



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

T E S I S

**SISTEMA INTEGRAL DE ADMINISTRACIÓN HABITACIONAL
(SIAH)**

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN**

PRESENTAN:

García Martínez Omar

González Escalera Marisol

Marañón Sánchez Pedro

Martínez Zanabria José Antonio

Pérez Rivera Adrián Alfonso



DIRECTOR DE TESIS: M.I. JUAN CARLOS ROA BEIZA.

CIUDAD UNIVERSITARIA

México D.F. 2015.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



Agradecimientos:

A la Universidad Nacional Autónoma de México quien nos formó como profesionistas, a los profesores, principalmente a nuestro asesor M.I. Juan Carlos Roa Beiza y al Programa de Apoyo a la Titulación (PAT) de la Facultad de Ingeniería de la UNAM que nos brindó esta facilidad para culminar una más de nuestras metas.

A mis compañeros de la Facultad de Ingeniería que participaron en la elaboración de esta tesis:

García Martínez Omar

González Escalera Marisol

Marañón Sánchez Pedro

Martínez Zanabria José Antonio

Pérez Rivera Adrián Alfonso



Agradecimientos:

A Dios.

Por darme vida y salud para llegar hasta aquí además de darme una familia maravillosa.

A mis padres.

Teófila y Roberto que me dieron la vida y han estado conmigo en todo momento. Gracias mamá por apoyarme en mi carrera, moral y económicamente y por creer en mí, gracias papá por todos los consejos que me das, con tu ejemplo me enseñaste el valor del esfuerzo. Los amo.

A mi hermana.

Mónica, por su apoyo, ánimos y porque sé que siempre cuento contigo.

A mis amigos.

Que conocí en la facultad de ingeniería y me apoyaron a lo largo de esta etapa, sobre todo a Antonio por darme animos para la realización de esta tesis y a mis amigos que conocí en el PAT, Adrián, Marisol y Pedro todo mi respeto.

A la UNAM, Facultad de Ingeniería y maestros.

Por todo el conocimiento y las experiencias vividas, sin duda soy una mejor persona, ¡infinitas gracias!.

García Martínez Omar



Agradecimientos:

Al PAT por darme ese empujón y el valor de sacar adelante este pendiente.

Al M. I. Juan Carlos Roa Beiza por cada una de sus aportaciones tanto teóricas como prácticas que nos ayudaron al desarrollo conceptual del presente trabajo.

A mis familiares por ser y estar.

Y por último y no la menos importante pero si la más difícil. A ti mi Pincesa mosha, TODO ESTO ES TUYO y sé que poco a poco estoy pagando mis deudas. Gracias por siempre estar a mi lado, por enseñarme a no rendirme a pesar de las tormentas (sabes a qué me refiero). Aquí no hay nada que no intercambiaría en un instante por tan sólo escuchar tu voz o ver tu sonrisa o sentir esos brazos cálidos una vez más. I MISS YOU SO MUCH.

A la UNAM, por el orgullo de pertenecer a la Máxima Casa de Estudios.

A mis compañeros de tesis Omar, Pedro, Antonio y Adrián, que a pesar de nuestras diferencias estamos a unos metros de la META.

A mis amig@s que me han brindado su apoyo de manera incondicional a lo largo de mi vida y han estado conmigo en las buenas y en las malas. Los amigos son LA FAMILIA QUE SI SE PUEDE ELEGIR.

Y a todos por aguantar mi carácter que no es nada fácil y tenerme paciencia.

González Escalera Marisol



Agradecimientos:

Agradezco a Dios por permitir mi existencia y tomar las decisiones que creo convenientes.

Agradezco a mis padres por su valioso apoyo que siempre me brindaron, mi cariño y respeto por siempre.

Mi agradecimientos infinito a todas aquellas personas que me apoyaron de manera incondicional para la realización de éste trabajo; así como a aquellas que me brindaron sus sabios consejos en mi formación; tanto de mi vida personal como profesional; seguro estoy que sin ellos todo hubiese tomado un rumbo diferente.

Marañón Sánchez Pedro



Agradecimientos:

A mis padres quiero agradecerles lo que ahora soy, gracias por darme la vida, por su amor, por las caricias, por el dolor, por las sonrisas, por el sufrimiento, por los regaños y por el aliento. Gracias por enseñarme a crecer, a través del sufrimiento, curándome las heridas y consolándome en mis lamentos. Gracias por el ejemplo de la honradez, del entusiasmo y la calidez, por los regaños y desacuerdos, por las verdades y descontentos. Gracias por enseñarme a dar de intensa forma y nada esperar, por los consejos y las caídas por enseñarme como es la vida. Gracias por estar a mi lado en el momento justo y el más anhelado, cuando más necesito de su apoyo, ¡Solo Dios sabe cómo los quiero! A mi madre la mujer más hermosa del planeta, la mujer más admirable que conozco que dice cada día en cada obra un sutil te quiero, a mi padre el ser más noble que he podido conocer, tienes un corazón muy bueno y puro, es capaz de transmitir su alegría, alegría que me llena de vida, de esperanza y me hace sentir muy bien. Gracias con todo mi corazón, gracias por ser como son por siempre estar ahí, que Dios no pudo escoger de una manera mejor, a mis padres, la pareja que ustedes son, la virtud de ustedes unos grandes padres vivirá para siempre.

A mi hermano la cosa horrorosa que sutilmente me apoya y se preocupa por mí, hoy quiero decirte que te quiero mucho, sin tu amistad y tu apoyo, todo hubiera sido más difícil... esta palabra sola lleva todo mi cariño y amor por todo lo que has hecho por mi...

Quiero darles las gracias a mis tíos por todo lo que me han dado, por su compañía, su apoyo, su comprensión y presencia. Por brindarme la oportunidad de tener a mi lado personas especiales, en quienes confío y con quienes me divierto.

Hay amigos a lo largo de nuestra vida, que nos demuestran mucho más que una familia, es algo que he aprendido con el tiempo, sabiendo que no hay nada más cierto. La amistad que me han brindado es única, por mucho o poco que hayamos podido compartir, quiero agradecer a Daniel, Grecia, David, Nancy, Rene, Omar, Pamela,



Ulises, Ignacio, Andrea y Eduardo, entre otros y espero no se enojen si no ven su nombre aquí (igual y ni lo leen ;)).

Debo agradecer de manera especial a mis compañeros de tesis: Omar (gracias por iniciar esta tesis conmigo), Marisol, Pedro y Adrián, por su apoyo, confianza, su amistad y su capacidad para trabajar en equipo, cada quien realizo un aporte invaluable, y no solamente en el desarrollo de esta tesis.

A la UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, en especial a la FACULTAD DE INGENIERÍA por darme la oportunidad de estudiar y ser un profesional.

Y quisiera agradecer a esa persona muy especial en mi vida, sé que lo leerás pero no estás segura si es para ti, pero lo sospechas, aunque no puedes confirmarlo, la duda me protege mientras a ti te hace sonreír.

Por eso les digo hoy a todos ellos, muchas gracias, gracias por su tiempo, su cariño, muchas gracias, porque de una u otra forma son mi impulso para seguir adelante.

Martínez Zanabria José Antonio



Agradecimientos:

Para Ana María, Alfonso y Mariana, por apoyarme en todo momento, siempre tendrán mi admiración, respeto y cariño.

Para Dolores y Alfonso, mis viejitos queridos, que me brindaron su amor desde pequeño.

Para Silvia, por ir de la mano conmigo en esta nueva etapa de nuestras vidas.

Pérez Rivera Adrián Alfonso

INDICE

CAPÍTULO I. ANTECEDENTES	1
1.1 Introducción	1
1.2 Conceptos generales de los términos usados por el mercado inmobiliario en la administración habitacional.	6
1.3 Normatividad de la administración habitacional	11
1.4 Descripción básica de la administración habitacional	18
1.5 Acotamiento del problema a resolver	25
CAPÍTULO II. MARCO TEÓRICO.....	30
2.1 Características, ventajas y desventajas de las bases de datos relacionales y metodología iterativa – incremental, así como del manejador de base de datos MySQL.....	30
2.2 Características, ventajas y desventajas de la herramienta de programación Google Charts.	38
2.3 Características, ventajas y desventajas del lenguaje de programación PHP	46
2.4 Características, ventajas y desventajas del framework Polymer.	51
2.5 Características, ventajas y desventajas del lenguaje de programación HTML5.....	60
CAPÍTULO III. DEFINICIÓN Y ANÁLISIS DEL PROBLEMA	67
3.1 Análisis del problema	67
3.2 Requerimientos de sistema.	73
3.3 Análisis y posicionamiento de los lenguajes para back-end y front-end.	79
3.4 Aplicación y análisis modular con base en la metodología iterativa-incremental.....	95
3.5 Diseño modular de la interfaz usuario	102
CAPÍTULO IV. DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN	110
4.1 Modelado del sistema.	110
4.1.1 Diagrama de contexto del sistema (SIAH).	114
4.1.2 Diagrama entidad-relación del sistema(SIAH).	117
4.1.3 Diagrama de flujo de datos y procesos	120
4.1.4 Diccionario de datos.....	133
4.1.5 Normalización de la base de datos	145



4.2 Construcción y prueba de la base de datos.	156
4.3 Desarrollo del front-end del SIAH.....	166
4.4 Aplicación de pruebas y soporte.	173
4.5 Obtención de reportes y estadísticas	181
Conclusiones	184
Bibliografía	187



CAPÍTULO I. ANTECEDENTES

1.1 Introducción

El Sistema Integral de Administración Habitacional (SIAH) surge con el objetivo de cubrir la problemática que se presentan en una inmobiliaria, ya que hasta ahora los reportes administrativos se generaban muy rudimentariamente con excel y formatos físicos, lo cual generaba retrasos en tiempo, afectando el análisis de la productividad de las diferentes sucursales y empleados. El sistema SIAH facilitará la compra-venta-venta de inmuebles en el rubro del mercado inmobiliario, para llevar una mejor administración de las diferentes áreas funcionales de la inmobiliaria como lo son ventas, compras y recursos humanos. El sistema concentrará los datos generados de las diferentes sucursales distribuidas en el territorio nacional, evaluará y comparará la información recabada de la compra-venta de inmuebles y generará gráficos. Con lo cual se tomarán decisiones referentes a las tácticas de corrección o mejoramiento de la empresa.

A continuación se dará una breve introducción a la información que se podrá encontrar en cada capítulo de la presente tesis para dar un panorama general al lector.

- El capítulo I. referente a los antecedentes, nos ayudará a entender los diferentes aspectos a tomar en cuenta para introducirnos en el mundo de las inmobiliarias, de esta forma conseguiremos interactuar con nuestros clientes de una mejor manera para conocer con el mayor detalle posible sus necesidades y el alcance del problema a resolver; ello implicará definir el lenguaje y los términos usados en la administración habitacional, de esta forma lograremos también tener un panorama que nos ayude a entender los términos (asociados a nuestro proyecto) usados en el mercado inmobiliario y apoyar al cliente para definir de mejor manera su sistema. Por otra parte se describirán de manera general las leyes, normas y diferentes documentos relacionados con la administración inmobiliaria para poder desarrollar nuestro proyecto con las suficientes bases legales, de esta forma daremos certeza de que el uso de nuestro sistema contará con esta información para arrojar los datos y la información precisa

para la interpretación de los usuarios que requieran dicha información. Describiremos de manera general todo lo que conlleva la administración habitacional, de tal forma que esta descripción nos ayude a acotar el problema a resolver y delimitar de manera conjunta con nuestro cliente las características del SIAH.

- En el capítulo II. referente al marco teórico, se describirán las tecnologías y las herramientas utilizadas para la realización de nuestro proyecto, se dará con ayuda de apuntes y bibliografía adecuada, una correcta definición de los lenguajes de programación utilizados, las bases de datos y el manejador de base de datos que vamos a utilizar. Hemos decidido utilizar para nuestro SIAH, el gestor de base de datos MySQL, con lo cual requeriremos conocimientos del lenguaje SQL Structured Query Language (lenguaje de consulta estructurado), también utilizaremos los lenguajes de programación HTML, siglas de HyperText Markup Language (lenguaje de marcas de hipertexto) y PHP que significa Hypertext Pre-Processor (preprocesador de hipertexto), como se muestra en la Figura 1.1.1, tanto HTML como PHP son lenguajes de código abierto muy populares, adecuados para el desarrollo web de nuestro sistema y se complementan el uno al otro, veremos las principales características, la versión utilizada, ventajas y desventajas de cada uno de estos lenguajes y del gestor MySQL.



Figura 1.1.1 Los lenguajes de programación PHP y HTML junto con MySQL son una potente combinación de tecnologías de código abierto para desarrollar aplicaciones web.



Continuando con el capítulo II, se verán conceptos teóricos que aplicaremos a nuestro proyecto de manera que sea para nosotros un recordatorio de cómo utilizar de manera correcta el uso de estas tecnologías y herramientas, por ejemplo, se darán las características de la metodología iterativa-incremental en el desarrollo de software y se describirán los puntos que deben cumplir las bases de datos relacionales.

También en el sistema SIAH se hará uso del **framework** (marco de trabajo) Polymer y la herramienta de programación Google Charts, éste último porque nos ofrece una forma fácil y sencilla para visualizar datos y gráficos en una aplicación web. Desde gráficos con líneas simples hasta gráficos de árboles jerárquicos complejos, además de que nos ofrece una galería gráfica con un gran número de tipos de gráficos listos para su uso. En cuanto a Polymer se describirán las características que nos ofrece este tipo de tecnología actual en el mundo de la programación. Para entender completamente Polymer necesitaremos conocer a fondo sus tres capas conceptuales, que son los componentes web, la biblioteca de Polymer y los elementos. Se darán las características, ventajas y desventajas de estas dos tecnologías.

Los capítulos III y IV, que a continuación se resumen, se enfocarán en proporcionarnos información que nos ayude a entender el problema y definirlo de manera que podamos plasmar las necesidades que se presentan en una inmobiliaria, de esta forma analizar de manera correcta las transacciones que realizan cada una de sus oficinas dentro de la República Mexicana, de esta forma con los datos almacenados en nuestro sistema mejoraremos el desarrollo de la administración habitacional y el SIAH arrojará información importante para mejorar las diferentes áreas funcionales de la empresa.

- El capítulo III. definición y análisis del problema, estudiará a detalle el problema a resolver, las problemáticas que se presentan en la administración de la inmobiliaria, sus oficinas y los puntos que se quieren llegar a cubrir con la solución que se proponga. Con base al análisis del problema se hará una

propuesta de los requerimientos del sistema que se ilustrarán con diferentes tipos de diagramas y esquemas. Ya con el problema definido y la solución propuesta daremos las razones por la cual en el SIAH se utilizarán las tecnologías informáticas descritas en el capítulo II tanto para el **front-end** (lo que ve el usuario) como para el **back-end** (lo que no ve el usuario) de nuestro sistema. También se explicará la estructura del SIAH en cuanto a programación, es decir la forma en cómo será hecho. Se mostrarán los módulos funcionales que conformarán la aplicación y cómo es que interactuarán entre ellos.

También mostraremos cómo la metodología iterativa-incremental nos ayuda en la creación de software, cuidando que nuestro producto cumpla con características universales y garanticen la calidad del producto.

Para nuestro SIAH, la metodología iterativa-incremental proporcionará entre otras cosas las siguientes características:

- Soporte para determinar la efectividad de los procesos y de la calidad del producto.
 - Estudiar y después mejorar y ajustar el proceso para el ambiente en particular.
 - En el diseño modular de la interfaz de usuario veremos cómo será conformado cada módulo en el front-end para interactuar con los usuarios de nuestro SIAH.
- En el capítulo IV. modelado y construcción de la aplicación, se hará con ayuda de esquemas y diagramas, un modelo básico del ambiente del SIAH, el objetivo de esto será plasmar de manera adecuada en una solución los requerimientos que recolectamos al definir y analizar el problema, de tal forma que el análisis de un problema A nos lleve a una correcta solución B, como se muestra en la Figura 1.1.2.

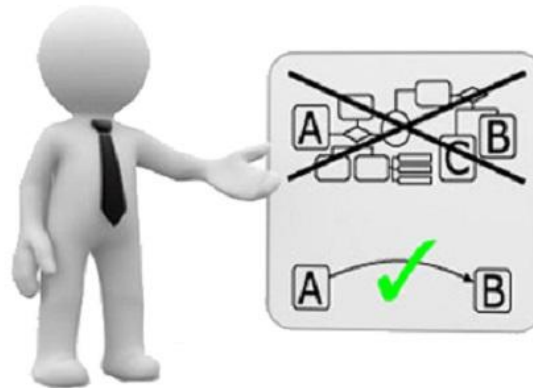


Figura 1.1.2 Análisis del problema y solución.

Haremos uso de diagramas que son muy utilizados en la Ingeniería del software, entre ellos el Diagrama de contexto de sistema el cual define los límites entre el sistema, o parte del sistema, y su ambiente, mostrando las entidades que interactúan con él. Otro diagrama del que haremos uso es del diagrama de flujo de datos y procesos para poder visualizar la forma en que el sistema funcionará, lo que el sistema va a lograr, y cómo el sistema se pondrá en práctica.

Además en cuanto a bases de datos se refiere, se desarrollará todo el proceso de estudio para obtener el correcto diagrama entidad relación, posteriormente normalizar nuestra base, para finalmente construirla y hacer las correspondientes pruebas. Teniendo la base ya funcionando de manera óptima, se construirá el front-end de nuestro SIAH con los lenguajes de programación elegidos y conforme al modelo establecido, conectaremos nuestra aplicación con la base de datos para que nuestro SIAH tome forma.

Finalmente se presentarán las pruebas que se realizarán sobre el sistema y los resultados que arrojarán, de igual manera se mostrarán los reportes que el sistema genera, se darán estadísticas y conclusiones.

1.2 Conceptos generales de los términos usados por el mercado inmobiliario en la administración habitacional.

Dentro del mercado inmobiliario existen diversos términos que serán definidos para determinar perfiles, transacciones y operaciones que se realizarán dentro de la plataforma.

El negocio de bienes raíces en México se define como la compra, venta o renta de inmuebles dentro del territorio nacional. Cualquier transacción realizada se rige por las leyes vigentes de los Estados Unidos Mexicanos.

Un bien inmueble comprende cualquier construcción, terreno o naturaleza de la cual se tenga propiedad legal y permiso para vender o rentar de acuerdo al uso de suelo definido por cada institución gubernamental o dependencia acreditada para emitirlo.

El suelo es el recurso más valioso y escaso de las ciudades, por lo que es indispensable gestionarlo en beneficio de la sociedad. La política de vivienda aplicada en años recientes, así como la provisión de servicios de forma reactiva para conectar zonas periféricas con la mancha urbana tuvieron importantes consecuencias en el mercado de suelo, por un lado, el suelo intraurbano incrementó su precio, haciendo prohibitivo para aquellos sectores de menos recursos, adquirir vivienda al interior de las ciudades; al mismo tiempo se promovió la especulación de suelo intraurbano debido a la carencia de instrumentos para aprovechar la plusvalía generada por las inversiones gubernamentales en favor de las ciudades.

La normativa únicamente regula usos del suelo, con obligaciones inadecuadas y limitadas en materia de concesiones, obras, servicios y contribuciones de los agentes inmobiliarios. Los gobiernos locales no cuentan con un instrumento que les permita asegurar la función social de la propiedad en una gestión urbana moderna, ni socializar los costos y beneficios del desarrollo urbano.

La Secretaría de Desarrollo Urbano y Vivienda (SEDUVI) es la dependencia del Gobierno del Distrito Federal encargada de diseñar, coordinar y aplicar la política urbana de la Ciudad de México. La planeación urbana de nuestra ciudad incluye la



orientación de su crecimiento, recuperación de espacios públicos, reactivación de zonas en desuso, protección y conservación del paisaje urbano, así como la promoción de construcción de vivienda social autosustentable, estas tareas buscan el desarrollo competitivo de la ciudad y fomentar proyectos que tengan un impacto positivo en la calidad de vida de los habitantes de la Ciudad de México.

A nivel nacional la Secretaría de Desarrollo Agrario, Territorial y Urbano (SEDATU) se encarga de promover el ordenamiento y la planeación territorial como articuladores del bienestar de las personas y el uso eficiente del suelo. Incentivar el crecimiento ordenado de los asentamientos humanos, los centros de población y las zonas metropolitanas. Consolidar ciudades compactas, productivas, competitivas, incluyentes y sustentables, que faciliten la movilidad y eleven la calidad de vida de sus habitantes. Así como fomentar el acceso a la vivienda mediante soluciones habitacionales bien ubicadas, dignas y de acuerdo a estándares de calidad internacional.

Para poder realizar transacciones inmobiliarias, la propiedad debe encontrarse dentro de suelo apto. Suelo apto se define como una extensión territorial con cualidades de ubicación, geología, topografía, disponibilidad de agua, costos de desarrollo y otros factores físicos que permiten que sea apropiado para el desarrollo urbano. Debe satisfacer las necesidades de las generaciones presentes sin comprometer las posibilidades de las del futuro para atender sus propias necesidades.

Los asentamientos humanos se definen como el establecimiento de un conglomerado demográfico, con el conjunto de sus sistemas de convivencia, en un área físicamente localizada, considerando dentro de la misma los elementos naturales y las obras materiales que lo integran.

Las zonas conurbadas son áreas de crecimiento urbano de dos o más centros de población que conforman un continuo edificado que normalmente rebasan los límites políticos de unos municipios.

Las zonas semiurbanas son las zonas que están en vía de ser una población urbana o trata de convertirse en urbana de acuerdo a si ya va a alcanzar el tamaño de la población para ser urbana, si se inclina más hacia la ciudad que hacia el campo.

La zonas metropolitanas son un grupo de dos o más municipios en los cuales se ubica una ciudad cuya área se extiende sobre los límites del municipio al cual pertenece originalmente; o un solo municipio dentro del cual se ubica totalmente una ciudad mayor de un millón de habitantes, o una ciudad con una población al menos 250 mil habitantes que forman una conurbación que forma con Estados Unidos.

El Banco Nacional de Obras y Servicios Públicos (BANOBRA) es una institución de banca de desarrollo que se tipifica como empresa pública con participación estatal mayoritaria, su objeto es financiar o refinanciar proyectos de inversión pública o privada en infraestructura y servicios públicos, así como coadyuvar al fortalecimiento institucional de los gobiernos Federal, Estatales y Municipales.

BANOBRA juega un papel decisivo para atenuar los efectos de la contracción económica al dirigir todos sus esfuerzos para fortalecer la inversión en:

- Infraestructura social básica, destinada a elevar las condiciones de vida de todas las familias y, con ello, sus capacidades y oportunidades de desarrollo personal, al poner a disposición de los gobiernos locales instrumentos de financiamiento para la atención de necesidades básicas de la población.
- Infraestructura para la competitividad y el desarrollo, que busca la creación de más y mejores empleos y un sostenido crecimiento económico al financiar proyectos con alta rentabilidad social.

La administración de plusvalías es el instrumento de gestión urbanística en la que la administración pública participa en la generación de valor sobre el suelo urbano susceptible de establecerse como bien de intercambio. Dado que el valor que este bien genera está construido socialmente por diferentes tipos de externalidades, parte del beneficio debería ser redituado a la sociedad que ha construido su valor, el mecanismo más importante para ello es el impuesto de plusvalía.



El impuesto de plusvalía aplica para propiedades con un valor superior a las 3.5 millones de pesos. El cálculo del monto se aplica sobre la ganancia que se obtiene al vender una propiedad por encima de la valuación inicial que tuvo al momento de su construcción.

En México existen instituciones de crédito creadas específicamente para transacciones inmobiliarias. Para los trabajadores del estado se creó el Fondo de la Vivienda del Instituto de Seguridad y Servicios Sociales de los trabajadores del Estado (FOVISSSTE) para ser encargado de administrar las aportaciones de las dependencias y entidades públicas, afiliadas al Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE), constituidas para la adquisición, reparación, ampliación o mejoramiento de viviendas. Los trabajadores de empresas privadas pueden hacer uso del Instituto del Fondo Nacional de la Vivienda para los Trabajadores (INFONAVIT) encargado de otorgar rendimientos eficientes a la subcuenta de vivienda de los trabajadores mediante esquemas que complementen la conformación del ahorro para el retiro de cada trabajador de acuerdo con sus necesidades, expectativas y decisiones.

Para la realización de transacciones inmobiliarias se cuenta con la figura de asesor inmobiliario. Un asesor inmobiliario se encarga de realizar todas las operaciones requeridas para la compra-venta o renta de una propiedad.

En México no existe una normativa específica que regule la profesión de asesor inmobiliario. La Asociación Mexicana de Profesionales Inmobiliarios A.C.(AMPI), es un organismo no gubernamental, rector de la actividad inmobiliaria en el país, que se ha construido con la visión de todos los asociados, consejeros y presidentes seccionales, consejeros y presidentes nacionales a través de su historia. Sin embargo no es requisito ser miembro de esta organización para ejercer la profesión de asesor inmobiliario.

Al realizar transacciones inmobiliarias, un asesor estipula el cobro de comisiones sobre la operación de venta o renta de una propiedad. Las comisiones se estipulan en

acuerdo mutuo y regularmente oscilan entre el 3% y el 10% del monto total de la transacción.

Los asesores inmobiliarios pueden pertenecer a una oficina concentradora de agentes o inmobiliarias, las cuales se encargan de estipular las comisiones que sus asesores deben cobrar por cada transacción.

Las operaciones inmobiliarias se clasifican de acuerdo al proceso de compra-venta en el que se encuentren; se denomina como operación abierta cuando la propiedad se incorpora al mercado y durante todo el tiempo que se mantiene en venta o renta. Las operaciones en trámite son aquellas que ya no se ofrecen a la venta o renta y que se encuentran en proceso de documentación y pago. Las operaciones cerradas son las propiedades de las cuales se han concluido todos los procesos de documentación y pagos.

Para poder realizar una venta o renta de una propiedad los asesores inmobiliarios deben realizar un proceso denominado “prospección”. La prospección es el método por el cual un asesor identifica clientes, propiedades y zonas que puedan ser adecuadas para poder realizar una venta inmobiliaria.

Las inmobiliarias pueden encargarse de la asesoría legal y comercial de las personas que desean realizar una transacción inmobiliaria, así como de alianzas estratégicas que faciliten la compra-venta de una propiedad. En México existen inmobiliarias que tienen plantillas laborales de 900 o más asesores.



1.3 Normatividad de la administración habitacional

Empezaremos diciendo que la normatividad es el conjunto de normas aplicables a una determinada materia o actividad; es la regla que se debe seguir o a que se deben ajustar las actividades, en este caso de la administración habitacional. Con esto entendemos la venta o alquiler de casas solas o en unidades habitacionales, departamentos en condominio, pent house, dúplex ó edificio.

Para la compraventa de un bien inmueble existen contratos para hacer la operación, en este se estipula quien es el vendedor (la parte vendedora) y quien el comprador (la parte compradora); en ella se estipula bajo protesta de decir la verdad, por parte de la parte vendedora, el lugar donde se encuentra ubicado el bien inmueble, el número de la escritura que ampara la posesión de la propiedad, se registra la fecha de la operación ante la fe del número de notario público que certifica la legalidad de la operación.

El inmueble cuenta con superficie, medidas y colindancias:

- Superficie
- Medidas y colindancias
- Al norte
- Al sur
- Al este
- Al oeste

Se nombra un representante legal debidamente capacitado y certificado ante el notario público correspondiente para la celebración del convenio, expuesto todo lo anterior se establecen las cláusulas correspondientes donde se establece el monto acordado por ambas partes y la manera de finiquitar dicha cantidad, el número de exhibiciones, la fecha del acuerdo, en el caso de que alguna de las partes no pudiese cumplir con lo establecido, podría llegar a sanciones por cada una de las partes; la situación final es que ambas partes queden de común acuerdo en base a lo acordado, dando por terminado lo establecido anteriormente. De tal manera que al momento de realizar la

compraventa del inmueble ambas partes deberán acordar el monto; establecer el representante legal debidamente capacitado y certificado así como en caso de no llegar a buenos términos la sanción correspondiente para cada una de las partes, lo anterior contando con la aprobación de las autoridades competentes basadas en el marco legal.

Después de llegar a finiquitar el proceso de la compra-venta del bien inmueble, dependiendo de la transacción que se haya llevado a cabo, se presentan los lineamientos acordes al bien inmueble. En caso de que la compraventa represente un condominio aplica reglas o lineamientos diferentes a los de una casa sola; un condominio es una construcción o terreno con construcciones acogido al régimen de copropiedad regulado por la ley 19.537.

Se distinguen dos tipos de condominio, los cuales no podrán estar emplazados en un mismo predio:

- Tipo A: las construcciones divididas en unidades, emplazadas en un terreno de dominio común.
- Tipo B: los predios con construcción o con proyectos de construcción aprobados en el interior de cuyos deslindes existan simultáneamente sitios que pertenezcan en dominio exclusivo a cada copropietario y terrenos de dominio común de todos ellos.

En esta ley se utiliza la expresión para referirse a los inmuebles que forman parte y sobre los cuales es posible construir dominio exclusivo.

Bienes de dominio común:

- Los que pertenezcan a todos los copropietarios por ser necesarios para su existencia, seguridad y conservación del condominio, tales como terrenos de dominio común, cimientos, fachadas, muros exteriores y soportantes estructura, cubiertas ascensores, montacargas, escaleras o rampas mecánicas
- A todos aquellos que permitan a todos y cada uno de los copropietarios el uso y goce de las unidades de su dominio exclusivo, tales como terrenos de dominio común diferentes a lo expuesto anteriormente.



- Los terrenos y espacios de dominio común colindantes con una unidad del condominio.
- Los bienes muebles o inmuebles destinados permanentemente al servicio, recreación y esparcimiento comunes de los copropietarios.
- Aquellos a los que se les otorgue tal carácter en el reglamento de copropiedad o que los copropietarios determinen.

Gastos comunes ordinarios, se tendrán por tales los siguientes

De administración: los correspondientes a remuneraciones del personal de servicio, conserje y administrador, y los de previsión que procedan

- De mantenimiento
- De reparación
- De uso o consumo

Gastos comunes extraordinarios: los gastos adicionales o diferentes a los gastos comunes ordinarios y las sumas destinadas a nuevas obras comunes.

Copropietarios hábiles: aquellos copropietarios que se encuentran al día en el pago de los gastos comunes.

Cada copropietario será dueño exclusivo de su unidad y comunero en los bienes de dominio común.

El derecho que corresponda a cada unidad sobre los bienes de dominio común se determinará en el reglamento de copropiedad, atendiéndose, para fijarlo al avalúo fiscal de la respectiva unidad.

Los avalúos fiscales de las diversas unidades de un condominio deberán determinarse separadamente.

Cada copropietario deberá contribuir tanto a los gastos comunes ordinarios como a los gastos comunes extraordinarios, en proporción al derecho que le corresponda en los

bienes de dominio común, salvo que el reglamento de copropiedad establezca otra forma de contribución.

Si el dominio de una unidad les corresponde en común a dos o más personas, cada una de ellas será solidariamente responsable del pago de la totalidad de los gastos comunes correspondientes a dicha unidad, sin perjuicio de su derecho a repetir lo pagado contra sus comuneros en la unidad, en la proporción que le corresponda.

Cada copropietario deberá pagar los gastos comunes con la periodicidad y en los plazos que establezca el reglamento de copropiedad, en el caso de incurrir en mora, la deuda devengará el interés máximo convencional para operaciones no reajustables o el inferior a este que establezca el reglamento de copropiedad.

En la administración de todo condominio deberá considerarse la formación de un fondo común de reserva para atender a reparaciones de los bienes de dominio común, a la certificación periódica de las instalaciones de gas, certificación de ascensores, montacargas y escaleras, o rampas mecánicas y sus instalaciones o a gastos comunes urgentes o imprevistos; el fondo se formará e incrementará con el porcentaje de recargo sobre los gastos comunes que fije la asamblea de copropietarios; con el producto de las multas e intereses que deban pagar los copropietarios y con los aportes por concepto de uso y goce exclusivos sobre bienes de dominio común.

Los ordenamientos jurídicos que regulan la edificación de la vivienda, desde el punto de vista técnico son los reglamentos de construcción elaborados por los gobiernos locales. En algunas localidades que no cuentan con un reglamento propio o que no abarcan los temas necesarios que regulen la construcción y ofrecen o adoptan otro reglamento de construcción, lo cual no siempre es adecuado a las condiciones específicas del lugar donde se adopta.

La existencia de una normatividad inadecuada, no actualizada o la falta de ella, tiene impactos negativos sobre la seguridad y la calidad de las edificaciones, la provisión de infraestructura básica y servicios, especialmente en la vivienda construida para personas de bajos recursos.



Un sistema de código modelo permite homologar y establecer estándares que facilitan establecer y medir la calidad y seguridad de las construcciones, así como un permanente sistema de actualización y funcionamiento. Que asegura tener una correcta aplicación y vigilancia de su cumplimiento. El código de edificación de vivienda se publicó en este contexto basándose en la experiencia de otros países con el objetivo de promover criterios y lineamientos generales para la edificación de vivienda en todo el país, respetando la autonomía estatal y municipal.

La normatividad de la administración habitacional depende del tipo de vivienda o construcción del que estemos hablando; al igual del tipo de si estamos hablando de una construcción aislada o se encuentra dentro de un conjunto habitacional, la normatividad es necesaria en todo momento, y posiblemente la tendencia a mejorar dichas normas para el bienestar de sus habitantes.

Se agrega como referencia de la ley de condóminos la siguiente dirección:

- <http://www.diputados.gob.mx/cronica57/contenido/cont6/leycondf.htm>

Y la del reglamento de construcción:

- http://www.poderjudicialdf.gob.mx/work/models/PJDF/Transparencia/IPO/Art14/Fr01/02Reglamentos/ReglConstruccionesDF_20040129.pdf

Además de un contrato general de compra-venta como se muestra en la Figura 1.3.1 y Figura 1.3.2

CONTRATO DE PROMESA DE COMPRAVENTA
CONTRATO DE PROMESA DE COMPRAVENTA. que celebran: de una parte la empresa _____, quienes en lo sucesivo se les denominará LOS PROMITENTES VENEDORES; y por la otra parte el señor _____ y su cónyuge señora _____ a quienes en adelante se les designará LOS PROMITENTES COMPRADORES."
Contrato que los otorgantes, por sus propios derechos, formalizan ante la presencia de los testigos señores licenciado _____ y _____, de conformidad con las siguientes declaraciones:



Sistema Integral De Administración Habitacional (SIAH)

DECLARACIONES

Manifiestan los promitentes vendedores, lo siguiente:

I. Que son dueños en legítima propiedad, posesión y pleno dominio del siguiente inmueble:

Casa habitación/Departamento marcada(o) con el número _____ de la calle _____, en el Fraccionamiento _____ de esta ciudad, construida en el lote de terreno marcado con el número _____, de la manzana número _____, cuyo predio tiene una superficie de _____ m² (_____ metros cuadrados) y las siguientes medidas y colindancias: al Norte _____ m (_____ metros), con frente a la calle _____; al Sur _____ m (_____ metros), con el lote _____; al Oriente _____ m (_____ metros), con el lote _____; y al Poniente _____ m (_____ metros) con la calle _____. La manzana de referencia está circundada por las calles de: _____ al Norte, _____ al Sur, _____ al Oriente y _____ al Poniente.

