MAPA DE UBICACIÓN VIRTUAL DE ESPACIOS Y ASIGNATURAS PARA LA FUNDACIÓN TECNOLOGICA AUTONOMA DE BOGOTÁ

YASMIN VIVIANA GONZALEZ GONZALEZ

JENNIFER ANDREA ROMANI JAMAICA

CESAR CAMILO GONZALEZ CUCHIVAGUEN

FUNDACIÓN TECNOLOGICA AUTONOMA DE BOGOTÁ

DEPARTAMENTO DE TECNOLOGÍA EN SISTEMAS

TECNOLOGIA EN SISTEMAS

BOGOTÁ D.C

2012

MAPA DE UBICACIÓN VIRTUAL DE ESPACIOS Y ASIGNATURAS PARA LA FUNDACIÓN TECNOLOGICA AUTONOMA DE BOGOTÁ

YASMIN VIVIANA GONZALEZ GONZALEZ

Código: 10431048

JENNIFER ANDREA ROMANI JAMAICA

Código: 10431006

CESAR CAMILO GONZALEZ CUCHIVAGUEN

Código: 10431046

Trabajo de grado presentado como requisito para obtener el título de Tecnólogo en sistemas

Tutor:

FREDY SANDOVAL

Ingeniero en Sistemas

FUNDACIÓN TECNOLOGICA AUTONOMA DE BOGOTÁ

DEPARTAMENTO DE TECNOLOGÍA EN SISTEMAS

TECNOLOGIA EN SISTEMAS

BOGOTÁ D.C

2012

Nota de aceptación

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

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

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

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

Firma del presidente del jurado

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

Firma del Jurado

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

Firma del Jurado

Bogotá D.C. Noviembre 29 de 2012.

**DEDICATORIAS**

*A Dios por ser tan bueno conmigo y*

*por la fortaleza que me da a diario.  
A mi familia por apoyarme y ayudarme*

*con mis proyectos de vida.*

*A mi mis amigos por compartir este*

*tiempo de estudio conmigo.*

*A las personas que hicieron posible*

*mi preparación profesional.*

*Yasmin Viviana Gonzalez*

*A la persona más especial de mi vida,*

*aún sin entender el propósito*

*de estos tres años, te amo Jesús.*

*A mis padres José Luis y Gilma,*

*por brindarme siempre*

*su apoyo incondicional.*

*A mi abuelito, por su gran ayuda.*

*Jennifer Andrea Romani Jamaica*

*A mis hermanas, Luisa y Paola,*

*quienes son la luz de mis ojos,*

*no lo supieron, pero siempre*

*me motivaron a seguir adelante.*

*Cesar Camilo González*

**AGRADECIMIENTOS**

En primer lugar, agradecemos a Dios, por abrirnos las puertas de la educación, llenarnos de fuerzas y conocimientos durante estos 3 años de estudio, por brindarnos la oportunidad de aprender, conocer y compartir con nuevas personas, por darnos la fortaleza que necesitamos para culminar con nuestros estudios y por no permitir que desfallezcamos a mitad del camino.

Especialmente a aquellos quienes hicieron posible nuestra formación profesional y personal; entre ellos se destacan nuestras familias, quienes por sus aportes motivacionales y económicos, siempre nos apoyan incondicionalmente y nos transmiten mediante su ejemplo disciplina y fuerza de voluntad en el trabajo.

Gracias a la ayuda de los maestros por compartir y enseñarnos sus conocimientos, especialmente a la profesora Elisa Santiago, quien se destacó por su buen manejo profesional en la profundización de los temas, sin contar con su buen trato para con nosotros (respetuoso, dedicado y atento) , supliendo así todas nuestras necesidades.

También agradecemos a la alcaldía local de Suba, la cual nos brindó la oportunidad de estudiar en la institución.

**NOTA DE SALVEDAD INSTITUCIONAL**

“La Fundación tecnológica Autónoma de Bogotá no se hace responsable por los conceptos omitidos por los investigadores en su trabajo, solo velará por el rigor científico, metodológico y ético del mismo en aras de la búsqueda de la verdad y la justicia”

CONTENIDO

**Pág.**

[**INTRODUCCIÓN** 20](#_Toc343079298)

[**RESUMEN** 21](#_Toc343079299)

[**ABSTRACT** 22](#_Toc343079300)

[**1. DESCRIPCIÓN DEL PROYECTO** 23](#_Toc343079301)

[1.1 Titulo 23](#_Toc343079302)

[1.2 Planteamiento del problema 23](#_Toc343079303)

[1.2.1 Identificación del problema y su descripción 23](#_Toc343079304)

[1.2.2 Formulación del problema 24](#_Toc343079305)

[1.2.3 Sistematización del problema 24](#_Toc343079306)

[**2. JUSTIFICACIÓN** 25](#_Toc343079307)

[**3. OBJETIVOS** 26](#_Toc343079308)

[3.1 Objetivo general: 26](#_Toc343079309)

[3.2 Objetivos específicos: 26](#_Toc343079310)

[**4. MARCO DE REFERENCIA** 27](#_Toc343079311)

[4.1 Marco teórico 28](#_Toc343079312)

[4.2 Estado del arte 37](#_Toc343079313)

[**5. METODOLOGÍA** 40](#_Toc343079314)

[5.1 Tipo de estudio 40](#_Toc343079315)

[5.1.1 Propósito 40](#_Toc343079316)

[5.1.2 Enfoque 40](#_Toc343079317)

[5.1.3 Diseño 41](#_Toc343079318)

[5.1.4 Seguimiento 41](#_Toc343079319)

[5.2 Unidad de estudio 42](#_Toc343079320)

[5.3 Variables 42](#_Toc343079321)

[5.3.1 Cuantitativas 42](#_Toc343079322)

[5.3.2 Cuantitativas 42](#_Toc343079323)

[5.4 Estrategia procedimental 42](#_Toc343079324)

[5.4.1 Recolección de los datos 42](#_Toc343079325)

[5.4.2 Procesamiento y análisis de los datos 43](#_Toc343079326)

[5.4.3 Instrumentos 43](#_Toc343079327)

[5.5 Etapa de investigación 43](#_Toc343079328)

[5.5.1 Investigación 43](#_Toc343079329)

[5.5.2 Diseño previo 44](#_Toc343079330)

[5.5.3 Desarrollo 45](#_Toc343079331)

[5.6 Cronograma de actividades 48](#_Toc343079332)

[**6. PRODUCTOS** 49](#_Toc343079333)

[6.1 Resultados obtenidos 49](#_Toc343079334)

[6.1.1 Generación de conocimiento y/o nuevos desarrollos tecnológicos o creativos. 49](#_Toc343079335)

[6.1.2 Fortalecimiento de la capacidad científica 49](#_Toc343079336)

[6.1.3 Apropiación social del conocimiento 49](#_Toc343079337)

[6.2 Impactos o proyecciones 49](#_Toc343079338)

[6.2.1 Temáticos: 49](#_Toc343079339)

[6.2.2 Territoriales: 49](#_Toc343079340)

[6.3 Requerimientos 50](#_Toc343079341)

[6.3.1 Del software usuario y administrador 50](#_Toc343079342)

[6.4 Casos de uso 51](#_Toc343079343)

[6.4.1 Casos de uso actor Usuario: 51](#_Toc343079344)

[6.5 Modelo relacional 74](#_Toc343079345)

[**7. RECOMENDACIONES** 75](#_Toc343079346)

[**8. CONCLUSIONES** 76](#_Toc343079347)

[**BIBLIOGRAFÍA DE REFERENCIA** 77](#_Toc343079348)

[**WEBGRAFÍA** 78](#_Toc343079349)

[**ANEXOS** 79](#_Toc343079350)

**LISTA DE FIGURAS**

**Pág.**

*Figura 1. Gráfica de Componentes de una base de datos28*

*Figura 2. Gráfica de Modelo Entidad-Relación (E-R)31*

*Figura 3. Gráfica de Ejemplo Modelo Entidad-Relación (E-R)31*

*Figura 4. Gráfica de Proyecto de georreferenciación de mapas antiguos38*

*Figura 5. Diagrama de casos de uso de usuario51*

*Figura 6. Diagrama de casos de uso de usuario administrador65*

*Figura 7. Modelo relacional…………………………………………………………………..74*

**LISTA DE TABLAS**

**Pág.**

*Tabla 1. Operaciones del usuario*52

*Tabla 2. Función ingresar*53

*Tabla 3. Función regresar*54

*Tabla 4. Función salir*55

*Tabla 5. Función ayuda*56

*Tabla 6. Función ver aula*57

*Tabla 7. Función buscar zona especial*58

*Tabla 8. Función ver búsquedas recientes*59

*Tabla 9. Función ver edificio (Ir al inicio)*60

*Tabla 10. Función ver piso*61

*Tabla 11. Función ver mapa del piso*62

*Tabla 12. Función ver zona especial*63

*Tabla 13. Función ver aula*64

*Tabla 14. Operaciones del usuario administrador*65

*Tabla 15. Función ingresar como administrador*66

*Tabla 16. Función seleccionar zona especial*67

*Tabla 17. Función seleccionar aula*68

*Tabla 18. Función editar historial de búsqueda*69

*Tabla 19. Función editar zona especial*70

*Tabla 20. Función editar aula*71

*Tabla 21. Función actualizar mapa*73

*Tabla 22. Función actualizar fotos*74

**ANEXOS**

**ANEXO A** Resultados de las encuestas para la recolección de datos.

**A1** Resultados de la encuesta para miembros de la comunidad FABA en gráficos de decisión.

**A2** Resultados de la encuesta para visitantes de la comunidad FABA en gráficos de decisión.

**ANEXO B** Fotos universidad por pisos (ver cd adicional).

**GLOSARIO**

**Aplicación:** es un [programa informático](http://www.terra.es/personal/lermon/cat/articles/evin0365.htm) diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Posee ciertas características que le diferencia de un [sistema operativo](http://www.terra.es/personal/lermon/cat/articles/evin0433.htm) (que hace funcionar al ordenador), de una [utilidad](http://www.terra.es/personal/lermon/cat/articles/evin0483.htm) (que realiza tareas de mantenimiento o de uso general) y de un [lenguaje](http://www.terra.es/personal/lermon/cat/articles/evin0275.htm) (con el cual se crean los programas informáticos).

**Atributo:** los atributos se utilizan para detallar las entidades (ver entidad) asignándoles propiedades descriptivas tales como nombre, color y peso. No sólo es posible especificar atributos en las entidades sino también en las relaciones. Los atributos también aparecen reflejados en el enunciado, generalmente, como nombres.

**ATK:** biblioteca para crear interfaces con características de una gran accesibilidad muy importante para personas discapacitadas o minusválidos. Pueden usarse utilerías como lupas de aumento, lectores de pantalla, o entradas de datos alternativas al clásico teclado o ratón.

**Base de datos:** la base de datos es un conjunto de [información](http://www.mastermagazine.info/termino/4012.php) que está almacenada en forma sistemática, de manera tal que los datos que la conforman puedan ser utilizados en forma fragmentada cuando sea necesario.

Los datos almacenados pueden ser muy diversos: nombres, números telefónicos, direcciones, años, etc. Todo depende de la finalidad para la que sea armada la base. Actualmente, en todos los quehaceres cotidianos se utiliza una base de datos: cajeros automáticos, catálogos de bibliotecas o librerías, páginas amarillas, listado de medicamentos, e incluso los mismos buscadores de [Internet](http://www.mastermagazine.info/termino/4012.php).

**Bytecode:** código intermedio entre el código fuente y el código máquina. Se suele tratarlo como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, en algunos casos existen traductores dinámicos o compiladores que traducen el bytecode a código máquina justo antes de ejecutar el programa para mejorar la velocidad, los bytecode suelen ser interpretados por programas que suelen llamarse máquina virtual.

**Código fuente:** es un texto escrito en un lenguaje de programación específico, que puede ser leído por un programador y debe traducirse a lenguaje máquina para que pueda ser ejecutado por la computadora.

**Compilación:** un compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es lenguaje de máquina, pero también puede ser un código intermedio (bytecode), o simplemente texto. Este proceso de traducción se conoce como compilación.

**Cairo:** biblioteca de renderizado avanzado de controles de aplicación.

**Casos de uso:** los casos de uso son una técnica para especificar el comportamiento de un sistema: “Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios.”

**Colapso:** destrucción, ruina de una institución, sistema, estructura, etc. Deformación o destrucción bruscas de un cuerpo por la acción de una fuerza.

**Comando:** un comando es una instrucción u orden que el usuario proporciona a un sistema informático, desde la línea de comandos (como una shell (intérprete de comandos)) o desde una llamada de programación. Puede ser interno (contenido en el propio intérprete) o externo (contenido en un archivo ejecutable).

**Dispositivos:** los dispositivos son elementos de la pc que se requieren en todo momento para que la máquina funcione de manera correcta. Es decir, son imprescindibles al momento de usar la PC. Estos dispositivos permiten al usuario ver los resultados de los cálculos o de las manipulaciones de datos de la computadora.

**DLS:** La idea básica de un lenguaje de dominio específico (DLS por sus siglas en ingles) es que en vez de ser una lengua de propósito general que se dirige a cualquier clase de problema de programación, es un lenguaje de programación que ataca a un problema particular. Los lenguajes de dominio específico se han discutido y utilizado casi desde los inicios de la computación.

