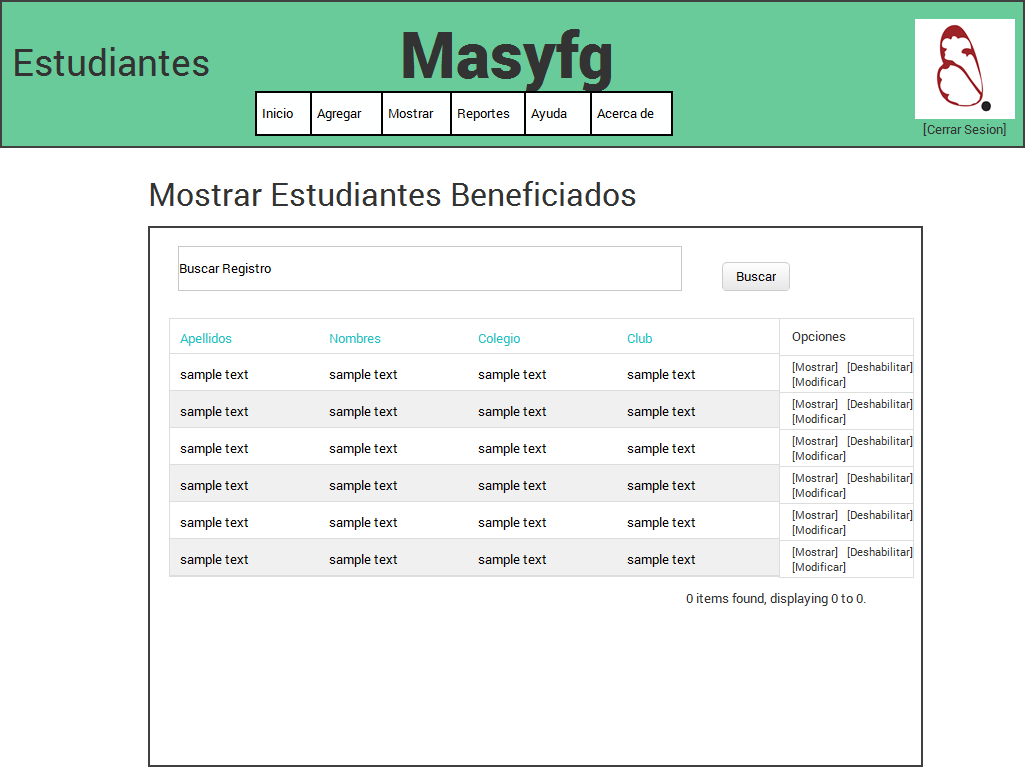
Vista de Pantalla Principal de Proyectos



Vista de Pantalla de Agregar Proyecto

Vista de Pantalla de Mostrar Proyecto

Mockups de Estudiantes

Vista de Pantalla Principal de Estudiantes

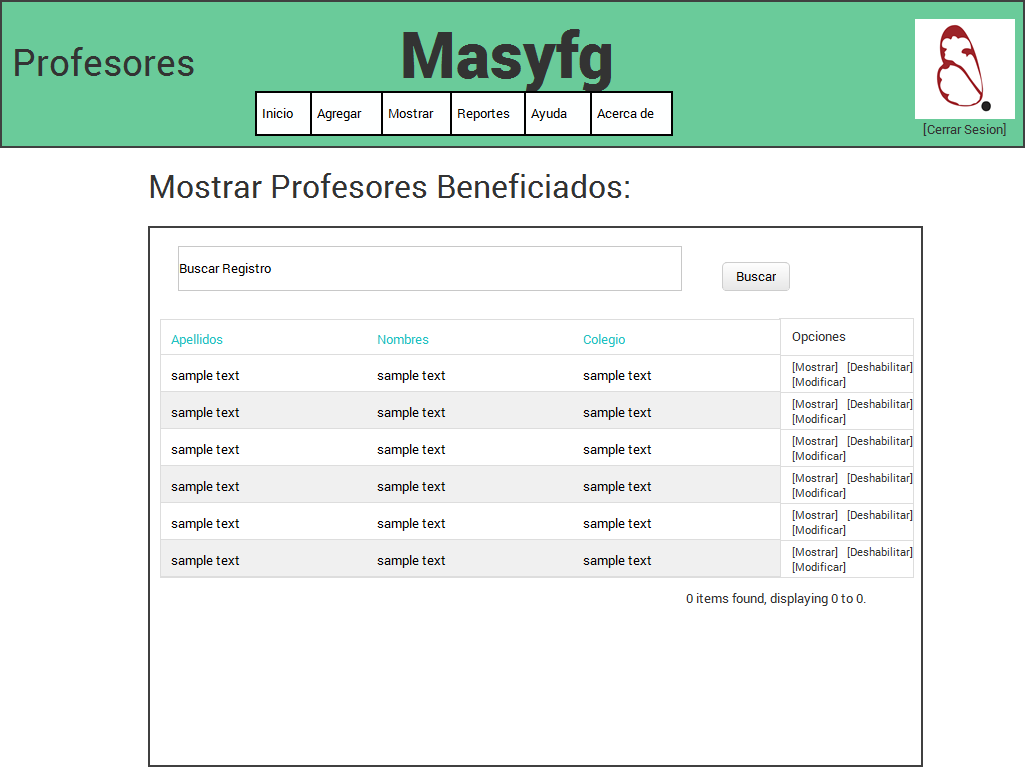
**Vista de Pantalla de Agregar Estudiante**