II. Que el inmueble anteriormente descrito lo adquirieron en cuanto al terreno, por compra efectuada a la _____, según consta en la escritura pública número _____, de fecha _____ de _____ de _____, pasada ante la fe del Sr. Lic. _____, titular de la Notaría Pública número _____, con ejercicio en este municipio, cuyo primer testimonio obra inscrito en el Registro Público de la Propiedad y del Comercio de esta ciudad, bajo el número _____, volumen _____, libro _____, sección _____, propiedad con fecha _____ de _____; y por cuanto a la Casa habitación/Departamento se refiere, ésta fue construida y/o adquirida dentro de todas las leyes y estatutos definidos en el Código Civil vigente del Estado de _____.

III. Que el inmueble de que se trata se encuentra libre de todo gravamen y responsabilidad y al corriente en el pago del impuesto predial que le corresponde, así como de los servicios de agua, drenaje, electricidad y gas con que cuenta.

IV. Que dicho inmueble actualmente se encuentra ocupado por los promitentes vendedores. Expresan conjuntamente los promitentes vendedores y los promitentes compradores, lo siguiente:

V. Que han concertado el otorgar el presente contrato de promesa de compraventa, en relación con el inmueble a que se refieren las declaraciones que anteceden y que con fundamento en lo anterior, sujetan el presente contrato al tenor de las siguientes:

CLAUSULAS

Primera: Los promitentes vendedores se comprometen a vender, en favor de los promitentes compradores, quienes por su parte, se comprometen a comprar el inmueble que ha quedado descrito e identificado en la declaración primera de este instrumento, con todos sus usos, costumbres, servidumbres, construcciones y demás derechos que concede la ley a todo propietario, con las medidas y superficie indicadas o con las que resultaren dentro de las colindancias expresadas, las cuales se tienen aquí por reproducidas, como si se insertasen a la letra, para todos los efectos legales a que hubiere lugar, de conformidad con lo dispuesto en el artículo _____



(_____) del Código Civil vigente en el estado de _____.
Segunda: Ambas partes fijan como precio de la operación la cantidad de \$_____. (_____. pesos 00/100 moneda nacional) importe que los promitentes compradores se obligan a pagar en favor de los promitentes vendedores, al momento en que se formalice el contrato definitivo.
Tercera: La posesión material del inmueble prometido en venta, se concederá a los promitentes compradores al momento en que se formalice en contrato definitivo.
Cuarta: Las partes contratantes convienen en que la fecha para la celebración del contrato definitivo será el día _____. de _____. de _____.
Quinta: Los impuestos, derechos, gastos y honorarios que se causen con motivo del otorgamiento del presente contrato, así como los que cause el contrato definitivo, serán pagados por los promitentes compradores, a excepción del impuesto sobre la renta por enajenación de bienes, el cual será pagado por los promitentes vendedores cuando se formalice el referido contrato definitivo.
Sexta: Los contratantes hacen constar su conformidad con lo estipulado en el presente contrato, por haberse redactado en los términos entre ellos convenidos; sometiéndose expresamente a la jurisdicción de los tribunales judiciales de esta ciudad, para todo lo relativo a la interpretación, cumplimiento y ejecución de lo convenido y, al efecto, renuncian al fuero que por razón de sus domicilios presentes o futuros pudieran favorecerles.
Para constancia de lo estipulado y en cumplimiento de lo dispuesto por el artículo _____ (_____) del Código Civil vigente en el estado de __, se firma el presente contrato, ante la presencia de los testigos al principio mencionados, quienes declaran conocer personalmente a los otorgantes, a quienes consideran con capacidad legal para contratar y obligarse válidamente, sin que les conste nada en contrario y manifestando los otorgantes que están bien enterados y conformes con el alcance y efectos legales del presente instrumento. En la ciudad de _____, del estado de __, a los _____. días del mes de _____. de _____. Conste.
Lugar, Fecha y firma

Figura 1.3.1 Contrato de compra venta

1.4 Descripción básica de la administración habitacional

Las casas, condominios, conjuntos residenciales, edificios de oficinas o de comercios generan a sus propietarios y habitantes una necesidad de: seguridad, servicios de infraestructura, calles limpias y bien iluminadas, jardines o áreas verdes y, en general, una buena apariencia, como se muestra en la Figura 1.4.1. La administración de condominios o inmuebles se encarga no solo de los recursos financieros (ingresos y egresos) sino también de los recursos inmobiliarios y materiales que el condominio tenga. El desarrollo integral de un inmueble depende de una administración responsable y sustentable, estableciendo un equilibrio entre el ejercicio de los recursos y la administración del factor humano. Una administración deficiente genera conflictos, si se deja deteriorar el inmueble por falta de planes adecuados de mantenimiento, a la larga las reparaciones y desperfectos fortuitos saldrán más caros, lo que también conduce a una mala convivencia vecinal.



Figura 1.4.1 Necesidades en casa habitación



Para la operación y mantenimiento de cualquier zona habitacional, se hace indispensable el trabajo de la comunidad obteniéndose muchas ventajas como son la integración de los vecinos y evitar gastos. Sin embargo, en ocasiones se hace imprescindible contratar a prestadores de servicio quienes son los encargados de realizar trabajos de mantenimiento y mejoramiento de las instalaciones físicas como: Infraestructura eléctrica, Instalaciones hidráulicas, Instalaciones de gas, albañilería y pintura, sistemas de seguridad, entre otros. No obstante que la contratación, supervisión de los trabajos y pago de estas personas es una responsabilidad del administrador de la zona habitacional, en caso de no existir esta figura, el asesor en organización vecinal, deberá orientar a los vecinos sobre la realización de estas acciones. Es importante considerar que deben existir lineamientos básicos a seguir para la contratación de prestadores de estos servicios, sean personas físicas o morales; es recomendable que se establezcan requisitos a cubrir por el prestador de servicios e incluso que se cuente con un registro de los mismos. Para contratar a los prestadores de servicios, habrá que considerar el monto destinado para el trabajo, el tiempo estimado para realizarlo, los materiales requeridos, verificar si es necesario un contrato o convenio, verificar si es conveniente dar anticipos o no, buscar al menos tres cotizaciones de diferentes proveedores, buscar asesoría especializada si se requiere, entre otros aspectos importantes.

Los ingresos de un condominio, son los recursos financieros que se reciben por el cobro de cuotas internas ordinarias y extraordinarias y cuotas externas extraordinarias.

- Las cuotas internas ordinarias de un condominio son aquellas que pagan los residentes del mismo, por ejemplo una cuota regular de mantenimiento mensual: “Cuota de mantenimiento de marzo de 2015”.
- Las cuotas internas extraordinarias son aquellas que pagan los residentes del condominio que no son regulares, normalmente las cuotas extraordinarias son ocupadas por los administradores para allegarse de recursos para imprevistos.
- Las cuotas externas extraordinarias son aquellas que pagan los que no son residentes del condominio.

Mientras que los egresos, son los gastos ordinarios y extraordinarios de un condominio. Por ejemplo, los gastos que cubren todos los habitantes por mantenimiento de las áreas de uso común o algún desperfecto que resulte necesario repararse (suministro de electricidad, agua, drenaje).

Necesidades generales de los condominios, edificios y residenciales.

- Administración y gestión condominal en general
- Seguridad (vigilancia y/o conserjería)
- Mantenimiento y reparación de áreas comunes.

Necesidades particulares de habitantes y usuarios:

- Reparación (carpintería, electricidad, plomería, etc.)
- Servicio de limpieza (personal calificado)
- Chofer y mensajero
- Lavado de autos
- Realización de pagos, servicios y gestión de trámites de todo tipo.

Sistema de Acceso a la Información

Lo más importante es la confianza por eso se debe ofrecer acceso a la información en todo momento:

- Estados de cuenta (saldos, egresos e ingresos)
- Reservación de áreas comunes (salones de eventos, ludotecas, canchas, etc.)
- Impresión de avisos y comprobantes de pago
- Aviso oportuno
- Agenda
- Buzón de quejas y sugerencias.



¿Qué es un administrador de condominio?

La persona física o moral que designa la asamblea general de condóminos en los términos de la Ley de Propiedad en Condominio de Inmuebles para el Distrito Federal y el reglamento interno del condominio. El administrador del condominio es el encargado de dar certeza en el manejo eficaz del dinero y recursos del condominio. En algunos inmuebles o condominios el administrador del condominio es fiscalizado por el comité de vigilancia, el cual está integrado por residentes del comité.

La administración de condominios o inmuebles es la que se encarga de la planificación, organización, dirección y control de los recursos de los condominios o inmuebles.

Su designación o nombramiento se da por el voto de la mayoría que represente el cincuenta y uno por ciento del valor del condominio, y puede ser removido cuando la misma asamblea general de condóminos lo estime conveniente, el administrador debe contar con experiencia en la administración de inmuebles y con amplia solvencia moral como se muestra en la Figura 1.4.2. La asamblea general de condóminos fija a mayoría de votos, y cada vez que lo juzgue conveniente, las remuneraciones ordinarias y extraordinarias del administrador, así como su forma de pago. Cuando la asamblea general de condóminos decida contratar servicios profesionales externos, el comité de vigilancia debe aplicar el contrato correspondiente conforme a la ley aplicable.



Imagen 1.4.2 Designación de administrador de condominios

Las medidas que adopte y las disposiciones que emita el administrador dentro de sus funciones y con base en la Ley de Propiedad en Condominio de Inmuebles para el Distrito Federal y el reglamento interno del condominio, son obligatorias para todos los condóminos.

Cuando la asamblea general de condóminos designa una nueva administración, la saliente debe entregar, en un término que no exceda de siete días naturales al día de la designación, todos los documentos incluyendo los estados de cuenta, valores, archivos, muebles, equipos y demás bienes que tuvo bajo su resguardo y responsabilidad. La entrega sólo puede posponerse por resolución judicial. Debiéndose levantar, en todos los casos, acta circunstanciada de la misma.



Cuotas de administración y mantenimiento

De acuerdo a la Ley de Propiedad en Condominio de Inmuebles para el Distrito Federal, las cuotas de administración y mantenimiento son las aportaciones a cargo de los condóminos que se establecen para:

- Constituir el fondo de mantenimiento y administración destinado a cubrir el gasto corriente que se genere en la administración, operación y servicios no individualizados de las áreas comunes del condominio. El importe de la cuota se establece distribuyendo los gastos entre el número de unidades de propiedad exclusiva, de acuerdo a su porcentaje de indiviso, y deben cubrirse conforme a la periodicidad y procedimiento estipulados en el reglamento interno del condominio.
- Constituir el fondo de reserva destinado a cubrir los gastos de adquisición de implementos y maquinarias con que debe contar el condominio, obras, mantenimiento y reparaciones mayores. El importe de esta cuota también se establece en proporción al valor estipulado en la escritura constitutiva del régimen de propiedad en condominio para cada unidad de propiedad exclusiva. El monto general y su periodicidad se establecen en el reglamento interno del condominio.

Efectuar gastos extraordinarios, cuando:

- El fondo de administración y mantenimiento no sea suficiente para cubrir un gasto corriente extraordinario. El importe de la cuota se establece, distribuyendo en partes iguales el importe del gasto entre todas las unidades de propiedad exclusiva.
- El fondo de reserva no sea suficiente para cubrir la compra de algún implemento, realización de obras, mantenimiento y reparaciones mayores. El importe de la cuota se distribuirá conforme a lo establecido para el fondo de reserva, como se muestra en la Figura 1.4.3 sobre el uso de las cuotas.



Figura 1.4.3 Uso de las cuotas

¿Es obligatorio el pago de las cuotas de mantenimiento y administración?

Sí al acordar la asamblea de condóminos el importe de las cuotas, todos los condóminos están obligados a pagar la que les corresponda de acuerdo al porcentaje de indiviso que represente su unidad de propiedad exclusiva.

Toda vez que las cuotas de mantenimiento y administración son utilizadas para efectuar pagos por: honorarios, sueldos y salarios de personal que trabaja en el condominio; trabajos necesarios para mantener el condominio en buen estado, que incluyen la operación cotidiana, reparación y conservación de las instalaciones y servicios generales; consumos de agua y energía eléctrica de áreas comunes, al no



cubrirse en su totalidad y oportunamente las cuotas se pone en riesgo el funcionamiento seguro y eficaz del condominio.

Además, al no constituirse el fondo de reserva se carece de los recursos económicos necesarios para solventar reparaciones mayores, comprar equipos para reponer los existentes o invertir en nuevas instalaciones o equipos.

1.5 Acotamiento del problema a resolver

La tecnología ha tenido grandes avances es por ello que es importante tener estas ventajas de nuestro lado y obtener el mayor provecho posible en nuestra vida cotidiana. El desarrollo de una aplicación para el sector inmobiliario nos facilitará íntegramente las tareas de administración, control y estado sobre las transacciones de compra-venta de casas habitación.

Este sistema se desarrollará como una propuesta tecnológica destinada a resolver una problemática específica en las empresas del sector inmobiliario dedicadas a la renta y compra-venta de bienes inmuebles, dicha propuesta será mostrada a través del presente proyecto, ofreciendo una visión detallada de cómo se construirá el flujo de información dentro de las diferentes áreas que constituyen dichas oficinas y la manera en cómo interactúan entre sí para lograr sus objetivos.

Con un auge cada día mayor y exponencial de desarrollos habitacionales, de oficina, industria, almacenaje, etc., y con ellos nuevas compañías encargadas de administrar, mejorar la eficacia, productividad, rentabilidad de inmuebles que se encuentran en compra-venta, surge la necesidad de diseñar y desarrollar un sistema que facilite el control y análisis del número de transacciones de propiedades que permita la toma de decisiones a corto, mediano y largo plazo por parte de los elementos que dirigen los esfuerzos, objetivos y misión de la empresa.

Las operaciones realizadas en las diferentes áreas de la inmobiliaria nos generan datos relevantes como lo son:

- El desempeño de cada una de ellas.
- El sector de mayores y menores utilidades para la empresa.
- El rubro donde no se necesita realizar un gran esfuerzo debido a la obtención de utilidades muy bajas.
- Visualizar nuevas estrategias para balancear las pérdidas.

Lo anterior nos sirve para contar con los instrumentos necesarios que permitan una toma de decisiones en condiciones de certidumbre y mejora de la misma al invitar a las oficinas con mayor utilidad a continuar con el mismo desempeño y a elevar el mismo en las oficinas donde este se encuentre bajo o permitir reorientar el camino que se había tomado hasta ese momento sentando las bases de elementos de control y evaluación, identificando situaciones que pudiesen alterar el resultado u obtención de mayores utilidades. Su uso será interno y se podrá obtener el grado de productividad de cada una de las oficinas así como del rubro de una manera visual (gráficas) para un mejor entendimiento e interpretación.

El sistema concentrará los datos generados de las diferentes sucursales distribuidas en el territorio nacional, evaluando y comparando la información recabada de la compra-venta de inmuebles de manera mensual paralelamente generando gráficas de la eficiencia de las diferentes oficinas, en cuanto a los inmuebles asignados a cada una de ellas y con base a estas se tomarán decisiones referentes a las tácticas de corrección y/o mejoramiento de la empresa, para incrementar la productividad en cada una de las oficinas según su desempeño.

Se busca que el sistema tenga una experiencia de usuario adecuada para el manejo de las transacciones, permitiendo una interactividad intuitiva disminuyendo el margen de posibles errores humanos en el manejo de cifras e información. Se pretende que el sistema genere análisis y reportes a través de gráficas amigables para el usuario, sin necesidad de tener conocimiento previo del mercado inmobiliario. Estos reportes permitirán observar el avance o retroceso de las ventas así como la productividad facilitando el seguimiento continuo de las transacciones.



Contará con un histórico de las transacciones efectuadas de los inmuebles de cada oficina por separado, además del análisis del comportamiento del conjunto de estas con lo que obtendremos comparativos del desempeño de cada oficina y elementos involucrados en las mismas.

El desarrollo de la aplicación estará basado en software libre, disminuyendo los costos inherentes a licencias, permisos y derechos y así no depender de terceros o de software propietario. Se elegirán las mejores herramientas de desarrollo libres en el mercado actual, haciendo uso de programación web, garantizando el acceso desde cualquier punto por los usuarios autorizados para ello, además de permitir la escalabilidad del sistema y la actualización de manera modular.

Como apoyo para realizar la administración y control de las transacciones de los inmuebles en cada una de las oficinas, se desarrollara un front-end amigable con ayuda de la herramienta de programación Google Charts para los gráficos, del lenguaje de programación PHP y HTML5 así como del uso del framework Polymer para definir los estilos.

En línea con un aprovechamiento y maximización de recursos usaremos un manejador de base de datos que no tenga costo, en este caso MySQL, para guardar toda la información generada por cada una de las oficinas para con ello generar las gráficas y las comparativas, estos datos se guardaran a partir de la realización de alguna transacción de renta o venta.

La ejecución del sistema tendrá que ser automática a partir de que se capture la transacción y que requiera generar algún gráfico.

Con ello esperamos obtener los siguientes resultados:

- La administración de las rentas y ventas de la inmobiliaria, teniendo un histórico de cada oficina.
- La generación y jerarquización de diferentes roles de usuario para tener una mejor administración y control de la actualización de las transacciones que realice cada oficina.

- La generación de gráficos y comparativas para el análisis y control de la toma de decisiones a mediano y largo plazo.
- La generación de reportes que permitan observar los avances o retrocesos en la inmobiliaria.
- Facilitar el seguimiento continuo de las transacciones.
- Optimizar la administración y manejo de la inmobiliaria así como de sus oficinas.
- La generación de reportes por oficina de los tipos de inmuebles que posee cada una de ellas.
- La clasificación de los inmuebles de acuerdo a su uso.
- Elementos de soporte y guía para la adecuada toma de decisiones sobre la productividad de cada elemento que integre el sistema.

De manera global el problema se visualiza con los siguientes aspectos como se muestra en la Figura 1.5.1

- Catalogar al inmueble (por su estructura, por su uso).
- Inmobiliaria.
- Oficinas.
- Transacciones (renta, compra y venta).
- Cliente.
- Desempeño.
- Utilidades.
- Toma de decisiones.
- Reportes y gráficos.

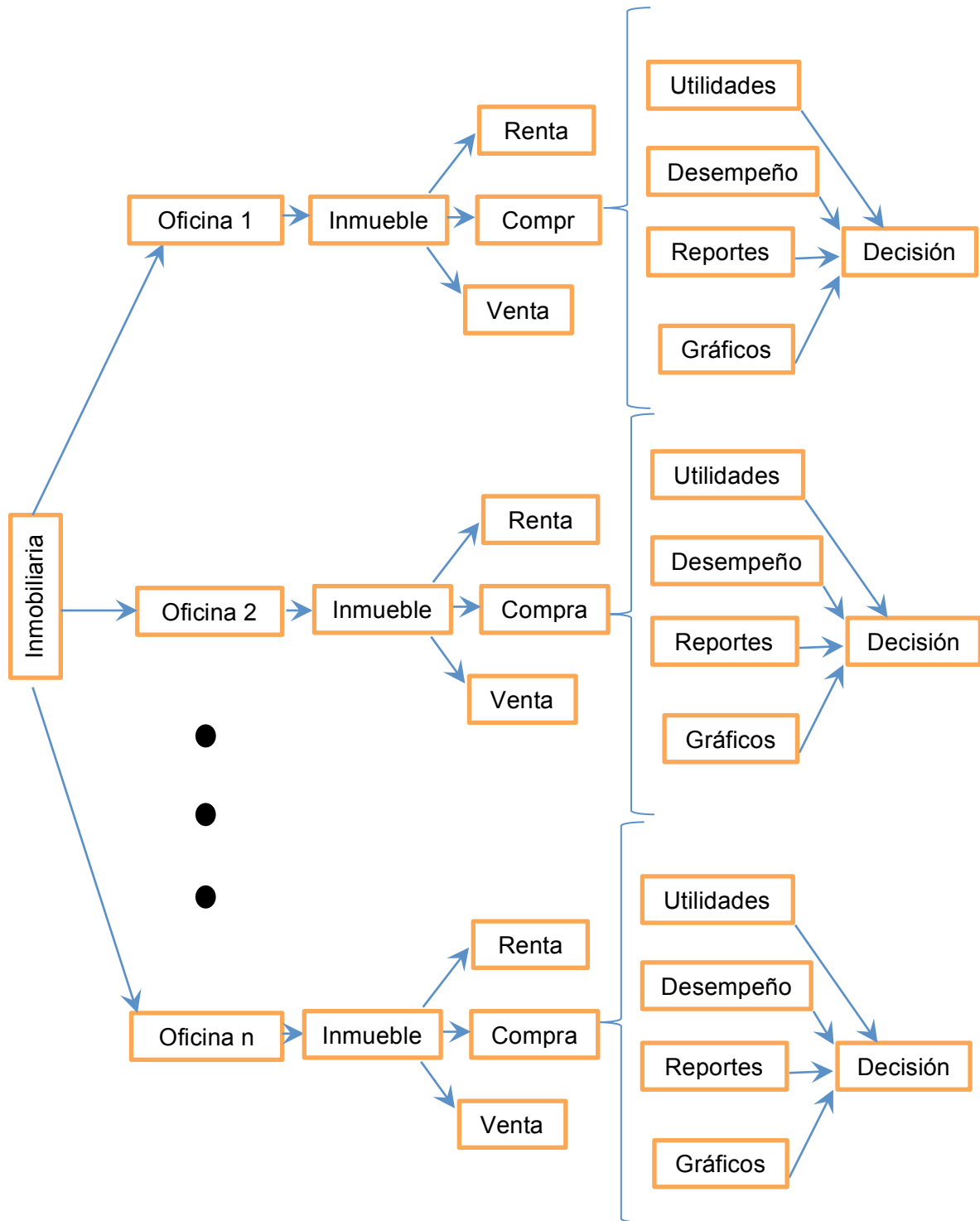


Figura 1.5.1 Diagrama de bloque

CAPÍTULO II. MARCO TEÓRICO

2.1 Características, ventajas y desventajas de las bases de datos relacionales y metodología iterativa – incremental, así como del manejador de base de datos MySQL.

Bases de datos relacionales

Una base de datos relacional, es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas ya que permite establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene el nombre: "modelo relacional".

Características

- Se compone de varias tablas o entidades.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las llaves primarias y foráneas.
- Las llaves primarias son la llave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- Las llaves foráneas se colocan en la tabla hija, contienen el mismo valor que la llave primaria del registro padre, por medio de éstas se relacionan.
- Una tabla bidimensional está constituida por líneas (tuplas) y columnas (atributos).
- Cada columna almacena información sobre una propiedad determinada de la tabla (se le llama también atributo).
- Cada fila posee una ocurrencia o ejemplar de la instancia o relación representada por la tabla (a las filas se las llama también registros o tuplas).



- Las entradas en las tablas tienen un solo valor (son atómicos); no se admiten valores múltiples, por lo tanto la intersección de un renglón con una columna tiene un solo valor y nunca un conjunto de valores.
- El orden de las columnas no es de importancia para la tabla.
- Debe haber consistencia en la información, es decir, no debe haber incongruencias en los datos que se almacenan. Frecuentemente los problemas de consistencia de datos se deben a la redundancia de éstos.
- La información debe ser no redundante, es decir, se debe de eliminar al máximo la existencia de duplicidad de los datos.

Ventajas

- Provee herramientas que garantizan evitar la duplicidad de registros (como llaves primarias).
- Garantiza la integridad referencial, al eliminar un registro elimina todos los registros relacionados dependientes (como llaves foráneas).
- Favorece la normalización por ser más comprensible y aplicable.
- Compatibilidad y estandarización.
- Conectividad con los lenguajes de programación estándar.
- Integridad de datos mediante las reglas o restricciones que se establezcan.
- Se puede agregar posteriormente datos al sistema sin tener que reescribir lo que ya se tiene.

Desventajas

- No se manipulan de forma manejable los bloques de texto como tipo de dato.
- Presentan deficiencias con datos gráficos, multimedia, **CAD** (Computer Aided Design - Diseño Asistido por Computadora) y sistemas de información geográfica.
- Las bases de datos orientadas a objetos (BDOO) se propusieron con el objetivo de satisfacer las necesidades de las aplicaciones anteriores y así, complementar pero no sustituir a las bases de datos relacionales.

- El acceso a los datos puede llegar a ser lento, dependiendo de la robustez de la información.

Metodología iterativa – incremental

Este modelo de desarrollo, no es más que un conjunto de tareas agrupadas en pequeñas etapas repetitivas (iteraciones),¹ es uno de los más utilizados en los últimos tiempos ya que, como se relaciona con novedosas estrategias de desarrollo de software y una programación extrema, es empleado en metodologías diversas.

El modelo consta de diversas etapas de desarrollo en cada incremento, las cuales inician con el análisis y finalizan con la instauración y aprobación del sistema².

Modelo iterativo

Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se entrega una versión mejorada como se muestra en la Figura 2.1.1.

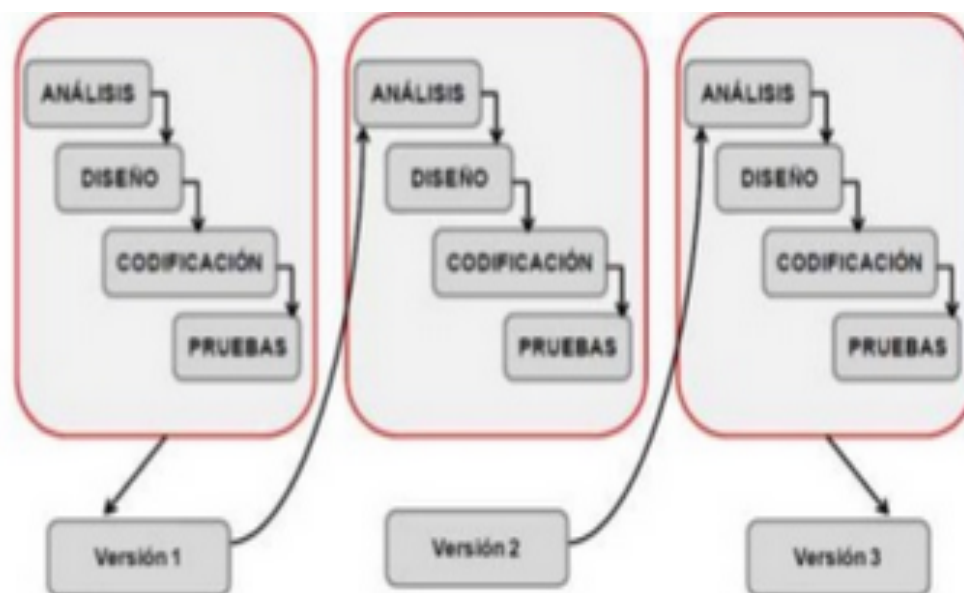


Figura 2.1.1 Modelo iterativo

¹ Proceso de Desarrollo Iterativo <http://fernandosoriano.com.ar/?p=13>

² . Desarrollo de software. Ciclo de vida iterativo incremental
<https://jummp.wordpress.com/2011/03/31/desarrollo-de-software-ciclo-de-vida-iterativo-incremental/>

² . Desarrollo de software. Ciclo de vida iterativo incremental
<https://jummp.wordpress.com/2011/03/31/desarrollo-de-software-ciclo-de-vida-iterativo-incremental/>

Modelo incremental

Combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos como se muestra en la Figura 2.1.2.

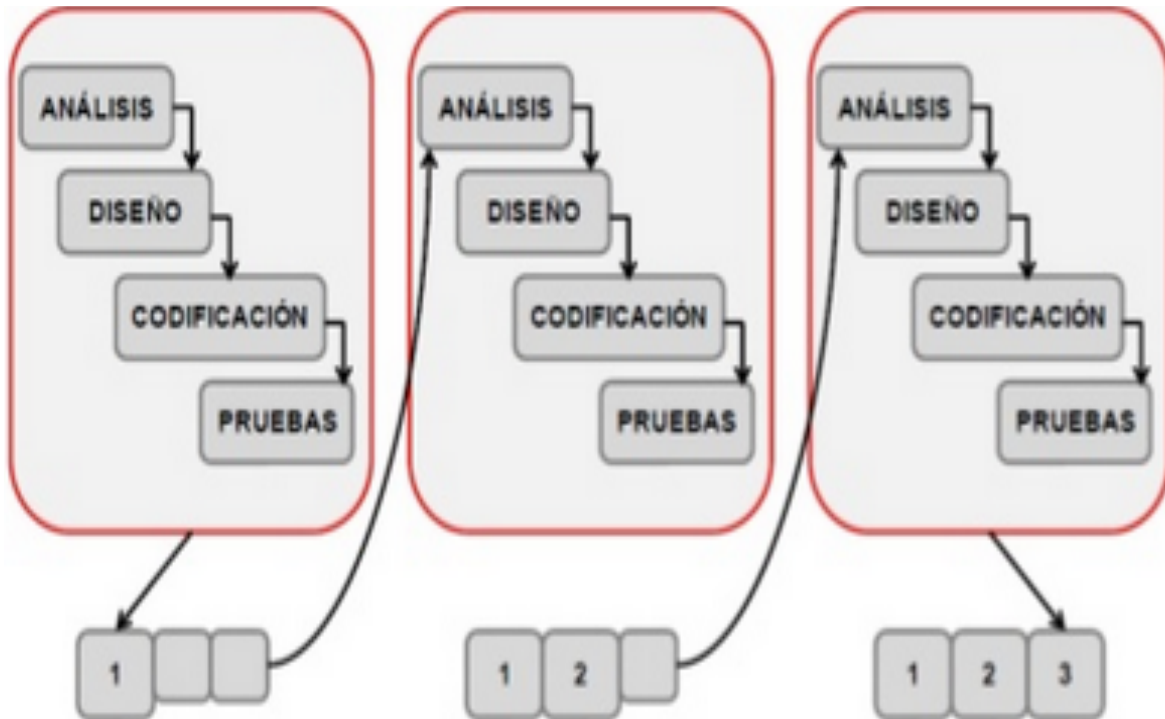


Figura 2.1.2 Modelo incremental

El producto se desarrolla por incrementos en el que cada iteración obtiene una versión funcional del producto, de esta forma el sistema se desarrolla poco a poco y obtiene un feedback continuo por parte del usuario.

Características

- Permite separar la complejidad del proyecto, gracias a su desarrollo por parte de cada iteración o bloque.
- El producto es consistente y puntual en el desarrollo.
- Se obtiene un aprendizaje en cada iteración que es aplicado en el desarrollo del producto y aumenta las experiencias para próximos proyectos.
- Visión del avance.
- Resolución de problemas en tiempos cortos.

Ventajas

- Los usuarios no tienen que esperar hasta que se entregue el sistema completo para hacer uso de él. El primer incremento cumple los requerimientos más importantes de tal forma que pueden utilizar el software al instante.
- Los usuarios pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.
- Existen muy pocas probabilidades de riesgo en el sistema. Aunque se pueden encontrar problemas en algunos incrementos, lo normal es que el sistema se entregue sin inconvenientes al usuario.
- Ya que los sistemas de más alta prioridad se entregan primero, y los incrementos posteriores se integran entre ellos, es muy poco probable que los sistemas más importantes sean a los que se les hagan más pruebas. Esto quiere decir que es menos probable que los usuarios encuentren fallas de funcionamiento del software en las partes más importantes del sistema.
- Se da la retroalimentación muy temprano a los usuarios.
- Los productos desarrollados con este modelo tienen una menor probabilidad de fallar.

Desventajas

- No garantiza por sí solo el éxito.
- Iteraciones costosas.
- Congelamiento de requerimientos.
- La entrega temprana de los proyectos produce la creación de sistemas demasiados simples que a veces se ven un poco monótonos a los ojos del personal que lo recibe.
- La mayoría de los incrementos se harán en base de las necesidades de los usuarios.
- Los incrementos no deben constar de muchas líneas de código ya que la idea de los incrementos es agregar accesorios al programa principal (o funcional), para



que este tenga una y mil formas de desenvolverse en su tarea; llenar los incrementos de muchas líneas de código provocaría que se perdiera la objetividad o base de lo que se trata el desarrollo incremental.

- Requiere de un cliente involucrado durante todo el curso del proyecto. Hay clientes que simplemente no estarán dispuestos a invertir el tiempo necesario.
- El trato con el cliente debe basarse en principios éticos y colaboración mutua, más que trabajar cada parte independientemente, defendiendo sólo su propio beneficio.
- La entrega de un programa que es parcial pero funcional puede hacer vulnerable al programa debido a la falta de robustez en su sistema, provocando que agentes ajenos puedan interferir con el correcto funcionamiento del programa en sí.
- Infunde responsabilidad en el equipo de desarrollo al trabajar directamente con el cliente, requiriendo de profesionales sobre el promedio.
- Sufre fuertes penalizaciones en proyectos en los cuales los requerimientos están previamente definidos, o para proyectos "todo/nada" en los cuales se requiere que se completen en un 100% el producto para ser implementado (por ejemplo, licitaciones) otro punto muy importante es asegurarnos de que el trabajo se pueda cumplir tomando en cuenta los costos que podamos usar en nuestros propios recursos.

Manejador de base de datos: MySQL

MySQL es un sistema de administración de bases de datos relacionales de gran robustez, con un excelente desempeño, diseñado para soportar sistemas de producción de misión crítica con altas cargas de procesamiento. Es además, el manejador de base de datos más popular entre los producidos bajo los términos del open source (software libre), aunque existe una versión comercial distribuida y soportada por la compañía MySQL AB.

Características

- Escrito en C y en C++.
- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.
- Proporciona sistemas de almacenamientos transaccionales y no transaccionales.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y el acceso a la información es ágil.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas.
- Los clientes pueden conectar con el servidor MySQL usando sockets.
- La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity).
- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.

Ventajas

- Velocidad y robustez.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.



- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que se encuentra.
- Conectividad y seguridad.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: uno de estructura, uno de datos, y uno de índice; y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de passwords (contraseñas) y gestión de usuarios, con un buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

Desventajas

- Lento con grandes bases de datos
- Un gran porcentaje de utilidades de MySQL no están documentadas.
- No es intuitivo como otros programas (SQL Server, ACCESS, etc.).
- Imposibilidad de hacer subconsultas o consultas anidadas.
- En versiones actuales, aparentemente tiene integridad referencial (aún no se explota realmente).
- Los índices son una desventaja en aquellas tablas en las que se utiliza frecuentemente operaciones de escritura (insert, delete, update), esto es porque los índices se actualizan cada vez que se modifica una columna.
- Los índices también suponen una desventaja en tablas demasiado pequeñas puesto que no necesitaremos ganar tiempo en las consultas.
- Tampoco son muy aconsejables cuando pretendemos que la tabla sobre la que se aplica devuelva una gran cantidad de datos en cada consulta.
- Por último hay que tener en cuenta que ocupan espacio y en determinadas ocasiones incluso más espacio que los propios datos.

2.2 Características, ventajas y desventajas de la herramienta de programación Google Charts.

Google Charts ofrece una forma perfecta para visualizar datos en una página web. Desde los gráficos de líneas simples a los mapas de árboles jerárquicos complejos, la galería gráfica ofrece un gran número de tipos de gráficos listos para su uso. Es una aplicación de Google para realizar estadísticas web, de fácil uso para desarrolladores web.