**Entidad:** una entidad representa una “cosa” u "objeto" del mundo real con existencia independiente, es decir, se diferencia unívocamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad.

Algunos ejemplos:

* Una persona. (Se diferencia de cualquier otra persona, incluso siendo gemelos).
* Un automóvil. (Aunque sean de la misma marca, el mismo modelo,..., tendrán atributos diferentes, por ejemplo, el número de chasis).
* Una casa (Aunque sea exactamente igual a otra, aún se diferenciará en su dirección).

Una entidad puede ser un objeto con existencia física como: una persona, un animal, una casa, etc. (entidad concreta); o un objeto con existencia conceptual como: un puesto de trabajo, una asignatura de clases, un nombre, etc. (entidad abstracta).

**GDK:** biblioteca que actúa como intermediario entre gráficos de bajo nivel y gráficos de alto nivel.

**GLib:** biblioteca de bajo nivel estructura básica de GTK+ y GNOME. Proporciona manejo de estructura de datos para C, portabilidad, interfaces para funcionalidades de tiempo de ejecución como ciclos, hilos, carga dinámica o un sistema de objetos.

**GTK:** biblioteca la cual realmente contiene los objetos y funciones para crear la interfaz de usuario. Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas, etc.

**GTK+:** es un conjunto de bibliotecas multiplataforma para desarrollar interfaces gráficas de usuario, GTK+ se basa en varias bibliotecas del equipo de GTK+ y de GNOME:

**Hardware:** son los componentes físicos del ordenador, es decir, todo lo que se puede ver y tocar. Clasificamos el hardware en dos tipos:

**1)** El que se encuentra dentro de la torre o CPU, y que por lo tanto no podemos ver a simple vista.

**2)** El que se encuentra alrededor de la torre o CPU, y que por lo tanto, si que vemos a simple vista, y que denominamos periféricos como el teclado, el ratón, una impresora, etc.

**HTML:** HyperText Markup Language, (lenguaje de marcado de hipertexto), HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

**IDE:** Entorno de desarrollo integrado, llamado también (**IDE**), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien poder utilizarse para varios.

Ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word.

**Interfaz:** conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el Sitio Web que está visitando. Por lo mismo, se considera parte de la interfaz a sus elementos de identificación, de navegación, de contenidos y de acción.

Todos ellos deben estar preparados para ofrecer servicios determinados al usuario, con el fin de que éste obtenga lo que vino a buscar cuando visitó el Sitio Web. Por lo anterior, cada uno de los elementos que sean integrados dentro de la interfaz debe estar pensado para causar un efecto sobre el usuario y deben ser utilizados con un propósito.

**Interfaz gráfica:** Es un tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos de menú. Las selecciones pueden activarse bien a través del teclado o con el ratón.

**JavaScript:** lenguaje multiplataforma orientado a objetos. Es un lenguaje pequeño y ligero; no es útil como un lenguaje independiente, más bien está diseñado para una fácil incrustación en otros productos y aplicaciones, tales como los navegadores Web. Dentro de un entorno anfitrión, JavaScript puede ser conectado a los objetos de su entorno para proveer un control programable sobre éstos.

**Lenguaje de programación:** se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora. Se asume que las instrucciones así escritas son traducidas luego a un código que la máquina pueda “comprender”. El proceso de traducción es realizado normalmente por la computadora, usando un programa especializado para tal fin.

**Linux:** Es un sistema operativo, compatible Unix. Dos características muy peculiares lo diferencian del resto de sistemas que podemos encontrar en el mercado, la primera, es que es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente.

**Microsystems:** es una Cooperativa de Trabajo Asociado, del sector social y solidario, con actividad de servicios que busca satisfacer las necesidades y expectativas de productividad de las empresas. En cumplimiento del Acuerdo Cooperativo, microsystems tiene como objeto socialgenerar y mantener puestos de labor que serán desarrollados por sus asociados de manera autogestionaria, en la prestación de servicios a terceros para el desarrollo de sus procesos.

**Modelo de Entidad Relación** es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre estos objetos, implementando en forma gráfica a través del Diagrama Entidad Relación.

**Módulo:** un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

**Ordenador:** Un ordenador es un dispositivo q recibe datos, los procesa y produce otros datos o información, de esta manera nosotros introducimos parámetros mediante los dispositivos de entrada (el teclado, el micrófono, el escáner, etc.). El ordenador los procesa en la CPU con gran velocidad y hace visibles los resultados en los dispositivos de salida (el monitor, la impresora, etc.).

**Pango:** Biblioteca para el diseño y renderizado de texto, hace hincapié especialmente en la internacionalización manejar las fuentes y el texto de GTK+2.

**Procesador:** El procesador es el cerebro del sistema, encargado de procesar toda la información. Es el componente donde es usada la tecnología más reciente.

Código nativo: En el contexto informático Código nativo se usa como seudónimo de lenguaje de máquina, este puede ser creado directamente para microcontroladores extremadamente sencillos o código fuente ya compilado, que puede ser interpretado por la máquina.

**Programa:** Un programa es un conjunto de instrucciones escritas en algún lenguaje de programación. El programa debe ser compilado o interpretado para poder ser ejecutado y así cumplir su objetivo.

**Programación:** es la acción de escribir un conjunto de instrucciones escritas en un determinado lenguaje, que dirigen a un ordenador para la ejecución de una serie de operaciones, con el objetivo de resolver un problema que se ha definido previamente.

**Protocolo:** Es un conjunto de reglas, convenciones o procedimientos establecidos de forma coincidente entre dos o más partes que van a interactuar para intercambiar información.

**Red:** Una [red](http://www.mastermagazine.info/termino/6496.php) es un sistema de comunicación entre computadoras que permite la transmisión de datos de una máquina a la otra, con lo que se lleva adelante entre ellas un intercambio de todo tipo de información y de recursos.

**RUP:** Los creadores de la metodología son los mismos que los del UML: Ivar Jacobson, Grady Booch y James Rumbaugh, que respectivamente eran autores de las metodologías: el método Booch y la metodología OMT. Como toda metodología de desarrollo software su finalidad es convertir las especificaciones que da el cliente en un sistema software. Las características que tiene el R.U.P. son:

1. Está basado en componentes. Estos componentes a su vez están conectados entre sí a través de interfaces.
2. Utiliza el UML como notación básica.
3. Dirigido por casos de uso.
4. Centrado en la arquitectura.
5. Ciclo de vida iterativo e incremental.

**Saturar:** Llenar u ocupar completamente o utilizar una cosa hasta el límite de su capacidad. Hartar, saciar.

**Server-Side Scripting:** Es un servidor Web de tecnología en la que se maneja la petición de un usuario por un script que se ejecuta en el servidor web para generar páginas web dinámicas. Por lo general se utiliza para proporcionar sitios web interactivos que se conectan a bases de datos o almacenes de otros datos.

**Servidor:** también llamado "Host" ó anfitrión, es una computadora con muy altas capacidades, encargada de proveer diferentes servicios a las redes de datos (una red es un conjunto de computadoras interconectadas entre sí), tanto inalámbricas como las basadas en cable, también permite accesos a cuentas de correo electrónico, administración de dominios empresariales, hospedaje y dominios Web entre otras funciones.

**Sistema de gestión de Bases de Datos (SGBD):** es una aplicación informática que permite a los usuarios deﬁnir, crear, mantener y consultar una base de datos; así como proporciona acceso controlado a la misma.

**Sistema Operativo:** Es el software básico de una computadora que provee una interfaz entre el resto de programas del ordenador, los dispositivos hardware y el usuario; las funciones básicas del Sistema Operativo son administrar los recursos de la máquina, coordinar el hardware y organizar archivos y directorios en dispositivos de almacenamiento.

**Software:** Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. Es el equipamiento lógico e intangible de un ordenador. En otras palabras, el concepto de software abarca a todas las aplicaciones informáticas, como los procesadores de textos, las planillas de cálculo y los editores de imágenes

**Software libre:** Este término no tiene ninguna relación con el precio. Lo que interesa es la libertad.

Un programa es software libre para el usuario siempre que, como usuario particular, tenga:

1. La libertad de ejecutar el programa sea cual sea el propósito.

2. La libertad para modiﬁcar el programa para ajustarlo a sus necesidades. (Para que se trate de una libertad efectiva en la práctica, se deberá tener acceso al código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil.)

3. La libertad de redistribuir copias, ya sea de forma gratuita, ya sea a cambio del pago de un precio.

4. La libertad de distribuir versiones modiﬁcadas del programa, de tal forma que la comunidad pueda aprovechar las mejora introducidas.

**Tecnología subyacente:** Es la tecnología subyacente es la que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios.

**Terabyte:** Un Terabyte es una unidad de almacenamiento de información y se representa con su símbolo respectivo el cual es “TB**”.**

**UML: Lenguaje Unificado de Modelado** o **UML**, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

**Unix:** es un sistema operativo multiusuario y multitarea, comprende programas que permiten compilar lenguajes de programación, editar texto, interpretar comandos, manejar archivos y discos, acceder a otras máquinas, establecer comunicaciones telefónicas, enviar y recibir correo electrónico, manejar las colas de impresión y un sinfín de tareas más. Algunos de estos programas pueden haber sido desarrollados por los propios usuarios.

# INTRODUCCIÓN

El presente proyecto pretende realizar el estudio de la georreferenciación, específicamente en la Fundación Tecnológica Autónoma de Bogotá (FABA). La georreferenciación es de vital importancia, ya que determina la localización de las personas en un lugar o dentro de un recinto.

Del mismo modo, la tecnología es una de las necesidades actuales más importantes, ya que permite realizar distintas actividades necesarias para el ser humano como la comunicación, educación, entre muchas otras. La tecnología hoy en día se puede aplicar en todo lo que hacemos, ayudando y facilitando nuestro diario vivir, desarrollando nuevas capacidades y a la vez nuevas tecnologías que sirven para brindar soluciones a las empresas, entidades o beneficiarios de esta.

La carencia de tecnologías en la actualidad, genera un impacto negativo en cualquier ámbito. En el caso de la universidad y trayendo a colación el tema de la georreferenciación, se requiere una mayor eficiencia con respecto a la localización de las personas dentro del plantel educativo.

Para analizar esta problemática, es necesario mencionar sus causas. Una de ellas es la desactualización de la guía de ubicación existente en la universidad, además de su diseño poco llamativo.

Con la realización de este proyecto, se pretende realizar un análisis del estado actual de la georreferenciación en la universidad, comenzando por una recolección de datos, la cual es pieza fundamental para realizar el estudio.

# RESUMEN

El proyecto MUVFABA - Mapa de Ubicación Virtual de espacios y asignaturas para FABA, se enfoca en el análisis de la georreferenciación en la Fundación Tecnológica Autónoma de Bogotá y propone una posible solución para mejorar la ubicación espacial dentro de sus instalaciones.

Trata de ser una base importante para futuras investigaciones en la universidad. Por lo tanto, se dejará el documento con los fundamentos de la base de datos y creación del prototipo para la interfaz de usuario.

De esta forma, si en un futuro se llegase a implementar el proyecto, la comunidad FABA y sus visitantes podrán acceder vía web a una herramienta tecnológica, llamativa y agradable, que les permitirá interactuar con la información que necesiten respecto a su localización espacial.

*Palabras claves: guía, ubicación, georreferenciación, base de datos.*

# ABSTRACT

The project MUVFABA - Location Map Virtual spaces and subjects for FABA, focuses on the analysis of geo Autonomous Technology Foundation in Bogotá and proposes a possible solution to improve the spatial location within your facility.

Try to be an important basis for future research in the university. Therefore, the document will be left with the fundamentals of database and prototyping for user interface.

Thus, if in the future they were to implement the project, FABA community and visitors can access via the web to a technological tool, striking and pleasant, allowing them to interact with the information they need about their spatial location.

*Keywords: guide, location, georeferencing, database.*

# DESCRIPCIÓN DEL PROYECTO

## 1.1 Titulo

Mapa de ubicación virtual de espacios y asignaturas para la Fundación Tecnológica Autónoma de Bogotá.

## 1.2 Planteamiento del problema

### 1.2.1 Identificación del problema y su descripción

Basándonos en las experiencias de los estudiantes de quinto semestre de tecnología de sistemas, con respecto a su ubicación espacial dentro del plantel educativo, durante sus primeros dos semestres de estudios, se ha evidenciado que ya no hay acercamiento de la comunidad a la guía de ubicación de las aulas, debido -entre otras cosas- a la desactualización de la información registrada en la misma desde hace mucho tiempo (por lo menos dos años), y a lo insulso de su diseño.

Por otro lado, se evidencia la carencia de implementación de herramientas tecnológicas en la universidad, situación que desfavorece mucho la imagen de la misma. Un caso muy frecuente en el que queda al descubierto esta realidad, es el vivido por la comunidad educativa a la hora de orientarse dentro de la universidad. Es innegable la ineficiencia de la guía de salones actual. Ahora bien, las guías no deben limitarse a ser un elemento decorativo en la institución, sino una ayuda eficaz para cualquier tipo de persona que necesite usarla, haciéndola una herramienta efectiva, cómoda, rápida y eficaz; ya que, una buena guía ayuda a disminuir el tiempo de búsqueda del recinto que el usuario desee visitar, dándole así, el lugar que debe tener esta herramienta en cualquier recinto en el que se necesite: el de guía.