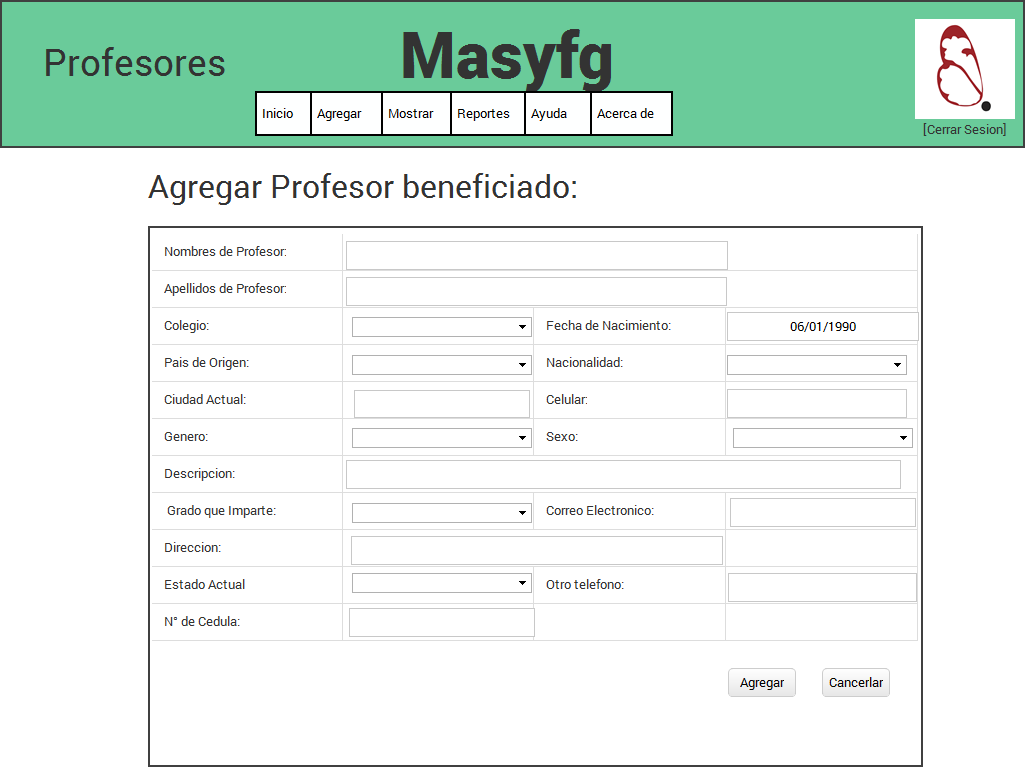
Vista de Pantalla de Mostrar Estudiante

****

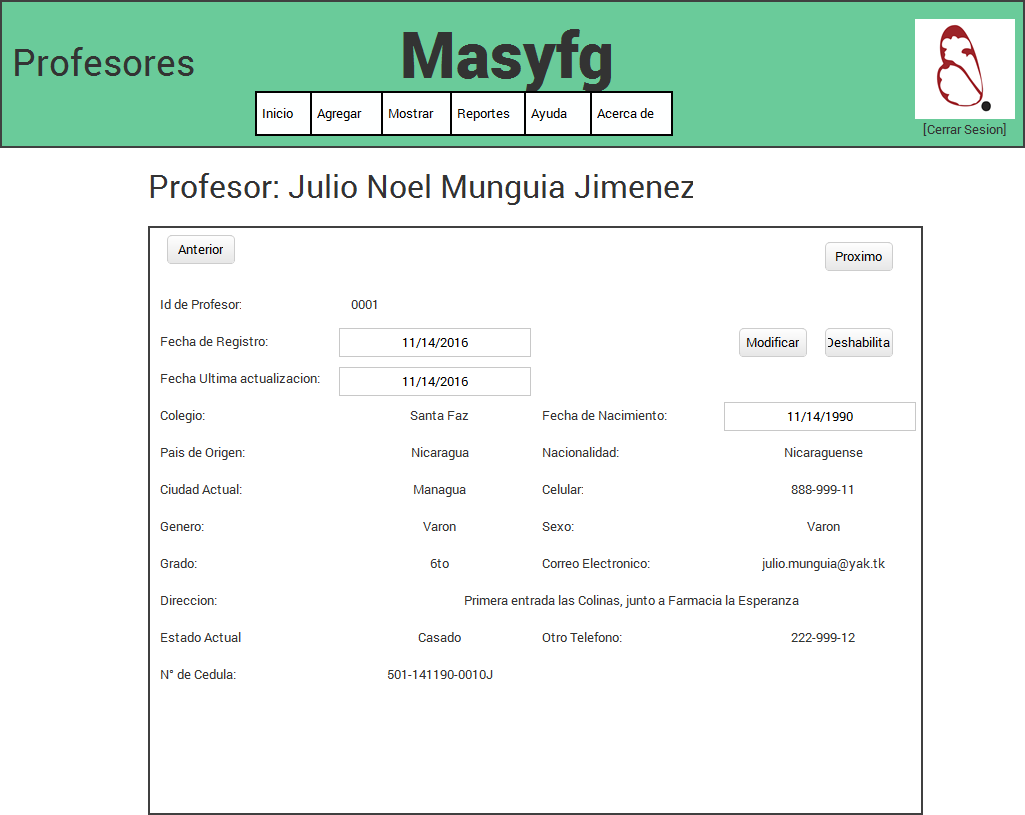
Mockups de Profesores

Vista de Pantalla Principal de Profesores

**Vista de Pantalla de Agregar Profesor**



Vista de Pantalla de Mostrar Profesor



Mapa de Sitio del Sistema (Para Manejo de Información)

1. Conclusiones
2. Recomendaciones

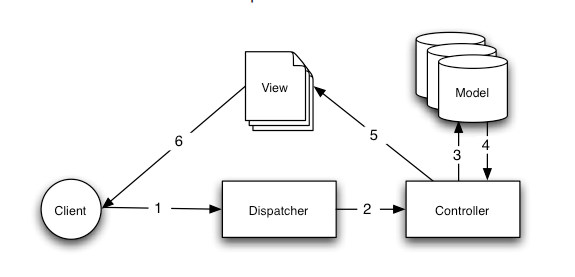
Referencias Bibliográficas

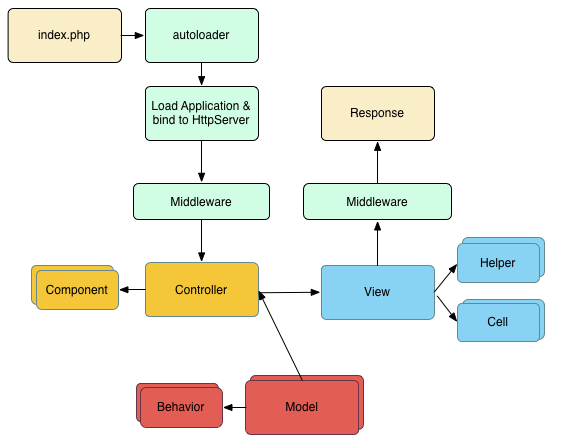
Referencias de Citas

Glosario

Anexos

1. El ciclo de una petición en CakePHP



Fuente: <http://book.cakephp.org/2.0/es/cakephp-overview/understanding-model-view-controller.html> 

Fuente: <http://mundocakephp.blogspot.com/2015/07/el-ciclo-de-una-solicitud-en-cakephp-3.html>

1. Preguntas de Entrevista para la recolección de requerimientos

**Parte 1.**

1. **¿Qué es Glasswing?**

Glasswing es una organización internacional sin fines de lucro, que ayuda en las comunidades de Centroamérica en infraestructura y educación.