La forma más común para usar Google Charts es con JavaScript, simplemente se incrusta en la página web como una biblioteca de JavaScript. Se cargan algunas bibliotecas de Google Charts, se enlistan los datos que se van a graficar, se seleccionan las opciones para personalizar la tabla de gráficos, y, finalmente, se crea un objeto de gráfico con un “Id” que se elija. Posteriormente, en la página web, se crea un “<div> “ con ese Id para mostrar la tabla de Google.

Para entender mejor cómo funciona Google Charts también tenemos que estudiar un poco del lenguaje de programación JavaScript, ya que la forma común de mostrar los gráficos en el sitio web es mediante la inserción de código JavaScript en un documento HTML.

JavaScript es un lenguaje ligero e interpretado, orientado a objetos más conocido como el lenguaje de script para páginas web, es un lenguaje de programación dinámico. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje. Las construcciones del lenguaje, tales como sentencias “if”, y bucles “for” y “while”, y bloques “switch” y “try” ... “catch” funcionan de manera muy parecida a estos lenguajes. Algunas cosas que se pueden hacer con JavaScript para hacer una página web dinámica, es lo siguiente:

- Cambiar los atributos HTML
- Puede cambiar los estilos CSS
- Cambiar el contenido HTML



- Validar datos de un formulario
- Acciones al pulsar botones
- Animaciones sencillas

El entorno más común para trabajar JavaScript es en un navegador web, Google Charts a través de su Interfaz de Programación de Aplicaciones, abreviada API (del inglés: **Application Programming Interface**), nos ofrece mediante un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) el uso de la biblioteca para escribir código en JavaScript que nos ayude a crear gráficos y ponerlos en nuestra página web como se muestra en la figura 2.2.1.



Figura 2.2.1 Gráficos con Google Charts

A continuación se describirá de una manera muy general cómo es que trabaja una página web con JavaScript, de esta forma nos daremos una idea de cómo se generarán nuestras gráficas para nuestro sistema con Google Charts, como se ilustra en la figura 2.2.2.

Como ya sabemos JavaScript es lo que se llama un lenguaje de script del lado del cliente. Eso significa que es un lenguaje de programación de computadoras que corre dentro de un explorador de internet (un explorador de internet es también conocido

como un cliente web porque se conecta a un servidor web para solicitarle páginas). La manera como JavaScript trabaja es interesante. Dentro de una página web normal se pone código JavaScript, cuando el explorador carga la página, este tiene un “motor de JavaScript” (también conocido como intérprete de JavaScript o implementación JavaScript), es un intérprete que interpreta (lee) el código fuente de JavaScript y ejecuta la secuencia de comandos en consecuencia en el cliente web.

Los navegadores web suelen crear objetos no nativos, dependientes del entorno de ejecución, para representar el **Document Object Model** (DOM, en español Modelo de Objetos del Documento) en JavaScript. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos etc.

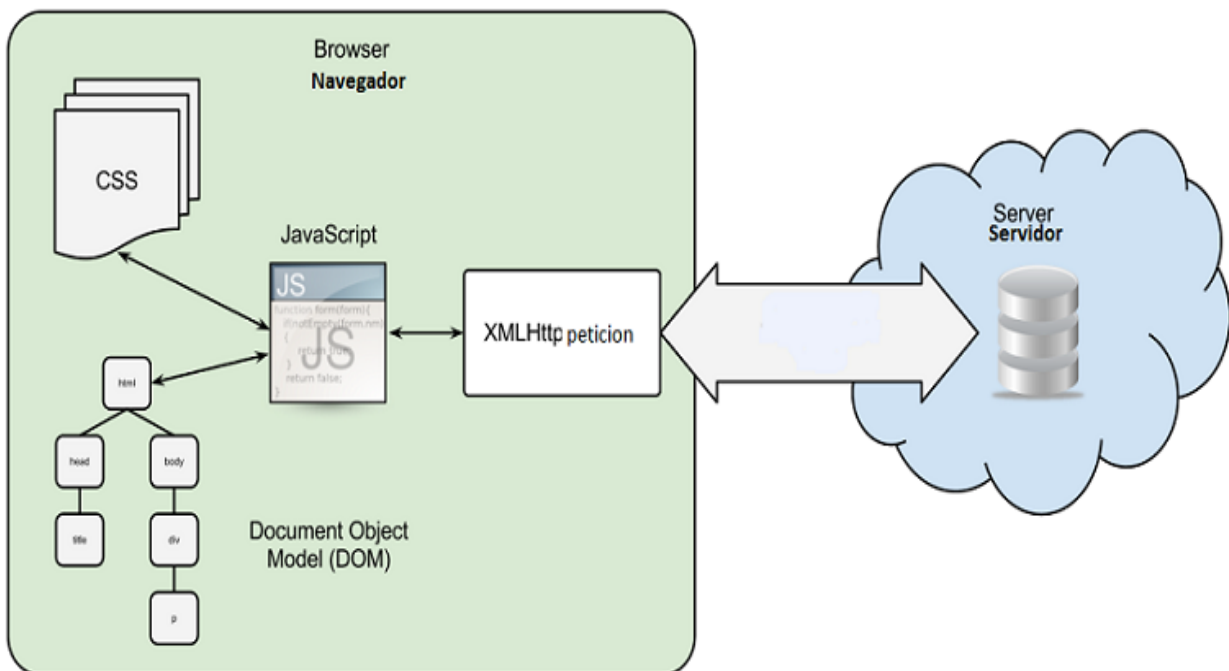


Figura 2.2.2 Procesamiento del lenguaje de programación JavaScript

Una vez que ya sabemos cómo trabaja a grandes rasgos JavaScript veremos las diferentes formas de integrarlo a una página HTML y las características más atractivas que nos ofrece.



La integración de JavaScript no solo se puede hacer con documentos HTML, sino que también se puede integrar con otros lenguajes de programación como PHP, ASP.NET, entre otros, a continuación se presentan las dos formas principales para incluir código JavaScript en páginas web HTML:

- La primera forma es insertar código JavaScript entre las etiquetas `<script>` y `</script>`.

Un ejemplo se muestra a continuación:

```
<script>
    document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

- La segunda forma es la que utilizaremos en las páginas HTML de nuestro SIAH, ya que utilizaremos una biblioteca externa de Google y haremos referencia a ella mediante la API Google Charts, por lo que cargaremos esta API como si fuera un archivo externo de JavaScript, con lo cual colocaremos el atributo `src` (fuente) en la etiqueta `<script>`, a continuación se muestra el ejemplo:

```
<html>
    <head>
        <!--Load the API-->
        <script type="text/javascript" src="https://www.google.com/jsapi"></script>
        .....
    </head>
    .....
</html>
```

A continuación se listan las principales características de JavaScript:

- Lenguaje de scripting (el más usado).
- Está diseñado con una sintaxis similar al C.
- Orientado a objetos.

- Interpretado.
- Basado en prototipos
- Imperativo y estructurado.
- Dinámico.
- Se utiliza principalmente en su forma del lado del cliente.
- Fue desarrollado por Netscape.

Ahora, ya que conocemos la información y las características de JavaScript se listan sus ventajas y desventajas, tratando de enfocar la selección de estas ventajas y desventajas, en los requerimientos de nuestro sistema.

Ventajas

- No se necesitan declarar los tipos de datos y se puede o no poner el punto y la coma al final de una sentencia.
- Es muy fácil de aprender.
- Todos los navegadores modernos interpretan el código JavaScript.
- Es muy usado para la creación de aplicaciones web.
- Es un lenguaje de programación muy liviano y utiliza poca memoria.
- JavaScript es rápido, ya que no requiere un tiempo de compilación
- Fácil de integrar con otros lenguajes de programación.

Desventajas

- Requiere que el intérprete esté activado en el explorador web.
- Es un lenguaje limitado. Por ejemplo, no es posible escribir aplicaciones independientes en JavaScript.
- Presenta problemas de seguridad. El código JavaScript una vez enviado a la parte del cliente y ejecutado, puede ser usado para explotar el sistema del usuario con propósitos maliciosos.



En cuanto a Google Charts se describe a continuación la forma en cómo se carga a un documento HTML la biblioteca y la creación de una gráfica muy sencilla utilizando esta API. Para realizar un gráfico se requieren de tres bibliotecas:

- Google JSAPI API
- Google Visualization
- La biblioteca del propio gráfico.

Estas bibliotecas son cargadas usando dos links `<script>` en la página HTML, como se muestra a continuación:

```
<html>
  <head>
    <!--Carga la API Google Charts-->
    <script type="text/javascript" src="https://www.google.com/jsapi"></script>
    <script type="text/javascript">

      // Carga Google Visualization API y la biblioteca del propio gráfico.
      google.load('visualization', '1.0', {'packages':['corechart']});
      google.setOnLoadCallback(drawChart);
      // ... Dibuja el gráfico...
    </script>
```

El siguiente pedazo de código crea y llena los datos de la tabla de donde se tomara la información para generar nuestra gráfica. También muestra cómo se instancia la gráfica a utilizar pasándole los datos y finalmente lo dibujaremos.

```
function drawChart() {
  var data = new google.visualization.DataTable();
  data.addColumn('string', 'Topping');
  data.addColumn('number', 'Slices');
```