Si se tiene en cuenta que la universidad tiene un énfasis tecnológico, tal como lo demuestra su nombre, la carencia de tecnologías novedosas genera un impacto negativo dentro y fuera de esta, lo cual disminuye su credibilidad ante la comunidad, dañando su prestigio y disminuyendo así el número de matrículas registradas en las distintas carreras tecnológicas, generando un efecto dominó. Por ende, es necesario aprovechar el conocimiento impartido por el departamento de tecnología en sistemas, por medio de los aportes académicos y los proyectos de investigación en curso del personal (docentes y estudiantes) a través del apoyo de la universidad.

Para conocer y analizar más específicamente la situación actual de la georreferenciación de la FABA, se realizaron dos encuestas, una a la comunidad FABA y la otra a los visitantes de la universidad. Los resultados de estas encuestas (Ver Anexo A) mostraron que un alto porcentaje de los estudiantes de las distintas carreras ofrecidas por el plantel educativo han tenido dificultad de ubicación de las aulas en donde tienen sus respectivas clases, mientras conocen la infraestructura de la universidad (lo cual toma un tiempo en promedio de uno a dos semestres), durante este tiempo se generan varios inconvenientes para los estudiantes (por su puntualidad), el personal docente (por los retrasos de sus clases, ya sea por desubicación de los alumnos o incluso de ellos mismos) y el personal visitante (padres de familia, aspirantes, conferencistas y particulares), situación que genera un ambiente desagradable dentro de la universidad.

El perfil profesional, el conocimiento, la capacidad de dar soluciones, el hecho de estar involucrados en la problemática y la disponibilidad de mejorar dichas carencias en este campo, además de conocer el enfoque que tiene el departamento de tecnología en sistemas respecto al software libre, hacen de los estudiantes del departamento de sistemas las personas más idóneas para corregir las deficiencias de investigación y desarrollo tecnológico presentadas dentro de las instalaciones.

### 1.2.2 Formulación del problema

¿Un mapa virtual facilitaría la ubicación de los miembros de la comunidad FABA dentro de sus instalaciones?

### 1.2.3 Sistematización del problema

¿Un mapa virtual de ubicación podría ahorrar tiempo a los miembros de la comunidad FABA en el momento de buscar un recinto dentro de la universidad?

¿Qué beneficios tendría la universidad, los visitantes y miembros de la fundación, al implementar un Mapa de Ubicación Virtual dentro de la comunidad FABA?

# 2. JUSTIFICACIÓN

Se ha evidenciado que los miembros de la comunidad FABA (personal administrativo, docentes, estudiantes y trabajadores) ya no se acercan a la guía de ubicación de las aulas, ya que la información registrada en esta guía está desactualizada desde hace algún tiempo (por lo menos dos años), además su diseño es poco llamativo.

Pensando en la comodidad de los miembros y visitantes de la FABA, este proyecto garantiza que la universidad tendrá un mejor ambiente. Los miembros y visitantes ya no tendrán inconvenientes para acercarse a un lugar específico del edificio, de modo que esta herramienta se acopla a las necesidades del usuario y busca satisfacerlas.

Vivenciando la problemática actual de la comunidad FABA, se pretende dar la solución más precisa y efectiva, desarrollando un mapa de ubicación virtual donde la gente pueda acercarse a buscar su lugar de destino, demostrando que este proyecto es de gran ayuda para el usuario.

# 3. OBJETIVOS

## 3.1 Objetivo general:

Analizar y diseñar un sistema gestor de base de datos correspondiente a un prototipo de mapa de ubicación virtual para la Fundación Tecnológica Autónoma de Bogotá (FABA).

## 3.2 Objetivos específicos:

* Evaluar los problemas específicos de ubicación espacio-temporal de los usuarios de las instalaciones de la FABA.
* Diseñar la base de datos correspondiente al mapa de ubicación virtual de FABA.
* Crear un prototipo de un mapa de ubicación virtual para FABA que muestre el funcionamiento del sistema, bajo una plataforma cómoda, llamativa y eficaz para el usuario.

# 4. MARCO DE REFERENCIA

La fundación tecnológica autónoma de Bogotá (FABA), ubicada en la Carrera 14 No 80-35 en el barrio el Lago de la ciudad de Bogotá, se constituyó como centro de educación superior a nivel tecnológico de carácter privado en el año de 1994 y entró en convenio con la Universidad de Santander (UDES) en el año de 1996.

FABA era un establecimiento reducido cuando comenzó a prestar sus servicios educativos. Con el transcurso del tiempo gracias a los avances de la tecnología y la ciencia se fueron ampliando la oferta educativa y la edificación del instituto, para brindar un mejor servicio a sus usuarios. Se evidencia que este cambio no es malo, pero se han dejado de lado algunas problemáticas y necesidades que han surgido con este cambio.

Según cita el objetivo principal del programa de Tecnología en Sistemas lo primordial para este es por lo cual es necesario que los estudiantes de la facultad tomen conciencia de las falencias de la institución y con su conocimiento puedan generar soluciones.

Considerando la problemática que se ve actualmente sobre la ubicación espacial de la comunidad de la FABA (visitantes, estudiantes, docentes, etc.) en sus instalaciones, además de la falta de un sistema de información dinámico de los usos de los espacios (asignaturas, eventos, ocupantes, etc.), acorde con el crecimiento de la institución, se busca con este proyecto mejorar estos aspectos, implementando la nueva tecnología y aplicando también software libre, conforme al enfoque del programa de tecnología en sistemas.

Bajo estas premisas, el presente proyecto propone el diseño y creación de una nueva base de datos para la institución, en la cual esté organizada la información referente a los cursos, salones, baños, oficinas, áreas comunes, etc., así como los registros de sus respectivos usos a través del tiempo, diseñando también un prototipo de mapa ubicación virtual y dinámica para los usuarios.

## 4.1 Marco teórico

**¿Qué es una base de datos?**

Base de datos se refiere a un conjunto de datos lógicos y coherentes que interactúan entre si, evitando redundancia alguna entre ellos.

Con lo anterior se puede concluir que una base de datos es una colección de tablas gestionadas por un ejemplar de SGBD donde se establecen colecciones y relaciones entre ellas; teniendo en cuenta que para realizar un base de datos es necesario utilizar un modelo de diagramar, por ejemplo, el modelo entidad relación (E/R).

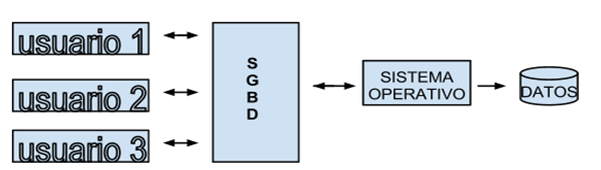
Los componentes de una base de datos son:

- Los datos: son el componente fundamental ya que con los que están interrelacionados entre si, formando un conjunto con un mínimo de redundancia.

- El software: para que los datos puedan ser utilizados por diferentes usuarios y diferentes aplicaciones, deben estar estructurados y almacenados de forma independiente de las aplicaciones. Para ello se necesita un software o programa que actúa de interfaz entre los datos y las aplicaciones. A este software se le denomina un sistema gestor de base de datos (SGBD), el cual crea y organiza la base de datos y además atiende todas las solicitudes de acceso hechas a la base de datos tanto por usuario como por aplicaciones.[[1]](#footnote-1)

En la siguiente gráfica se pueden evidenciar los componentes de una base de datos, como lo son, los usuarios, un sistema de gestión de base de datos, un sistema operativo y los datos.

***Figura 1. Gráfica de Componentes de una base de datos.***



***Fuente: Elaboración propia.***

**Ventajas de una Base de Datos (B.D).**

Las bases de datos permiten mejorar la calidad de presentaciones de los sistemas informáticos y aumentar su rendimiento.

- Independencia de los datos, programas y procesos; es decir que permite modificar los datos sin modificar el código de las aplicaciones.

- Menor redundancia; quiere decir que no hace falta tanta repetición de datos, solo se indica la forma en la que se relacionan los datos.

- Integridad de los datos; hace referencia a que es menos probable que se pierdan los datos o que se realicen incoherencias entre ellos.

- Coherencia de los resultados; al recogerse y almacenarse la información una sola vez, en los tratamientos se utilizan siempre los mismos datos, por lo que los resultados coherentes.

- Hay mayor seguridad en los datos, ya que al permitir limitar el acceso a los usuarios, cada tipo de usuario podrá acceder a algunas cosas, según corresponda.

- Acceso a los datos más eficientemente: gracias a la organización de los datos, se produce un resultado más óptimo en rendimiento.

- Reducción del espacio de almacenamiento; esto se da gracias a una buena estructuración de los datos.

- Acceso simultáneo a los datos: es más fácil controlar el acceso de usuario de forma concurrente.

**Desventajas de una Base de Datos (B.D).**

Además de las ventajas que posee una base de datos, también encierra ciertas desventajas, como:

- Instalación costosa, de modo que el control y la administración de una base de datos requiere de un software y hardware de elevado coste, Además de la adquisición y el mantenimiento del sistema gestor de base de datos (SGBD).

- Implantación larga y fácil; debido a los puntos anteriores, se podría concluir que la adaptación del personal es mucho más complicada y requiere de bastante tiempo.

- Ausencia de estándares reales: significa una excesiva dependencia hacia los sistemas comerciales del mercado. Aunque, hoy en día, una buena parte de esta tecnología está aceptada como estándar de hecho.

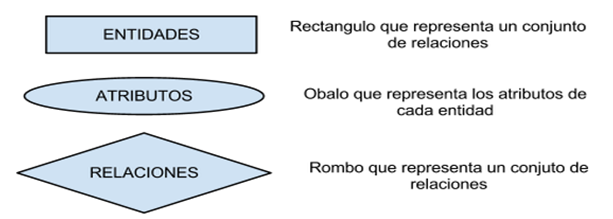
- Falta de rentabilidad a corto plazo; esto debido al coste del personal y equipos, además del tiempo que tarda en estar operativa.[[2]](#footnote-2)

**¿Qué es el modelo entidad relación?**

El modelo entidad relación, es el modo lógico de datos más extendido para el diseño conceptual de una base de datos, porque siendo un modelo relativamente simple y fácil de emplear, posee una gran capacidad expresiva. Las técnicas que este modelo emplea, sirven básicamente para especificar las necesidades de información de una organización.

El modelo entidad relación en su forma más simple implica identificar los asuntos de importancia dentro de una organización (entidades), las propiedades de los asuntos (atributos) y cómo se relacionan entre sí (relaciones), pero esto tiene valor solamente dentro del contexto de lo que se realiza en la empresa y en la forma de actuar de estas funciones de gestión sobre el modelo de información.

El siguiente gráfico representa los componentes de un modelo Entidad - Relación, a los cuales se les llama: entidades (nombre de una tablas, ej.: “Alumnos”), atributos (la información que requiere cada entidad, ej.: “Código alumno, Nombre alumno”) y relaciones (es la relación que hay entre entidades, ej.: entidad Alumno relacionada con entidad grado que cursa).

***Figura 2. Gráfica de Modelo Entidad-Relación (E-R).***

***Fuente: Elaboración propia.***

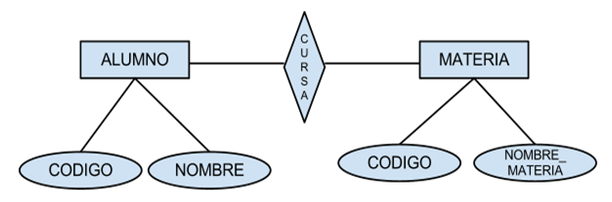
Los objetivos más fundamentales que tiene el modelo entidad relación, son dos:

- Ofrecer un modelo que refleje fielmente las necesidades de información de una organización, el cual será usado como base para el desarrollo de un sistema.

- Ofrecer un modelo independiente del posterior almacenamiento de los datos y sus métodos de acceso, lo que permitirá tomar decisiones objetivas acerca de la implementación más idónea. [[3]](#footnote-3)

En la siguiente gráfica se representa un ejemplo del modelo Entidad - Relación, el cual da una idea más clara de como trabaja el modelo junto con sus componentes (Entidades, Atributos y Relaciones), siendo los rectángulos las entidades, los óvalos los atributos y el rombo la relación.

***Figura 3. Gráfica de Ejemplo Modelo Entidad-Relación (E-R).***



***Fuente: Elaboración propia.***

**¿Qué es un SGBD?**

SGBD, es un software que tiene la capacidad de definir, mantener y utilizar una base de datos, además un sistema de gestión de base de datos debe proveer las siguientes operaciones fundamentales:

- Crear, modificar, eliminar y obtener la estructura asociada al esquema lógico de una base de datos.

- Instanciar datos operativos de una base de datos, modificar dichas instancias, eliminarlas y recuperarlas bajo diferentes criterios de búsqueda.

**Objetivos de un SGBD.**

- Garantiza la independencia de los datos.

- Diseño y utilización oriente al usuario; los datos y aplicaciones deben ser accesibles de la manera más amigable posible. Para ello, los SGBD han de fundamentarse en un modelo de datos teórico coherente y proporcionar lenguajes que permitan definir la estructura de la base de datos y acometer la generación, mantenimiento y acceso a los datos en términos que resulten apropiados.