1. **¿Cuántos años tienes Glasswing de trabajar en el país y a nivel internacional?**

En el Salvador tiene 9 años, y en el país dos.

1. **¿Cuáles son las funciones que Glasswing realiza en Nicaragua?**

Proyectos en escuelas con infraestructura, remodelación de aulas, jardinería con materiales reciclados clubes extracurriculares en las escuelas, y proyectos de tecnología.

**Parte 2.**

1. **¿Qué tipo de información manejan?**

Información de proveedores, donantes; sobre voluntariado, una base de datos con su información general, horas que realizan, datos sobre las escuelas, población estudiantil, y cuestiones de proyectos que se ejecutan y las inversiones que se realizan.

1. **¿Se ha implementado un sistema parecido anteriormente?**

Actualmente se está desarrollando un sistema, donde se manejarán donantes, voluntarios y escuelas. Es de carácter local.

1. **¿Cuentan actualmente con un sistema propio o herramienta, para el manejo de Información?**

Actualmente se está desarrollando uno a nivel local, framework Php-elementary, mysql. Administrador web local apache, bajo el nombre de **Infoglass**

1. **¿Si cuentan con un sistema actualmente, o herramienta para el apoyo a la toma de decisiones, cuales son las deficiencias que esta presenta?**

No se cuenta con un servidor propio, el servidor estará local.

1. **¿Si cuentan con un sistema actualmente, o herramienta para el apoyo a la toma de decisiones, cuales son las tareas que esta no logra ejecutar?**

Está en prueba y desarrollo.

NOTA: No cuenta con un ciclo de desarrollo ni diseño pre desarrollo.

1. **¿Si cuentan con un sistema actualmente, o herramienta para el apoyo a la toma de decisiones, cuales son las tareas que esta logra ejecutar?**

Búsqueda de información relevante de los datos de la escuela, de los voluntarios. Se buscará que agregue, modifique, elimine. Crea estadísticas menores.

**Parte 3.**

1. **¿Podría Ud. describir las tareas o procesos que desea que el sistema ejecute?**

Que cierre los ciclos de cada proyecto, ejemplo un proyecto puede durar 2 años, que procese la información más relevante por cada ciclo (un año ejemplo) que calcule cuan factible resulto ser.

Poder anexar a las empresas que quieran trabajar con RSE.

Que cree informes estadísticos relacionados a las tareas de y actividades que la organización realiza.

Monitoreo de evaluaciones de los beneficiados.

1. **¿Cuáles son las tareas primordiales que el nuevo sistema ejecute?**

* Creación de respaldo de base de datos periódicamente.
* Que la base de datos conecte con las otras filiales.
* Elaboración de informes en base a combinación de variables.
* Filtrar reportes específicamente.

Que este en línea.

1. **¿Cuáles son los problemas que se solventarían con el Nuevo Sistema?**

* La información almacenada y tener la información más accesible en cualquier momento.
* Minimizar gastos en papelería.
* Tener el sistema en línea y así llenar los datos de las escuelas y proveedores que quieran participar.
* Monitorear el avance de cada proyecto.

1. **¿Cuál va a ser la información procesada?**

* Información proveedores, escuelas, donantes, voluntarios. Estudiantes.
* Información individual por objeto.

1. **que función se desea para el software a crear?**
2. **¿Quiénes serán los usuarios de la solución?**

Coordinadores de proyecto, coordinadores escolares, asistente administrativo y otras personas que se les dé acceso a la información.

1. **¿Quién tendría control total del sistema?**

El responsable de proyectos (gerente de proyectos)

1. **¿Quiénes son los beneficiados por el sistema?**

Personal de la organización.

1. **¿Quién usara el sistema?**

Personal de Glasswing (coordinares de proyectos, administrativos, gerente de proyectos, comunicadores)

1. **¿Habrá varios tipos de usuarios?**

tres niveles: superusuario, usuarios normales, y visitantes.