```
data.addRow([
  ['Mushrooms', 3],
  ['Onions', 1],
  ['Olives', 1],
  ['Zucchini', 1],
  ['Pepperoni', 2]]);
var options = {'title':'How Much Pizza I Ate Last Night',
              'width':400,
              'height':300};
var chart = new
google.visualization.PieChart(document.getElementById('chart_div'));
  chart.draw(data, options);
}
</script>
</head>
<body>
  <div id="chart_div" style="width:400; height:300"></div>
</body>
</html>
```

Teniendo ya la definición de lo que es Google Charts y las características de su implementación, se presentan las ventajas y desventajas al usar esta herramienta:

Ventajas

- Trabaja bien, sin problemas. Esto se refiere a que como programadores nos beneficiamos del trabajo de Google, del mantenimiento y las nuevas características que proporciona Google Charts.
- Es software libre por parte de Google.
- Es muy interactivo comparado con otras herramientas para generar gráficos.
- Fácil de personalizar y diseñar.
- Gran variedad y tipos de gráficas.



- No se necesita instalar ningún componente en nuestro entorno o servidor.
- Excelente documentación sobre su uso.

Desventajas

- No tiene una gran actividad por lo que no evoluciona rápidamente.
- Un poco lento a la hora de retornar la imagen que contiene el gráfico.
- Existen otros productos que ofrecen otras características que podrían ser de mayor utilidad en proyectos con más requerimientos en cuanto a gráficos.

2.3 Características, ventajas y desventajas del lenguaje de programación PHP

Los orígenes de PHP están fechados en torno al año 1995, PHP es el acrónimo de Hypertext Pre-processor, es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, con una gran librería de funciones, fue creado por Rasmus Lerdorf en el año de 1995. La primera versión fue conocida como PHP, la cual revolucionó al mercado, generando consigo un gran uso de tecnologías vía web. En torno a estadísticas se maneja que 60 por ciento de los sitios en la red utilizan PHP, este es un lenguaje de gran capacidad que puede ser utilizado desde pequeñas páginas web hasta grandes portales; como el caso de Wikipedia o Yahoo, sitios de gran cantidad de visitas.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, similar al ASP de Microsoft o JSP de Sun, incrustado en páginas HTML y ejecutado en el servidor, el objetivo del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas, aunque también podría ser una página WML (WAP), como se muestra en la Figura 2.3.1. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, sin embargo para que las páginas funcionen el servidor donde están alojadas si deben soportar PHP.



Figura 2.3.1 Secuencia de PHP en la generación de páginas Web



Al ser PHP un lenguaje libre, dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas.

Características

- Integración con varias bibliotecas externas
- XFORMS define una variable de los webforms tradicionales, los cuales permiten ser usados por una gran variedad de plataformas y navegadores o incluso en medios no tradicionales como documento PDF.
- Analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del web de fácil programación.
- Es un lenguaje multiplataforma
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador; esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan, destaca su conectividad con Adabas D, Empress, IBM DB2, Informix, Ingres, Interbase, Frontbase, mSQL, Direct MS-SQL, Oracle (OCI7, OCI8), Raima Velocis, Solid, Sybase, MySQL y PostgreSQL
- Posee una amplia documentación en el sitio oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un archivo de ayuda.
- PHP soporta cookies HTTP de forma transparente; las cookies son un mecanismo por el que se almacenan datos en el navegador remoto y poder monitorear o identificar a usuarios que regresan al sitio
- Es de código abierto, libre, por lo que se presenta como una alternativa de fácil acceso; puede ser descargado de la siguiente dirección <http://www.php.net/downloads.php> (como se muestra en la Figura 2.3.2).

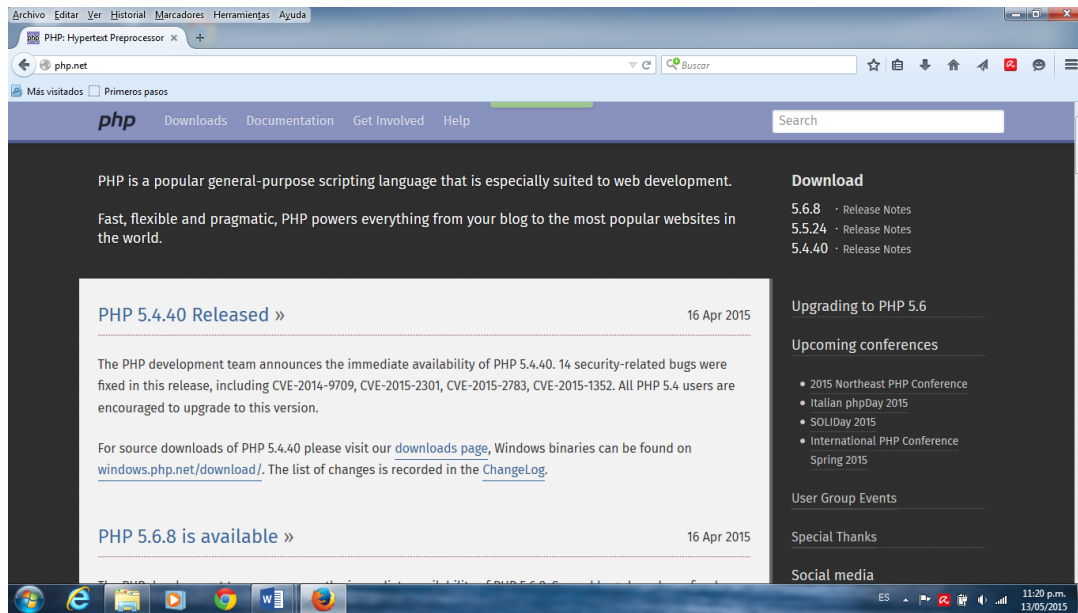


Figura 2.3.2 Página de Descarga de PHP

- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia.
- No necesita definición de tipos de variables aunque sus variables pueden ser evaluadas también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.
- PHP no tiene una metodología de programación, el desarrollador puede aplicar cualquiera, y puede escribir código ordenado, estructurado y manejable.
- PHP soporta ocho tipos de datos primitivos:
 - Boolean
 - Integer
 - Double
 - String
 - Array
 - Object
 - Resource
 - Null



Ventajas

- Es un lenguaje multiplataforma
- Está completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos
- Lenguaje totalmente libre y abierto
- Los entornos de desarrollo son de rápida y fácil configuración
- Es de fácil despliegue: paquetes totalmente autoinstalables que integran PHP
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos
- Cuenta con una comunidad muy grande para alguna incontinencia que pudiera presentarse
- PHP corre casi en cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en aproximadamente 25 plataformas, incluyendo diferentes versiones de AS/400, Mac OS X, Novell Netware, OS 2, RISC OS, SGI IRIX 6.5x, Solaris (SPARC, INTEL), GNU/Linux y Microsoft.
- El código fuente escrito en PHP es invisible al navegador y al cliente; ya que el servidor es el responsable de ejecutar el código y enviar el resultado HTML al navegador, lo cual hace que la programación en PHP sea segura y a la vez confiable.
- Tiene la posibilidad de extender su potencial utilizando la enorme cantidad de extensiones que pueden ser utilizadas para mejorar el rendimiento en el desarrollo del trabajo que estemos haciendo.
- Es totalmente gratuito y con documentación muy amplia en la red, no es difícil encontrar algún tutorial sobre el uso del lenguaje.
- Posee versatilidad y sencillez, es un lenguaje con una sintaxis sencilla que no es muy difícil de entender.
- El lenguaje es de uso común en la red y sin costo, una cantidad considerable de desarrolladores que utilizan el programa se encuentran constantemente trabajando para la mejora del motor de PHP, a medida que transcurre el tiempo

y aumenta la versión del programa, se consigue mayor seguridad y estabilidad en el mismo.

Desventajas

- El sistema necesita contar con la actualización de archivos del sistema operativo para evitar vulnerabilidades en la red.
- No cuenta con códigos prediseñados, esto en el momento de estar desarrollando es un pequeño inconveniente.
- Es un lenguaje que se interpreta en ejecución para ciertos usos puede resultar un punto en contra que el código fuente no pueda ser ocultado.
- La ofuscación es una técnica que puede dificultar la lectura del código, y en ciertos casos representa un costo durante la ejecución
- Su actual modelo de objetos los trata como si fuesen un tipo simple de dato, y al pasarse como parámetro a una función, se hace una copia del objeto y no tan solo un paso de una referencia.

2.4 Características, ventajas y desventajas del framework Polymer.

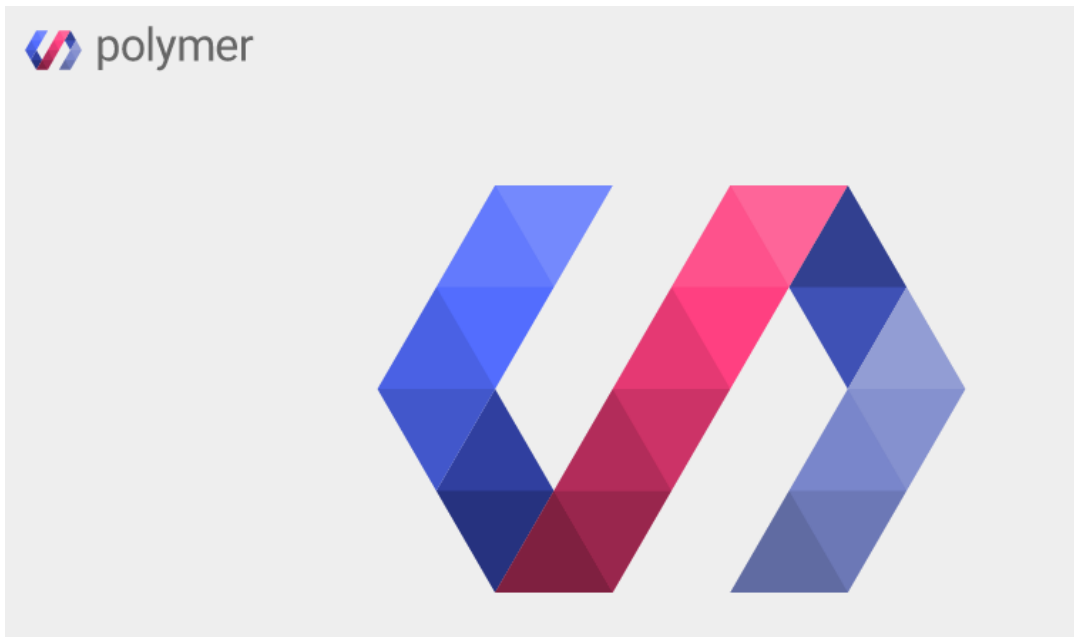


Figura 2.4.1 Logotipo de framework Polymer.

La web está evolucionando a pasos agigantados en los últimos tiempos. Una de las direcciones de avance en este sentido apunta hacia su modularización. En lugar de construir pesadas aplicaciones monolíticas, parece estar más en sintonía con los principios arquitectónicos de la web, hacer un desarrollo dirigido por la construcción de componentes web sencillos, atómicos y funcionales que fomenten su reutilización transversal y puedan utilizarse como etiquetas personalizadas para articular soluciones elaboradas.

Con la llegada de HTML5, el desarrollo web ha evolucionado a una velocidad brutal, pero ahora ha nacido un nuevo concepto en el desarrollo web, los llamados **web components** (componentes web, también conocido como elemento), los cuales permiten llevar la velocidad del desarrollo web a otro nivel y uno de los frameworks de desarrollo de web components que está tomando mucho auge es Polymer como se muestra en la figura 2.4.1. Cuando hablamos de Polymer tenemos que pensar en web components, ya que Polymer de por sí es una librería para la creación fácil de web components.

Web Components

No es un tema para nada nuevo, desde hace un tiempo se empiezan a desarrollar una nueva forma de hacer la web y al parecer se refiere a esto. Cuando se plantea la teoría de web components, lo primero que se comenta es que para desarrollar la web en la actualidad se manejan un sin fin de **plugins** (complemento, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.), archivos CSS de gran peso y un desarrollo en HTML que en la mayoría de los casos puede resultar un caos cuando de proyectos grandes se trata ya que termina siendo para nada entendible y gigante.

Lo que propone web components es simple, crear nuestros propios componentes web, los cuales se le pueden agregar configuraciones a manera de atributos para poder dar mayores funcionalidades. Además es un escenario donde se crean nuestros propios elementos para que estos puedan ser reutilizables, mantenibles y de fácil lectura.

En internet existen muchas opciones para hacer menús desplegados, lighthbox, mapas, sliders, en donde en algunos casos son difíciles o tediosos de implementar. Entonces para lograr la construcción de un web component se tiene que tener en cuenta lo siguiente:

- **Templates** : es donde se define la estructura del componente.
- **Decorators** : es el css que se aplica al componente.
- **Custom Elements** : permite crear nuevos componentes a la cual se le puede agregar funcionalidades.
- **Shadow Dom** : encapsula un sub árbol DOM para la composición más fiable de elementos de la interfaz de usuario.
- **Imports** : se define las plantillas, decoradores y elementos con los que se trabajara.

En pocas palabras web components usa un comportamiento de encapsulamiento.



Shadow DOM

Ya que el shadow DOM es una de las piedras angulares de Polymer. El **shadow DOM** es la habilidad que tiene los Browser de incluir un subárbol de elementos **DOM**, dentro de un documento renderizado, pero no dentro del Documento principal. Se trata de una capa de abstracción que se introduce en el DOM del navegador, que "esconde" los elementos que se encuentren dentro. Cuando creamos un elemento Polymer, la plantilla que lo forma se encuentra directamente dentro de una capa de shadow DOM.

Material Design

Material Design es un lenguaje de diseño que utiliza los principios clásicos de diseño, e incorpora la ciencia y la tecnología para dar un aspecto material a los componentes de **UI** (interfaz de usuario), como sombras y aceleración, de modo de lograr una metáfora de los componentes con objetos reales que le transmita al usuario información en forma intuitiva y que permita usar los años de diseño aplicado a objetos materiales a una nueva rama de diseño, que es el diseño de UI, que solo lleva pocos años de vida. Material Design es además el resultado de la evolución de Android a través de sus diferentes versiones.

¿Qué es Polymer?

Polymer es una tecnología surgida de la necesidad de dotar a las aplicaciones web de modularidad que permita un desarrollo sencillo y ágil de las nuevas aplicaciones que se están realizando en este entorno. Polymer es un framework de desarrollo web basado en el encapsulamiento e interoperabilidad de elementos personalizados que se extienden del HTML; los web components que se construyen con Polymer, terminan siendo etiquetas HTML particulares que encapsulan un funcionamiento y que pueden o no presentar atributos que permitan la interoperabilidad con la misma.

Es una librería muy completa que nos ayuda a la creación de web components de una manera más fácil y pensando siempre en crear en un ecosistema de componentes. Para crear un web component (también conocido como elemento) con Polymer primero se tiene que saber manejar **bower** (es un gestor de librerías, es decir, te permite agregar, actualizar y eliminar paquetes: librerías, plugins, frameworks, etc., de las cuales depende el proyecto) ya que así será más fácil manejar las dependencias.

Polymer³ es una librería web creada por Google construida sobre los estándares establecidos por la **W3C** (World Wide Web Consortium) con respecto a la creación de componentes web. Permite crear tanto componentes web aislados como complejas aplicaciones web de escritorio o móviles. Debido a que los estándares de la W3C no disponen de soporte por parte de todos los navegadores web, Polymer trae consigo un conjunto de implementaciones de los estándares de la W3C referentes a componentes web (templates, custom elements, shadow DOM e imports). Estas implementaciones son denominadas **polyfills** (Un Polyfill puede ser un trozo de código o un plugin que permite tener las nuevas funcionalidades de HTML5 en aquellos navegadores que nativamente no lo soportan) y proveen soporte como se ha comentado a navegadores que no soportan de manera nativa dichos estándares, permitiendo a los navegadores que sí los soporten usar su implementación nativa.

Características

- Se implementa para nuevos navegadores. Google está trabajando mucho en este desarrollo, de modo que hay una comunicación continua entre los equipos de desarrollo de los nuevos frameworks como AngularJS o Polymer con el equipo de desarrollo de Chrome.
- Es algo reciente (2013). En YouTube existen los vídeos de presentación de Google I/O 2013 y muchos otros comentando polymer, pero hay pocos manuales y documentación referente al tema.

³ "Polymer"; <https://www.polymer-project.org/>



- Existe cierta compatibilidad hacia atrás a través de código Javascript gracias a los Polyfills. Aun así en la implementación de esta tecnología se tiene que tener cuidado a la hora de utilizarla porque no está muy desarrollada (versión 0.5) y compatibilidad no muy alta. Quizá sea algo demasiado experimental, aunque hay algunas páginas que empiezan a usarlo (como GitHub). Actualmente (27 de marzo 2015) se lanzó la versión Polymer 0.8 (alpha).
- Dispone de un catálogo de elementos basados en material design (Android 5) pero para la Web.

Elementos de Polymer



Polymer - Elements

Polymer core elements

Polymer's core elements are a set of visual and non-visual utility elements. They include elements for working with layout, user input, selection, and scaffolding apps.

[DOCS](#) [DEMOS](#)

Core Elements Demo 0

- Demo 1
- Demo 2
- Demo 3
- Demo 4
- Demo 5
- Demo 6
- Demo 7
- Demo 8
- Demo 9
- Slider
- Toolbar
- Form

`<core-ajax>`
`<core-selector>`
`<core-toolbar>`
`<core-scaffold>`
 ...

Paper elements

Polymer's paper elements collection implements material design for the web. They're a set of highly visual, highly interactive elements that include things like controls, layouts, hero transitions, and scrolling effects.

[DOCS](#) [DEMOS](#)

Paper Elements Toolbar

- Dropdown
- Radio button
- Toggle button
- Text
- Toolbar
- Scroll Header Panel
- Progress bar
- Slider
- Title
- Button
- Icon Button
- File

`<paper-slider>`
`<paper-tabs>`
`<paper-dialog>`
`<paper-button>`
 ...

Figura 2.4.2 Elementos de Polymer⁴

Polymer dispone de elementos visuales y no visuales como se muestra en la figura 2.4.2, apoyando estos últimos el funcionamiento de los primeros. Con respecto a los elementos visuales un subconjunto de ellos, los **paper elements** (elementos de papel), son implementan el diseño llevado a cabo por Google, el Material Design. Dicho

⁴Página de Polymer Elements <https://www.polymer-project.org/0.5/docs/elements/>

diseño, se inspira en los elements paper y **core elements** (elementos básicos) para crear un diseño interactivo, visual y lleno de movimiento, además de multiplataforma.

Polymer cuenta actualmente con dos bibliotecas de elementos. Por una parte los “Core elements” y por otra los “Paper elements”.

Tenemos:

- **Core elements:** para la funcionalidad básica que se espera en una página web de forma independiente a su apariencia. Se trata de componentes básicos como llamadas Ajax, y efectos de animaciones como Collapse, iconos, barras... estructuración de la página, etc...
- **Paper elements:** Es la implementación del Material Design usando web components que es la apariencia que Google ha adoptado a partir de Android 5.0.

En la página de Polymer hay demos del funcionamiento de cada uno de ellos así como la documentación para utilizarlos.

Las capas de Polymer

Hay tres capas conceptuales a Polymer:

- **Componentes web:** Polymer se construye en los estándares web components. No todos los navegadores soportan estas características, sin embargo, la capa de polyfill llena los vacíos, además de la implementación de las API de JavaScript. En tiempo de ejecución, Polymer recoge automáticamente la ruta más rápida - la aplicación o JavaScript nativo.
- **La biblioteca de Polymer :** Proporciona una sintaxis declarativa que hace que sea más sencillo para definir elementos personalizados. Y añade características



como enlace de datos bidireccionales, designación de propiedad y soporte de gestión para ayudar a construir poderosos elementos reutilizables.

- **Elementos:** Core elements y Paper elements forman un conjunto completo de la interfaz de usuario. Estos elementos dependen de la biblioteca Polymer, pero están separados y es opcional. Puede utilizar los elementos sin utilizar directamente Polymer, o puede utilizar Polymer para crear sus propios elementos y no utilizar los elementos básicos o de papel en absoluto. Usted puede mezclar y combinar los Core elements y Paper elements con otros elementos, incluyendo una función de elementos y otros elementos personalizados.

Componentes de polymer

Polymer tiene polyfills para muchas de las tecnologías web necesarias para implementar con éxito aplicaciones con componentes web como se muestra en la figura 2.4.3, incluyendo:

- **HTML imports:** (Importaciones HTML) : una manera de incluir y reutilizar documentos HTML dentro de otro documento HTML.
- **Custom elements:** (Elementos personalizados) : permite a los desarrolladores definir y utilizar elementos DOM personalizados.
- **Shadow DOM:** (Sombra DOM) : proporciona encapsulación en el DOM.
- **MDV** (Model Driven Views): ofrece angularjs-como el enlace de datos .
- **Web animations** (Animaciones Web) : API para implementar animaciones complejas.
- **Pointer events** (Eventos de puntero) : unifica eventos para mouse.

Estos polyfills pueden utilizarse por separado, sin la necesidad de utilizar el resto del Polymer. Sin embargo, además, Polymer proporciona:

- **polymer.js** : El tiempo de ejecución de polímero de núcleo, por lo que es más fácil crear atributos y eventos personalizados
- Elementos visuales y no visuales.

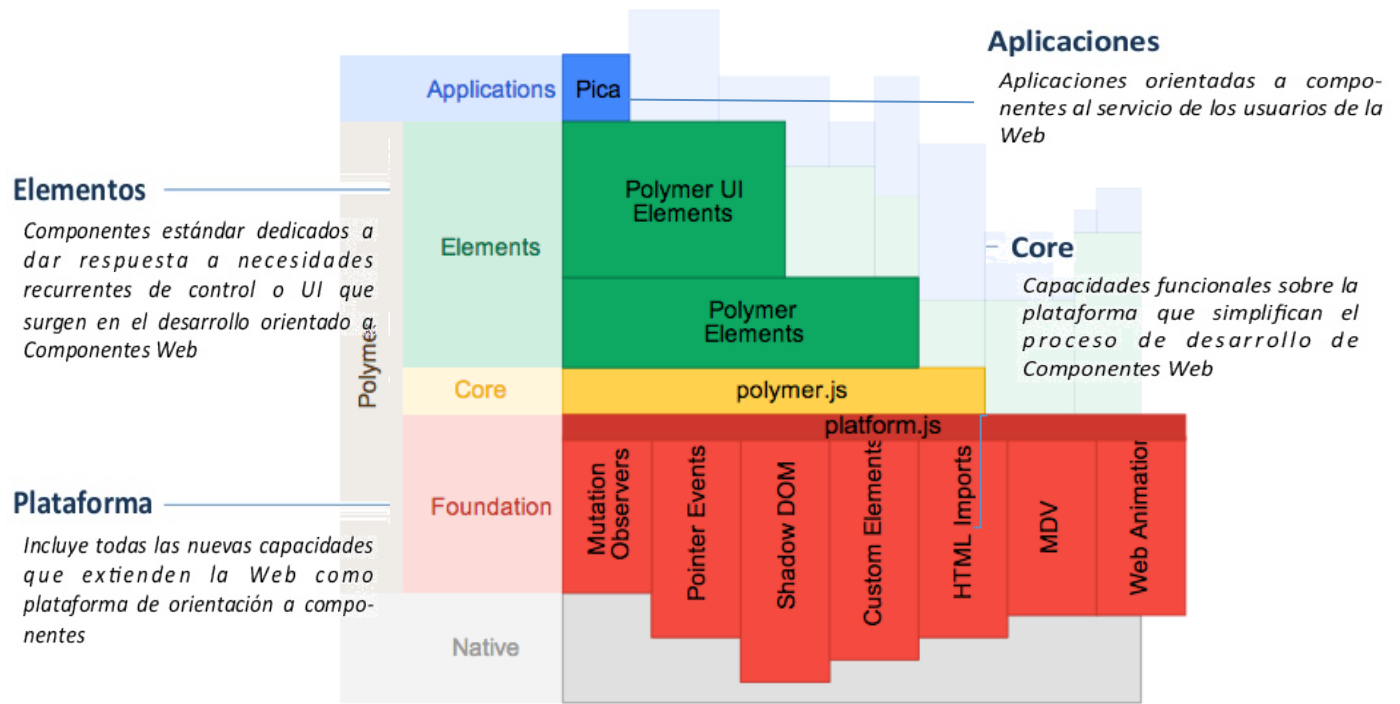


Figura 2.4.3 . Componentes de Polymer

Ventajas

- Permite a los desarrolladores crear sus propios componentes reutilizables que todo navegador admite.
- Permite crear aplicaciones web utilizando los servicios de Google de forma sencilla.
- Permite utilizar componentes propios y de terceros, y manipularlos para construir aplicaciones de forma simple, reduciendo tiempos y esfuerzo.
- Con Polymer, puedes crear tus propios elementos HTML y componerlos en aplicaciones completas y complejas, que son escalables y mantenibles.
- Hace que sea mucho más fácil para crear grandes sitios web y aplicaciones.
- Polymer implementa los nuevos estándares de la W3C acerca de los componentes web con una sintaxis simple y orientada a usuarios finales



- Permitir un alto grado de reutilización de sus componentes, disminuyendo así el costo de desarrollo y mantenimiento de las aplicaciones.
- Facilita la creación de web components, así como aplicaciones web complejas de escritorio o móviles
- Es más entendible nuestro código html, ya que en proyectos muy grandes es muy difícil de mantener y por ende entender
- Dentro de la página oficial existe un tutorial donde te muestra paso a paso los elementos, para que sirven y como se utilizan.

Desventajas

- Polymer se construye en los estándares web components, no todos los navegadores soportan estas características
- Polímero es una tecnología intrigante, pero es innegable que aún está en pleno desarrollo.
- No hay muchos guías o tutoriales disponibles para el desarrollo web Polymer.js.
- Es altamente dependiente de los navegadores que se emplean.
- Polymer requiere un nivel bastante significativo de experiencia en el uso de tecnologías de navegación como el DOM
- Es un proyecto que está en desarrollo y que tiene que mejorar sobre todo en compatibilidad con los navegadores que lo soportan.

2.5 Características, ventajas y desventajas del lenguaje de programación HTML5

El lenguaje de programación del **World Wide Web** (WWW) siempre ha sido **Hyper Text Mark-up Language** (HTML). HTML fue diseñado como un lenguaje para describir semánticamente documentos científicos, sin embargo su diseño general y adaptaciones a través de los años ha permitido que pueda ser utilizado para varios tipos de documentos más. Desde su creación en 1989, HTML ha sufrido varias modificaciones hasta llegar a su última variación en 2014 como se muestra en la tabla 2.5.1.

Versión	Año
Tim Berners-Lee inventa www	1989
Tim Berners-Lee inventa HTML	1991
Dave Raggett boceta HTML+	1993
Grupo de trabajo de HTML define HTML 2.0	1995
Revisión de W3C sobre HTML 3.2	1997
Revisión de W3C sobre HTML 4.01	1999
Revisión de W3C XHTML 1.0	2000
Primer boceto público de WHATWG sobre HTML5	2008
Standard público de WHATWG sobre HTML5	2012
Recomendación final de W3C sobre HTML5	2014

Tabla 2.5.1



El **W3C** es el organismo encargado de realizar las revisiones y validaciones de HTML, el objetivo de W3C es llevar el WWW a su máximo desarrollo implementando protocolos y guías que aseguren el crecimiento a largo plazo de la web.

El **Web Hypertext Application Technology Working Group** (WHATWG) es una comunidad interesada en el crecimiento de las tecnologías web. Se enfoca principalmente en el desarrollo de HTML y APIs necesarios para las aplicaciones web.

Muchos desarrolladores entienden que los documentos HTML deben trabajar de la misma manera en todas las plataformas y navegadores. La compatibilidad de HTML es la clave de los beneficios de su desarrollo. Cada edición de HTML busca alcanzar mayor unanimidad dentro de la industria sin importar la plataforma, navegador o dispositivo.

HTML5 es la última evolución de la norma que define HTML. El término representa dos conceptos diferentes

- Se trata de una nueva versión del lenguaje HTML, con nuevos elementos, atributos y comportamientos.
- Un conjunto más amplio de tecnologías que permite a los sitios web y a las aplicaciones ser más diversas y de gran alcance. Este conjunto se le llama HTML5.

Los pilares en los que se basan las características de HTML5 son:

- Semántica: Permite describir con mayor precisión cuál es su contenido.
- Conectividad: Permite comunicarse con el servidor de formas nuevas e innovadoras.
- Fuera de línea y almacenamiento: Permite a páginas web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.
- Multimedia: Nos otorga un excelente soporte para utilizar contenido multimedia como lo son audio y video nativamente.

- Gráficos y efectos 2D/3D: Proporcionar una amplia gama de nuevas características que se ocupan de los gráficos en la web como lo son el lienzo 2D, WebGL, SVG, etc.
- Rendimiento e Integración: Proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- Acceso al dispositivo: Proporciona APIs para el uso de varios componentes internos de entrada y salida de nuestro dispositivo.
- CSS3: Nos ofrece una gran variedad de opciones para la sofisticación del diseño.

Ventajas

- Compatibilidad con dispositivos mobile.
- No necesita instalación de plugins
- Nuevos elementos incluidos en el estándar como se muestra en la Tabla 2.5.2.

Tag	Descripción
<article>	Define un artículo en el documento
<aside>	Define contenido lateral al contenido principal
<bdi>	Define una parte de texto que será formateada de manera diferente al resto
<details>	Define detalles adicional que el usuario puede ver o esconder
<dialog>	Define un cuadro de texto o ventana
<figcaption>	Define encuadre para el elemento <figure>



<code><figure></code>	Define un elemento auto contenido con objetos como ilustraciones, diagramas, fotos, código, etc.
<code><footer></code>	Define la parte inferior de un documento o sección
<code><header></code>	Define la parte superior de un documento o sección
<code><main></code>	Define el contenido principal del documento
<code><mark></code>	Define un texto resaltado
<code><menuitem></code>	Define un comando o menú que puede ser utilizado para invocar un menú popup
<code><meter></code>	Define una medida escalar o rango
<code><nav></code>	Define las ligas de navegación de un documento
<code><progress></code>	Define el progreso de una tarea
<code><rp></code>	Define que mostrar en los navegadores que no soportan anotaciones de Ruby
<code><rt></code>	Define una explicación o pronunciación de caracteres (para tipografía asiática)
<code><ruby></code>	Define anotaciones Ruby
<code><section></code>	Define una sección de un documento
<code><summary></code>	Define un sección superior visible para elementos <code><details></code>
<code><time></code>	Define una hora y fecha

<wbr> Define un posible salto de línea

Tabla 2.5.2

- Define nuevos elementos de formas de contacto como se muestra en la Tabla 2.5.3.

Tag	Descripción
<datalist>	Define una lista de opciones para controles de entrada
<keygen>	Define un campo de paridad de llaves
<output>	Define el resultado de un cálculo

Tabla 2.5.3

- Define nuevos elemento de generación de gráficos 2D y 3D como se muestra en la Tabla 2.5.4

Tag	Descripción
<canvas>	Define un elemento gráfico para dibujar utilizando JavaScript ⁵
<svg>	Define un gráfico para dibujar utilizando SVG ⁶

Tabla 2.5.4

⁵ JavaScript es un lenguaje de programación de scripts multiplataforma orientado a objetos. Es un lenguaje liviano y pequeño. Dentro de un ambiente JavaScript, los objetos pueden ser conectados para permitir a través de programación control sobre ellos.

⁶ SVG es un lenguaje de etiquetas para describir gráficos e imágenes bidimensionales y un set de interfaces de programación. SVG es soportado por todos los navegadores modernos para Desktop y Mobile.