- Centralización; los datos deben gestionarse de forma centralizada e independiente de las aplicaciones. Para satisfacer este objetivo un SGBD habrá de proporcionar una serie de utilidades que faciliten la administración de este fondo común.

- Evitar la redundancia y gestionar la concurrencia; como un SGBD provee un fondo de datos común que puede ser compartido por varias aplicaciones, por lo tanto no es preciso duplicar datos. Sin embargo, puesto que varias aplicaciones pueden acceder a los mismos datos al mismo tiempo, el SGBD debe disponer de los mecanismos adecuados para gestionar esas concurrencias.

- Mantener la integridad semántica de los datos; un SGBD debe y puede proveer de mecanismos para evitar las alteraciones originadas por los datos.

- Mantener la seguridad; un SGBD debe proveer un mecanismo que evite que algún usuario puede acceder a información no autorizada, utiliza mecanismos de identificación y gracias a ellos se podrá definir qué usuario accede a que recurso y cual no.

- Mantener la fiabilidad del sistema; algunos sistemas tienen mecanismos redundantes a nivel de hardware y de software para garantizar que el fallo de un componente sea asumido por otro sin que se pierda la información ni la disponibilidad de la base de datos.[[4]](#footnote-4)

**Lenguajes de programación, herramientas de modelado, entornos de desarrollo y gestor de base de datos**

El proyecto MUVFABA se podría desarrollar con las siguientes herramientas, lenguajes de programación y sistemas gestores de base de datos. Dependiendo de las características y requisitos del proyecto, se podrá escoger la herramienta que cumpla con estos requisitos, sea más apropiada y eficaz para su desarrollo.

**Lenguajes de programación**

* **PHP**

En el caso de implementar la guía virtual como página web se recomienda utilizar el lenguaje de programación PHP, el cual es uno de los lenguajes que permiten a la mayoría de los programadores crear aplicaciones robustas de una forma rápida, eficiente y asequible. PHP corresponde a las iniciales de personal home page tools (herramientas para páginas iniciales personales).

PHP es un lenguaje de programación muy potente que, junto con html, permite crear sitios web dinámicos. Se instala en el servidor y funciona con versiones de Apache, Microsoft IIs, Netscape Enterprise Server y otros.

La forma de usar php es insertando código php dentro del código html de un sitio web. Cuando un cliente (cualquier persona en la web) visita la página web que contiene este código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador.[[5]](#footnote-5)

* **Java**

En el caso de implementar la guía virtual como aplicación se recomienda utilizar el lenguaje de programación Java, que gracias a sus ventajas se convierte en una herramienta eficaz para el desarrollo de un software con una interfaz amigable para el usuario.

“Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.”

“Las aplicaciones de Java están típicamente compiladas en un bytecode, aunque la compilación en código nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.” [[6]](#footnote-6)

* **Netbeans (Entorno de desarrollo-IDE)**

Se recomienda utilizar esta herramienta en el caso de elegir el lenguaje de programación JAVA.

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Es un protocolo libre y gratuito sin restricciones de uso. MicroSystems fundó el proyecto en Junio de 2000.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos.[[7]](#footnote-7)

* **Bluegriffon**

Se recomienda utilizar esta herramienta en el caso de escoger PHP como lenguaje de programación.

BlueGriffon es una aplicación que proporciona a los desarrolladores una interfaz de usuario sencilla y permite crear sitios web atractivos sin necesidad de amplios conocimientos técnicos sobre estándares web.[[8]](#footnote-8)

**Gestor de base de datos**

* **Mysql**

El gestor de base de datos Mysql probablemente es el gestor más usado en el mundo de software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.[[9]](#footnote-9)

* **PostgreSQL**

PostgreSQL es un potente sistema objeto-relacional de bases de datos de código abierto. Se ha ganado una sólida reputación por su fiabilidad, integridad de datos y la corrección, además se ejecuta en todos los principales sistemas operativos, incluyendo Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows.

Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, triggers y procedimientos almacenados (en varios idiomas). Se incluye la mayor parte de SQL: 2008 los tipos de datos, incluyendo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, incluyendo imágenes, sonidos o vídeo.

Cuenta con interfaces nativas de programación C / C + +, Java, NET, Perl, Python, Ruby, Tcl, ODBC, entre otros, y documentación excepcional.

Es altamente escalable, tanto en la gran cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede acomodar. Hay sistemas activos de PostgreSQL en entornos de producción que manejan más de 4 terabytes de datos.[[10]](#footnote-10)

* **Apache**

Se recomienda utilizar como servidor web y también es de código abierto. Apache es un servidor de web. Un servidor web es un software que responde a las solicitudes de los navegadores web. En estos momentos, Apache es uno de los servidores web más populares del mundo. Ello se debe, entre otras cosas, a que Apache es un software de alta calidad y de código abierto (open source), lo que significa que puede descargarse de forma gratuita desde Internet.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido. Tiene amplia aceptación en la red y desde 1996, es el más usado.[[11]](#footnote-11)

**Herramienta para Uml**

* **Opción 1: Umbrello UML modeller**

Es una herramienta libre para crear y editar diagramas UML, que ayuda en el proceso del desarrollo de software. Fue desarrollada por Paul Hensgen, y está diseñado principalmente para KDE, aunque funciona en otros entornos de escritorio.

Umbrello UML Modeller ayuda en el proceso de desarrollo de software usando el estándar 'Unified Modelling Language' (UML) que le permitirá crear diagramas para diseñar y documentar los sistemas.[[12]](#footnote-12)

* **Opción 2: Star UML**

StarUML es un proyecto de código abierto para el desarrollo rápido, flexible, extensible, con muchas características, y libremente disponible-UML / MDA plataforma se ejecuta en la plataforma Win32. El objetivo del proyecto StarUML es construir una herramienta de modelado de software y plataforma también que es un remplazo convincente de herramientas comerciales de UML como Rational Rose, Together, etc.[[13]](#footnote-13)

## 4.2 Estado del arte

**Georreferenciación de mapas antiguos**

**Introducción: ¿para qué georeferenciar?**

La adopción masiva de tecnología en las cartotecas ha facilitado la digitalización de los mapas antiguos para su preservación y difusión. Escáneres de alta resolución, repositorios digitales o recolectores de metadatos son conceptos habituales en las cartotecas. En el año 2007 el Institut Cartografic de Catalunya (ICC) inauguró sus colecciones digitales con el repositorio de la Cartoteca Digital. En 2012 ha superado las 36.000 imágenes en alta resolución de mapas y fotografías antiguas, documentos muy valiosos por su información histórica y geográfica. Tras esta experiencia hemos constatado que existe la necesidad de disponer de información georeferenciada en múltiples ámbitos de actividad, tanto académica como de negocio.

En la bibliografía sobre patrimonio cartográfico se lleva años tratando el tema de la georreferenciación. Las principales ventajas de georeferenciar mapas antiguos son:

1. Mejorar las interfaces de visualización. Los documentos georeferenciados pueden ser visualizados con sistemas familiares como Google Maps, Google Earth o servicios de mapas que utilicen estándares abiertos OGC, como los visores de mapas oficiales (Visor del IGN, Vissir ICC, IDEs). En el caso de las series cartográficas en papel, georeferenciar índices y hojas será el primer paso para crear un mapa continuo, que posteriormente también podrá ser mostrado como una capa de referencia.

2. Renovar los catálogos de las cartotecas, convirtiéndolos en buscadores geográficos.

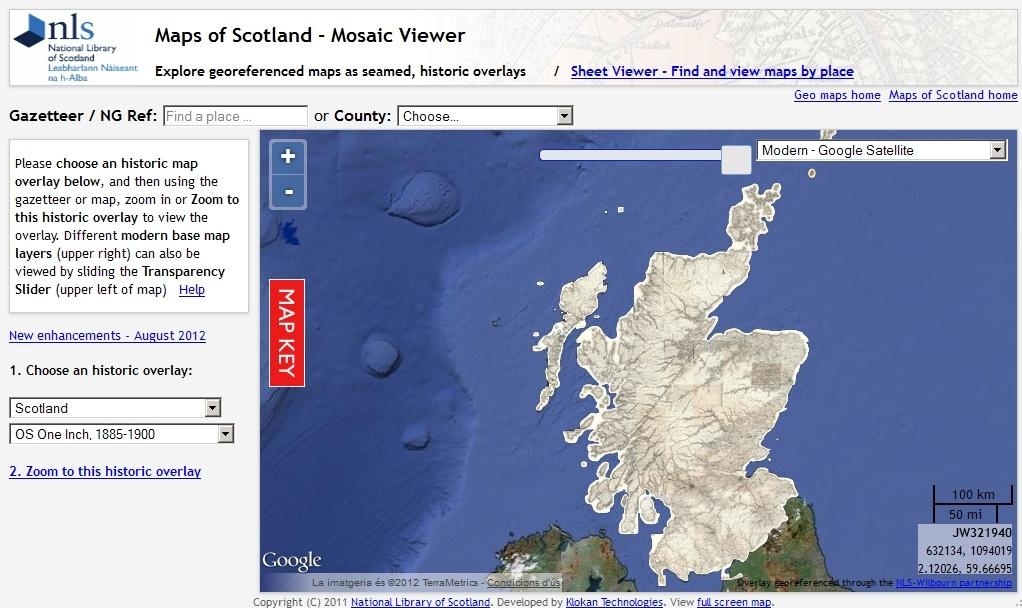
Asignar áreas geográficas alfanuméricas a los documentos ha sido una ardua tarea para las bibliotecas con un resultado muchas veces frustrante para los usuarios. Describir la dimensión espacial de determinados materiales con uno o varios topónimos resulta ambiguo y está destinado la obsolescencia. Las coordenadas geográficas en cambio, aportan precisión y estabilidad a la recuperación de la información. Soluciones de búsquedas geográficas como Map Rank, han sido implementadas con éxito en la colección David Rumsey o el portal federado de mapas.

3. Innovar en la investigación de la historia de la cartografía, gracias a software específico para el estudio de la exactitud y la geometría de las piezas. Es el caso de MapAnalyst, cuyo objetivo es evaluar la distorsión y la precisión de mapas antiguos.

4. Enriquecer los mapas antiguos mediante la integración con otra información científica, humanística o social. La cartografía antigua sirve de base para la investigación histórica en muchos campos científicos. Muestra de ello es la corriente de las humanidades digitales y los proyectos concretos como la iniciativa sobre historia urbana, (VUG). VUG, impulsado por la University of Edinburgh, es una experiencia para facilitar a los historiadores la inserción de datos históricos estadísticos y factuales sobre mapas antiguos georeferenciados.

5. Modernizar la difusión del patrimonio cartográfico, sumando útiles y demandadas opciones como comparar mapas antiguos y/o contemporáneos mediante transparencias, herramientas de medida y dibujo.

***Figura 4. Gráfica de Proyecto de georreferenciación de mapas antiguos.***



***Fuente:*** *Revista digital de cartografía y ciencias de la tierra*

<http://www.rcg.cat/images/46_12_01.jpg>

**Georeferenciar desde las cartotecas**

Georeferenciar requiere de tiempo, software específico, entrenamiento y pericia por parte del operador. Las bibliotecas luchamos continuamente con la falta de personal y presupuesto, lo que conlleva una serie de obstáculos que determinarán la planificación de un programa de georreferenciación dentro de una cartoteca.

Algunas instituciones han trabajado esta vertiente e iniciado la georreferenciación de materiales cartográficos, tanto para facilitar la visualización -mediante Google Maps o Google Earth como en la colección de David Rumsey, como para poner en marcha un buscador geográfico como en el caso de Oldmapsonline liderado por la University of Portsmouth o Maps of Australia promovido por la National Library of Australia.

Excepcional es el caso de la red de bibliotecas suizas (Oehrli, M. et al, 2011), donde desde 1995 es obligatorio describir la información geográfica según su adaptación de las Reglas Angloamericanas. Todos sus registros bibliográficos MARC 21contienen coordenadas geográficas, con lo que fue relativamente rápido lanzar su buscador geográfico Kartenportal.ch

La mayor parte de la georreferenciación se lleva a cabo localmente con sistemas de información geográfica (SIG) de escritorio, tanto comerciales - ArcGis, Autodesk, MapInfo, etc.- como de código abierto - GvSig, Qgis, GDAL...-. Para ello, se requiere de recursos económicos (más altos incluso en el caso de software privativo) además de personal disponible y entrenado. Si se dedica tiempo y recursos, los resultados son de muy alta calidad.

La posibilidad de usar sistemas en línea para asignar coordenadas a mapas antiguos, la inició la empresa MetaCarta en 2006 con Map Rectifier, aplicación online que permite subir una imagen ráster, rectificarla y luego exportarla en GeoTiff o un WMS estándar. El proceso es más simple que con el uso de un SIG profesional, así que la tarea de georeferenciar resulta asequible también para personal poco entrenado.[[14]](#footnote-14)

# 5. METODOLOGÍA

## 5.1 Tipo de estudio

### 5.1.1 Propósito

#### **5.1.1.1 Básico:**

Desarrollo a nivel avanzado de bases de datos bajo programas de software libre.

#### **5.1.1.2 Aplicado:**

Migración del sistema de archivos a un sistema de gestión de base de datos, con interfaz gráfica para el usuario final.

#### **5.1.1.3 Desarrollo Tecnológico:**

Creación de un mapa de georreferenciación virtual desarrollado en software libre.