2. Comparativa de Base de datos para Selección

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tipo | Nombre | Costo | Beneficio | Ventaja | Desventaja | Licencia |
| Open source | María DB | Gratis | Introduce dos motores de almacenamiento Aria y XtraDB. | * Compatible con MySql. * Robusto * Escalable | * No ofrece manera de aprender a manejarla. * Puede presentar errores con MySql | GPL |
| PostgreSQL | Gratis | El gestor de base de datos más seguro de todos | * De las más seguras * Comunidad grande. * Soporta procedimientos almacenados. | * Lento para procesar datos y/o solicitudes. * Limitado en herramientas gratuitas. | Licencia PostgreSQL |
| MySql community | Gratis | Rapidez a la hora de responder a una solicitud | * De las más rápidas * De los gestores más populares que hay. * Baja probabilidad de corromper datos. * Consume pocos recursos. | * De las menos seguras. * Puede provocar errores cuando hay alta concurrencia. * No todas las utilidades están documentadas | GPL |
| Oracle Express | Gratis | De los gestores de base de datos “serios” más usados. | * Orientada hacia internet * Motor objeto-relacional más usado * Usado desde Pc común hasta súper computador * Permite uso de particiones. | * Muchas versiones presentan varios fallos * Puede llegar a ser muy lento * No hay mucha documentación | Apache |
| De pago | Oracle DB | $ 165 por 25 licencias al año  $ 825 por 25 licencias  perpetuas |  | * Muy bien documentada. * Paralelismo en las sentencias. * Todas las funcionalidades están permitidas: | * De paga * Compra mínima de 25 licencias. | Proprietary OTN Standard License |
| IBM DB2 | U$$ 530-22800 |  | * Arquitectura similar a Oracle * Base de datos histórica | * Similar en precio a Oracle * Pude llegar a necesitar el uso de lenguaje externo |  |
| MS SQL Server | Standard $931 |  | * Soporte de transacciones. * Soporta procedimientos almacenados. * Incluye también un entorno gráfico de administración | * Limitaciones de uso de memoria en sistemas de 32bits | Microsoft EULA |
| MySql Enterprise | U$$ 600-5000 |  | * Sin las limitaciones de la versión community |  | GPL/Comercial |

1. Comparativa de Frameworks para Selección

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Framework | Versión Actual | Licencia | Costo | Ventajas | Desventajas |
| PHP |  | | | | |
| CakePHP | 3.2.8 | MIT | Gratis | - Tiene muy buena documentación  - Reutiliza Código, y soporta ORM, limpieza de datos  - Soporta CRUD y validación integrada  - Componentes de email, sookie, seguridad, sesión y manejo de solicitudes  - Funciona en cualquier subdirectorio del sitio web, con poca o ninguna configuración de Apache | - Algunos ficheros del Framework son algo pesado.  - Algunos puntos de la documentación inducen a malos hábitos  Exceso de consultas |
| Symfony | 3.1.0 RC-1 | MIT | Gratis | - Orientado a objetos, e independiente de la base de datos.  - Basado en la premisa de "convenir en vez de configurar"  - Preparado para aplicaciones empresariales, suficientemente estable. | - Necesita un VPS para funcionar apropiadamente.  - Uso excesivo de cache  - Mucha flexibilidad afecta el diseño de las aplicaciones  - Utiliza demasiada Memoria |
| Python |  | | | | |
| Django | 1.9.4 | BSD License | Gratis | - Programas y desarrollar “Módulos es fácil”  - Trabaja bajo el concepto DRY  - SU ORM es potente, es flexible  - Soporte WYSWYG | - Curva de Aprendizaje más elevada |
| Web2py | 2.14.6 | GNU Lesser GPL v3.0 | Gratis | - Desarrollo ágil de aplicaciones web de pequeño y mediano tamaño.  - Aplica el principio de DRY | - falta de flexibilidad |
| Ruby |  | | | | |
| Rails | 5.0.0.1 | MIT | Gratis | - Permite escribir menos código que otros frameworks  - Basado en DRY  - Closures (blocks, lambdas): Se pueden pasar trozos de código como  Argumento.  - Mixin: Permite hacer includes de una clase dentro de otra.  - Creación de grandes sitios web con pocas líneas de código. | - Desarrollo lento comparado con PHP  - Procesamiento Lento  - poca escalabilidad para sitios grandes |
| Sinatra | 1.2.6 | MIT |  | - El segundo más popular después de Rails.  - Se define como un DSL  -Permite construir aplicaciones de cierta complejidad  - Potente ORM | -No es basado en MVC |