- Define nuevos elementos de multimedia como se muestra en la Tabla 2.5.5
- Permite el almacenamiento local de datos
- Permite geolocalización
- Permite reproducción de audio y video sin necesidad de plugins

Tag	Descripción
<audio>	Define contenido de sonido o música
<embed>	Define contenedores para aplicaciones externas
<source>	Define la localización del recurso de <video> y <audio>
<track>	Define pistas de <video> y <audio>
<video>	Define contenido de video o película

Tabla 2.5.5

Desventajas

- No todos los navegadores tienen compatibilidad con los nuevos elementos
- Los navegadores de versiones anteriores no pueden hacer uso de ninguna de las nuevas funcionalidades
- La eficacia y velocidad del código se ve reducida por no ser código nativo
- La diferencia de tiempo de ejecución entre plataformas puede causar problemas en navegadores mobile.
- El estándar aún no se ha terminado por completo
- Los estándares de seguridad son limitados.
- El acceso al hardware a través de APIs es limitado

- El almacenamiento de datos locales es de espacio limitado
- Las aplicaciones offline tienen muchas complicaciones para hacer las sincronizaciones
- No hay un estándar de video y audio que pueda ser utilizado en todas las plataformas y navegadores
- Las implementaciones de HTML5 dependen del navegador que se esté utilizando
- La utilización de algunas etiquetas del lenguaje cambiaron por lo que se requiere migración como se muestra en la tabla 2.5.6.

HTML4	HTML5
<code><div id="header"></code>	<code><header></code>
<code><div id="menu"></code>	<code><nav></code>
<code><div id="content"></code>	<code><section></code>
<code><div id="post"></code>	<code><article></code>
<code><div id="footer"></code>	<code><footer></code>

Tabla 2.5.6



CAPÍTULO III. DEFINICIÓN Y ANÁLISIS DEL PROBLEMA

3.1 Análisis del problema

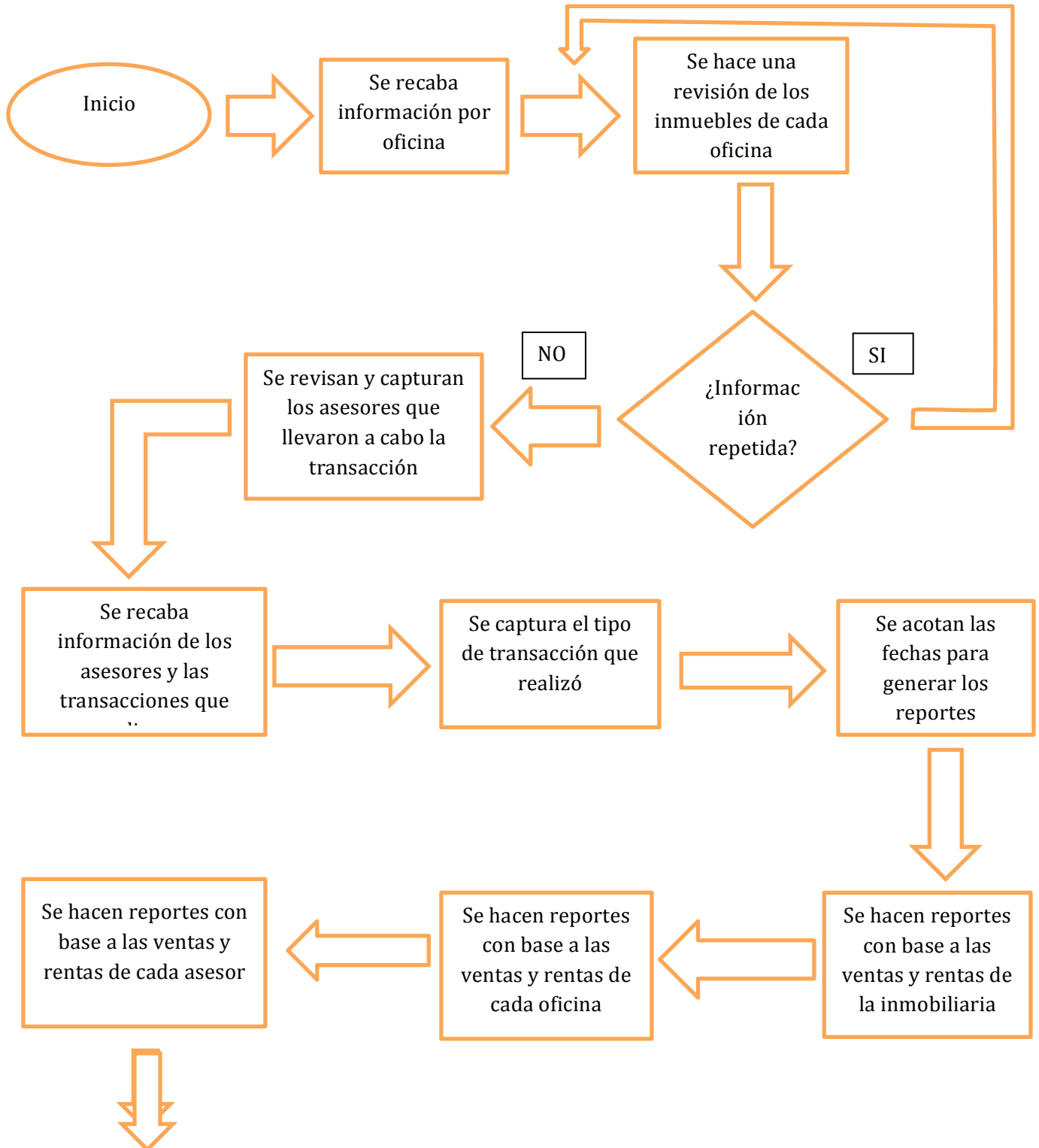
Las problemáticas que se nos presentan y lo que ha motivado el desarrollo de esta tesis y la creación de un SIAH se describen a continuación.

La administración de transacciones en la inmobiliaria y sus oficinas, necesita del control y análisis de las operaciones efectuadas para la toma de decisiones. Las operaciones realizadas en las diferentes áreas de la inmobiliaria generan datos relevantes, reflejan el desempeño de cada una de estas áreas por oficina, lo cual implica la necesidad de generar reportes con los datos recabados para valorar qué tan eficaces están siendo las diferentes oficinas, además de todo esto también se debe llevar una administración de las propiedades que cada oficina tiene a su cargo y con base a esto, se realizan reportes sobre las transacciones que se realicen. Además de lo antes mencionado también se presentan problemas de manera cotidiana, básicamente debido a temas de comunicación oportuna entre las áreas funcionales de la empresa, las oficinas y demás circunstancias que se presentan en los procesos que tiene la empresa

Tener que analizar todo esta información se vuelve tedioso y se pierde mucho tiempo en ello, además de que hay bastante desorganización en cada área de la empresa y por cada oficina que tiene esta inmobiliaria a lo largo de la República Mexicana, ya que cada quien aplica un proceso distinto en cuanto a administrar propiedades e inmuebles. Todo esto impide tener un reporte confiable que presente datos actualizados de cada oficina y las transacciones que se realizan por cada inmueble que pertenece a esa oficina, ya que se presentan inconsistencias a la hora de analizar los datos, datos repetidos en la información ya que nadie se preocupa por seguir reglas a la hora de capturar los datos en Excel o en un sistema independiente que cada oficina pueda llegar a tener, no se incluyen datos específicos que ayuden a identificar al personal que llevo a cabo cierta transacción y el tipo de transacción que realizó.

Por todas estas problemáticas antes mencionadas, se presenta el diagrama de flujo para analizar el proceso actual que se sigue en la mayoría de las oficinas que

pertenecen a la inmobiliaria, para llevar la administración de los inmuebles y como es que estos informes de administración son analizados por el corporativo, como se muestra en la figura 3.1.1.



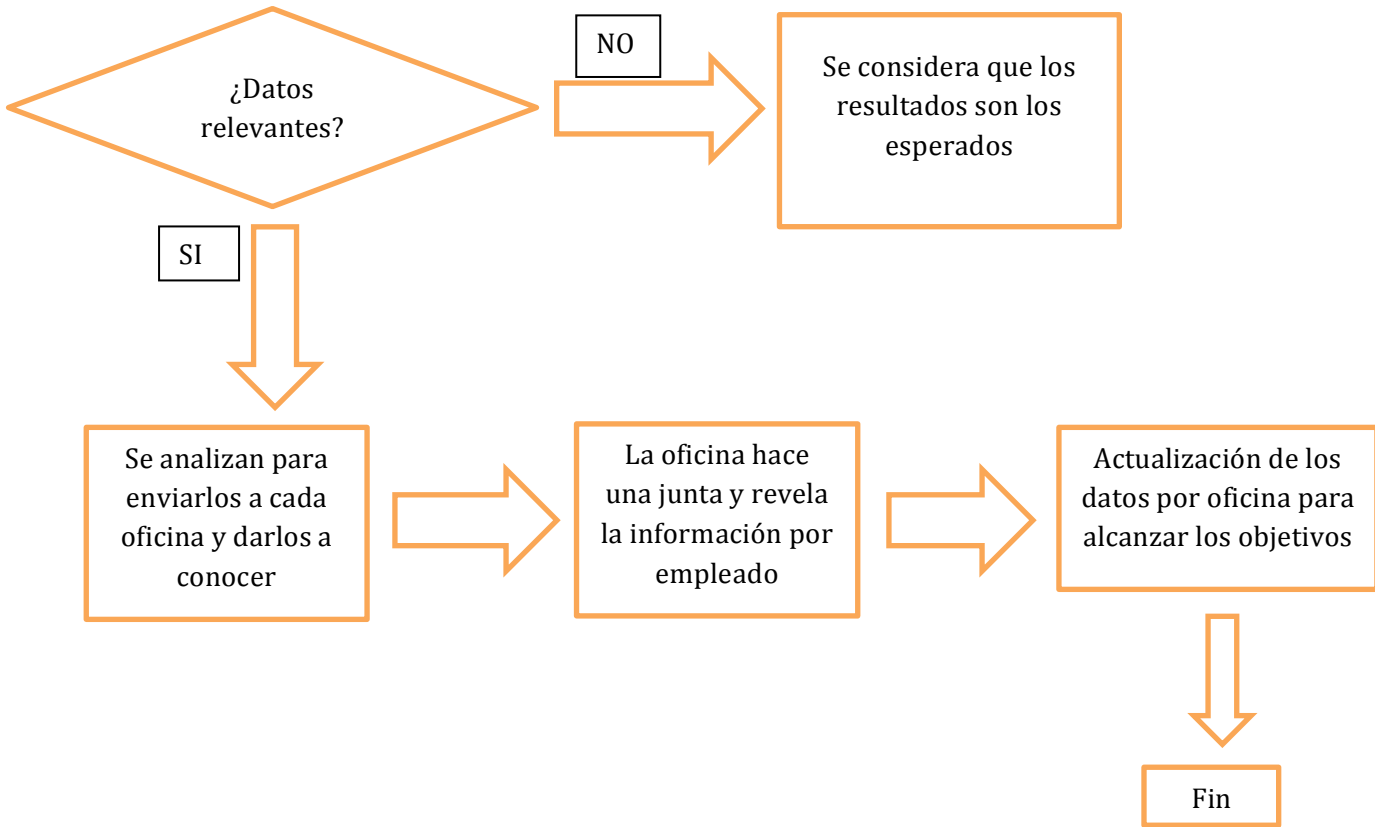


Figura 3.1.1 Diagrama de flujo del proceso actual.

Por lo tanto, nuestros clientes requieren de un sistema que ayude a poner en un solo lugar toda esta información, a continuación iremos describiendo cada bloque a fin de clarificar mejor las actividades y los pasos involucrados durante el proceso de administración actual.

- **Se recaba información por oficina.** Se refiere a que se lleva un registro de las propiedades-inmuebles de cada oficina, es decir, cada oficina tiene la obligación de llevar los registros de los inmuebles que tiene a su cargo, también es responsable de llevar el registro de los asesores comerciales que trabajan en esa oficina y las transacciones (ventas-rentas) que lleva a cabo cada uno de estos empleados. Estos reportes se entregan cada cierto periodo al corporativo ubicado en la Ciudad de México.

- **Se hace una revisión de los inmuebles de cada oficina.** En el corporativo se encargan de revisar los registros que llegan por parte de las oficinas y lo comparan con la base de datos que ellos tienen. Si hay información repetida se encargan de filtrarla hasta verificar que la información es correcta.
- **Se revisan y capturan los asesores que llevaron a cabo la transacción.** Una vez que se tienen las menos inconsistencias posibles, los asesores que llevaron a cabo la venta o renta de un inmueble son registrados para tener idea del desempeño del personal de la inmobiliaria y tener registro de ellos en nómina.
- **Se recaba información de los asesores y las transacciones que realizaron.** Este paso del proceso va muy de la mano con el anterior, ya que se ligan las transacciones y los asesores de manera que se pueda hacer un análisis de las ventas-rentas que realiza en cierto tiempo cada asesor.
- **Se captura el tipo de transacción que realizó.** Esto se refiere primero a los asesores encargados de las ventas-rentas, y después a cada oficina en particular, este paso es para tener bien claro que es lo que más se está solicitando en cada región del país si una venta o una renta de un inmueble y como es que las oficinas y los empleados están respondiendo a estas demandas.
- **Se acotan las fechas para generar los reportes.** Los datos que llegan al corporativo por parte de las oficinas no siempre llegan en las fechas estipuladas y llegan desfasados de tiempo, por lo que para analizar los datos el corporativo requiere de acotar las fechas y de esta forma generar datos que se acerquen lo más posible a la realidad de lo que está sucediendo en el mercado inmobiliario.
- **Los siguientes tres pasos (Se hacen reportes con base a las ventas y rentas de la inmobiliaria, de cada oficina y de cada asesor de ventas).** Se refieren a los reportes finales que genera el corporativo con base a los datos que recibió de cada una de sus oficinas ubicadas en la República Mexicana, de esta



manera la inmobiliaria conoce su posicionamiento a nivel nacional y también puede darse cuenta en que regiones del país tiene que mejorar o mantenerse, con todo esto genera un reporte para cada oficina. Si la oficina en términos generales salió con buenos números y cumplió con los objetivos simplemente se le envía un mail informándoles que continúen de esta manera, de lo contrario si los datos son relevantes sucede lo descrito en el siguiente párrafo.

- **Se analizan para enviarlos a cada oficina y darlos a conocer.** Si la información que arroja el estudio resulta ser “negativa”, se envía por parte del corporativo un informe a las respectivas oficinas, dando a conocer en qué aspectos están fallando.
- **La oficina hace una junta y revela la información a cada empleado.** Con la información que envía el corporativo se hace una junta para dar a conocer los puntos débiles y a cubrir por parte de la oficina y a los empleados se les entrega sus reportes para que sepan en qué mejorar y de qué manera hacerlo.
- **Actualización de los datos por oficina para alcanzar los objetivos.** Finalmente cada oficina implementa las recomendaciones dadas por el corporativo para alcanzar los objetivos deseados.

En base al proceso de administración y generación de reportes descritos anteriormente, nos podemos percatar de los problemas que se quieren llegar a evitar con el desarrollo de un SIAH, los cuales se listan a continuación:

- Errores humanos (reducirlos lo más posible).
- Excesivo tiempo para conocer la información que es de primera importancia para la inmobiliaria.
- Pérdida de documentos.
- Duplicidad de reportes.
- Poca comunicación entre las distintas áreas y oficinas.

- Datos inconsistentes.
- Inexactitud para poder medir el desempeño de los trabajadores.
- Información dispersa por todas partes.

Con base en este análisis del problema presentado, podremos colaborar conjuntamente con la empresa para desarrollar un sistema que ayude a mejorar la administración habitacional y haga crecer a nuestro cliente de una manera importante, como se muestra en la figura 3.1.2.



Figura 3.1.2 Colaboración en el desarrollo de una empresa.



3.2 Requerimientos de sistema.

El SIAH debe ser un sistema que facilite a las oficinas y asesores inmobiliarios la administración de las propiedades así como todas las operaciones asociadas con la venta o renta.

Las principales características con los que debe contar son:

- Sistema en plataforma web. Los asesores inmobiliarios realizan su operación diaria regularmente fuera de su oficina de trabajo, por lo que el sistema debe permitir conectarse y realizar operaciones desde cualquier computadora o dispositivo móvil sin necesidad de realizar la instalación de un software o plugin externo.
- Conexión a Internet. Para poder realizar los movimientos, actualizaciones o revisión de resultados, la computadora o dispositivo móvil que se utilice para ingresar al sistema debe contar con conexión a Internet con velocidad mínima de conexión de 1Mb o 3G dependiendo del dispositivo utilizado.
- Dispositivo para visualizar la aplicación. La aplicación deberá poder utilizarse en la mayoría de los dispositivos de escritorio, laptop y móviles. La compatibilidad deberá abarcar dispositivos con requerimientos mínimos para la visualización de páginas web. Se recomiendan equipos con al menos un procesador superior a Intel Pentium 4 o similares, 1 GB de memoria RAM y 256 GB de disco duro para computadoras de escritorio y laptop. Para equipos móviles se recomienda un smartphone.
- Sistema operativo. La aplicación deberá poder ser visualizada en cualquier sistema operativo vía navegador web.
- Navegadores web. La aplicación debe poder ser utilizada en todos los navegadores web modernos. Se recomienda como mínimo utilizar Explorer 8, Chrome 25 o Safari 4.
- Dispositivos de entrada. Para poder ingresar datos al sistema, el dispositivo debe contar con dispositivos de entrada como mouse y teclado, o en caso de ser dispositivo móvil debe contar con pantalla táctil y/o teclado alfanumérico.

- Servidor. Se recomienda un servidor de al menos 1 CPU, 1 GB de memoria física, con capacidad mínima de 100 procesos de entrada. Un servidor de menor capacidad podría ocasionar mal funcionamiento del sistema y/o pérdida de datos.
- Hosting. El servicio de hosting recomendado debe incluir ancho de banda ilimitado de manera que las transacciones entre los usuarios y el sistema no se vean interrumpidas o enviadas a cola de procesos.
- Almacenamiento. El almacenamiento mínimo requerido para que el sistema pueda funcionar de manera adecuada es de 100 GB de espacio en disco. Se recomienda un plan de almacenamiento ilimitado para evitar problemas de saturación por un volumen grande de información.
- Base de datos. Se requiere una base de datos de al menos 1 GB de espacio para poder almacenar todos los registros de propiedades y asesores que se requieran.
- Diseño de la aplicación. Los colores de la aplicación deberán ser acordes a la imagen institucional de la empresa. Los logos, nombres o imágenes deberán apearse a los lineamientos de uso y distribución de la empresa.
- Funcionalidad de la aplicación. La aplicación deberá tener un desempeño adecuado en cuanto a utilización de servicios de comunicación entre bases de datos y servicios de actualización o modificación de datos.
- Experiencia de usuario. La interfaz de la aplicación deberá ser de fácil uso de manera que permita al usuario final poder realizar todas las operaciones necesarias rápida y eficazmente.
- Menús y submenús. Los menús y submenús deberán contar con nombres claros y relevantes a la operación que se va a realizar. Deberán ser ubicados de manera que el usuario pueda acceder a ellos en cualquier momento durante toda su experiencia de uso de la aplicación.
- Descarga de imágenes. Las imágenes utilizadas dentro del sistema deberán estar optimizadas para web para que la descarga sea rápida y continua acorde a la conexión de internet mínima requerida.



- Fuentes. Las fuentes utilizadas dentro del sistema deberán ser legibles y contar con un tamaño que permita la lectura adecuadamente, se recomienda un tamaño mínimo de 12px y un color acorde a la imagen institucional de la empresa.
- Secciones. Las secciones deberán estar delimitadas de manera que el usuario pueda identificar de manera clara en qué sección se encuentra.
- Navegación. La navegación del sitio deberá ser intuitiva, clara y coherente para que el usuario pueda recorrer el sitio sin dificultades. Se deberá incluir un mapa de sitio que permita visualizar el árbol de direcciones del sitio.
- Ventanas emergentes. Se recomienda no hacer uso de ventanas emergentes que puedan ser bloqueadas por los navegadores y derive en la pérdida de información o visibilidad para el usuario.
- Multiusuario. El sistema debe ser capaz de dar servicio a múltiples usuarios simultáneos.
- Acceso a la aplicación. Para acceder a la aplicación es necesario un nombre de usuario y contraseña.
- Tipos de usuario. Deberán existir 3 tipos de usuario.
 - o Administrador
 - o Director
 - o Asesor
- Administradores. Los administradores de sistema pueden realizar cualquier cambio de datos de la plataforma.
- Directores. Los directores pueden realizar cambios a los registros de propiedades y asesores asignados a su oficina.
- Asesores. Los asesores pueden realizar cambios únicamente a los registros de propiedades que tenga asignadas.
- Permitir a los administradores poder dar de alta nuevas oficinas que se incorporen al sistema.
- Los administradores podrán dar de baja las oficinas que ya no se quieran activas dentro del sistema. Los datos asociados a la oficina permanecerán almacenados en la aplicación para fines de comparación y estadísticas. Al darse de baja una

oficina, las propiedades y asesores asociados con esa oficina serán puestos como inactivos dentro del sistema de manera automática.

- Las oficinas que se den de baja podrán darse de alta nuevamente. Las propiedades y asesores asociados a esta oficina se reactivarán de manera automática.
- Los administradores podrán agregar o desactivar los perfiles de otros administradores.
- Se deberán agregar los siguientes tipos de propiedad:
 - o Bodega
 - o Casa en condominio
 - o Casa para oficina
 - o Casa
 - o Departamento
 - o Inmueble para productos
 - o Local comercial
 - o Oficina
 - o Terreno
 - o Rancho/ Hacienda
 - o Hotel
 - o Edificio
- Las propiedades deberán poder se catalogadas por la condición en la que se encuentren con la siguiente clasificación:
 - o Excelente
 - o Bueno
 - o Regular
 - o Malo
- Las propiedades se podrán clasificar de acuerdo a la operación que se esté realizando:
 - o Venta
 - o Renta



- Las propiedades deberán incluir los siguientes datos:
 - o Dirección
 - o Precio
 - o Años del inmueble
 - o Medidas
 - o Niveles
 - o Número de Estacionamientos
 - o Número de Cuartos
 - o Número de Baños
- Las oficinas deberán incluir los campos de:
 - o Nombre
 - o Dirección
 - o Teléfono
 - o Correo electrónico
- Los registros de asesores deben contar con los datos de:
 - o Nombre
 - o Teléfono
 - o Dirección
 - o Correo Electrónico
 - o Oficina a la que pertenece
- Modificaciones a propiedades. Las propiedades podrán modificarse mientras se encuentren como activas.
- Modificaciones a asesores. Los asesores podrán actualizar sus datos en cualquier momento.
- Modificaciones a oficinas. Los datos de las oficinas sólo podrán ser modificados por los administradores de la plataforma.
- Búsqueda de propiedades. Las propiedades podrán buscarse por colonia, ciudad, estado, precio, oficina y/o asesor a la que pertenecen, tipo de transacción (venta o renta) y por tipo de operación (abierta, trámite, cerrada).
- Búsqueda de asesores. Los asesores podrán buscarse por nombre y oficina a la que estén asociados.

- Búsqueda de oficinas. Las oficinas podrán buscarse por nombre, ciudad, y estado.
- Reportes. El sistema podrá generar reportes en cuanto a oficinas, propiedades y asesores.
- Reportes de oficinas. Los reportes de oficinas deberán incluir gráficas en donde pueda visualizarse el progreso o retroceso en cuanto a la operación de la oficina. Deberán incluir análisis de ventas y rentas por parte de los asesores así como un análisis del número de propiedades con el que cuenta la oficina.
- Reportes de propiedades. Los reportes de propiedades deberán incluir gráficos donde se indique el número de propiedades y el tipo de operación en el que se encuentran, estadísticas de precios y tipo de transacciones.
- Reportes de asesores. Los reportes deberán incluir gráficas que permitan analizar el desempeño del asesor en cuanto a ventas, rentas y precios de las transacciones realizadas.

3.3 Análisis y posicionamiento de los lenguajes para back-end y front-end.

Back-End (base de datos).

Para la elección de este componente se analizaron tres bases de datos, todas son de distribución y utilización libre con el fin de obtener bases suficientes que nos permitan elegir alguno como la mejor opción para la instalación y ejecución del proyecto SIAH.

PostgreSQL



Figura 3.3.1 Logotipo de PostgreSQL

Su logotipo es como se muestra en la figura 3.3.1. PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

MySQL



Figura 3.3.2 Logotipo de MySQL

Su logotipo es como se muestra en la figura 3.3.2. Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

Es muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.



El lenguaje de programación que utiliza MySQL es **SQL** (Structured Query Language) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

MySQL es veloz, multi-hilo, multiusuario y robusto.

Firebird



Figura 3.3.3 Logotipo de Firebird

Su logotipo es como se muestra en la figura 3.3.3. Deriva del código fuente de Interbase 6.0 de Borland. Es open source y no hay licencias duales. Tanto para uso comercial como para aplicaciones open source, es totalmente libre. La tecnología de Firebird lleva 20 años funcionando, esto hace que sea un producto muy maduro y estable.

Es un software de administración de una base de datos relacional, similar en propósito a productos tales como DB2; Oracle o PostgreSQL. Tiene dos componentes principales el servidor de base de datos, que vive en la misma máquina que la base de datos, y la interface de aplicación, comúnmente referida como la librería del cliente.

La tabla 3.3.1 muestra la comparación de algunas características de las tres bases de datos anteriormente mencionadas.

CARACTERISTICAS	FIREBIRD	MYSQL	POSTGRESQL
Licenciamiento libre	Sí	Sí (Modelo dual)	Sí
Plataformas	Larga lista de plataformas	Larga lista de plataformas	Larga lista de plataformas
Soporte de 64 bits	Sí	Sí	Sí
Limite por base de datos	Limitado	Ilimitado	Ilimitado
Limite por tabla	Limitados	16 TB en Solaris	32 TB
Número de índices por tabla	Limitados	64	Ilimitados
Número máximo de decimales que puede manejar en tipos decimales	18	65	131072
Límites numéricos	Enteros 64 bits	Enteros 64 bits, 38 dígitos para numéricos y decimales	Precisión de 1000 dígitos
Límites de cadenas (VARCHAR)	32767 bytes	65,535 bytes	1 Gb
Operación de concatenación	(doble pipe)	CONCAT()	(doble pipe)
Carácter delimitador	N/A	"	N/A
Precisión en el tipo TIMESTAMP	4 dígitos	6 dígitos	14 dígitos
Tipos BINARY y VARBINARY	No	Sí	Sólo binary



Soporte de unicode (UTF-8)	Sí (también UTF-16)	Sí	Sí
Transacciones (conformidad ACID)	Sí	Sí	Sí
Candados a nivel renglón de tabla (conurrencia)	Modelo MVCC	Modelo MVCC solo para tablas InnoDB	Modelo MVCC
Funcionalidad de TIMEOUT	No	No	Sí
Soporte de stores procedures	Sí	Sí, con restricciones	Sí
Soporte de triggers	Sí	Sí, con restricciones	Sí
Escalabilidad/soporte SMP	Sólo en el classic Server	Sí	Sí
Balanceo de cargas	No	Parcial	Sí
Replicación	Sí	Sí	Sí
Respaldo en línea (Online Backup)	Soportada con comando en línea	Soportado con la utilidad Mysqldump	Puede realizar backups continuos

Tabla 3.3.1 Manejadores de bases de datos.

Front-end (lenguaje de programación y manejador sitio web).

C Sharp



Figura 3.3.4 Logotipo de C#

Su logotipo es como se muestra en la figura 3.3.4. Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la **ECMA** (European Computer Manufacturers Association) (ECMA-334) e **ISO** (International Organization for Standardization) (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés sharp) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de cuatro signos '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono -



DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

PHP



Figura 3.3.5 Logotipo de PHP

Su logotipo es como se muestra en la figura 3.3.5. Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

Fue creado originalmente por Rasmus Lerdorf en 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la licencia pública general de GNU debido a las restricciones del uso del término PHP.

Java

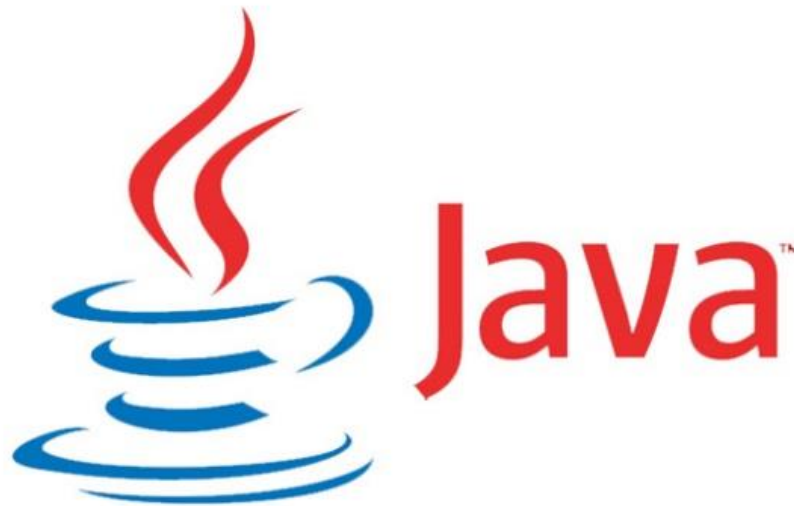


Figura 3.3.6 Logotipo de Java

Su logotipo es como se muestra en la figura 3.3.6. Es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.



El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales, y librerías de clases en 1991 y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento con las especificaciones del proceso de la comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la licencia pública general de GNU. Otros también han desarrollado implementaciones alternas a estas tecnologías de Sun, tales como el compilador de Java de GNU y el GNU classpath.

La Tabla 3.3.2 muestra la comparación de algunas características de los tres lenguajes de programación anteriormente mencionados.

CARACTERISTICAS	PHP	C Sharp	Java
Creador	Libre	Microsoft	Sun
Programación orientada a objetos	Sí	Sí	Sí
Conectividad con base de datos	Todas	Todas	Todas
Herramienta de generación de SQL	Sí	Sí	Sí

Gestión de excepciones	Sí	Sí	Sí
Asistencia del IDE	Muchas limitaciones, instalación de plugins para IDE's no nativos. Inexistencia de IDE's nativos pre-construidos.	IDE gratuito y con mantenimiento mucha asistencia.	Muchos IDE's con mantenimiento y asistencia, múltiples foros y comunidades
Uso asíncrono	No está pensado como un lenguaje de procesamiento asíncrono, no explota el multi-threading del procesador	Escritura tediosa de funciones	Existencia de librerías para uso asíncrono
Paquetes adicionales disponibles	pear, pecl	nuget	Maven
Posibilidades de optimización a nivel de ejecución	Bajas	Altas con el uso de templates	Optimización basada en la máquina virtual de cada sistema.
Plataforma de ejecución	Windows, Linux, Solaris	Sólo Windows	Múltiples

Tabla 3.3.2 Lenguajes de Programación.

Apache

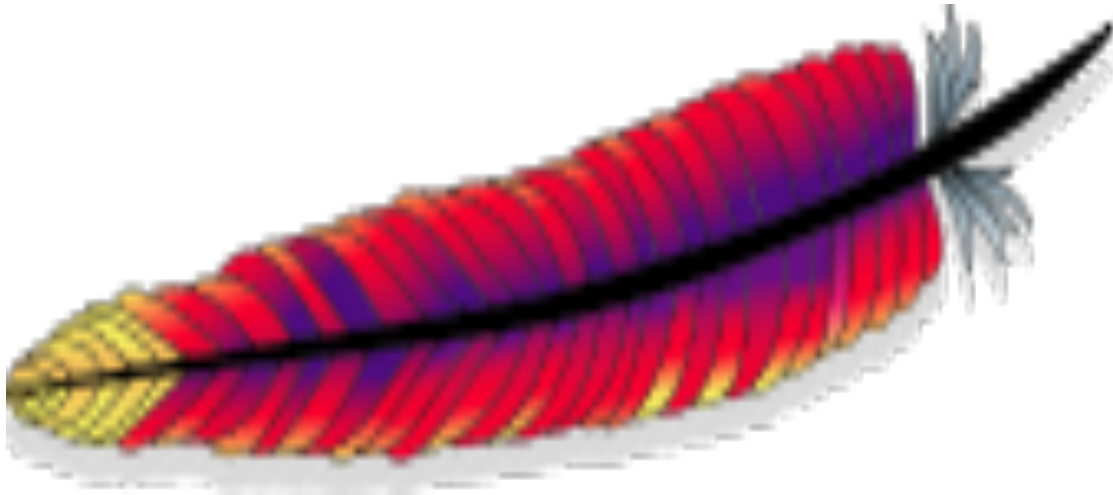


Figura 3.3.7 Logotipo de Apache

Su logotipo es como se muestra en la Figura 3.3.7. El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

NGINX



Figura 3.3.8 Logotipo de Nginx

Su logotipo es como se muestra en la figura 3.3.8. **Nginx** (pronunciado en inglés “engine X”) es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3).

Es software libre y de código abierto, licenciado bajo la licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows.

El sistema es usado por una larga lista de sitios web conocidos, como: WordPress, Netflix, Hulu, GitHub, Ohloh, SourceForge, TorrentReactor y partes de Facebook (como el servidor de descarga de archivos zip pesados).



Es el segundo servidor web más usado en dominios activos (14,35%) superando a Microsoft Information Server. Además, pasó la marca de ser usado en más de 100 millones de sitios.

LIGHTTPD



Figura 3.3.9 Logotipo de Lighttpd

Su logotipo es como se muestra en la figura 3.3.9. **Lighttpd** (pronunciado lighty) es un servidor web diseñado para ser rápido, seguro, flexible, y fiel a los estándares.

Está optimizado para entornos donde la velocidad es muy importante, y por eso consume menos CPU y memoria RAM que otros servidores. Por todo lo que ofrece, es apropiado para cualquier servidor que tenga problemas de carga.

Es software libre y se distribuye bajo la licencia BSD. Funciona en GNU/Linux y UNIX de forma oficial. Para Microsoft Windows actualmente hay una distribución conocida como Lighttpd For Windows mantenida por Kevin Worthington.

Permite comunicarse con programas externos mediante FastCGI o SCGI, que son mejoras al CGI original (también soportado). De esta forma, se pueden usar programas en prácticamente cualquier lenguaje de programación.

Tiene una importancia especial PHP, para el que se han hecho mejoras específicas. También es habitual combinarlo con Ruby on Rails.

La tabla 3.3.3 muestra la comparación de algunas características de los tres manejadores de sitio web anteriormente mencionados.

CARACTERÍSTICAS	APACHE	NGINX	LIGHTTPD
Servidor basado en	Procesos	Eventos	Eventos
Ejecuta funciones de servidor proxy	No	Si	No
Ejecución de código	estático/dinámico	estático/dinámico	estático/dinámico
Cantidad de componentes	Muchos	Medio	Pocos
Trabajo asíncrono	No	Si	Si
SSL/TLS	SI	SI	Si
Virtual host	SI	SI	Si
Balanceo de cargas	SI	SI	No
Compresión	SI	SI	Si
Control de acceso	Si	Si	Si
Reescritura de	Si, desde .htaccess	Si	No (sólo vía config)