#### **5.1.1.4 Innovación Tecnológica:**

Fusionar el mapa de ubicación virtual con el sistema de gestión de salones y materias de FABA.

### 5.1.2 Enfoque

#### **5.1.2.1 Conducido:**

Aprovechar el conocimiento impartido por el Departamento de sistemas, por medio de los aportes académicos y los proyectos de investigación en curso del personal (docentes y estudiantes) a través del apoyo de la universidad.

#### **5.1.2.2 Participativo:**

Teniendo en cuenta el enfoque que tiene el departamento de sistemas respecto al software libre, los proponentes del proyecto (estudiantes del departamento de sistemas), por medio de la investigación y el desarrollo tecnológico pretenden resolver la problemática que tiene la universidad con respecto a la geo-referenciación. Tanto la interfaz del proyecto como la base de datos serán desarrollados teniendo en cuenta el enfoque del departamento de sistemas (software libre).

### 5.1.3 Diseño

#### **5.1.3.1 Observacional:**

#### Como fuente de información directa para reunir datos, se recurrirá al personal docente y administrativo de la fundación, quienes proporcionarán la información requerida con respecto a la geo-referenciación, piedra angular del proyecto. Para cumplir este objetivo se realizarán encuestas para la comunidad de la FABA y para los visitantes de la fundación.

#### **5.1.3.2 Intervencional:**

Se realizarán encuestas a la comunidad (estudiantes, docentes, personal administrativo y visitantes) para determinar los inconvenientes que tiene la comunidad FABA respecto a la ubicación espacio temporal. Después de la implementación del mapa de ubicación virtual, se realizará una nueva encuesta para analizar cuál seria el cambio que generaría el proyecto en la universidad.

### 5.1.4 Seguimiento

#### **5.1.4.1 Transversal:**

Se observa que dentro de la universidad se presentan problemas de ubicación (espacio-tiempo). Se ha evidenciado que los miembros de la comunidad FABA, ya no se acercan a la guía de ubicación de las aulas, ya que la información registrada en esta guía está desactualizada desde hace algún tiempo (por lo menos dos años), además su diseño es poco llamativo; dilema que se evidencia de primera impresión.

#### **5.1.4.2 Longitudinal:**

**1.** Se realizaron dos encuestas, una a la comunidad Faba y la otra a los visitantes de la universidad para estudiar el porcentaje de dificultad que han tenido los miembros de la comunidad, con respecto a la ubicación de los lugares de destino, resultados fundamentales para la elaboración del proyecto en beneficio de la fundación.

**2.** Para reunir información histórica sobre la universidad (“La fundación tecnológica autónoma de Bogotá (FABA), ubicada en la Carrera 14 No 80-35 en el barrio el Lago de la ciudad de Bogotá, se constituyó como centro de educación superior a nivel tecnológico de carácter privado en el año de 1994 y entró en convenio con la Universidad de Santander (UDES) en el año de 1996”), se requirieron varios días (información difícil de conseguir por la desactualización de datos dentro del plantel educativo).

**3.** Se tomaron fotos de la universidad como muestra de la desactualización de la guía de ubicación existente en la FABA. Además, estas fotografías se utilizarán para el desarrollo de la interfaz del proyecto.

## 5.2 Unidad de estudio

## 5.3 Variables

### 5.3.1 Cuantitativas

**1.** Carencias de una guía de ubicación actualizada y llamativa dentro de la universidad.

**2.** Uso de la guía ubicada en el primer piso de la universidad.

### 5.3.2 Cuantitativas

**1.** Pérdida de tiempo al momento de ubicar algún recinto en la universidad.

**2.** Si se tiene en cuenta que la universidad tiene un énfasis tecnológico, tal como lo demuestra su nombre, la carencia de tecnologías novedosas genera un impacto negativo dentro y fuera de esta, lo cual disminuye su credibilidad ante la comunidad, dañando su prestigio y disminuyendo así el número de matrículas registradas en las distintas carreras tecnológicas.

## 5.4 Estrategia procedimental

### 5.4.1 Recolección de los datos

Como método de investigación, se iniciará por la observación. Cabe mencionar que este método no tiene algún registro tangible pero será de ayuda para para estudiar el problema tal y como se presenta en realidad. Luego, como método de medición ya mencionado anteriormente se realizarán registros con las dos encuestas realizadas virtualmente (miembros comunidad y visitantes).

### 5.4.2 Procesamiento y análisis de los datos

Se pretende hacer unas encuestas a los miembros de la comunidad FABA, con el fin de reunir información sobre la ubicación dentro de la universidad, datos importantes y fundamentales para el desarrollo del proyecto; lo expuesto anteriormente se hará por medio de una herramienta de Google (Docs “Formularios”), la cual mostrará un análisis estadístico de los resultados de dichas encuestas.

### 5.4.3 Instrumentos

Encuestas virtuales en dos formatos, el primero para los miembros directos de la comunidad FABA (estudiantes, docentes) y el segundo para los visitantes.

## 5.5 Etapa de investigación

Este proyecto se desarrollará bajo las siguientes etapas:

### 5.5.1 Investigación

A partir del planteamiento del problema, el cual se determinó en la descripción del proyecto, se iniciará una investigación que debe ir acompañada de una revisión bibliográfica, consulta de expertos y recolección de datos.

#### **5.5.1.1 Limitación del problema**

Se reconocerán las carencias que tiene la universidad con respecto a la georreferenciación y se plantearán posibles soluciones para el mejoramiento de la ubicación dentro del plantel educativo.

Este proyecto se limitará a analizar el estado de la georreferenciación de la FABA, excluyendo edificios aledaños y zonas que no tengan que ver directamente con la universidad.

Busca ser un piloto para futuras investigaciones de georreferenciación, por lo tanto se dejarán los fundamentos de la base de datos, así como un prototipo de la aplicación interfaz de usuario.

#### **5.5.1.2 Métodos de desarrollo**

Como método de desarrollo se utilizará el modelo “RUP” y el lenguaje de modelado unificado “UML” para los casos de uso.

Para el desarrollo del software se recomienda emplear las siguientes herramientas de programación: php con html, bajo el entorno de desarrollo bluegriffon y como gestor de base de datos postgresql bajo su propio IDE.

#### **5.5.1.3 Recolección de datos**

Como fuente de información directa para reunir datos, se recurrirá al personal docente y administrativo de la fundación, quienes proporcionarán la información requerida con respecto a la georreferenciación, piedra angular del proyecto. Para cumplir este objetivo se realizarán encuestas para la comunidad de la FABA y para los visitantes de la fundación.

Además, se tomarán las respectivas fotos de la universidad, piso por piso para la elaboración del prototipo de la página web. (Ver anexo B en el cd adicional)

### 5.5.2 Diseño previo

Creación del diseño previo del software. En esta etapa se estipularán las bases del software. Luego, se hará una revisión y de acuerdo a ella se realizarán los ajustes necesarios, de ahí se pasará a la etapa de aprobación del diseño.

#### **5.5.2.1 Revisión y ajustes**

Se entregará un primer informe al tutor del proyecto, quien determinará las falencias y ajustes necesarios que se deben corregir para que se cumplan las expectativas.

#### **5.5.2.2 Aprobación**

Posteriormente a los ajustes se harán las revisiones necesarias hasta que el tutor dé el visto bueno del diseño.

### 5.5.3 Desarrollo

En función del modelo decidido en el módulo anterior y teniendo en cuenta las herramientas que se propondrán para el desarrollo del programa, se realizarán los menús y módulos. Ahí, se dividirán las labores entre los integrantes del proyecto y se hará una prueba al programa, seguidamente una revisión y ajustes por parte del tutor el cual dará el visto bueno.

#### **5.5.3.1 Primera fase**

Se hará la división de labores que realizará cada miembro del grupo, luego se realizarán los casos de uso del prototipo y la base de datos para el usuario, los cuales serán entregados al tutor. Después se elaborarán los esquemas del prototipo para el software y la base de datos, estableciendo parámetros referentes a la elaboración del proyecto. Posteriormente, una prueba al prototipo y una revisión del diseño de la base de datos, la cual servirá para corregir los errores que se vayan presentando en el proceso, hasta obtener un resultado satisfactorio. Después se entregará el esquema de la base de datos y esquema del prototipo al tutor del proyecto quien dará el visto bueno.

##### 5.5.3.1.1 División de labores

Para una mayor eficacia en el desarrollo de la primera fase, se asignarán las labores que van a asumir los integrantes del grupo. Base de datos, diseño e interfaz y manuales.

##### 5.5.3.1.2 Creación de los casos de uso de la interfaz para usuario

Se hará un documento que especifique los casos de uso bajo el Lenguaje Modelado Unificado (UML), su descripción y relación (actores, módulos) para especificar el comportamiento del prototipo, posteriormente se entregará al tutor del proyecto.

##### 5.5.3.1.3 Creación de los casos de uso de la base de datos para usuario

Se hará un documento que especifique los casos de uso bajo el Lenguaje Modelado Unificado (UML), su descripción y relación (actores, módulos) para especificar el comportamiento de la base de datos, posteriormente se entregará al tutor del proyecto.

##### 5.5.3.1.4 Esquema de la base de datos

Se realizará el diseño de la base de datos conforme al modelo entidad-relación y se definirán las entidades, atributos y relaciones, que se puedan presentar en el proceso.

##### 5.5.3.1.5 Estructuración del prototipo del software y prueba

Se hará un prototipo de la página web, en el que se definirán los módulos de acceso y tipos de usuario que van a utilizar el programa, dependiendo de su rol en la universidad para probar la buena interacción del usuario con el programa.

Se realizará una prueba del prototipo del software a una muestra de la población (profesores que tengan conocimiento en el área de sistemas).

##### 5.5.3.1.6 Entrega del esquema de la base de datos

Se hará la entrega del esquema de la base de datos al tutor del proyecto, quien comprobará la efectividad del esquema y definirá las falencias existentes.

##### 5.5.3.1.7 Entrega del prototipo del software

Se hará la entrega del prototipo del software al tutor del proyecto, quien comprobará la efectividad de la estructura y el ajuste de su diseño, además definirá las falencias existentes.

##### 5.5.3.1.8 Ajustes del esquema de la base de datos

Posteriormente a la revisión del esquema, se realizarán las correcciones necesarias determinadas por el tutor para que la base de datos sea efectiva.

##### 5.5.3.1.9 Ajustes al prototipo del software

Se analizarán los resultados obtenidos en la prueba y de acuerdo a las conclusiones se realizarán los ajustes pertinentes que determine el tutor del proyecto.

##### 5.5.3.1.10 Aprobación del esquema de la base de datos

El tutor dará el visto bueno a la base de datos, cuando considere que se han cumplido las expectativas.

##### 5.5.3.1.11 Visto bueno del prototipo del software

Se entregará un informe de resultados al tutor, quien hará una revisión general de los módulos modificados y designará el visto bueno, cuándo el prototipo cumpla las expectativas establecidas anteriormente.

#### **5.5.3.2 Segunda fase**

Una vez realizados la base de datos y el prototipo del software, se hará una entrega del informe del software y se realizará la muestra del proyecto al comité de proyectos.

##### 5.5.3.2.1 Entrega del informe (prototipo)

Se hará la entrega del informe del software al tutor del proyecto, quien dirá que correcciones se deben realizar.

##### 5.5.3.2.2 Aprobación del software

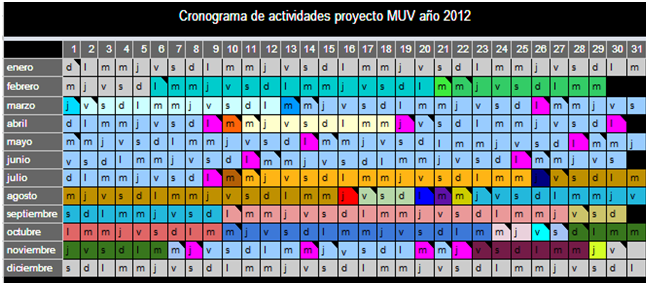
El tutor dará el visto bueno al software, cuando considere que se han cumplido todas las expectativas del software.

##### 5.5.3.2.3 Muestra del proyecto

##### 

Después de la aprobación del software, se hará una muestra al comité de proyectos.

## 5.6 Cronograma de actividades





# 6. PRODUCTOS

## 6.1 Resultados obtenidos

### 6.1.1 Generación de conocimiento y/o nuevos desarrollos tecnológicos o creativos.

* Aplicativo: Mapa de Ubicación Virtual para FABA (ver cd adicional).
* Aportes creativos de acuerdo a la metodología escogida (fases).
* Diseñar el esquema de la base de datos de acuerdo al prototipo.

### 6.1.2 Fortalecimiento de la capacidad científica

Generación de conocimientos científicos y tecnológicos para la comunidad FABA a través de la investigación y el desarrollo tecnologías.

### 6.1.3 Apropiación social del conocimiento

Motivación del departamento de sistemas a seguir creciendo e innovando.

## 6.2 Impactos o proyecciones

### 6.2.1 Temáticos:

Demostrar a la comunidad externa que es posible la implementación de tecnologías dentro de la universidad, despertando así el interés de futuros posibles miembros de la comunidad.

### 6.2.2 Territoriales:

**1.** Posicionamiento de la imagen de la universidad con respecto al desarrollo tecnológico.

**2.** Mayor eficiencia con respecto a la ubicación espacial de la comunidad FABA.

**3.** Demostrar las viabilidades del uso del software libre para beneficio de la universidad.

## 6.3 Requerimientos

Teniendo en cuenta que el software será vía web, el proyecto “MUVFABA” necesita los siguientes requerimientos:

### 6.3.1 Del software usuario y administrador

#### **6.3.1.1 Funcionales del usuario**

**1.** El usuario puede ingresar a la página web.

**2.** El usuario puede crear una zona desde la página web.

**3.** El usuario puede editar, cambiar o modificar alguna zona desde la página web.

**4.** El usuario puede eliminar alguna zona desde la página web.

**5.** El usuario puede buscar zona especial.

**6.** El usuario puede buscar aula.

**7.** El usuario puede ver el edificio (ir al inicio).

**8.** El usuario puede ver las aulas de su interés.

**9.** El usuario puede ver la zona especial que desee.

**10.** El usuario puede ver el mapa de un piso.

**11.** El usuario puede ver un piso del edificio (universidad).

**12.** El usuario puede ver las búsquedas recientes.

**13.** El usuario tiene una opción de ayuda.

**14.** El usuario puede regresar a la acción que realizó anteriormente.

**15.** El usuario puede salir en cualquier momento de la página.

#### **6.3.1.2 Funcionales del administrador**

Además de las siguientes funciones, el usuario administrador puede ejecutar las mismas funciones que realiza un usuario normal.

**1.** El administrador puede ingresar a la página para actualizar la información de la base de datos.

**2.** El administrador puede editar el historial de búsqueda de la página.

**3.** El administrador puede editar zona especial de la página.

**4.** El administrador puede editar aula de la página.

**5.** El administrador puede actualizar mapa de la página.

**6.** El administrador puede actualizar fotos de la página.

**7.** El administrador puede seleccionar aula de la página.

**8.** El administrador puede seleccionar zona especial de la página.

#### **6.3.1.2 No funcionales**

**1.** Capacidad de respuesta ante cualquier eventualidad presente en tiempo de ejecución (máximo 10 segundos), permitiéndole al usuario identificarlo y corregir su falla.

**2.** El diseño de la página web debe ser llamativo.

**3.** La página web debe ser rápida, por ende se debe cargar en máximo 30 segundos.

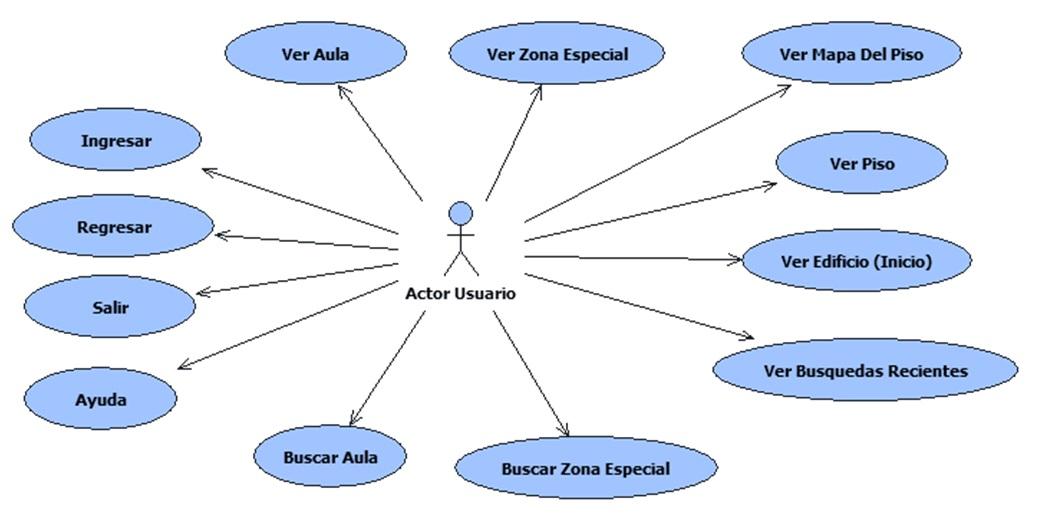
**4.** Se debe lograr una buena interacción del usuario con el software.

## 6.4 Casos de uso

Para la interfaz grafica de la base de datos, se han determinado dos tipos de actores, el actor usuario y el actor administrador. El administrador puede ejercer las mismas operaciones del actor usuario, al mismo tiempo que cuenta con operaciones propias de su rol.

### 6.4.1 Casos de uso actor Usuario:

***Figura 5. Diagrama de casos de uso de usuario***



***Fuente: Elaboración propia***

|  |  |  |
| --- | --- | --- |
| **ID** | **OPERACION** | **DESCRIPCION** |
| **1** | **Ingresar** | **El usuario ingresa a la interfaz de la base de datos para efectuar su búsqueda** |
| **2** | **Regresar** | **El usuario vuelve al módulo visitado anteriormente** |
| **3** | **Salir** | **El usuario sale de la interfaz de la base de datos** |
| **4** | **Ayuda** | **Abre una ventana de ayuda para el usuario** |
| **5** | **Buscar Aula** | **Efectúa la búsqueda de un aula específica** |
| **6** | **Buscar Zona Especial** | **Efectúa la búsqueda de una zona especial específica** |
| **7** | **Ver Búsquedas Recientes** | **Abre un módulo en el cual aparecen registradas las búsquedas del día.** |
| **8** | **Ver Edificio (Inicio)** | **Regresa a la página de inicio de la interfaz, en la cual hay imágenes del edificio** |
| **9** | **Ver Piso** | **Abre un módulo el cual aparece el piso seleccionado** |
| **10** | **Ver Mapa Del Piso** | **Abre un módulo el cual aparece el mapa del piso seleccionado** |
| **11** | **Ver Zona Especial** | **Abre un módulo el cual aparece la zona especial seleccionada** |
| **12** | **Ver Aula** | **Abre un módulo el cual aparece el aula seleccionada** |

***Tabla 1. Operaciones del usuario.***

**Ingresar**

**Contrato:**

|  |  |
| --- | --- |
| **Ingresar** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ingresar a la interfaz para realizar búsquedas** |
| **Resumen** | **Al hacer clic, el usuario ingresa a la página de inicio de la interfaz, en la cual puede iniciar a hacer su búsqueda.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ingresar** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Envío al usuario de la página principal** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: no conectado con el servidor** |

***Tabla 2. Función ingresar.***

**Regresar**

**Contrato:**

|  |  |
| --- | --- |
| **Regresar** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Regresar a la ventana anterior** |
| **Resumen** | **Al hacer clic, el usuario regresa a la página anterior.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Regresar** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Regreso a la página anterior** | |
| **Flujo Erróneo** |  |

***Tabla 3. Función regresar.***

**Salir**

**Contrato:**

|  |  |
| --- | --- |
| **Salir** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Sale de la interfaz de consulta** |
| **Resumen** | **Al hacer clic en este botón, el usuario inmediatamente saldrá de la interfaz gráfica de la base de datos, volverá al módulo iniciar** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Salir** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Sale de la interfaz de la base de datos** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor** |

***Tabla 4. Función salir.***

**Ayuda**

**Contrato:**

|  |  |
| --- | --- |
| **Ayuda** |  |
| **Actores** | **Ayuda.** |
| **Propósito** | **Mostrar una ventana de ayuda al usuario.** |
| **Resumen** | **Al hacer clic, el usuario puede ver un módulo de ayuda, en el cual aparecen explicitas las opciones que puede tomar en el módulo en el que se encuentre.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ayuda** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Regreso a la página anterior** | |
| **Flujo Erróneo** | **Error: Modulo Ayuda en actualización.**  **Error: no conectado con el servidor** |

***Tabla 5. Función ayuda.***

**Buscar Aula**

**Contrato:**

|  |  |
| --- | --- |
| **Buscar Aula** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Iniciar la búsqueda de un aula** |
| **Resumen** | **Al hacer clic, el usuario ingresa al módulo “buscar aula”, en el cual puede iniciar la búsqueda del aula requerida.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Buscar Aula** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: no ha hecho ninguna consulta**  **Error: página en mantenimiento**  **Error: aula no existente**  **Error: no conectado con el servidor** |

***Tabla 6. Función buscar aula.***

**Buscar Zona Especial**

**Contrato:**

|  |  |
| --- | --- |
| **Buscar Zona Especial** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Iniciar la búsqueda de una zona Especial** |
| **Resumen** | **En el caso de que el usuario necesite realizar una búsqueda de una zona especial (biblioteca, salón de bienestar, psicología, auditorios, cafetería, entre otros),** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Buscar Zona Especial** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: no ha hecho ninguna consulta**  **Error: página en mantenimiento**  **Error: Zona Especial no existente**  **Error: no conectado con el servidor** |

***Tabla 7. Función buscar zona especial.***

**Ver Búsquedas Recientes**

**Contrato:**

|  |  |
| --- | --- |
| **Ver Búsquedas Recientes** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ver el registro de las búsquedas recientes del día** |
| **Resumen** | **Al hacer clic, el usuario regresa a la página anterior.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ver Búsquedas Recientes** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: página en mantenimiento**  **Mensaje: No se han efectuado Búsquedas en el día de hoy**  **Error: no conectado con el servidor** |

***Tabla 8. Función ver búsquedas recientes.***

**Ver Edificio (Ir Al Inicio)**

**Contrato:**

|  |  |
| --- | --- |
| **Ver Edificio** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ir a la página de inicio** |
| **Resumen** | **Si el usuario desea hacer una nueva consulta, deberá ir a la página de inicio, la cual lleva como nombre “edificio”.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ver Edificio** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Página de Inicio** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: página en mantenimiento**  **Error: no conectado con el servidor** |

***Tabla 9. Función ver edificio (Ir al inicio).***

**Ver Piso**

**Contrato:**

|  |  |
| --- | --- |
| **Ver Piso** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ver el Piso del edificio consultado** |
| **Resumen** | **El usuario ve las imágenes del piso seleccionado anteriormente.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ver Piso** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: Vista En Actualización** |

***Tabla 10. Función ver piso.***

**Ver Mapa Del Piso**

**Contrato:**

|  |  |
| --- | --- |
| **Ver Mapa Del Piso** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ver Mapa del Piso Consultado** |
| **Resumen** | **El usuario ve las imágenes del mapa del piso seleccionado anteriormente.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ver Mapa Del Piso** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: Vista En Actualización** |

***Tabla 11. Función ver mapa del piso.***

**Ver Zona Especial**

**Contrato:**

|  |  |
| --- | --- |
| **Ver Zona Especial** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ver zona especial consultada** |
| **Resumen** | **El usuario ve las imágenes de la zona especial seleccionada anteriormente.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón ver zona especial** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: Vista En Actualización** |

***Tabla 12. Función ver zona especial.***

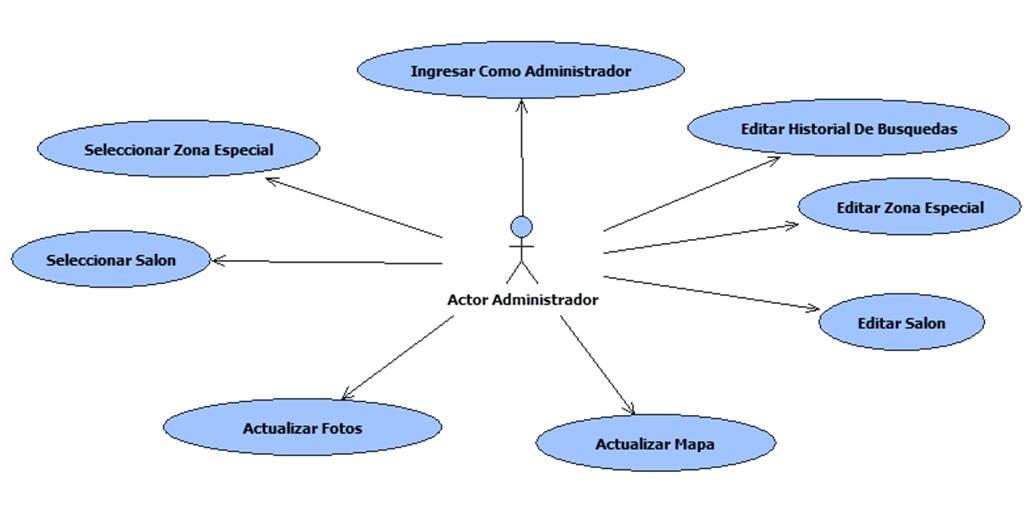
**Ver Aula**

**Contrato:**

|  |  |
| --- | --- |
| **Ingresar** |  |
| **Actores** | **Usuario** |
| **Propósito** | **Ver zona especial consultada** |
| **Resumen** | **El usuario ve las imágenes del aula seleccionada anteriormente.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ver Aula** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: Vista En Actualización** |