URL			
Personalización de log	Si	Si	No
Transmisión de FLV	Si	Si	Si
Ejecución de FastCGI	Si	Si	Si
Estable	Muy estable	Estable	Estable
Popularidad	Muy popular	poca popularidad	poca popularidad
Licenciamiento	Apache Licence 2.0	Simple BSD	Revised BSD
Uso de memoria RAM	Muy alta	Baja	Baja

Tabla 3.3.3 Servidores Web.

Por lo que podemos concluir con respecto al entorno de desarrollo y al manejador de base de datos, las herramientas que utilizaremos serán PHP y MySQL, porque son software de uso libre y no se requieren licencias para su empleo, mismos que son compatibles entre sí.

Para una mejor justificación, estas son las razones por las que se eligió MySQL:

- Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro y de buscadores de aplicaciones.

Las razones por las que se eligió PHP:

- Independencia de plataforma.
- Portabilidad.
- Facilidad de instalación, configuración y utilización.
- Soporte para una gran cantidad de bases de datos.
- Tiene una mejor interacción con MySQL.

Las razones por las que se eligió Apache:

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/soporte)
- Ejecuta código PHP
- Portabilidad

Y sobre todo, sabiendo que este trío trabaja de la mejor forma PHP, MySQL y Apache.



3.4 Aplicación y análisis modular con base en la metodología iterativa-incremental

El SIAH consistirá de varios módulos, que interrelacionados entre sí nos da pauta a un sistema de gran relevancia en el área administrativa. El sistema se encontrará alojado en un hosting, para ello será necesaria una clave de usuario y un password; de otra manera no habría restricción al sitio y la seguridad estaría limitada, situación no conveniente a ningún sitio, una vez que se tenga el acceso aparecerá un menú con cuatro módulos principales:

- Oficinas
- Propiedades
- Usuarios
- Reportes

La estructura aparecerá como se muestra en la figura 3.4.1:

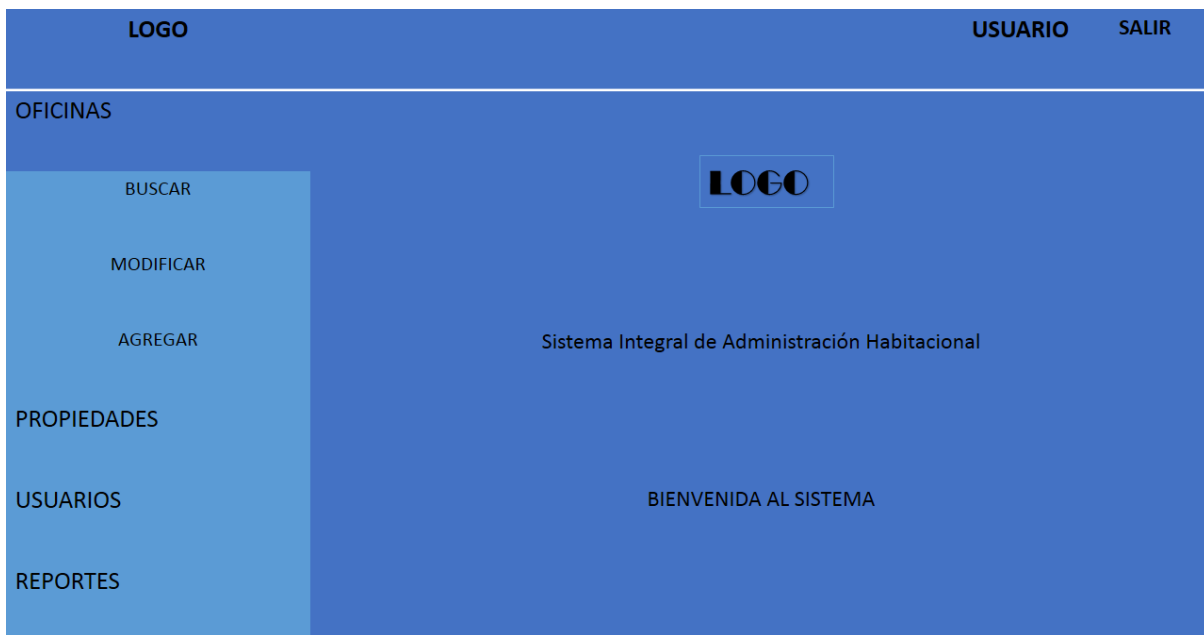


Figura 3.4.1 Formato de la pantalla principal del sistema

En caso de que el acceso sea negado después de tres oportunidades, el usuario no tendrá oportunidad de acceder a este, los módulos principales serán:

- Oficinas
- Propiedades
- Usuarios
- Reportes

Los tres primeros módulos contarán con las opciones de

- Buscar
- Modificar
- Agregar

La cuarta opción

- Ventas
- Rentas
- Oficina
- Asesores

El árbol es como se muestra en la tabla 3.4.2:

Sistema Integral de Administración Habitacional

Oficinas
▪ Buscar
▪ Modificar
▪ Agregar
Propiedades
▪ Buscar
▪ Modificar
▪ Agregar
Usuarios
▪ Buscar
▪ Modificar
▪ Agregar
Reportes
▪ Ventas
▪ Rentas
▪ Oficina
▪ Asesores

Tabla 3.4.2 Estructura del SIAH.



En el primer módulo de “oficinas” se tendrá la posibilidad de buscar, modificar o agregar; en la opción de “buscar” se tendrá la posibilidad de hacerlo mediante nombre, ciudad o estado.

Búsqueda de oficinas

- Nombre
- Ciudad
- Estado

En la opción de modificar se necesitará hacerlo mediante la oficina, al hacerlo se desplegarán todas las oficinas registradas al momento, se selecciona la oficina a afectar, en ese momento se desplegarán los datos que se encuentra ya precargados, el cambio la hacemos donde sea necesario, después de ello tenemos la posibilidad de aceptar los cambios o cancelarlos; como se muestra en la Figura 3.4.3:

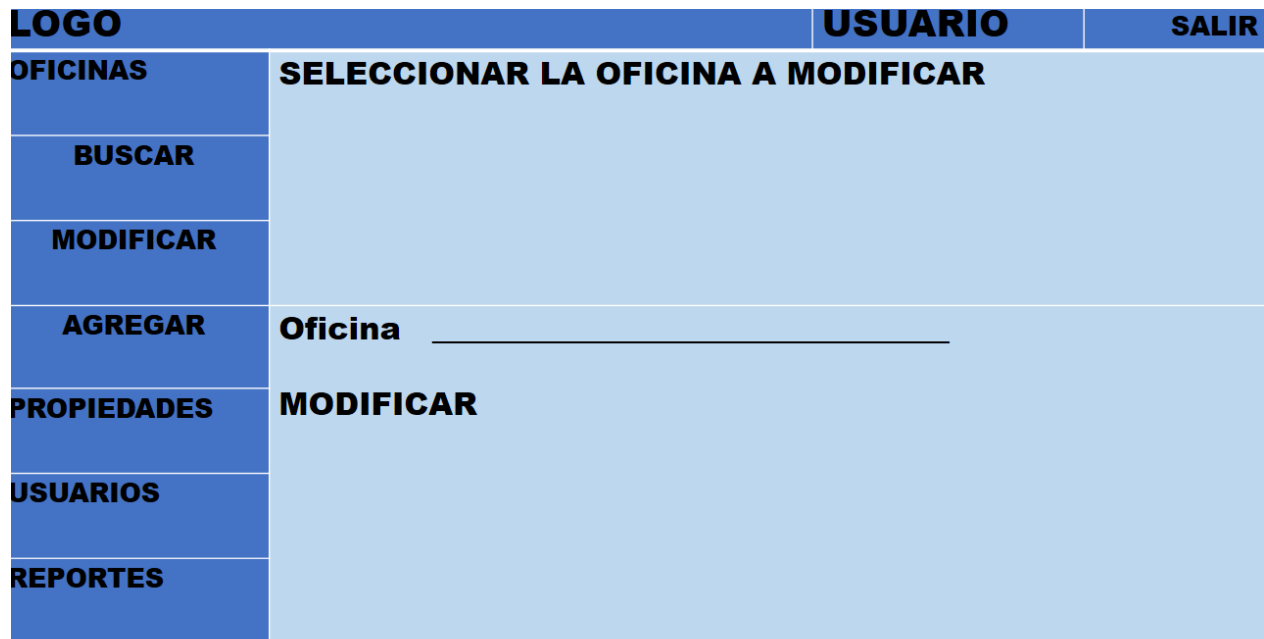


Figura 3.4.3 Opción de Modificar la Oficina

Enseguida desplegará los datos de la oficina, como se muestra en la Figura 3.4.4:

LOGO		USUARIO	SALIR
OFICINAS	Despliega Dato1	Despliega Dato8	
BUSCAR	Despliega Dato2	Despliega Dato19	
MODIFICAR	Despliega Dato3	Despliega Dato10	
AGREGAR	Despliega Dato4	Despliega Dato11	
PROPIEDADES	Despliega Dato5	Despliega Dato12	
USURAIOS	Despliega Dato6	...	
REPORTES	Despliega Dato7	Despliega Dato1N	

Figura 3. 4.4 Desplegado de datos en la modificación de oficina

En la opción de agregar se desplegarán los datos necesarios en el registro de oficina, los cuales debe ser llenados uno a uno, en el caso de que la oficina ya exista, se desplegará un mensaje de oficina ya registrada; como se muestra en la figura 3.4.5:

LOGO		USUARIO	SALIR
OFICINAS	Solicita Dato1	Solicita Dato8	
BUSCAR	Solicita Dato2	Solicita Dato19	
MODIFICAR	Solicita Dato3	Solicita Dato10	
AGREGAR	Solicita Dato4	Solicita Dato11	
PROPIEDADES	Solicita Dato5	Solicita Dato12	
USURAIOS	Solicita Dato6	...	
REPORTES	Solicita Dato7	Solicita Dato1N	

Figura 3.4.5 Solicita datos a ser llenados para la oficina



El siguiente módulo es el de “propiedades”, que al igual que el de oficina conllevan las opciones de “buscar”, “modificar” o “agregar”; para hacer la búsqueda de la propiedad será necesario indicar que tipo de propiedad se está buscando:

- Bodega
- Casa en condominio
- Casa para oficina
- Casa
- Departamento
- Inmueble para productos
- Local comercial
- Oficina
- Terreno
- Rancho / Hacienda
- Hotel
- Edificio

En la opción de modificar la propiedad será necesario indicar que oficina la tiene registrada, además del asesor; como se muestra en la figura 3.4.6:

LOGO		USUARIO	SALIR
OFICINAS	OFICINA	_____	
PROPIEDADES	ASESOR	_____	
BUSCAR			
MODIFICAR	DIRECCIÓN	_____	
AGREGAR			
USUARIOS	MODIFICAR	_____	
REPORTES			

Figura 3.4.6 Opción de “modificar” la propiedad

Una vez que se selecciona la propiedad los datos contenidos son desplegados, pudiendo ser modificados, al terminar es necesario grabar los cambios hechos o descartarlos, que el proceso es parecido al caso de la oficina.

En la siguiente opción se tiene la posibilidad de agregar la propiedad, que al igual que en la opción de “oficinas” serán necesarios algunos datos a ser llenados para la tabla correspondiente. La siguiente opción corresponde a “usuarios”; que igual que las opciones anteriores, oficinas y propiedades, se contará con la posibilidad de buscar, modificar o agregar, para el caso de búsqueda en usuarios será necesario digitar el nombre así como la oficina correspondiente, puede presentarse el caso que un usuario se encuentre en más de una oficina. Para modificar un usuario se presentarán dos opciones: nombre y asesor, usar el nombre será la opción más directa de hacer la búsqueda; la otra será mediante un asesor, este tendrá una relación directa, o lo que es lo mismo, en la cartera de clientes del asesor se encontrará el usuario, al efectuar la selección se hace el filtro y los datos correspondientes son desplegados; así, solo se modifican los campos correspondientes para después aceptar la operación o hacer la cancelación de la operación. En el caso de agregar un usuario aparecerá la máscara con los datos necesarios, en caso de que el usuario ya se encuentre registrado se enviará un mensaje indicando que el usuario ya se encuentra registrado, en caso contrario se llenará para ser vaciado a la tabla correspondiente.

En cualquier organización la parte estadística es de vital importancia, ello ayuda a la toma de decisiones, el SIAH no es la excepción, contará con una serie de estadísticos que conforman una ayuda única para la toma de decisiones, los estadísticos son los siguientes:

- Ventas
- Rentas
- Oficina
- Asesores



Entre los estadísticos que se encontrarán en la parte de “ventas” están:

- Precio de venta
- Años de la propiedad
- Tamaño de la propiedad
- Número de metros construidos
- Número de niveles
- Número de cuartos
- Número de baños
- Número de estacionamientos
- Condiciones del inmueble

En lo referente a “rentas” se contará con la misma estructura que en ventas: precio de venta, años de la propiedad, tamaño de la propiedad, número de metros construidos, número de niveles, número de cuartos, número de baños, número de estacionamientos y condiciones de Inmueble, Con todo ello los reportes estadísticos de rentas contarán con todo lo necesario en la toma de decisiones. En la parte de “oficinas” habrá que seleccionar la oficina para poder generar el estadístico, los puntos a tratar son los mismos que el de ventas y el de rentas; por lo que no será necesario detallarlos nuevamente. Por otra parte en lo referente a los reportes de asesores será necesario seleccionar la oficina y el asesor, con la generación de los estadísticos de los mismos puntos que se trataron en ventas, rentas y oficinas, de lo anterior se desprende todo lo referente a la aplicación y análisis modular del sistema.

3.5 Diseño modular de la interfaz usuario

El término resolver un problema significa, a grandes rasgos, llevar a cabo todo el proceso de definición del problema y desarrollo de una solución, a modo de programa, que lo resuelva.

Diseño modular

Una vez que los requerimientos generales y particulares del sistema SIAH han sido establecidos, ya se puede iniciar la fase de diseño. En esta etapa se tiene que encontrar una solución informática al problema planteado. Dicha resolución determinará cómo se va a resolver el problema y para encontrarla, hemos de hacer uso del diseño modular.

En la fase de diseño del ciclo de vida de un programa, la solución a un problema suele venir dada por un programa representado por un módulo principal, el cual se descompone en subprogramas (módulos), los cuales, a su vez, también se pueden fraccionar, y así sucesivamente, es decir, el problema se resuelve de arriba hacia abajo. A este método se le denomina diseño modular o descendente (top-down).

Dado un problema a resolver, en primer lugar hay que estudiar la posibilidad de dividirlo en otros más pequeños, llamados subproblemas (este método es conocido como "divide y vencerás"). Cada uno de ellos puede tratarse de manera aislada; por tanto, la complejidad global del problema disminuirá considerablemente.

Identificación de los posibles interfaces

El SIAH se definirá por los siguientes módulos para su funcionamiento operacional:

- Autenticación al sistema.
- Pantalla principal del sistema SIAH
- Módulo de oficinas.



- Módulo de propiedades.
- Módulo de usuarios.
- Módulo de reportes.

Diseño de la página de autenticación al sistema

El primer módulo debe de estar definido para la autenticación del usuario. Aquí se debe de proporcionar una pantalla para que el usuario pueda ingresar su información de autenticación:

- Identificación
- Contraseña.

El sistema debe de validar la información ingresada y en caso de ser inválida deberá mandar un mensaje de error; en caso de ser información valida el sistema deberá desplegar en el menú los módulos correspondientes al perfil asignado al usuario, como se muestra en la figura 3.5.1



Figura 3.5.1 Autenticación al sistema

Pantalla principal del sistema SIAH

La pantalla principal del sistema SIAH se plantea de la siguiente forma como se muestra en la figura 3.5.2.

El cual contendrá:

1. Logotipo el sistema.
2. Identificación de usuario que ingreso al sistema, este apartado contendrá el nombre del usuario que ingrese.
3. Botón de salir, el cual cerrará el sistema.



4. Menús del sistema: oficinas, propiedades, usuarios y reportes; los cuales a la vez están conformados por submenús.
5. Cuadro de resultados, en caso de la pantalla principal este mostrará un mensaje de bienvenida así como la información del sistema SIAH.

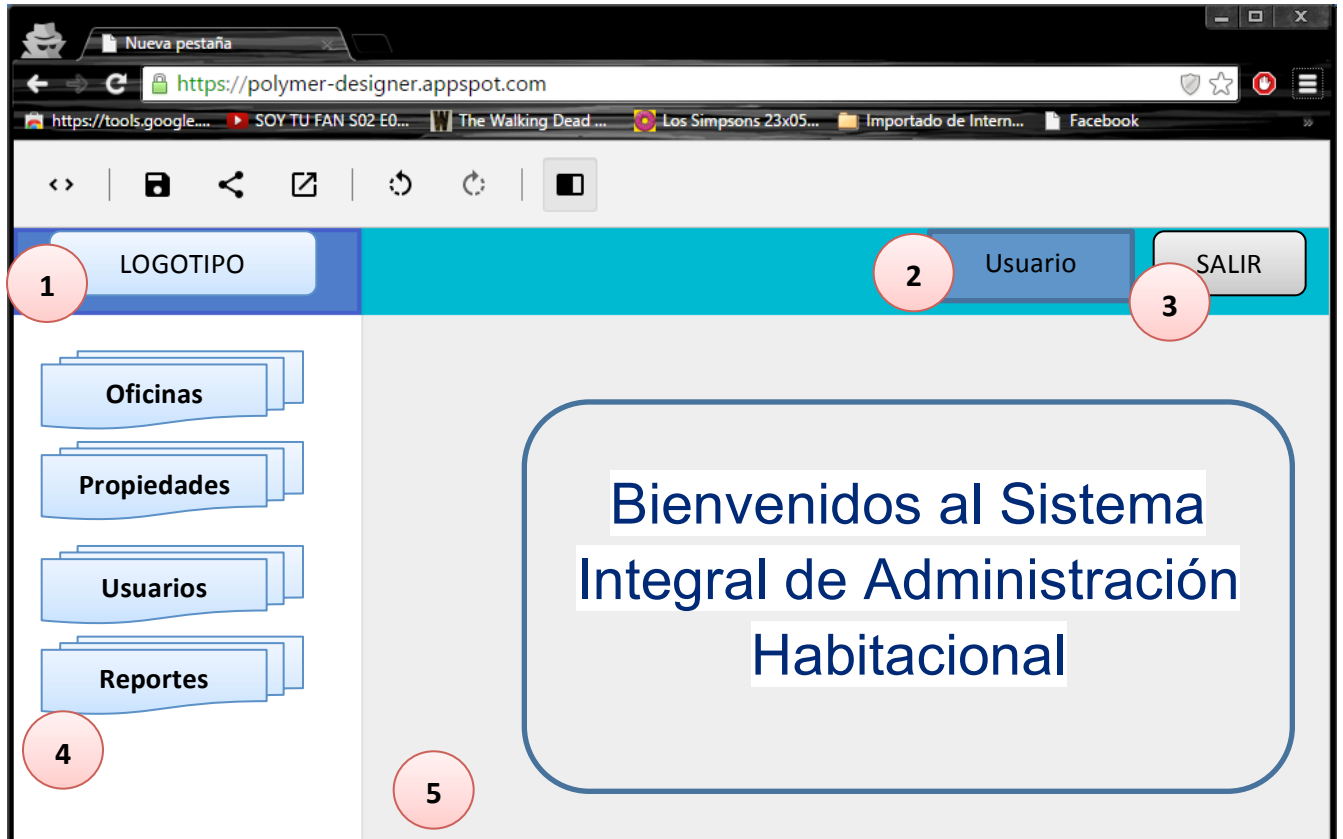


Figura 3.5.2 Pantalla principal del sistema SIAH

Submenú oficinas

El submenú como se muestra en la figura 3.5.3 desplegará información de las oficinas, el cual estará compuesto de los siguientes apartados:

- Buscar: el submenú desplegará un formulario de búsqueda en el cual realizará una búsqueda de oficinas de acuerdo a los parámetros ingresados.
- Modificar: el submenú desplegará las oficinas y dará la opción de modificar los campos de cada una.

- Agregar: el submenú nos dará la opción de ingresar nuevas oficinas, desplegando un formulario el cual se ingresarán los datos de la nueva oficina.

En el cuadro de resultados se mostrará la acción de acuerdo al submenú que se haya elegido.

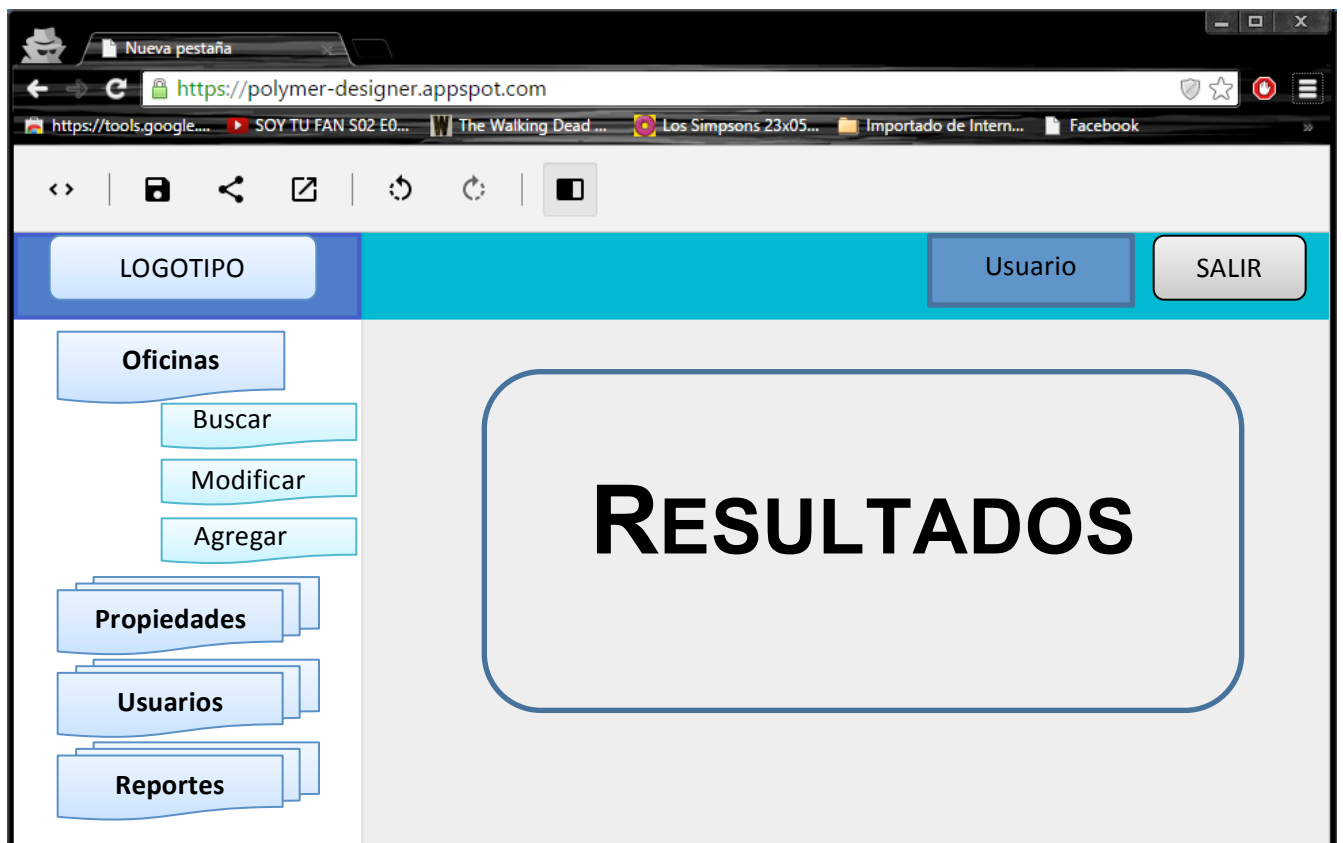


Figura 3.5.3 Submenú oficinas del sistema SIAH

Submenú propiedades

El submenú propiedades desplegará información de las propiedades registradas en el sistema SIAH como se muestra en la figura 3.5.4, el cual estará compuesto de los siguientes apartados:



- **Buscar:** el submenú desplegará un formulario de búsqueda el cual realizará una búsqueda de las propiedades de acuerdo a los parámetros ingresados.
- **Modificar:** el submenú desplegará las propiedades y dará la opción de modificar los campos de cada una.
- **Agregar:** el submenú dará la opción de ingresar nuevas propiedades, desplegando un formulario el cual se ingresarán los datos de la nueva propiedad.

En el cuadro de resultados se mostrará la acción de acuerdo al submenú que se haya elegido.

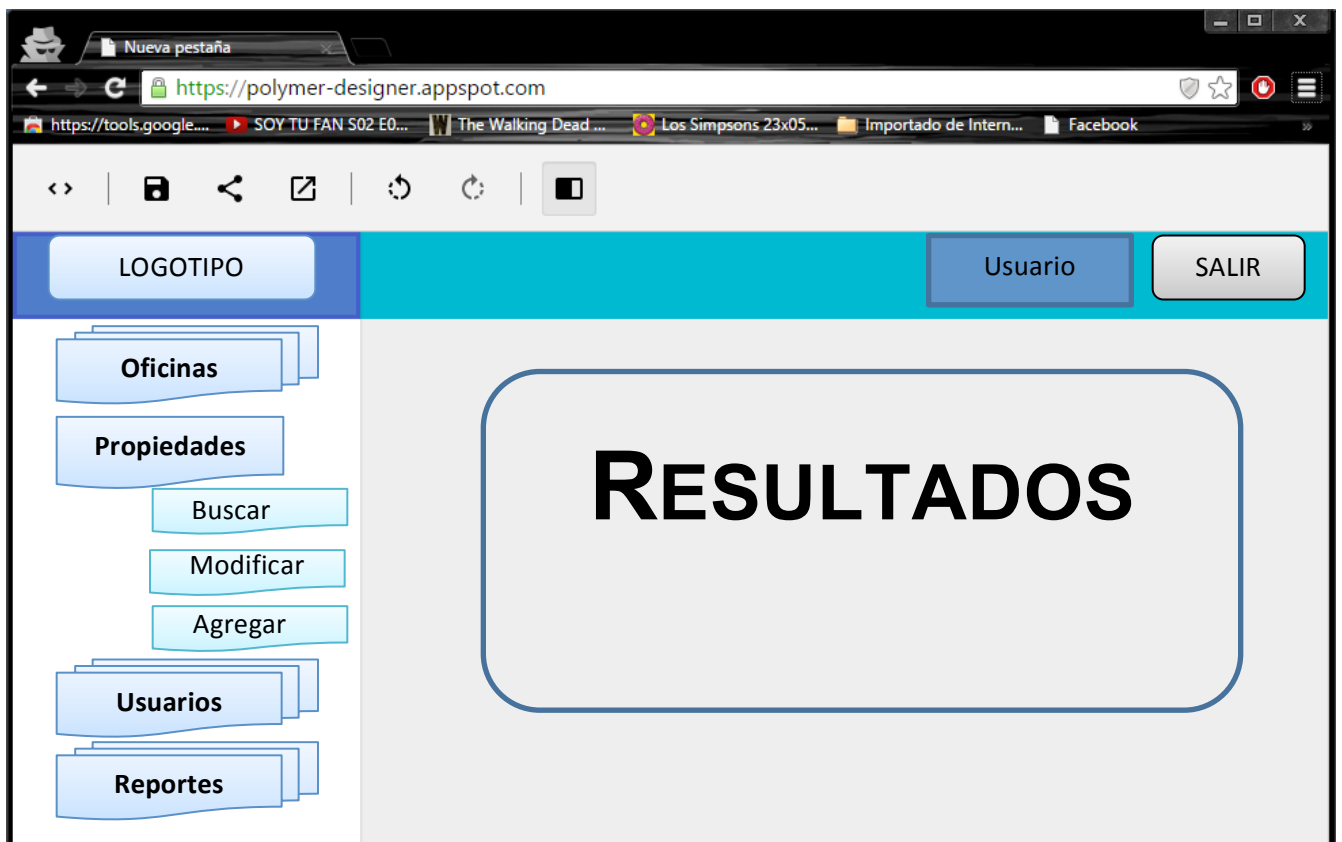


Figura 3.5.4 Submenú propiedades del sistema SIAH

Submenú usuarios

El submenú Usuarios como se muestra en la figura 3.5.5, estará compuesto de los siguientes apartados:

- **Buscar:** el submenú desplegará un formulario de búsqueda en el cual realizará una búsqueda del usuario de acuerdo a los parámetros ingresados.
- **Modificar:** el submenú desplegará los usuarios y dará la opción de modificar las características de los mismos.
- **Agregar:** el submenú mostrará la opción de ingresar nuevos usuarios, desplegando un formulario el cual se ingresarán los datos del nuevo usuario.

En el cuadro de resultados se mostrará la acción de acuerdo al submenú que se haya elegido.

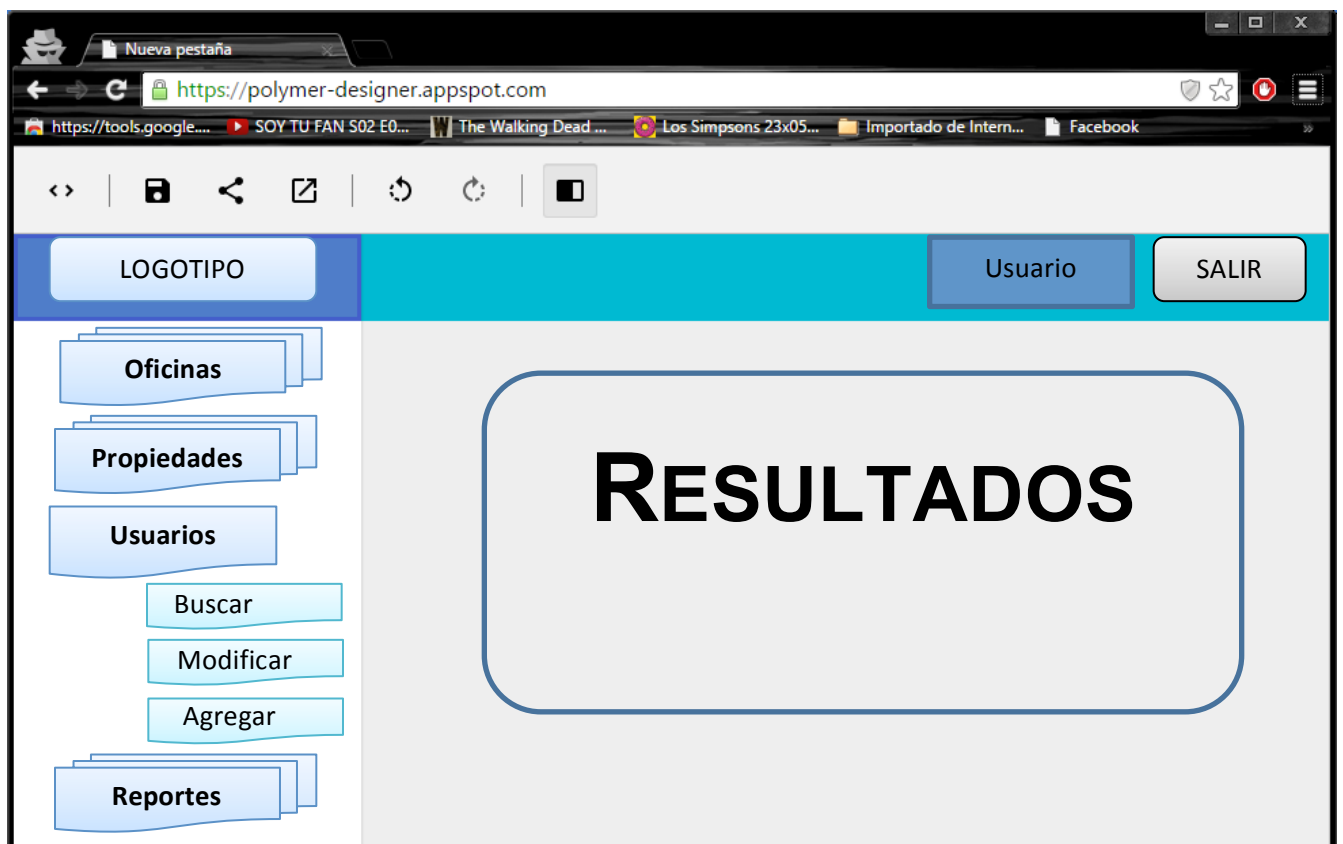


Figura 3.5.5 Submenú usuarios del sistema SIAH



Submenú reportes

El submenú reportes es de gran importancia, ya que aquí se desplegará toda la información en gráficos como se muestra en la figura 3.5.6, esta información se obtendrá de acuerdo a las ventas o rentas de los asesores de cada oficina.

El submenú “reportes” está compuesto de los siguientes apartados:

- Ventas: el submenú de ventas desplegará gráficos de las ventas obtenidas.
- Rentas: el submenú rentas desplegará gráficos de acuerdo a las rentas obtenidas.
- Oficinas: el submenú oficinas desplegará gráficos de acuerdo a los resultados obtenidos en cada oficina.
- Asesores: el submenú asesores mostrará reportes de los resultados de cada asesor.

En el cuadro de resultados se mostrará la acción de acuerdo al submenú que se haya elegido.

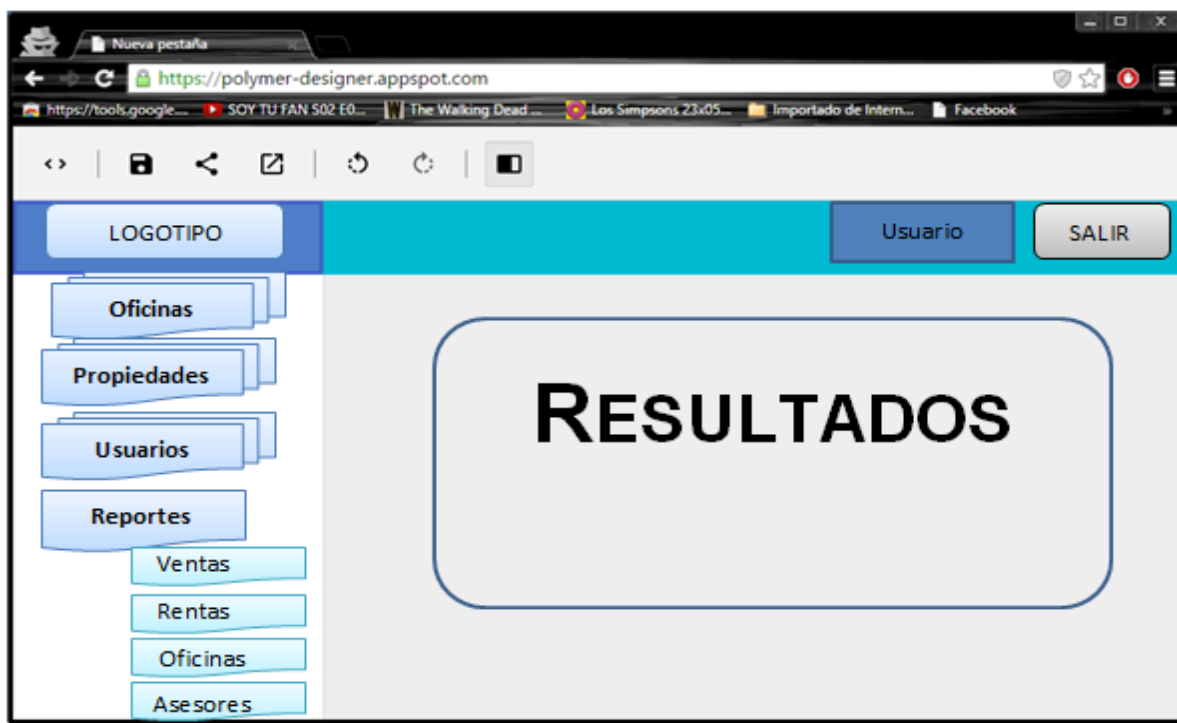


Figura 3.5.6 Submenú reportes del sistema SIAH.

CAPÍTULO IV. DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

4.1 Modelado del sistema.

Introducción al modelado.

Por modelo habremos de entender, que es la descripción analógica para ayudar a visualizar algo que no se puede observar directamente, y que se realiza con un propósito determinado y se destina a un público específico. Es decir, el modelo no es aquello que se quiere observar, sino una representación simplificada; el propósito o la perspectiva es lo que determina para qué realizamos el modelo, y se dirige a un público específico ya que hay un público potencial que es quién va a usar el modelo en cuestión.

La finalidad última de un modelo es la comunicación de algo: un proyecto, un sistema, un concepto, la descripción física de algún elemento, etc. Precisamente, por esta necesidad de comunicar y porque, además, se destina a un determinado público, y con cierto propósito, un modelo es, en definitiva, una abstracción. Mediante la técnica de abstracción se simplifica una realidad compleja, priorizando lo que realmente se quiere comunicar y evitando datos innecesarios para el modelado.

Modelos de software.

Los modelos de software tienen los siguientes elementos:

- Descripción analógica: el modelo no es el sistema en sí, sino su representación.
- El software nunca puede ser observado directamente porque es intangible e invisible por su propia naturaleza; de hecho los modelos son la única forma de observar el mismo.
- Un modelo de software puede servir, entre otras cosas, para construir una aplicación todavía inexistente, para validar conceptos con otros interesados en el



desarrollo o para documentar un programa existente con el fin de facilitar su entendimiento.

- Al destinarse a un determinado público, puede ser un modelo para usuarios finales, para analistas, para desarrolladores, para testers, etc.

Por lo tanto, un mismo sistema puede tener varios modelos, que dependen del propósito y del público al que se dirige.

Y en efecto, el modelo más detallado de un producto de software es el código fuente, sin embargo, esto no sirve para concebirlo antes de la construcción, ni para entender sus aspectos más ocultos con vistas al mantenimiento.

Sin embargo, en el software el modelado es aún más importante que en otras ingenierías. Esto tiene varias razones de ser:

- El software es invisible e intangible, sólo se ve su comportamiento, y sus efectos en el medio.
- El software es mucho más modificable que otros productos realizados en la industria, y por la naturaleza misma del hombre esto incentiva que haya peticiones para muchas más modificaciones.
- El software se desarrolla por proyectos únicos, esto hace que cada vez que construyamos un producto de software estemos enfrentándonos a un problema nuevo.
- El software es sustancialmente complejo, con cientos o miles de partes interactuando, diferentes entre sí, y que pueden ir cambiando de estados a lo largo de su vida; esto hace que analizar un producto de software requiera mecanismos de abstracción y de un lenguaje para representarlo.
- El desarrollo de software es inherentemente complejo a sus componentes, es decir, que la complejidad del producto lleva a la complejidad del proyecto, a la complejidad del equipo de desarrollo, y a la complejidad de la administración del proyecto.

Por lo tanto para producir software que cumpla su propósito, los requisitos del sistema deben estar claros, con este fin se trabaja activamente con los usuarios haciéndolos participar para que no queden requerimientos sin atender. Para conseguir un software de calidad, que sea duradero y fácil de mantener hay que idear una sólida base arquitectónica que sea flexible al cambio. Para desarrollar software de forma rápida y eficientemente, minimizando el trabajo de recodificación y evitando creación de código inútil, hay que disponer, de gente, herramientas necesarias y lo más importante un enfoque apropiado.

Para conseguir, desarrollar software de manera profesional y se obtenga un producto de calidad, es completamente necesario seguir una metodología y no abordar los problemas de manera somera, con el fin de obtener un modelo que represente bien la solución al problema. El modelado es la espina dorsal del desarrollo de software de calidad. Se construyen modelos para explicar el comportamiento del sistema a desarrollar, para comprender mejor ese sistema, para controlar el riesgo y en definitiva para poder atacar problemas que sin el modelado su resolución sería imposible, tanto desde el punto de vista de los desarrolladores (no se pueden cumplir los plazos estimados, no se consigue ajustar los presupuestos...) como desde el punto de vista del cliente, el cual, si finalmente se le entrega el producto del desarrollo, se encontrará con infinidad de problemas, desde que no se cumplen las especificaciones hasta fallos que dejan inutilizado el sistema.

El modelado del sistema

El modelo proporciona los planos de un sistema, desde los más generales, que proporcionan una visión general del sistema, hasta los más detallados. En un modelo se han de incluir los elementos que tengan más relevancia y omitir los que no son interesantes para el nivel de abstracción que se ha elegido. El modelado permitirá plantear los siguientes objetivos:

- Ayudan a visualizar cómo es o cómo será el sistema.
- Especificar la estructura o el comportamiento de un sistema.



- Obtener las plantillas que guían en la construcción de un sistema.
- Documentar las decisiones que se adoptan.

Un modelo representa a un sistema de software desde una perspectiva específica. Cada diagrama permite observar en un aspecto distinto el sistema.

A través del modelado se plantearán los siguientes objetivos:

- Visualizar cómo es o cómo será el sistema.
- Especificar la estructura o el comportamiento del sistema.
- Obtener las plantillas que guían la construcción del sistema.
- Documentar las decisiones adoptadas.

Cada diagrama permite observar un aspecto distinto el sistema. Por lo tanto, nuestro sistema estará documentado por medio de los diagramas que proporciona la metodología iterativa-incremental que ya se ha descrito con anterioridad.

Los diagramas de iterativa-incremental con los que vamos a trabajar son los siguientes:

- Diagrama de contexto.
- Diagrama entidad-relación.
- Diagrama de flujo de datos y procesos.
- Diccionario de datos.
- Normalización de la base de datos.

4.1.1 Diagrama de contexto del sistema (SIAH).

Un diagrama de flujo de datos especial es el diagrama de contexto del sistema, representa el sistema de forma global, sólo pueden aparecer entidades externas, flujos de datos y un único proceso que representa el sistema en su conjunto. Pueden aparecer almacenes de datos cuando son compartidos entre nuestro sistema y el exterior. Un modelo de flujo de datos puede aplicarse a cualquier sistema basado en computadora independientemente del tamaño o complejidad.

El sistema acepta entradas de distintas formas; aplica un hardware, software y elementos humanos para transformar la entrada en salida; y produce una salida en distintas formas. La entrada puede ser una señal de control transmitida por un transductor, una serie de números escritos por un operador humano, un paquete de información transmitido por un enlace a red, o un voluminoso archivo de datos almacenado en memoria secundaria. La transformación puede comprender una sencilla comparación lógica, un complejo algoritmo numérico, o un método de inferencia basado en la regla de un sistema experto. La salida puede encender un sencillo led o producir un informe de 200 páginas. Un ejemplo de lo descrito anteriormente como se muestra en la Figura 4.1.1.1.

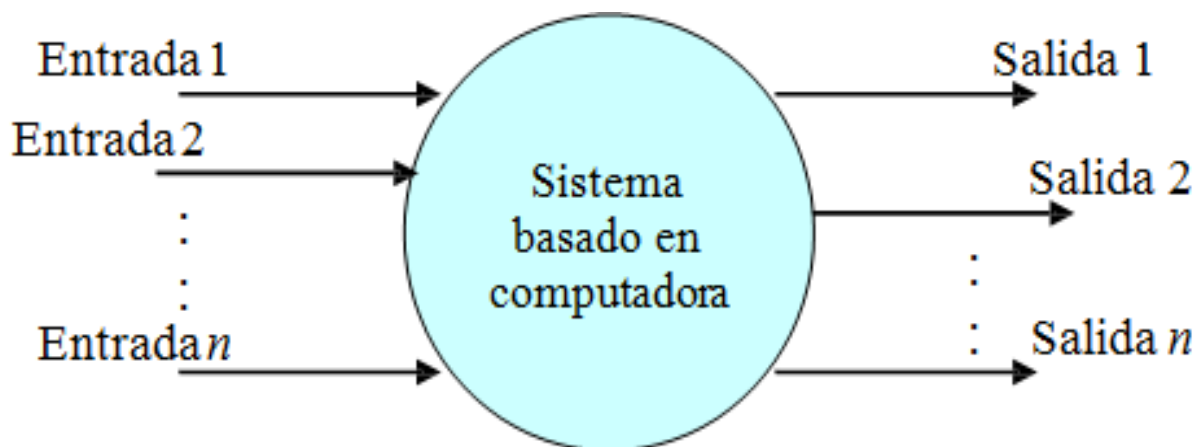


Figura 4.1.1.1 Cualquier sistema computacional se puede representar mediante un diagrama de contexto.



Teniendo en cuenta que este diagrama debe de ser comprensible, no es posible representar todos los flujos de datos del sistema en él, sino más bien debe representarse en él una visión general del sistema desde la perspectiva de los propietarios de sistemas siguiendo dos lineamientos básicos:

- Representar únicamente los flujos de datos que tengan algo que ver con el objetivo principal del sistema.
- Utilizar flujos de datos compuestos que representen a aquellos que sean similares.

Dentro de éste diagrama se enfatizan varias características importantes del sistema:

- Las personas, organizaciones y sistemas con los que se comunica el sistema. Son conocidos como terminadores.
- Los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma.
- Los datos producidos por el sistema y que se enviarán al exterior.
- Los almacenes de datos que el sistema comparte con los terminadores.
- La frontera entre el sistema y el resto del mundo.

Elementos del diagrama de contexto

El diagrama de contexto consiste de terminadores, flujos de datos y flujos de control, almacenes de datos y un solo proceso, que consiste en una sola burbuja. El nombre dentro del proceso suele ser el nombre del sistema completo o un acrónimo convenido.

Los terminadores se representan por medio de rectángulos y se comunican con el sistema utilizando flujos de datos o de control, los cuales son representados por flechas, o a través de almacenes externos. Hay que recalcar que los terminadores no se comunican entre sí, al menos no en el diagrama de contexto, ya que denotarían interacciones externas al sistema.

A continuación, como se muestra en la figura 4.1.1.2, presenta el diagrama de contexto de nuestro SIAH:

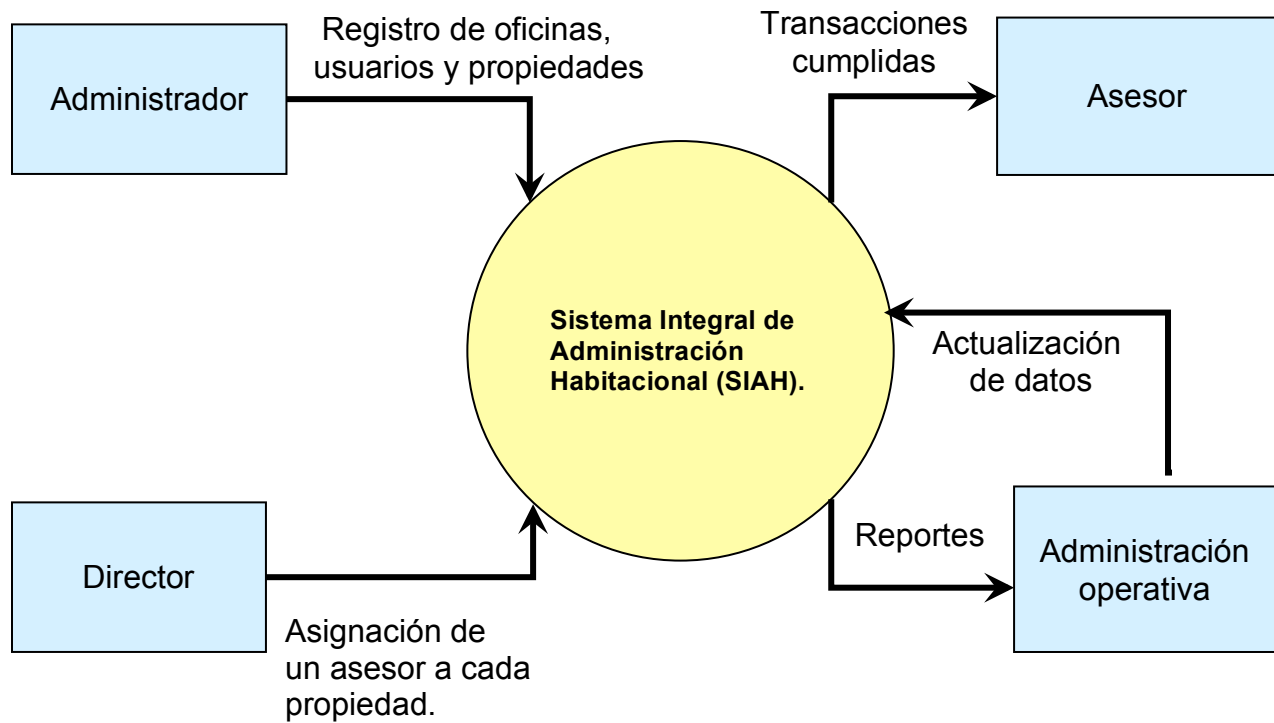


Figura 4.1.1.2 Diagrama de contexto del sistema (SIAH).



4.1.2 Diagrama entidad-relación del sistema(SIAH).

Un modelo es una representación del mundo real al mundo abstracto. Para modelar nuestro negocio, es decir, nuestro proyecto, en cuestión de bases de datos utilizamos el modelo lógico y el modelo físico.

El modelo lógico apoya las necesidades de los datos que deben representar la información del negocio y definir sus reglas. El modelo físico apoya las necesidades del administrador de la base de datos, que se centra en la puesta en práctica del modelo en una base de datos.

El modelo entidad-relación (ER) es uno de los modelos lógicos más populares utilizados en las bases de datos. Se basa en una representación del mundo real en que los datos se describen como entidades, relaciones y atributos. Hay múltiples notaciones para representar un diagrama entidad-relación, para nuestros propósitos y por comodidad ya que vamos a utilizar el manejador MySQL, explicaremos la notación “**crow’s foot**” (pie de cuervo), ya que es la notación que se utiliza en el software de MySQL WorkBench con el que creamos nuestro diagrama entidad-relación.

Componentes usados en la creación de un diagrama entidad-relación (DER):

- Entidad: Una persona, lugar o cosa del cual queremos recoger y almacenar sus atributos, también queremos tener registro de varias abstracciones que se le hacen a esa entidad. Tiene un identificador, que identifica de forma exclusiva una instancia de una entidad, éste se identifica con algún símbolo como un asterisco o en el caso de WorkBench con un signo de exclamación amarillo. Vea figura 4.1.2.1.

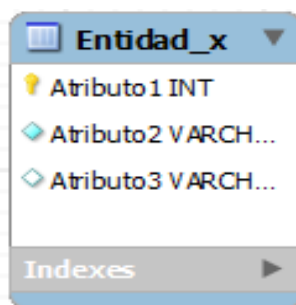


Figura 4.1.2.1 Representación de una Entidad.

- Relación: Ilustra asociación entre dos entidades. Tiene también cardinalidad. Se representa con una línea como se muestra en la figura 4.1.2.2.



Figura 4.1.2.2 Representación de una Relación.

- Cardinalidad: Son los indicadores de las reglas del negocio de toda una relación. Se refiere al número máximo de veces que una instancia en una sola entidad puede estar asociada con instancias en la entidad relacionada. Como se muestra en la figura 4.1.2.3., las cardinalidades pueden ser:

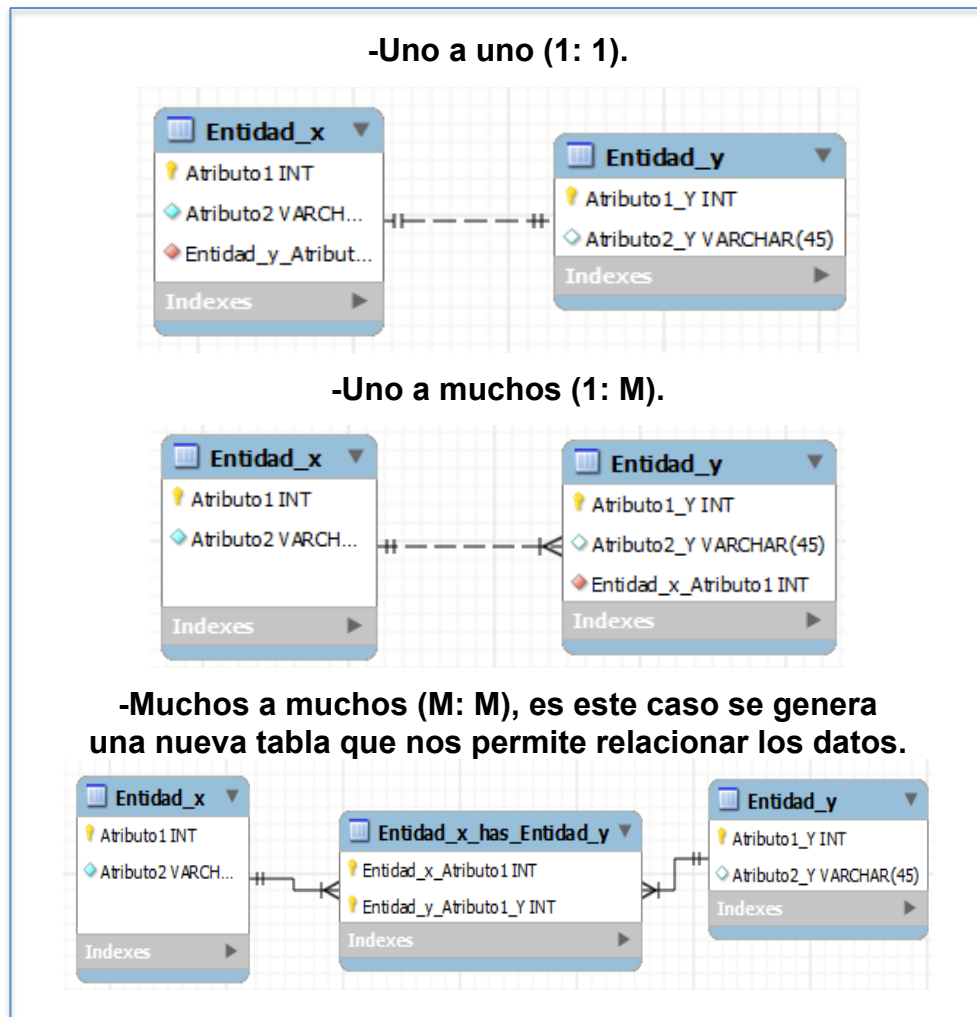


Figura 4.1.2.3 Tipos de cardinalidad 1: 1, 1: M y M: M.



Cada uno de los atributos de una entidad posee un dominio, el que corresponde al tipo de atributo. Se define dominio como un conjunto de valores que puede tomar un determinado atributo dentro de una entidad.