***Tabla 13. Función ver aula.***

**6.2.2 Casos de uso actor Administrador:**

***Figura 6. Diagrama de casos de uso de usuario administrador.***

***Fuente: Elaboración propia.***

|  |  |  |
| --- | --- | --- |
| **ID** | **OPERACION** | **DESCRIPCIÓN** |
| **13** | **Ingresar Como Administrador** | **El Administrador ingresa a la interfaz de la base de datos para Configurar la interfaz gráfica** |
| **14** | **Seleccionar Zona Especial** | **El Administrador selecciona una zona especial** |
| **15** | **Seleccionar Aula** | **El Administrador selecciona un aula** |
| **16** | **Editar Historial De Búsquedas** | **El Administrador edita el historial de búsquedas de la interfaz grafica** |
| **17** | **Editar Zona Especial** | **El Administrador edita una zona especial** |
| **18** | **Editar Aula** | **El Administrador edita un aula** |
| **19** | **Actualizar Mapa** | **El Administrador actualiza un mapa** |
| **20** | **Actualizar Fotos** | **El Administrador actualiza las fotos almacenadas en la base de datos** |

***Tabla 14. Operaciones del usuario administrador.***

**Ingresar como Administrador**

**Contrato:**

|  |  |
| --- | --- |
| **Ingresar Como Administrador** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Ingresar a la interfaz para realizar búsquedas** |
| **Resumen** | **Al hacer clic, el administrador ingresa a la página de inicio de la interfaz, en la cual puede iniciar a hacer su búsqueda.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz de la base de datos** | | **1** | **Clic en el botón Ingresar Como Administrador** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Envío al Administrador de la interfaz grafica para el administrador** | |
| **Flujo Erróneo** | **Error: Pagina no disponible**  **Error: no conectado con el servidor** |

***Tabla 15. Función ingresar como administrador.***

**Seleccionar Zona Especial**

**Contrato:**

|  |  |
| --- | --- |
| **Seleccionar Zona Especial** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Seleccionar una zona especial** |
| **Resumen** | **Al hacer clic, el administrador ingresa a un modulo en el cual aparecen las zonas especiales de la universidad, para posteriormente, editar una de ellas.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Seleccionar Zona Especial.** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en pantalla de opción requerida.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento** |

***Tabla 16. Función seleccionar zona especial.***

**Seleccionar Aula**

**Contrato:**

|  |  |
| --- | --- |
| **Seleccionar Aula** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Seleccionar una aula** |
| **Resumen** | **Al hacer clic, el administrador ingresa a un módulo en el cual aparecen las aulas de la universidad, para posteriormente, editar uno de ellos.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Seleccionar Aula.** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en pantalla de opción requerida.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento** |

***Tabla 17. Función seleccionar aula.***

**Editar Historial de Búsqueda**

**Contrato:**

|  |  |
| --- | --- |
| **Editar Historial De Búsqueda** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Editar el historial de búsquedas de la interfaz.** |
| **Resumen** | **Al hacer clic, el administrador puede ver y modificar el historial de búsqueda de la interfaz, esencialmente se eliminará el historial periódicamente.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Editar Historial de Búsqueda** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en pantalla de opción requerida.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento**  **Error: no hay nada seleccionado** |

***Tabla 18. Función editar historial de búsqueda.***

**Editar Zona Especial**

**Contrato:**

|  |  |
| --- | --- |
| **Editar Zona Especial** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Editar las zonas especiales de la interfaz.** |
| **Resumen** | **Al hacer clic, el administrador puede modificar las zonas especiales de la interfaz.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Editar Zona Especial** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en pantalla de opción requerida.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento**  **Error: no hay nada seleccionado** |

***Tabla 19. Función editar zona especial.***

**Editar Aula**

**Contrato:**

|  |  |
| --- | --- |
| **Editar Aula** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Editar las aulas de la interfaz.** |
| **Resumen** | **Al hacer clic, el administrador puede modificar las aulas de la interfaz.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Editar Aula** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en pantalla de opción requerida.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento**  **Error: no hay nada seleccionado** |

***Tabla 20. Función editar aula.***

**Actualizar Mapa**

**Contrato:**

|  |  |
| --- | --- |
| **Actualizar Mapas** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Actualizar los planos de los pisos de la universidad** |
| **Resumen** | **Al hacer clic, el administrador podrá actualizar los planos del piso que verán los usuarios de la interfaz grafica.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Actualizar Mapa** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento**  **Error: no hay nada seleccionado** |

***Tabla 21. Función actualizar mapa.***

**Actualizar Fotos**

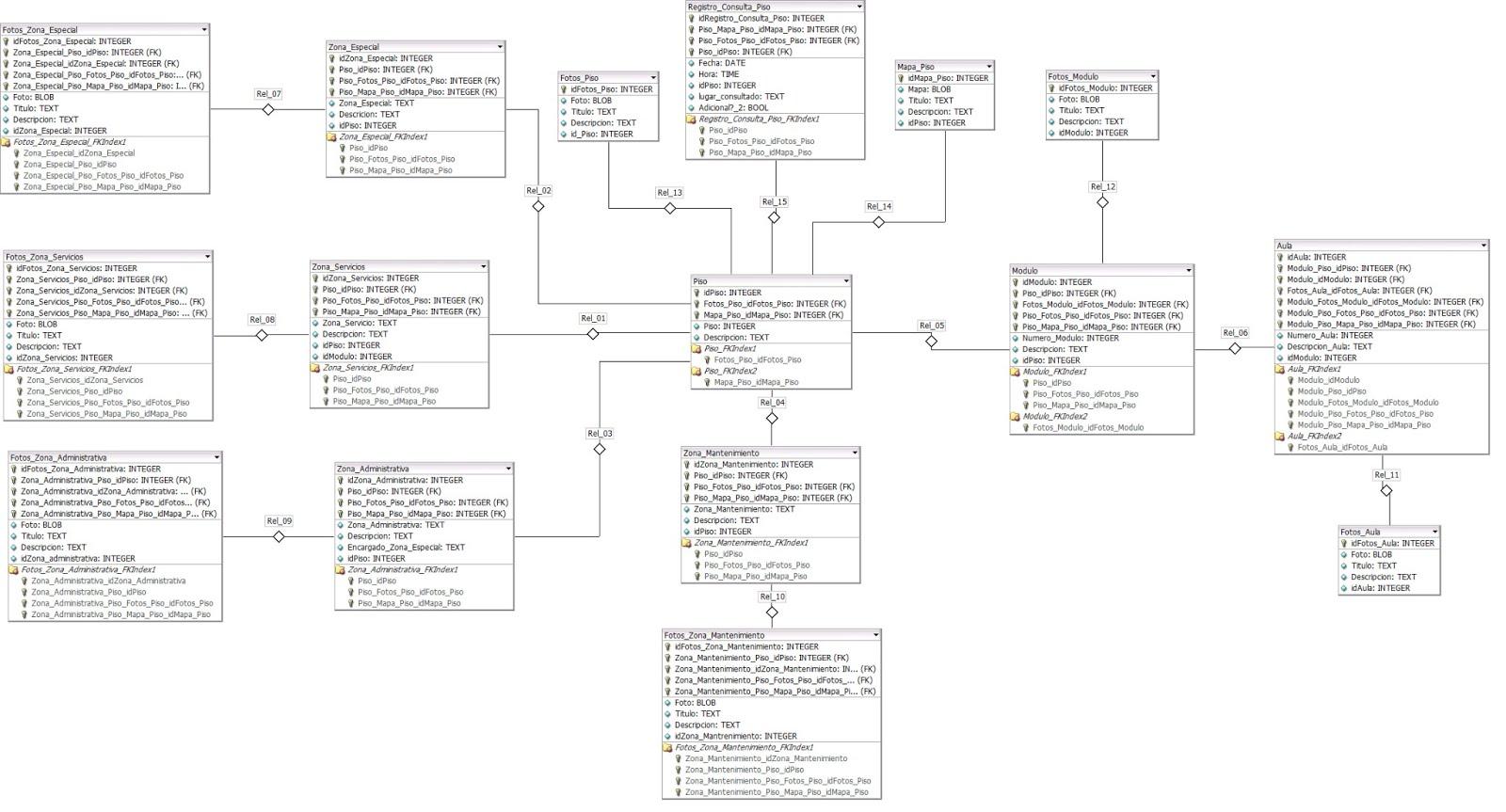
**Contrato:**

|  |  |
| --- | --- |
| **Actualizar Fotos** |  |
| **Actores** | **Administrador** |
| **Propósito** | **Actualizar las fotos de los módulos de la universidad.** |
| **Resumen** | **Al hacer clic, el administrador podrá actualizar los las fotos que verán los usuarios de la interfaz grafica. Con esta opción se pueden modificar todas las fotos de la interfaz.** |
| **Flujo Normal** | |  |  |  | | --- | --- | --- | |  | **Usuario** | **Interfaz del Administrador** | | **1** | **Clic en el botón Actualizar Fotos** |  | | **2** |  | **La interfaz analiza la opción escogida** | | **3** |  | **Impresión en la pantalla de la consulta realizada.** | |
| **Flujo Erróneo** | **Error: no conectado con el servidor**  **Error: página en mantenimiento**  **Error: no hay nada seleccionado** |

***Tabla 22. Función actualizar fotos.***

## 6.5 Modelo relacional

***Figura 7. Modelo relacional***



***Fuente: Elaboración propia***

**7. RECOMENDACIONES**

* Se recomienda utilizar herramientas de software libre para la ejecución del proyecto.
* Para la creación de la interfaz gráfica se recomienda utilizar como lenguaje de programación PHP con Html 5.
* Para la implementación de la base de datos, se sugiere tomar como base el esquema diseñado. (Ver cd adicional).
* Se aconseja que la interfaz de la página web sea cómoda y llamativa para el usuario.

# 8. CONCLUSIONES

1. El uso del software libre genera un impacto positivo para la universidad respecto a la reducción de costos, el cual se ve reflejado en este proyecto.
2. Este proyecto demostrará a la comunidad externa que es posible la implementación de tecnologías dentro de la universidad y despertará así el interés de futuros posibles miembros para la comunidad.
3. El proyecto no necesita de una infraestructura compleja para su desarrollo.
4. El proyecto es diseñado con un modelo escalable, para que sea compatible con cualquier tipo de sistema.
5. En el proyecto se reflejan por lo menos seis materias vistas en el recorrido de la carrera, entre ellas: seguridad informática, redes, bases de datos, programación, análisis y diseño de software y elementos del computador.

# BIBLIOGRAFÍA DE REFERENCIA

* PONS CAPONTE Olga, MARÍN Nicolás, MEDINA Juan Miguel, ACID Silvia, VILA Mª Amparo. *Introducción a las bases de datos: El modelo relacional*. Editorial Paraninfo. 2005.
* RIVERO Cornelio Enrique, GUARDIA RIVAS Carlos, REIG HERNANDEZ José Carlos. *Bases de datos relacionales: Diseño físico (Orientado a BD2 para z/OS de IBM).*
* BARKER Richard. *El modelo entidad-relación.* Ediciones Díaz de Santos. EEUU.1994.
* SANCHEZ Jorge. *Diseño conceptual de Base de Datos.* 2004.
* DUARTE Manuel Palomo, MONTERO PÉREZ Ildefondo. *Programación en PHP a través de ejemplos*. Autoedición. Cádiz. 2007.
* BELMONTE FERNÁNDEZ Oscar, GRANELL CANUT Carlos, ERDOZAIN María del Carmen. *Desarrollo de proyectos informáticos con tecnología Java.* Editorial Universitat Jaume I. 2012.
* MILLET Piero Berni, GIL DE LA IGLESIA Didác. *Laboratorio de PHP y MySQL*, material docente de la UOC (Universidad Oberta de Catalunya). Editorial UOC. Catalunya. 2009.
* POSTGRESQL TEAM. *PostgreSQL 9 Documentation.* Autoedición. California. 2011.

* CERIA Santiago. *Casos de uso, un método práctico para explorar requerimientos.*
* NEVADO CABALLERO María Victoria. *Introducción a las bases de datos relacionales.* Editorial Visión libros. Madrid.
* SÁNCHEZ ASENJO Jorge. *Sistemas gestores de bases de datos. Autoedición. 2009.*
* VANCHOXCHK. *Java: Bases + SQL.* Autoedición. 2011.
* SANCHEZ Sergio Hugo. *Ejercicios y prácticas PHP.* Universidad de Matamoros & IDSystems. 2009-2011
* DE LA IGLESIA Guillermo. *Guía de aprendizaje de fireworks CS5.* Editorial UOC. 2011.

# WEBGRAFÍA

* Referenciación de mapas antiguos <http://www.rcg.cat/articles.php?id=257>
* Modelo entidad-relación.

<http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf>

* Netbeans <http://netbeans.org/>
* Bluegriffon <http://www.bluegriffon.org/>
* Apache <http://www.apache.org/>
* Umbrello <http://uml.sourceforge.net/>
* Poseidon Linux <http://www.es.poseidonlinux.org/>

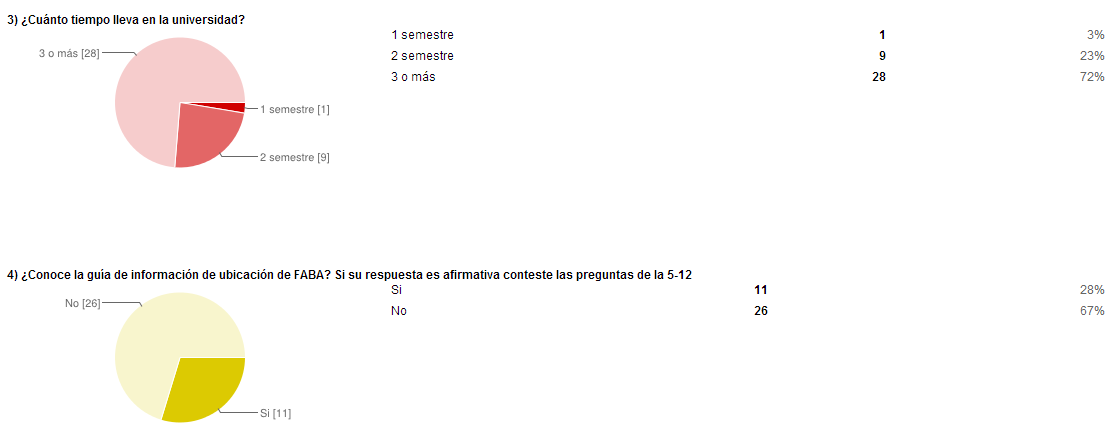
# ANEXOS

**Anexo A: Resultados encuestas**

**A1. Resultados de la encuesta para miembros de la comunidad FABA, en gráficos de decisión.**

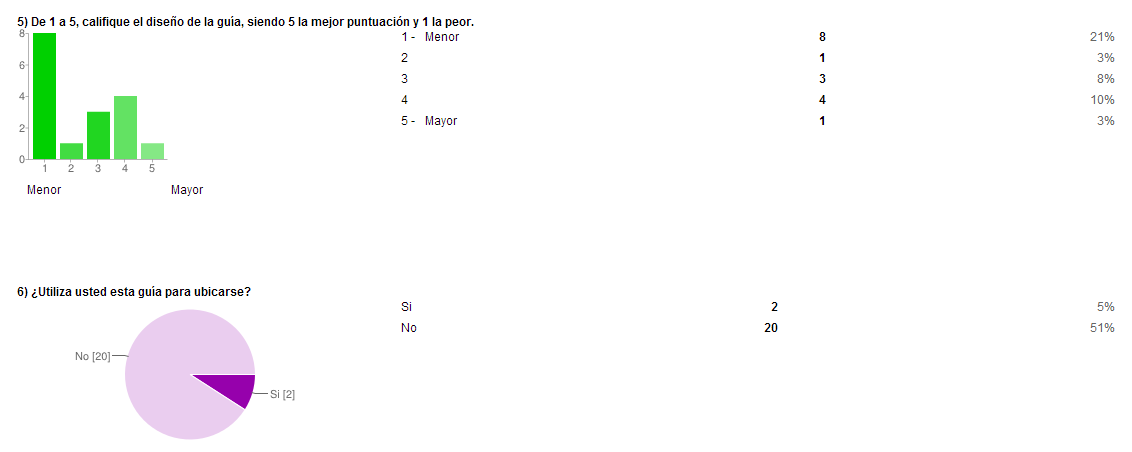
*\*Miembros de la comunidad FABA: estudiantes, docentes y personal administrativo respondiendo la encuesta para la fase de investigación.*



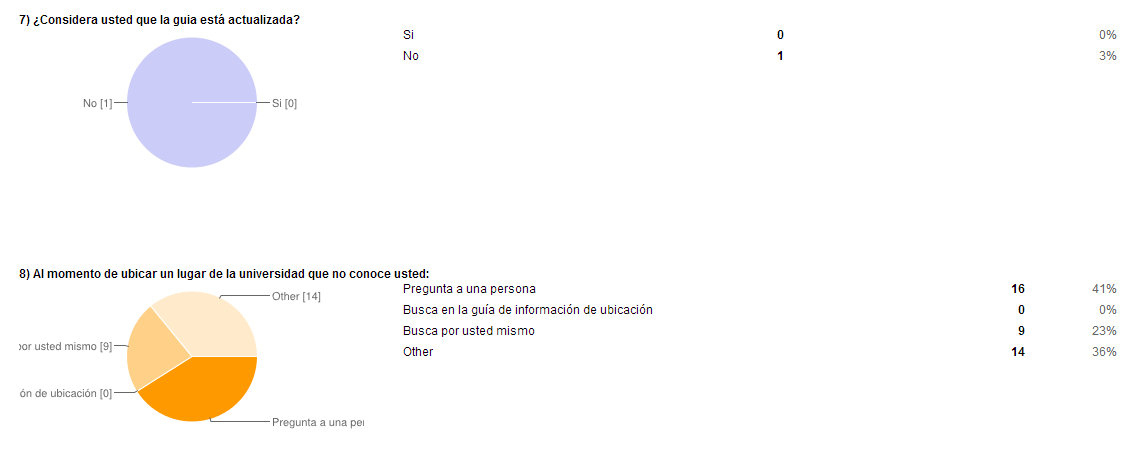
*\*Como se puede observar, un porcentaje alto de los encuestados desconoce la guía de información ubicada en FABA. Y también, un porcentaje alto lleva más de 3 semestres en FABA.*

*\*Del porcentaje que tiene conocimiento de la guía (porcentaje bajo), califican el diseño como insulso o simple.*

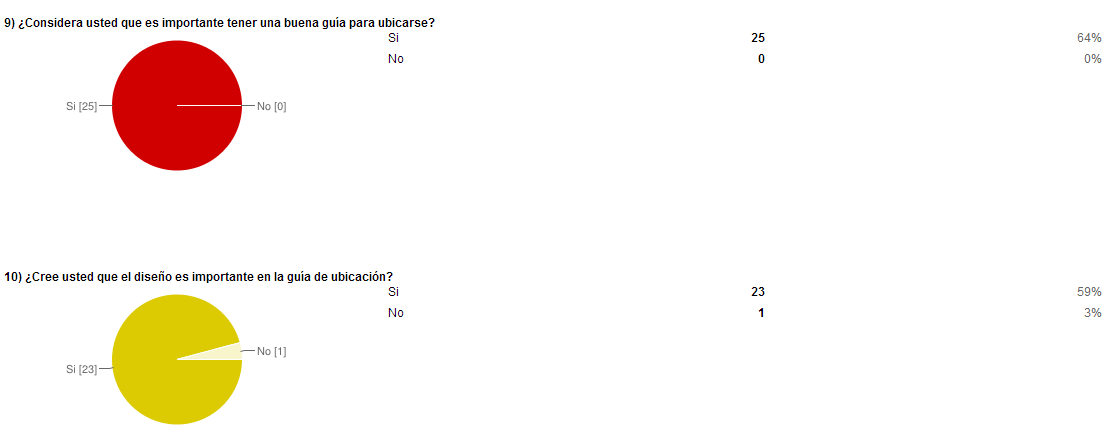
*\*No hay acercamiento de la comunidad a la guía de ubicación de las aulas.*



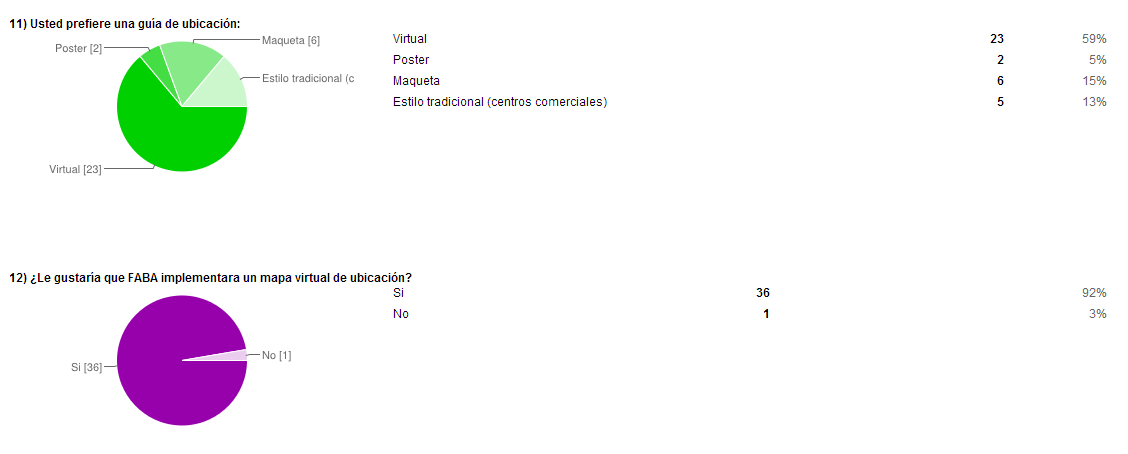
*\*Se evidencia completamente la desactualización de la guía de ubicación.*



*\*Una buena guía de ubicación reduce el tiempo de búsqueda. Para la totalidad de encuestados, la guía es de suma importancia, como también su diseño.*

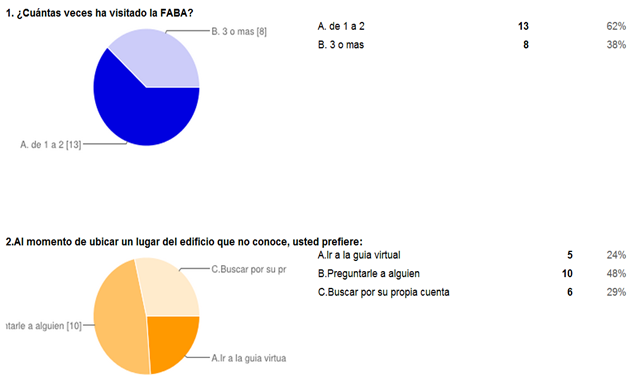


*\*Un alto porcentaje prefiere una guía de ubicación virtual y le gustaría que en FABA se implementara una.*



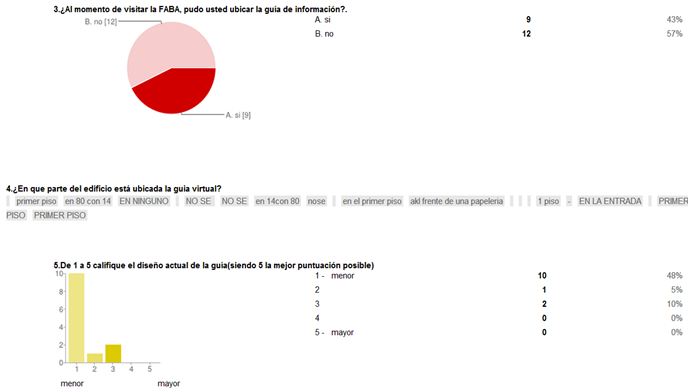
**A2 Resultados de la encuesta para visitantes de la comunidad FABA, en gráficos de decisión.**

*\*Encuesta realizada a visitantes de la comunidad, entre ellos padres de familia.*

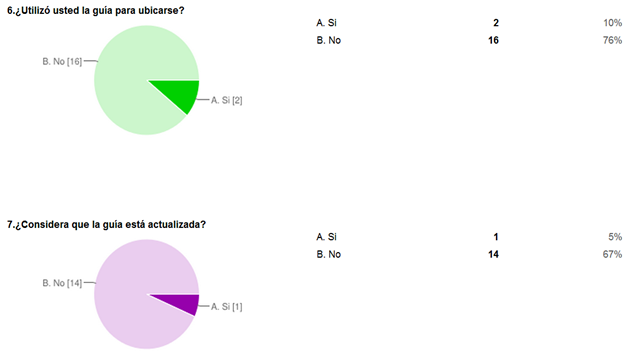


*\*Un porcentaje alto no ubican la guía actual en FABA.*

*\*También, califican el diseño como insulso.*



***\*****Las guías no deben limitarse a ser un elemento decorativo en la institución.*



1. *Pons Olga, Marín Nicolás. Introducción a las bases de datos: El modelo relacional. Editorial Paraninfo.*  [↑](#footnote-ref-1)
2. *Rivero Cornelio, Guardia Carlos, Reig José. Bases de datos relacionales: Diseño físico (Orientado a BD2 para z/OS de IBM).*  [↑](#footnote-ref-2)
3. *Richard Barker. El modelo entidad-relación. Ediciones Díaz de Santos.* [↑](#footnote-ref-3)
4. *Sánchez Jorge. Diseño conceptual de Base de Datos.*  [↑](#footnote-ref-4)
5. *Duarte Manuel, Montero Ildefondo. Programación en PHP a través de ejemplos.* [↑](#footnote-ref-5)
6. *Belmonte Oscar, Granell Carlos, Erdozain María. Desarrollo de proyectos informáticos con tecnología Java. Editorial Universitat Jaume I.* [↑](#footnote-ref-6)
7. *Netbeans* [*http://netbeans.org/*](http://netbeans.org/) [↑](#footnote-ref-7)
8. *Bluegriffon* [*http://www.bluegriffon.org/*](http://www.bluegriffon.org/) [↑](#footnote-ref-8)
9. *Millet Piero, Gil de la Iglesia Didác. Laboratorio de PHP y MySQL, material docente de la UOC (Universidad Oberta de Catalunya)* [↑](#footnote-ref-9)
10. *PostgreSQL team. PostgreSQL 9 Documentation.*  [↑](#footnote-ref-10)
11. *Apache* [*http://www.apache.org/*](http://www.apache.org/) [↑](#footnote-ref-11)
12. Umbrello UML <http://uml.sourceforge.net/> [↑](#footnote-ref-12)
13. Star UML <http://staruml.sourceforge.net/en/about.php>/ [↑](#footnote-ref-13)
14. *Revista digital de cartografía y ciencias de la tierra* [*http*://www.rcg.cat/articles.php?id=257](http://www.rcg.cat/articles.php?id=257) [↑](#footnote-ref-14)