Cada uno de los atributos tiene asociado un tipo de dato que puede ser un entero, carácter, float, etc...

Como se muestra en la figura 4.1.2.4 se presenta nuestro DER resultante de las necesidades de negocio para nuestro SIAH.

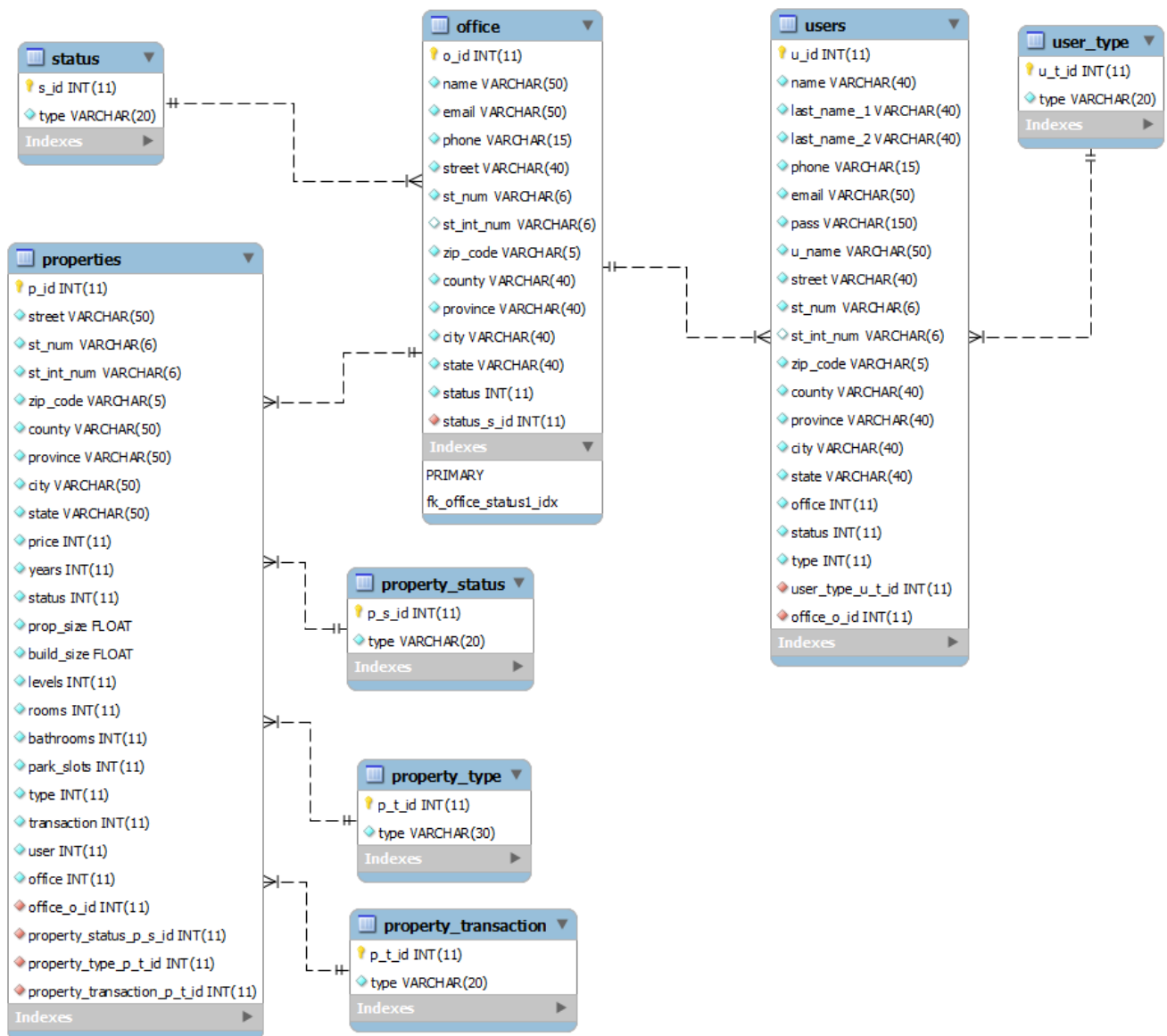


Figura 4.1.2.4

4.1.3 Diagrama de flujo de datos y procesos

Los diagramas de procesos representan de forma gráfica las funcionalidades que se esperan sean solucionadas por la plataforma. Al realizar un análisis de los requerimientos solicitados para el desarrollo de la plataforma, se tienen los siguientes procesos como se muestra en la figura 4.1.3.1

- Proceso de seguridad y acceso
- Proceso de menú principal
- Proceso de oficinas
- Proceso de propiedades
- Proceso de usuarios
- Proceso de reportes

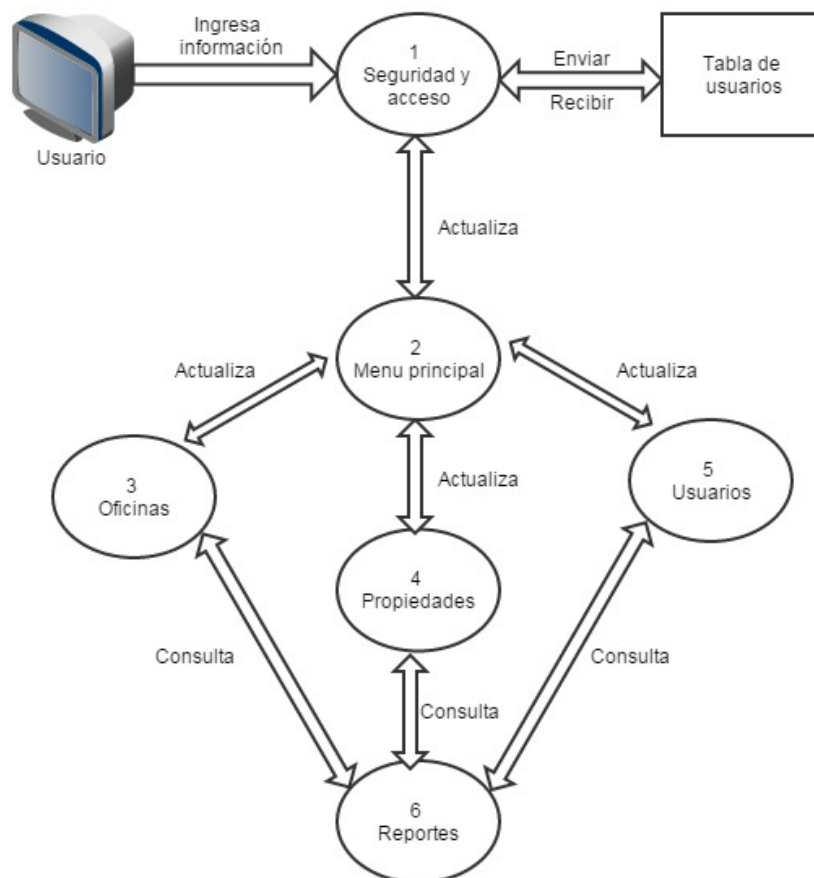


Figura 4.1.3.1 Diagrama de flujos y procesos

El proceso de seguridad y acceso del SIAH se muestra en la figura 4.1.3.2

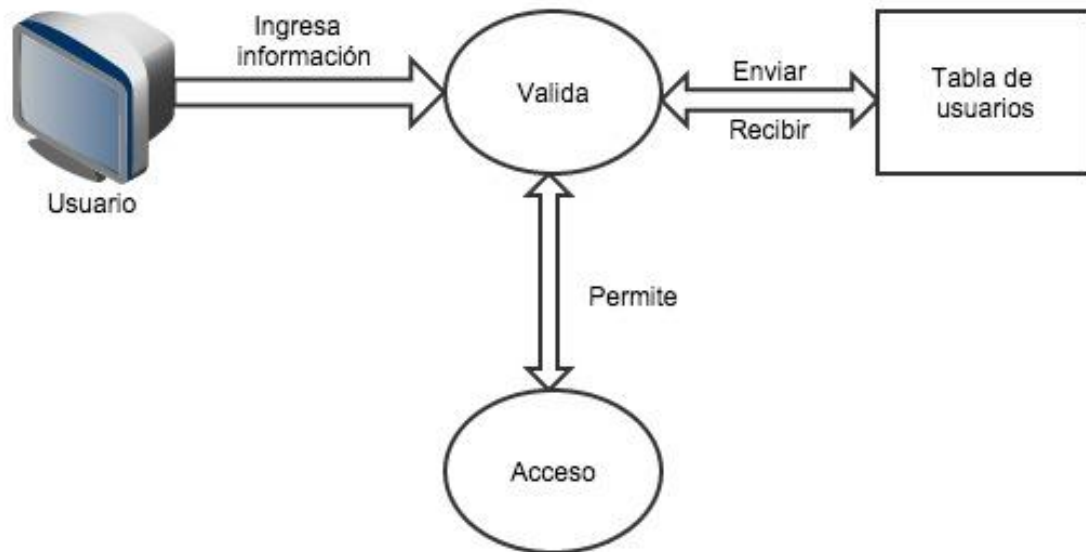


Figura 4.1.3.2 Proceso de seguridad y acceso

Este proceso consta de dos subprocesos:

- Validación. A partir de los datos que ingresa el usuario se realiza una consulta a la tabla de usuarios para determinar si se trata de un acceso válido.
- Acceso. Cuando se realiza un acceso válido, se recopila la información de sesión con los datos de su perfil, credenciales, nivel de acceso e información asociada al usuario. Este proceso otorga el ingreso al usuario.

El proceso de menú principal permite realizar al usuario diversas tareas de acuerdo al nivel de acceso otorgado como administrador de la plataforma, director o asesor, como se muestra en la figura 4.1.3.3

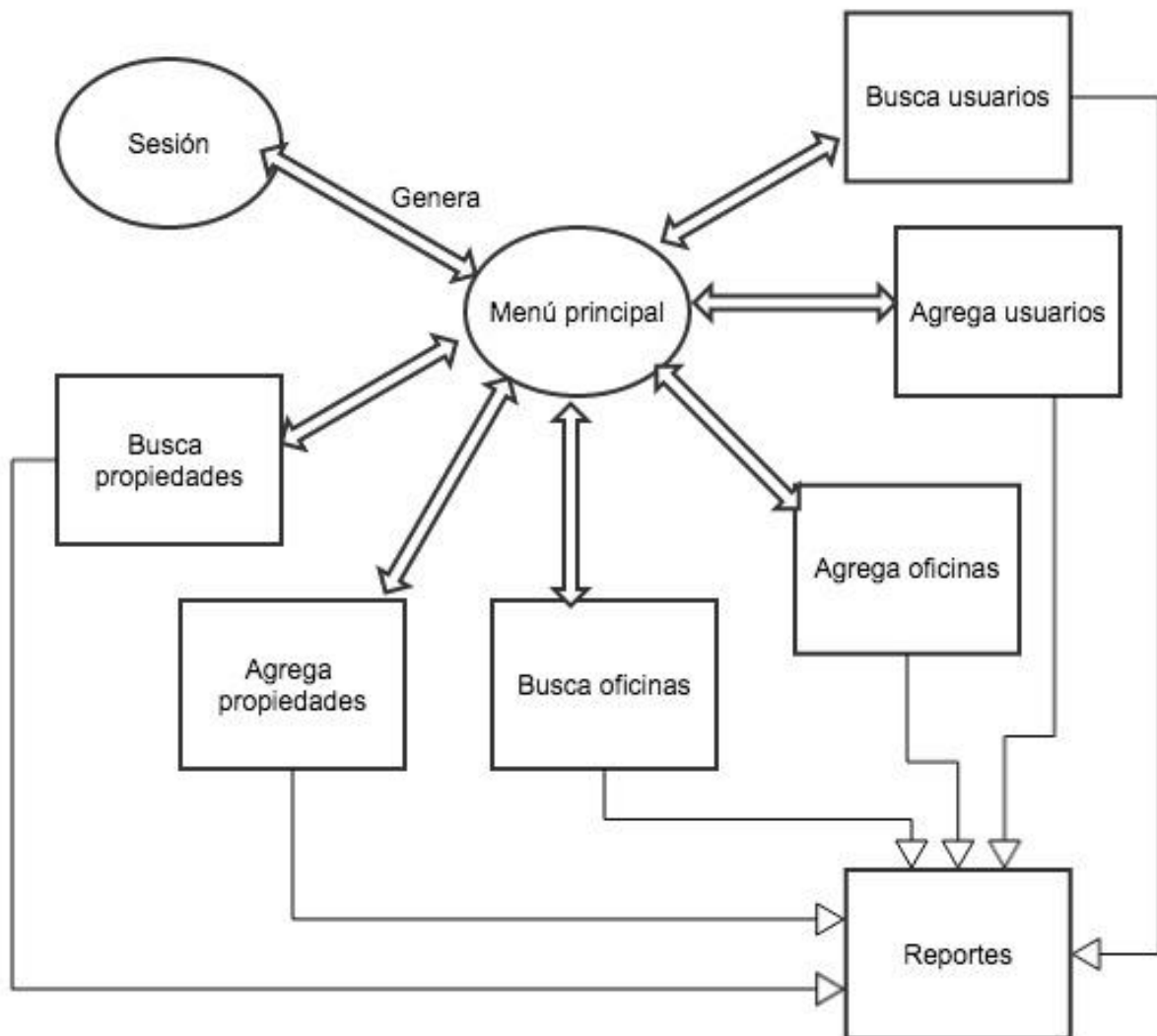


Figura 4.1.3.3 Proceso de menú principal

El proceso de oficinas permite buscar, agregar o modificar los registros de las oficinas que se encuentren dentro de la tabla de datos. Para realizar una modificación a los datos de las oficinas se debe verificar los usuarios asignados dentro de la tabla de usuarios para que puedan permanecer ligados de manera correcta.

Para agregar o modificar oficinas se debe verificar el nivel de acceso que el usuario tenga asignado como se muestra en la figura 4.1.3.4.

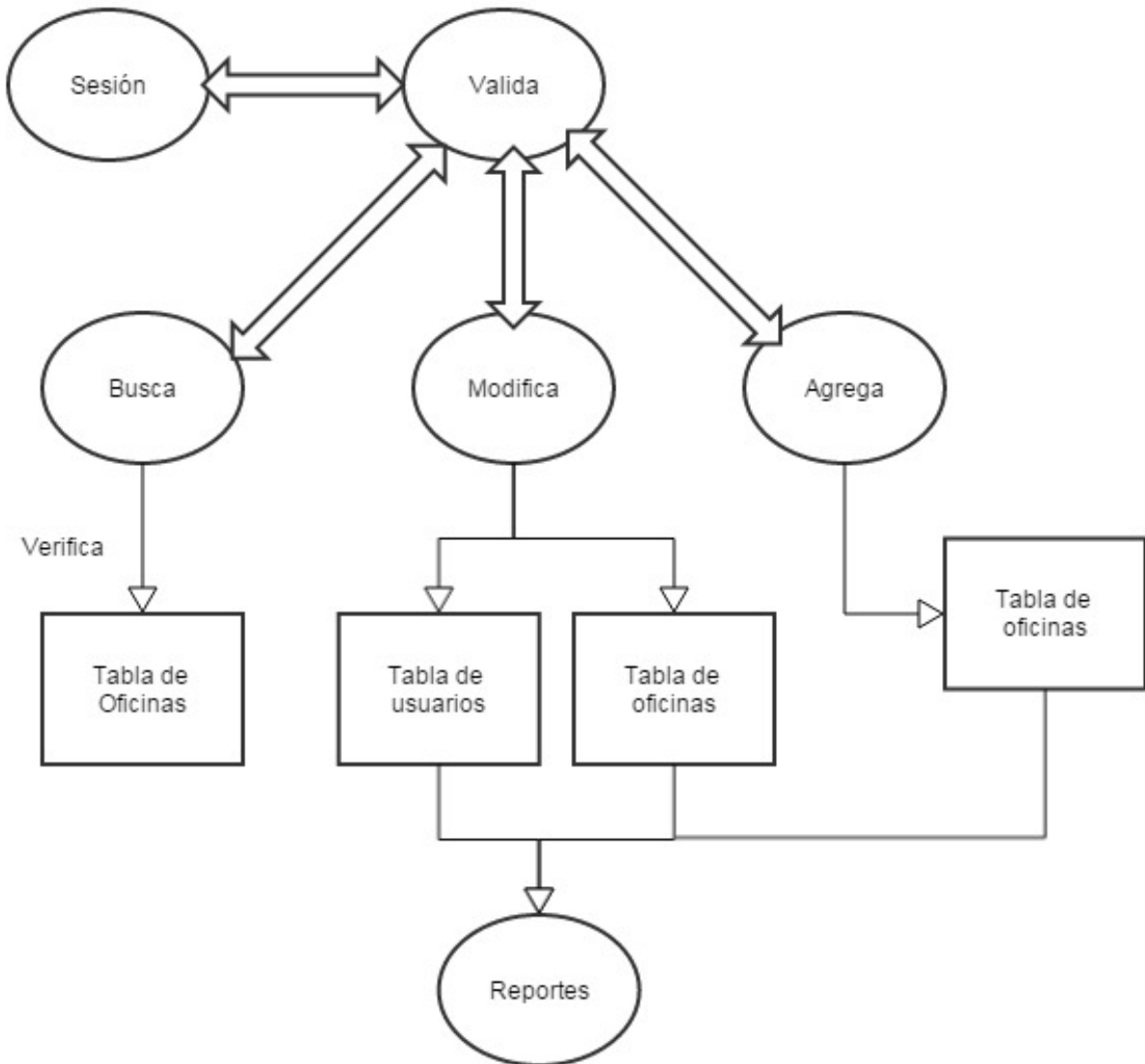


Figura 4.1.3.4 Proceso agregar, modificar y buscar oficinas

El proceso de propiedades permite buscar, agregar o modificar los registros de las propiedades que se encuentren dentro de la tabla de datos. Para realizar una modificación a los datos de las propiedades se debe verificar el usuario y la oficina asignados a esa propiedad dentro de la tabla de usuarios y oficinas para que puedan permanecer ligados de manera correcta.

Para agregar o modificar propiedades se debe verificar el nivel de acceso que el usuario tenga asignado como se muestra en la figura 4.1.3.5

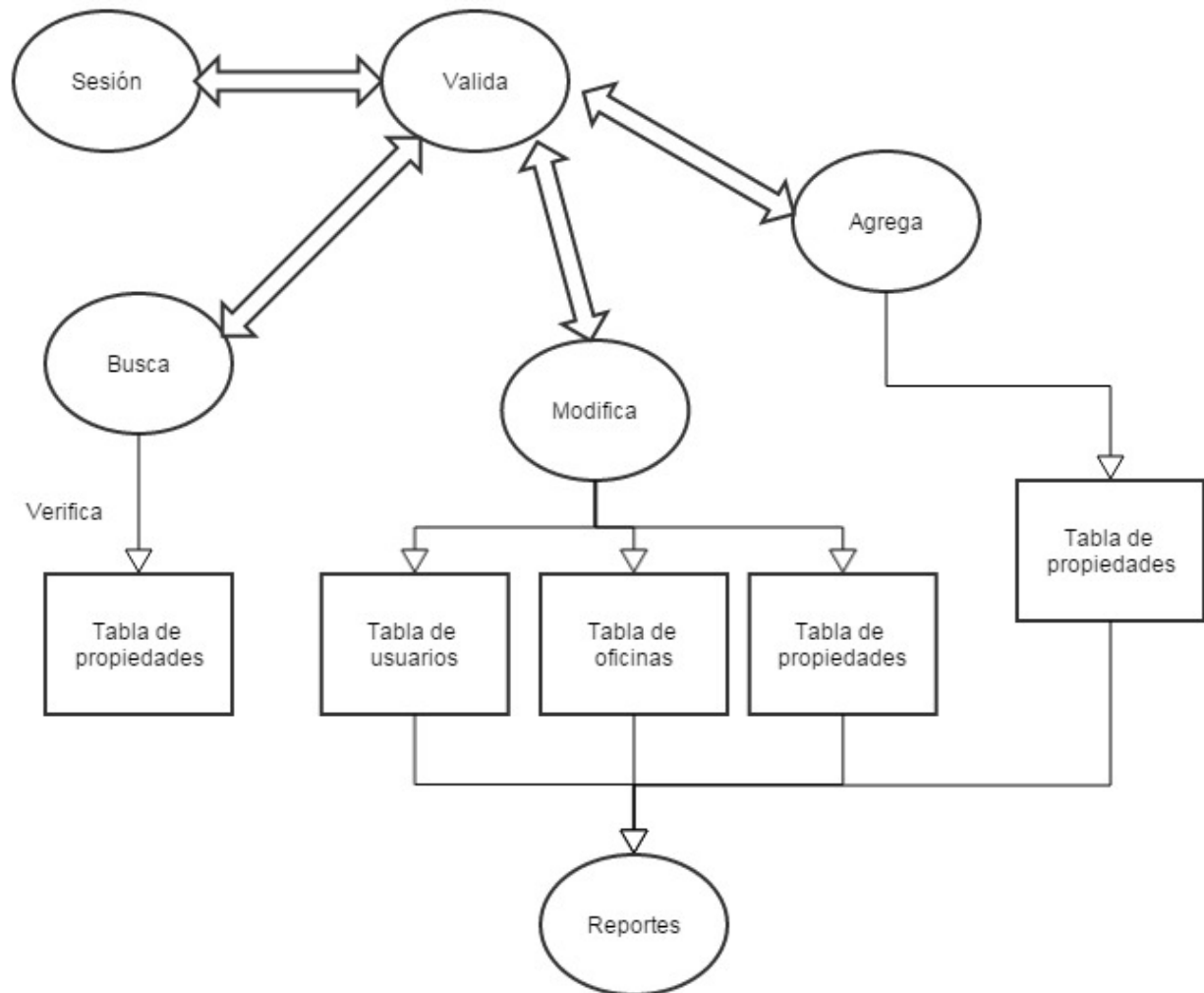


Figura 4.1.3.5 Proceso de agregar, modificar y buscar propiedades

El proceso de usuarios permite buscar, agregar o modificar los registros de los usuarios que se encuentren dentro de la tabla de datos. Para realizar una modificación a los datos de los usuarios se debe verificar la oficina y las propiedades asignadas a ese usuario dentro de la tabla de propiedades y oficinas para que puedan permanecer ligados de manera correcta.



Para agregar o modificar usuarios se debe verificar el nivel de acceso que se tenga asignado como se muestra en la figura 4.1.3.6

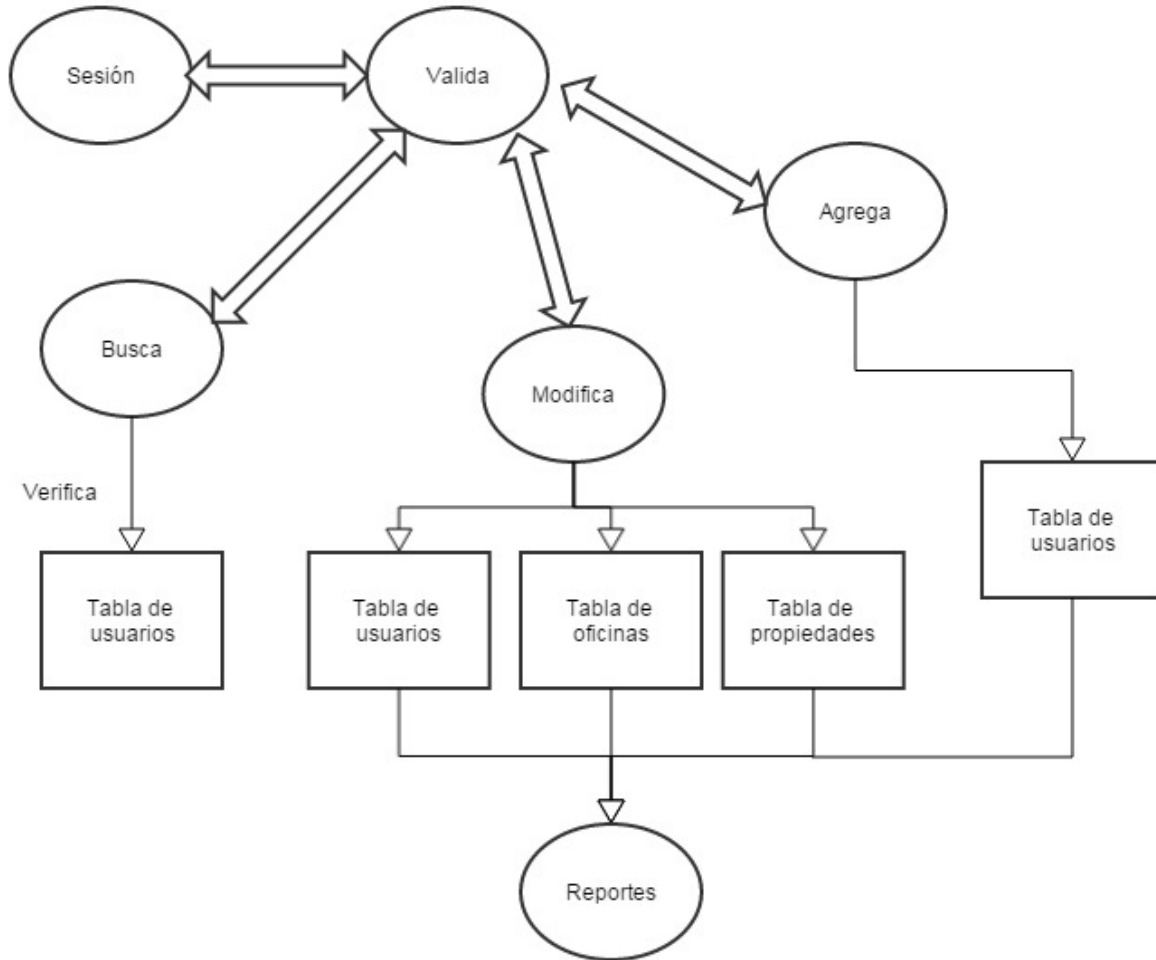


Figura 4.1.3.6 Proceso de agregar, modificar y buscar usuarios

El proceso de reportes permite mostrar de manera gráfica y entendible para el usuario todas las transacciones de venta o renta que se realizan dentro de la plataforma. El proceso permite filtrar por oficina, usuario, tipo de propiedad y monto de la transacción realizada.

Para poder realizar este proceso es necesario que el usuario tenga los permisos correspondientes como se muestra en la figura 4.1.3.7

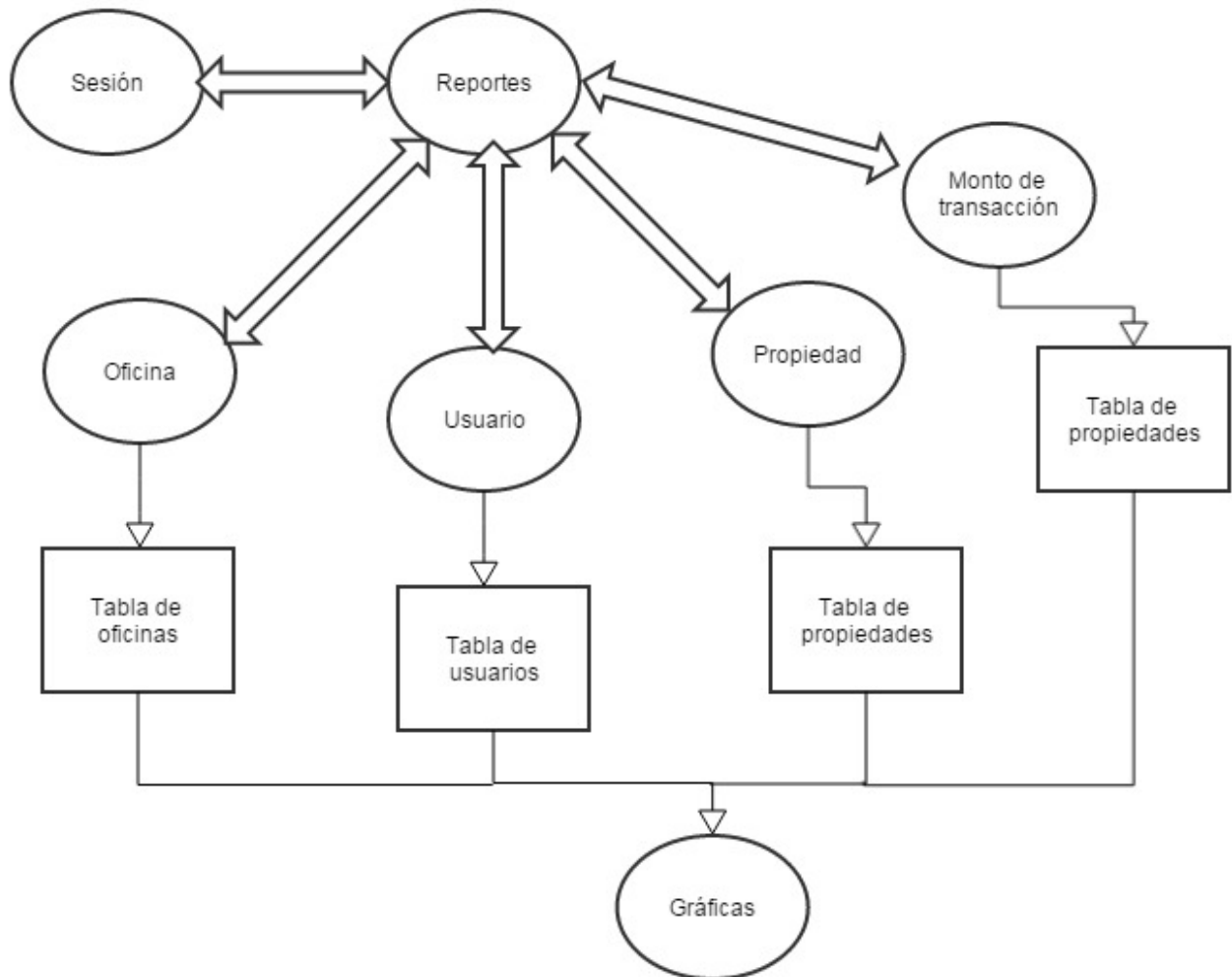


Figura 4.1.3.7 Proceso de reportes

Los diagramas de flujo permiten analizar a detalle las entradas y salidas que tendrá el usuario al realizar los diversos procesos definidos dentro de la plataforma.



El diagrama de flujo de la validación de usuario muestra las acciones tomadas de acuerdo a las credenciales de ingreso que presente el usuario como se muestra en la figura 4.1.3.8

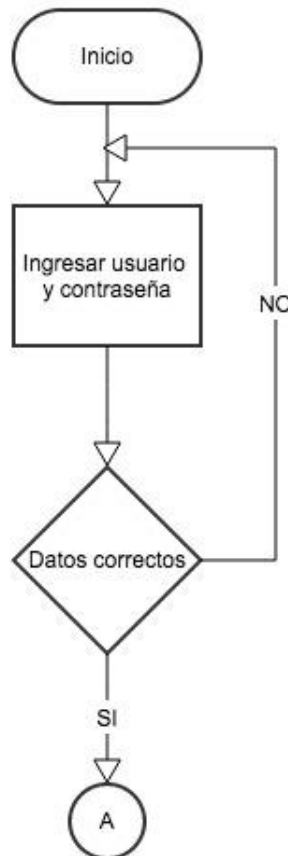


Figura 4.1.3.8 Validación de usuario

Como se muestra en la figura 4.1.3.9 al ingresar de manera correcta el usuario podrá seleccionar entre las siguientes opciones:

- Home
- Propiedades
- Oficinas
- Usuarios
- Reportes
- Salir

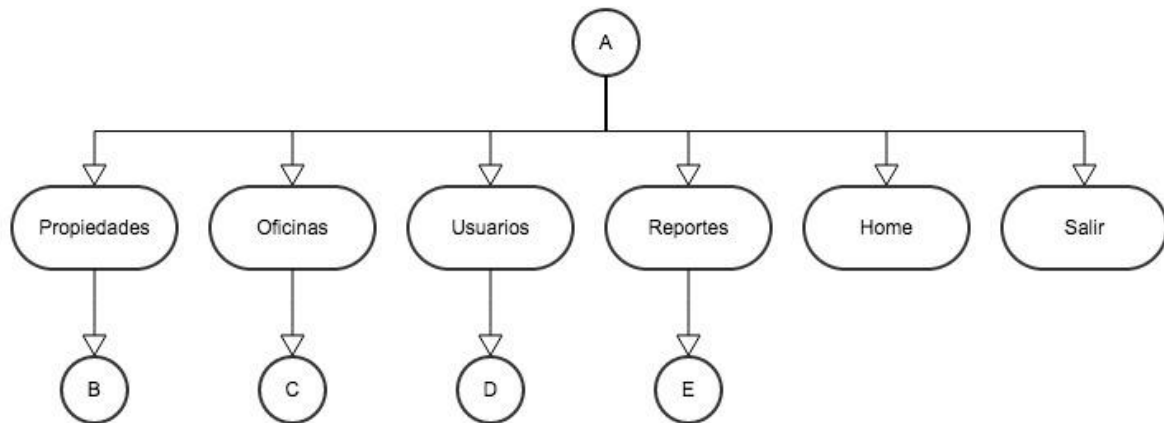


Figura 4.1.3.9 Ingreso

Como se muestra en la figura 4.1.3.10 dentro del módulo de propiedades las acciones a realizar son:

- Agregar
- Modificar
- Buscar

El módulo de oficinas puede realizar las siguientes acciones como se muestra en la figura 4.1.3.11

- Agregar
- Modificar
- Buscar

De la misma manera, para el módulo de usuarios se pueden realizar las siguientes acciones como se muestra en la figura 4.1.3.12

- Agregar
- Modificar
- Buscar

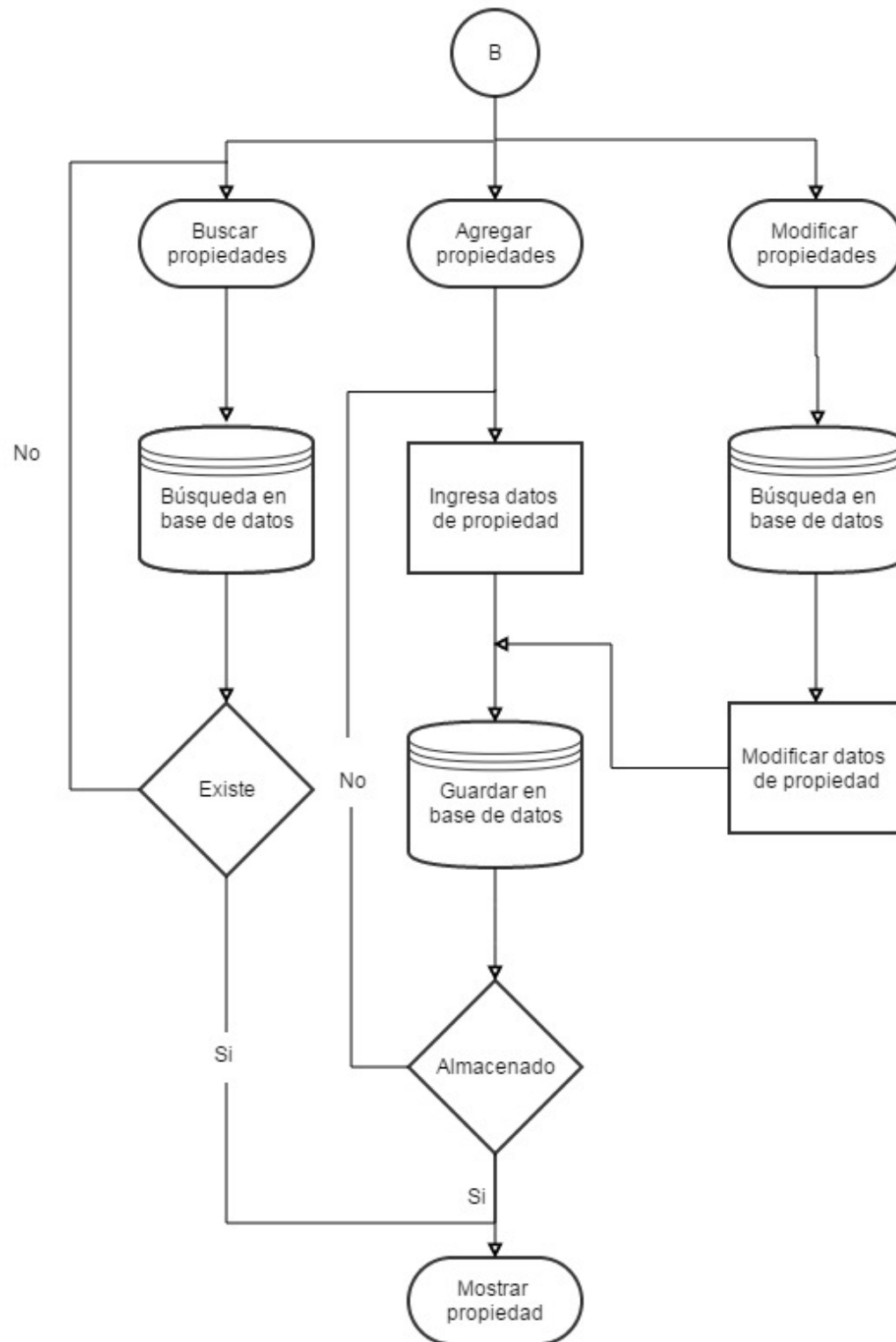


Figura 4.1.3.10 Módulo de propiedades

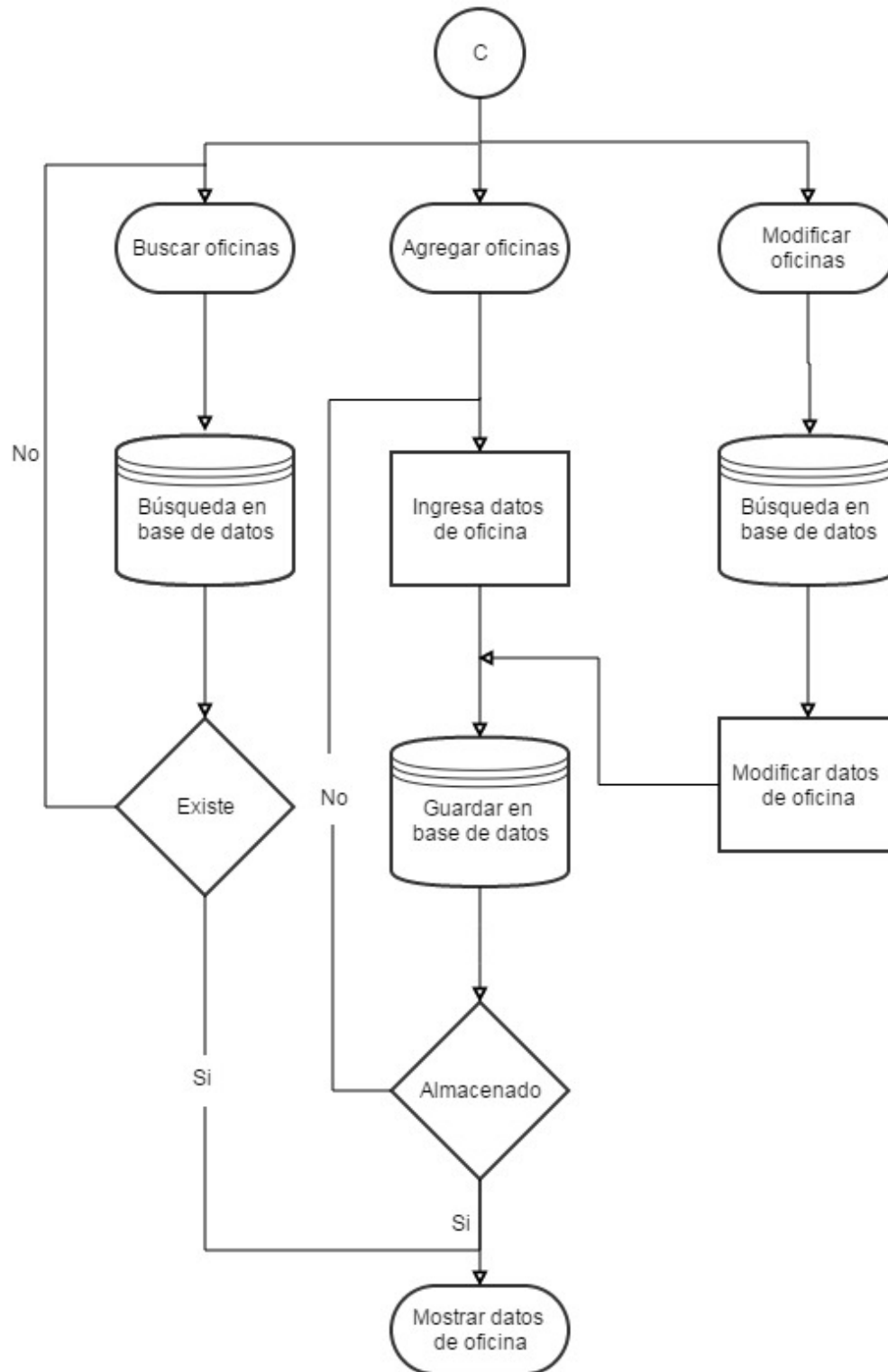


Figura 4.1.3.11 Módulo de oficinas

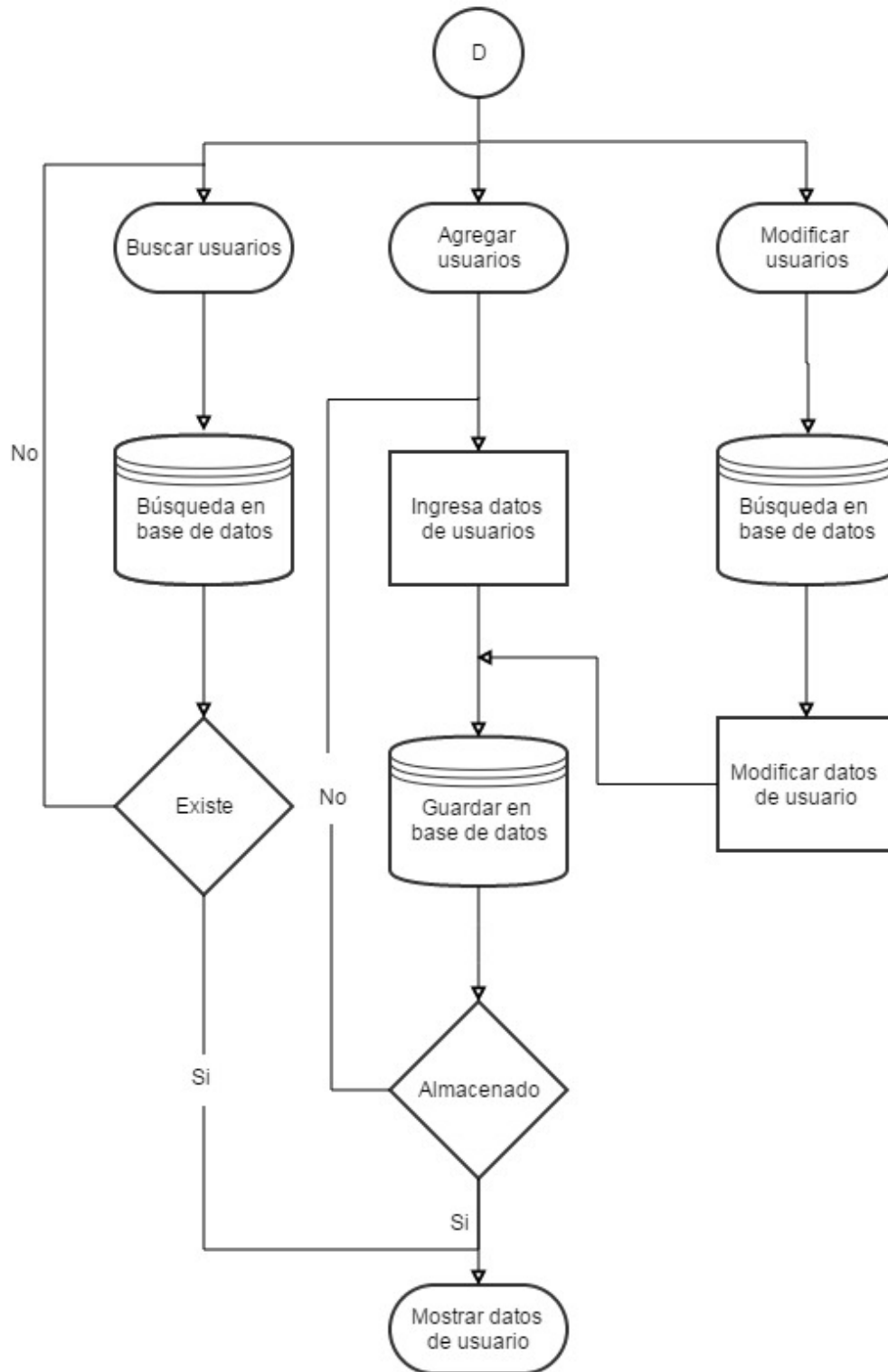


Figura 4.1.3.12 Módulo de usuarios

El módulo de reportes puede generar gráficos adecuados para obtener reportes de rendimiento de las oficinas, usuarios y propiedades como se muestra en la figura 4.1.3.13.

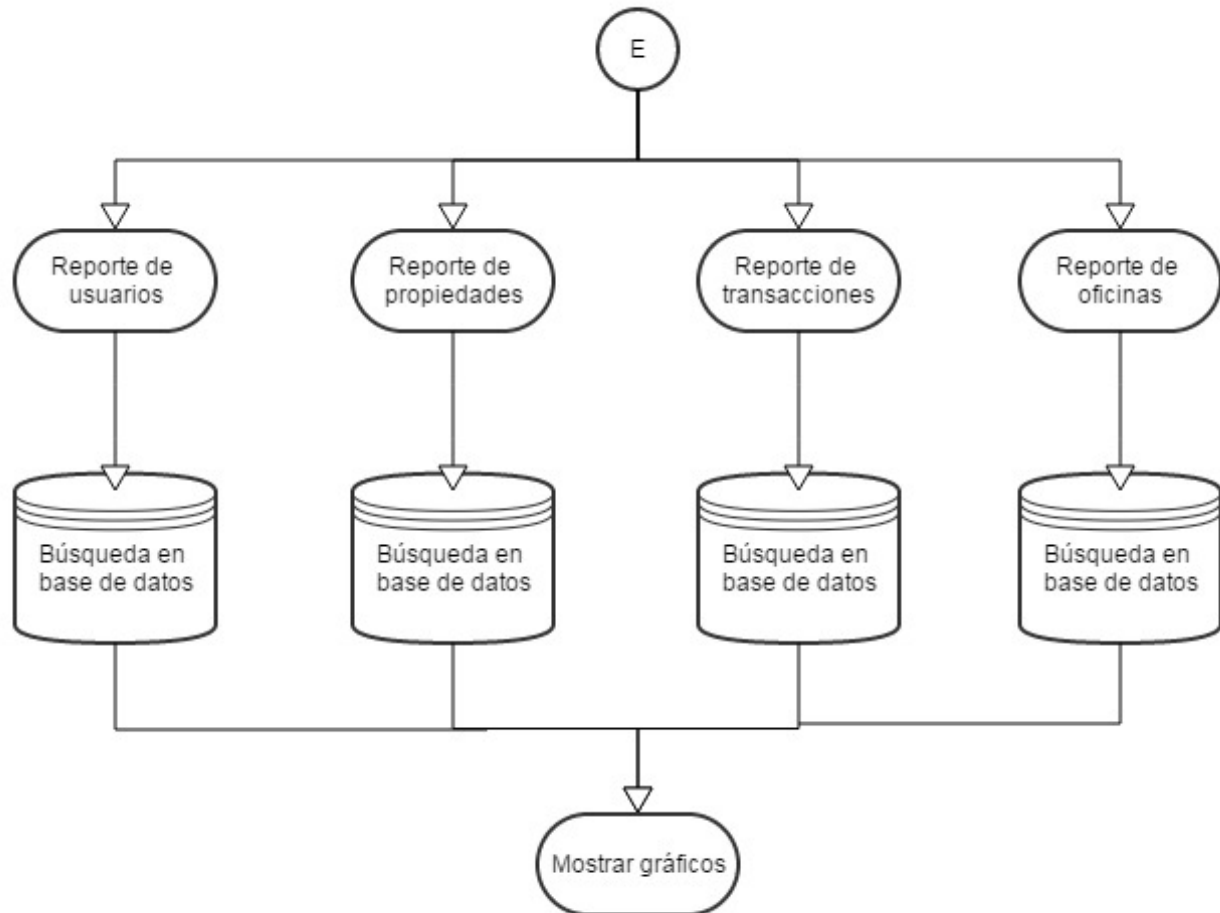


Figura 4.1.3.13 Módulo de reportes



4.1.4 Diccionario de datos.

Un diccionario de datos es un conjunto de registros que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades. Define con precisión los datos de entrada, salida, componentes de almacenes, flujos, detalles de las relaciones entre almacenes, etc. Los diccionarios de datos son buenos complementos a los diagramas de flujo de datos, los diagramas de entidad-relación, etc.

Estos diccionarios se desarrollan durante el análisis de flujo de datos y ayuda a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño del proyecto. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

En el diccionario de datos se documenta la información generada por el diseñador de bases de datos, sus principales utilidades son:

- Describir el propósito de la base de datos.
- Son datos comúnmente llamados metadatos.
- Proveer una detallada descripción de cada tabla dentro de la base de datos y explicar su propósito.
- Descripción detallada de cada campo de la tabla dentro de la base de datos.



- Documentar la estructura interna de cada tabla, incluyendo sus campos y sus tipos de datos con comentarios, todos los índices y todas las vistas.
- Describir reglas, como pueden ser el uso de valores no nulos o valores únicos.

Este diccionario es muy importante pues contiene todos los nombres y características de los atributos de cada objeto de la base de datos del sistema, en resumen contiene metadatos y frecuentemente es utilizado por el administrador de la base de datos para el registro de las decisiones tomadas en cuanto a la estructura, y nombre de los objetos.

En la tabla PROPERTIES, ver tabla 4.1.4.1, se ingresa información de la propiedad.

La tabla OFFICE, ver tabla 4.1.4.2, contiene información de las oficinas.

La tabla USERS, ver tabla 4.1.4.3, contiene información de los usuarios que manejan el sistema.

La tabla STATUS, ver tabla 4.1.4.4, contiene los detalles del status que están dentro de una oficina.

En la tabla de PROPERTY_STATUS, ver tabla 4.1.4.5 se genera de las propiedades de los estatus

La tabla de PROPERTY_TYPE, ver tabla 4.1.4.6, se genera con el tipo de status.

La tabla de PROPERTY_TRANSACTION, ver tabla 4.1.4.7, se genera con las características de las transacciones.

La tabla de USER_TYPE, ver tabla 4.1.4.8, contiene información de los tipos de usuario.



Tabla PROPERTIES

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de propiedad	p_id	Int 11	Primaria	No		Es el ID de la propiedad
ID de oficina	o_id	Int 11	Foránea	No	office	Es la llave primaria de la tabla office
ID de propiedades de status	p_s_id	Int 11	Foránea	No	Property_status	Es la llave primaria de la tabla Property_status
ID de propiedades de type	p_t_id	Int 11	Foránea	No	Property_type	Es la llave primaria de la tabla Property_type
ID de propiedades de transacción	p_t_id	Int 11	Foránea	No	Property_transaction	Es la llave primaria de la tabla Property_transaction
Calle	street	Varchar 50		No		Es la calle de la propiedad
Número	st_num	Varchar 6		No		Es el número de la dirección de la propiedad
Numero interior	st_int_num	Varchar 6		No		Es el numero

						interior de la propiedad
Código postal	zip_code	Varchar 5		No		Es el código postal de la dirección de la propiedad
Ciudad	contry	Varchar 50		No		Es la ciudad donde está localizada la propiedad
Provincia	province	Varchar 50		No		Es la provincia donde está localizada la propiedad
Años	year	Int 11		No		Años de propietario
Estatus	status	Int 11		No		Status de la propiedad
Tamaño de la propiedad	prop_size	Float		No		Muestra el tamaño de la propiedad
Superficie construida	build_size	Float		No		Nos indica el tamaño de la superficie construida
Niveles en Pisos	levels	Int 11		No		Nos indica los pisos que tiene la



					propiedad
Cuartos	rooms	Int 11		No	Nos indica cuantos cuartos tiene la propiedad
Baños	bathrooms	Int 11		No	Indica cuantos baños hay en la propiedad
Espacios libres	park_slots	Int 11		No	Muestra cuantos espacios libres tiene la propiedad
Tipo	type	Int 11		No	Nos muestra el tipo de inmueble
Transacción	transaction	Int 11		No	Nos muestra la transacción del inmueble
Usuario	user	Int 11		No	Nos muestra el usuario a cargo
Oficina	office	Int 11		No	Nos muestra en qué oficina se está manejando la propiedad

Tabla 4.1.4.1 PROPERTIES

Tabla OFFICE

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla office	o_id	Int 11	Primaria	No		Es el ID de la tabla office
Estatus de las propiedades	s_id	Int	Foránea	No	Status	Nos muestra el estatus de la propiedad
Nombre	name	Varchar 50		No		Nos muestra el nombre de la oficina
Email	email	Varchar 50		No		Nos muestra el email de la oficina
Teléfono	pone	Varchar 15		No		Nos muestra el teléfono de la oficina
Calle	street	Varchar 40		No		Nos muestra la dirección de la oficina
Numero	st_num	Varchar 6		No		Es el número de la



						dicción de la oficina
Numero interior	st_int_num	Varchar 6		Si		Es el número interior de la oficina
Código postal	zip_code	Varchar 40		No		Es el código postal de la oficina
Ciudad	contry	Varchar 50		No		Es la ciudad donde está localizada la oficina
Provincia	province	Varchar 50		No		Es la provincia donde está localizada la propiedad
Estatus	status	Int 11		No		Status de la oficina

Tabla 4.1.4.2 OFFICE

Tabla USERS

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla users	u_id	Int 11	Primaria	No		Es el ID de la tabla users

Tipo de usuario	U_t_id	Int 11	Foránea	no	User_type	Es el Id de la tabla user type, que hace referencia al tipo de usuario
ID de oficina	O_id	Int 11	Foránea	No	office	Nos muestra a que oficina pertenece el usuario
Nombre	Name	Varchar 50		No		Nos muestra el nombre del usuario
Apellido paterno	Last_name_1	Varchar 40		No		Es el apellido paterno del usuario
Apellido materno	Last_name_2	Varchar 40		No		Es el apellido materno del usuario
Email	Email	Varchar 50		No		Nos muestra el email del usuario
Contraseña	Pass	Varchar 150		No		Es la contraseña de sesión del usuario
Alias	U_name	Varchar 50		No		Es el alias del usuario



Teléfono	Pone	Varchar 15		No		Nos muestra el teléfono del usuario
Calle	Street	Varchar 40		No		Nos muestra la dirección del usuario
Numero	St_num	Varchar 6		No		Es el número de la dirección del usuario
Numero interior	St_int_num	Varchar 6		Si		Es el número interior de donde vive el usuario
Código postal	Zip_code	Varchar 40		No		Es el código postal del usuario
Ciudad	contry	Varchar 50		No		Es la ciudad donde está localizado el usuario
Provincia	province	Varchar 50		No		Es la provincia donde está localizada la propiedad
Oficina	office	Int 11		No		Nos muestra

						en qué oficina se está manejando la propiedad
Estatus	status	Int 11		No		Status de la oficina
Tipo	type	Int 11		No		Nos muestra el tipo usuario

Tabla 4.1.4.3 USERS

Tabla STATUS

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla users	s_id	Int 11	Primaria	No		Es el ID de la tabla status
Tipo	type	Varchar 20		No		Nos muestra el tipo de status

Tabla 4.1.4.4 STATUS



Tabla PROPERTY_STATUS

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla property_status	p_s_id	Int 11	Primaria	No		Es el ID de la tabla property_status
Tipo	type	Varchar 20		No		Nos muestra el tipo del status de la propiedad

Tabla 4.1.4.5 PROPERTY_STATUS

Tabla PROPERTY_TYPE

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla property_type	p_t_id	Int 11	Primaria	No		Es el ID de la tabla property_type
Tipo	type	Varchar 30		No		Nos muestra el tipo del status de las características de la propiedad

Tabla 4.1.4.6 PROPERTY_TYPE

Tabla PROPERTY_TRANSACTION

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla property_transaction	p_t_id	Int 11	Primaria	No		Es el ID de la tabla property_transaction
Tipo	type	Varchar 30		No		Nos muestra las transacciones de la propiedad

Tabla 4.1.4.7 PROPERTY_TRANSACTION

Tabla USER_TYPE

Nombre	Acrónimo	Tipo y longitud	Llave	Nulos	Tablas asociadas	Descripción
ID de la tabla user_type	u_t_id	Int 11	Primaria	No		Es el ID de la tabla user_type
Tipo	type	Varchar 30		No		Nos muestra las el tipo de usuarios

Tabla 4.1.4.8 USER_TYPE



4.1.5 Normalización de la base de datos

La normalización es el proceso de organizar los datos en una base de datos. Esto incluye la creación de tablas y establecer relaciones entre las tablas según reglas diseñadas tanto para proteger los datos y hacer que la base de datos sea más flexible, eliminando la redundancia y la dependencia incoherente.

Los datos redundantes desperdician espacio en disco y crea problemas de mantenimiento. Los datos que existen en más de un lugar se deben evitar, ya que generan inconsistencias. Un cambio de dirección del cliente es mucho más fácil de implementar si los datos que se almacenan sólo son en la tabla de clientes y en ningún otro lugar en la base de datos.

Hay algunas reglas para la normalización de bases de datos. Cada regla se denomina una "forma normal". Si se cumple la primera regla, se dice que la base de datos para estar en "la primera forma normal". Si se cumplen las tres primeras reglas, la base de datos se considera que está en "tercera forma normal." Aunque son posibles otros niveles de normalización, la tercera forma normal se considera el nivel más alto necesario para la mayoría de aplicaciones.

Al igual que con muchas reglas y especificaciones formales, los escenarios del mundo real no siempre permiten el cumplimiento perfecto. En general, la normalización requiere tablas adicionales y algún trabajo para su aplicación correcta.

Primera Forma Normal

- Eliminar grupos repetidos en tablas individuales.
- Crear una tabla separada para cada conjunto de datos relacionados.
- Identificar cada conjunto de datos relacionados con una clave principal.

Segunda Forma Normal

- Crear tablas independientes para conjuntos de valores que se aplican a varios registros.

- Relacionar estas tablas con una clave externa.
- Los registros no deben depender de otra cosa que no sea la clave principal de una tabla (una clave compuesta, si es necesario).

Tercera Forma Normal

- Eliminar los campos que no dependan de la clave.
- Los valores en un registro que no son parte de la clave de ese registro no pertenecen a la tabla (dependencias transitivas).

Como se muestra en la tabla 4.1.5.1 se tomará como muestra 25 datos para realizar la normalización de la base de datos

Id propiedad	Dirección propiedad	Id oficina	Nombre oficina	Dirección oficina	Id usuario	Nombre usuario
34	Lago Alberto 45 Col. Anahuac	3	DF	Reforma 25 Col. Cuauhtémoc	6	Luis Montes
67	Mirador 15 Col. Coapa	4	Guadalajara	Minerva 15 Col. Moderna	3	Juan Gutiérrez
33	Florida 24 Col. San Antonio	1	Monterrey	Fundidora 365 Col. San Pedro	2	Lucas González
56	Balbuena 45 Col. San Lázaro	7	Mérida	Paseo Montejo 34 Col. Álamos	7	Pedro Ortiz
23	Horacio 38 Col. Escobedo	7	Mérida	Paseo Montejo 34 Col. Álamos	8	Carlos Medina



90	Norte 15 Col. Blvd Puerto	10	Querétaro	Balbuena 45 Col. San Lázaro	10	Adrián Juárez
123	Minerva 15 Col. Moderna	3	DF	Reforma 25 Col. Cuauhtemoc	12	María Espinoza
45	Boturini 14 Col. Irrigación	25	Jalapa	Puerto 67 Col. Irrigación	10	Adrián Juárez
89	Refugio 12 Col. Centenario	2	Saltillo	Horacio 38 Col. Escobedo	15	Raúl Sánchez
35	Tláhuac 987 Col. Moctezuma	6	Los Cabos	Florida 24 Col. San Antonio	3	Juan Gutiérrez
2	Alvaro Obregón 863 Col. Madero	20	San Miguel	Mesones 91 Col. Bolívar	2	Lucas González
5	Paseo Montejo 34 Col. Alamos	12	Tlaxcala	Molina 190 Col. Corzo	2	Lucas González
17	Fundidora 365 Col. San Pedro	10	Querétaro	Balbuena 45 Col. San Lázaro	4	Sandra Lima
29	Oaxaca 18 Col.	9	Cancún	Norte 15 Col. Blvd Puerto	1	Silvia Chávez

	Universidad					
58	Plateros 62 Col. Pirules	12	Tlaxcala	Molina 190 Col. Corzo	1	Silvia Chávez
41	Rio Lerma 56 Col. Cuauhtémoc	2	Saltillo	Horacio 38 Col. Escobedo	7	Pedro Ortiz
70	Progreso 76 Col. San Miguel	4	Guadalajara	Minerva 15 Col. Moderna	7	Pedro Ortiz
40	Calle 15 789 Col. San Pedro	6	Los Cabos	Florida 24 Col. San Antonio	10	Adrián Juárez
11	Mesones 91 Col. Bolívar	4	Guadalajara	Minerva 15 Col. Moderna	12	María Espinoza
9	Molina 190 Col. Corzo	20	San Miguel	Mesones 91 Col. Bolívar	4	Sandra Lima
167	Boston 78 Col. Nápoles	9	Cancún	Norte 15 Col. Blvd Puerto	7	Pedro Ortiz
103	Zapata 467 Col. Revolución	10	Querétaro	Balbuena 45 Col. San Lázaro	4	Sandra Lima
115	Av. Coyoacán 293 Col. Romero	9	Cancún	Norte 15 Col. Blvd Puerto	10	Adrián Juárez
82	Zempoala 23	20	San Miguel	Mesones 91	4	Sandra



	Col. Odesa			Col. Bolívar		Lima
20	Tajín 892 Col. Narvarte	3	DF	Reforma 25 Col. Cuauhtemoc	6	Luis Montes

Tabla 4.1.5.1 Tabla original

Tomando en cuenta la tabla podemos observar que se repiten los datos de nombre de oficina, dirección de oficina y nombre de usuario. Aplicando la primera forma normal se obtiene la tabla de propiedades como se muestra en la tabla 4.1.5.2,

Id propiedad	Dirección propiedad	Id oficina	Id usuario
2	Alvaro Obregón 863 Col. Madero	20	2
5	Paseo Montejo 34 Col. Alamos	12	2
9	Molina 190 Col. Corzo	20	4
11	Mesones 91 Col. Bolívar	4	12
17	Fundidora 365 Col. San Pedro	10	4
20	Tajín 892 Col. Narvarte	3	6
23	Horacio 38 Col. Escobedo	7	8
29	Oaxaca 18 Col. Universidad	9	1
33	Florida 24 Col. San Antonio	1	2
34	Lago Alberto 45 Col. Anahuac	3	6
35	Tláhuac 987 Col. Moctezuma	6	3

40	Calle 15 789 Col. San Pedro	6	10
41	Rio Lerma 56 Col. Cuauhtemoc	2	7
45	Boturini 14 Col. Irrigación	25	10
56	Balbuena 45 Col. San Lázaro	7	7
58	Plateros 62 Col. Pirules	12	1
67	Mirador 15 Col. Coapa	4	3
70	Progreso 76 Col. San Miguel	4	7
82	Zempoala 23 Col. Odesa	20	4
89	Refugio 12 Col. Centenario	2	15
90	Norte 15 Col. Blvd Puerto	10	10
103	Zapata 467 Col. Revolución	10	4
115	Av. Coyoacán 293 Col. Romero	9	10
123	Minerva 15 Col. Moderna	3	12
167	Boston 78 Col. Nápoles	9	7

Tabla 4.1.5.2

También de la primera forma normal se obtiene la tabla de oficinas como se muestra en la tabla 4.1.5.3



Id oficina	Nombre oficina	Dirección oficina
1	Monterrey	Fundidora 365 Col. San Pedro
2	Saltillo	Horacio 38 Col. Escobedo
3	DF	Reforma 25 Col. Cuauhtemoc
4	Guadalajara	Minerva 15 Col. Moderna
6	Los Cabos	Florida 24 Col. San Antonio
7	Mérida	Paseo Montejo 34 Col. Álamos
9	Cancún	Norte 15 Col. Blvd Puerto
10	Querétaro	Balbuena 45 Col. San Lázaro
12	Tlaxcala	Molina 190 Col. Corzo
20	San Miguel	Mesones 91 Col. Bolívar
25	Jalapa	Puerto 67 Col. Irrigación

Tabla 4.1.5.3

Por último también se obtiene la tabla de usuarios como se muestra en la tabla 4.1.5.4

Id usuario	Nombre usuario
1	Silvia Chávez
2	Lucas González
3	Juan Gutiérrez
4	Sandra Lima

6	Luis Montes
7	Pedro Ortiz
8	Carlos Medina
10	Adrián Juárez
12	María Espinoza
15	Raúl Sánchez

Tabla 4.1.5.4

Aplicando la segunda forma normal se obtiene la tabla de propiedades como se muestra en la tabla 4.1.5.5

Id propiedad	Dirección propiedad
2	Alvaro Obregón 863 Col. Madero
5	Paseo Montejo 34 Col. Alamos
9	Molina 190 Col. Corzo
11	Mesones 91 Col. Bolívar
17	Fundidora 365 Col. San Pedro
20	Tajín 892 Col. Narvarte
23	Horacio 38 Col. Escobedo
29	Oaxaca 18 Col. Universidad
33	Florida 24 Col. San Antonio
34	Lago Alberto 45 Col. Anahuac



35	Tláhuac 987 Col. Moctezuma
40	Calle 15 789 Col. San Pedro
41	Rio Lerma 56 Col. Cuauhtémoc
45	Boturini 14 Col. Irrigación
56	Balbuena 45 Col. San Lázaro
58	Plateros 62 Col. Pirules
67	Mirador 15 Col. Coapa
70	Progreso 76 Col. San Miguel
82	Zempoala 23 Col. Odesa
89	Refugio 12 Col. Centenario
90	Norte 15 Col. Blvd Puerto
103	Zapata 467 Col. Revolución
115	Av. Coyoacán 293 Col. Romero
123	Minerva 15 Col. Moderna
167	Boston 78 Col. Nápoles

Tabla 4.1.5.5

Al aplicar la tercera forma normal se obtiene la tabla que relaciona los datos de las tablas anteriormente creadas y al eliminar los registros que no se relacionan con una llave externa como se muestra en la tabla 4.1.5.6



Sistema Integral De Administración Habitacional (SIAH)

Id propiedad	Id oficina	Id usuario
2	20	2
5	12	2
9	20	4
11	4	12
17	10	4
20	3	6
23	7	8
29	9	1
33	1	2
34	3	6
35	6	3
40	6	10
41	2	7
45	25	10
56	7	7
58	12	1
67	4	3
70	4	7
82	20	4



89	2	15
90	10	10
103	10	4
115	9	10
123	3	12
167	9	7

Tabla 4.1.5.6

De esta forma observamos que los datos ya se encuentran en tercera forma normal y por tanto se normalizaron cada una de las tablas que conforman la aplicación, dando como resultado que nuestro esquema de datos pueda manejar fácilmente los datos de toda la plataforma.

La misma normalización se aplicó a todos los demás datos y registros de la aplicación.

4.2 Construcción y prueba de la base de datos.

El SIAH, se encuentra basado en MySQL, que es un sistema de gestión de bases de datos relacional, el cual es un software libre y usado en el desarrollo de un sinnúmero de sitios Web, desde modestos sitios hasta muy sofisticados, además de popular. MySQL emplea el lenguaje SQL para consultas a la base de datos; SQL es un lenguaje de bases de datos normalizado utilizado por los diferentes motores de bases de datos para realizar determinadas operaciones sobre los datos o sobre la estructura de los mismos. El lenguaje SQL está compuesto por comandos, cláusulas, operadores lógicos y de comparación y funciones de agregado; estos elementos se combinan en los elementos para crear, actualizar y manipular las diferentes bases de datos.

Para cualquier referencia al sistema de gestión de base de datos podemos acceder a la dirección www.mysql.com, donde encontraremos la información que pudiésemos necesitar.

El sistema a desarrollar se enfoca a la administración de la compra-venta de bienes inmueble, esto se realiza a través de una inmobiliaria, la cual se encargará de toda la logística necesaria, posiblemente sea necesario buscar lo que el cliente necesita y se cuenta con ello. Dado lo anterior, se propone una tabla para la administración, la cual involucra desde las mismas oficinas, propiedades casas, departamentos, local comercial, terreno, el bien inmueble del que estemos hablando, al igual que los usuarios.

XAMPP es un servidor independiente de la plataforma de código libre; no tiene costo; que consiste principalmente en la base de datos MySQL, el servidor web Apache y los interpretes para lenguajes de script PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl, está liberado bajo licencia GNU, actúa como un servidor web libre, es sencillo en su instalación y capaz de interpretar páginas dinámicas, en la actualidad está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X. primeramente empezaremos por bajar el sistema,



Para descargar XAMPP buscamos la dirección www.apachefriends.org y descargamos el archivo, como se muestra en la figura 4.2.1

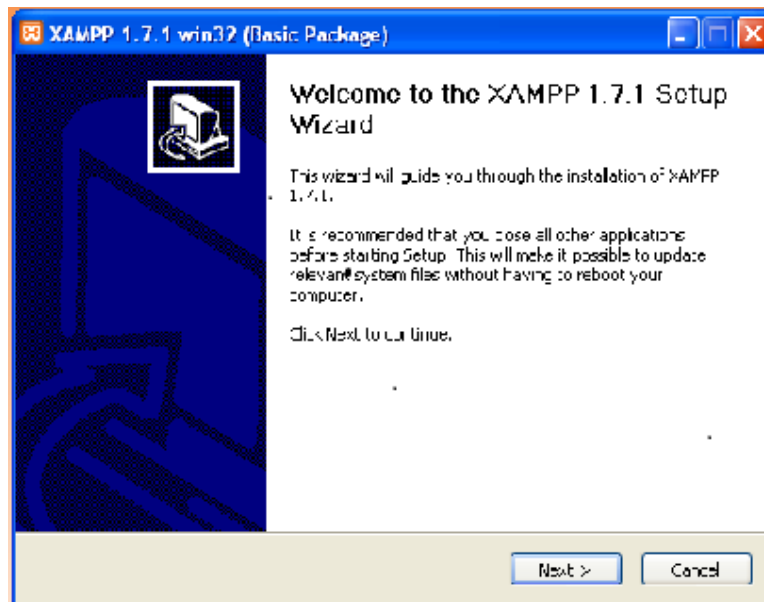


Figura 4.2.1 Pantalla de instalación de XAMPP

Seleccionamos la ruta para hacer la instalación, como se muestra en la figura 4.2.2

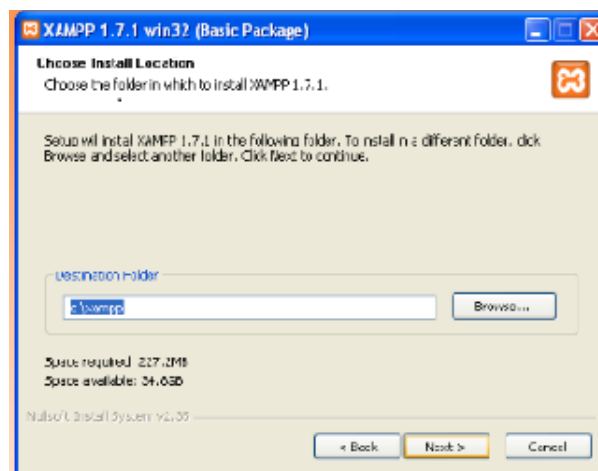


Figura 4.2.2 Selección de ruta de instalación

Configuramos las opciones necesarias, como se muestra en la figura 4.2.3

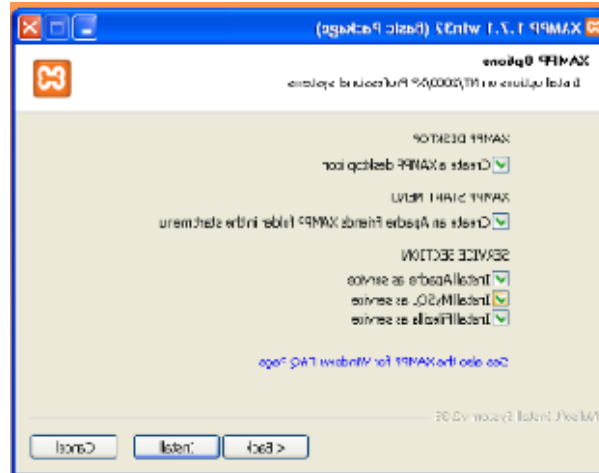


Figura 4.2.3 Configuración de XAMPP

Finalmente la instalación estará lista, como se muestra en la figura 4.2.4

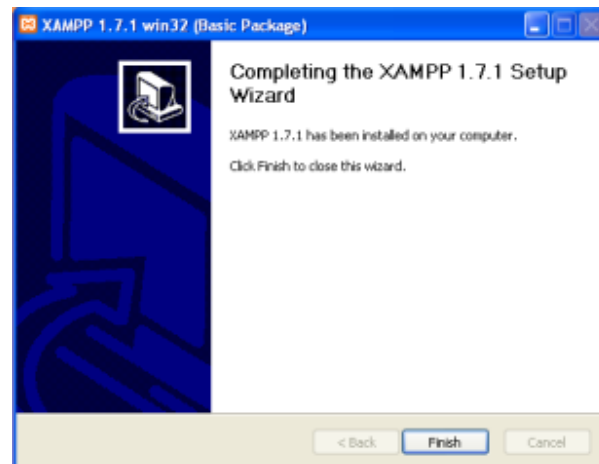


Figura 4.2.4 Finalizando la instalación de XAMPP

El SIAH contará con una base de datos para la manipulación de la información, una vez que XAMPP se tenga instalado es necesario hacer la configuración, iniciamos Apache y después MySQL, el sistema mismo se encargará de la configuración, como se muestra en la figura 4.2.5

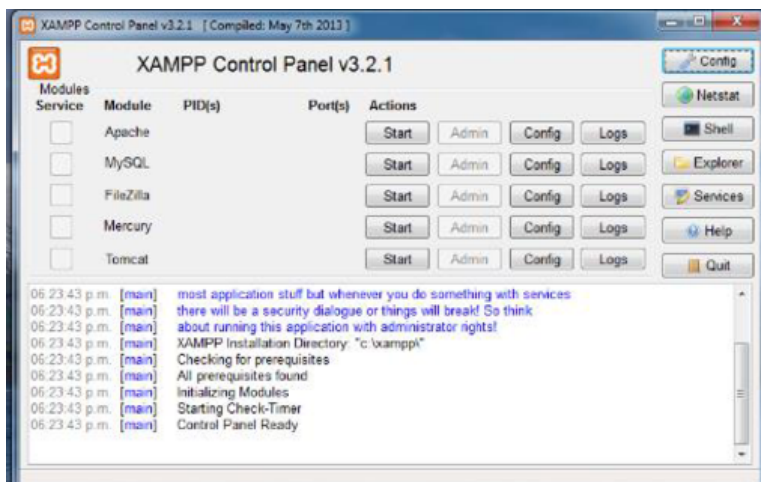


Figura 4.2.5 Configuración de Apache y MySQL

Una vez iniciado XAMPP procedemos a la creación de la base de datos, en la opción de Admin de MySQL; posteriormente creamos las tablas correspondientes, para la creación de la base de datos lo podemos hacer a través de PHPMyAdmin, primeramente inicializamos Apache y después MySQL, una vez hecho esto nos dirigimos a Admin de MySQL, vamos a la pestaña de base de datos, enseguida crear base de datos, damos nombre a la base de datos y la opción cotejamiento, ahora crear; y la base de datos estará creada a través de la misma aplicación, como se muestra en la figura 4.2.6:

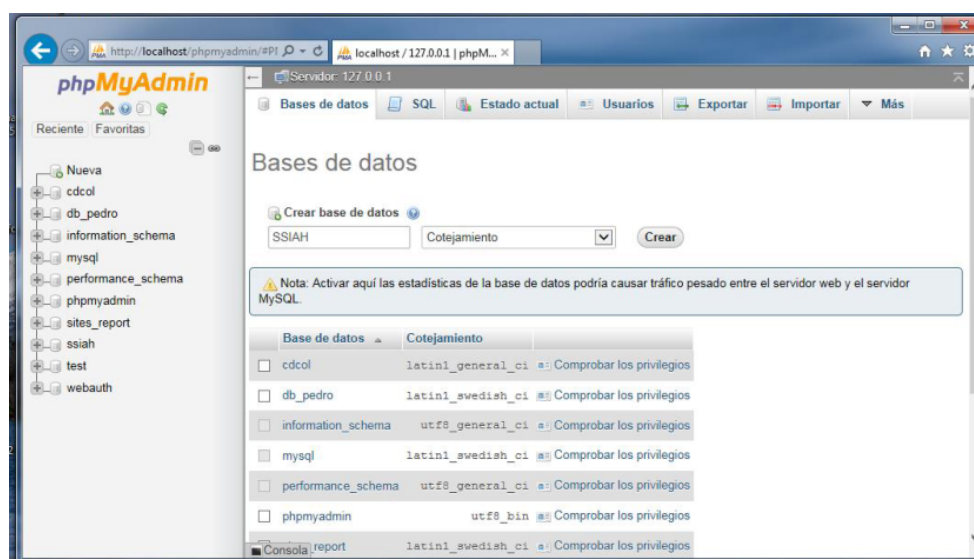


Figura 4.2.6 Creación de la base de datos a través de PHPMyAdmin

Para crear las tablas hacemos lo siguiente, ya creada la base de datos ahora la abrimos, enseguida aparece la pestaña estructura, crear tabla, nos pide el nombre de la tabla y nos pregunta el número de columnas que tendrá la tabla, le damos nombre y decimos continuar, después nos pide la estructura de la tabla con todas las propiedades, llenamos y guardamos los cambios, como se muestra en la figura 4.2.7:

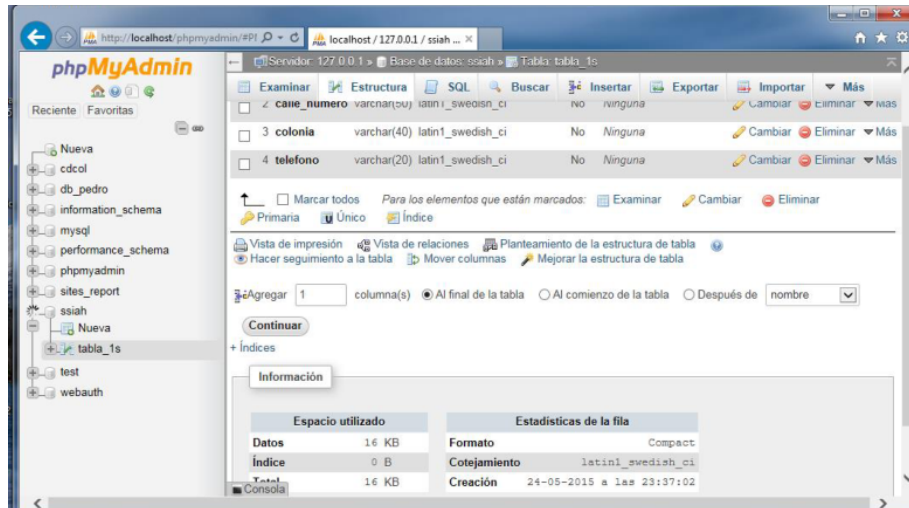


Figura 4.2.7 Creación de tablas a través de PHPMyAdmin

A través de MySQL la sentencia es:

```
CREATE TABLE 'ssiah'. 'tabla_1s'
```

```
('nombre' VARCHAR (60) NOT NULL,
```

```
'calle_numero' VARCHAR(50),
```

```
'colonia' VARCHAR(60) NOT NULL,
```

```
'telefono' VARCHAR(20) NOT NULL,
```

```
PRIMARY KEY ('nombre' (60))) ENGINE = InnoDB ;
```

PHPMyAdmin nos permite generar consultas, esto significa, una vista de algunos campos de la alguna tabla de la base de datos que cumplan con un criterio dado, para hacer esto necesitamos primeramente iniciar nuestro PHPMyAdmin; si es que no lo



hubiésemos hecho, activamos Apache y MySQL, accedemos en la parte de MyAdmin de MySQL, para llegar a la pantalla de PHPMyAdmin,

Un trigger en MySQL es un programa almacenado, creado para ejecutarse automáticamente cuando ocurra un evento en nuestra base de datos, los eventos son generados por los comandos INSERT, UPDATE y DELETE los cuales son parte del DML (Data Modeling Lenguaje) de SQL. Para la creación de triggers en MySQL lo podemos hacer de la siguiente manera:

```
CREATE [DEFINER={usuario|CURRENT_USER}]  
TRIGGER nombre_del_trigger {BEFORE|AFTER} {UPDATE|INSERT|DELETE}  
ON nombre_de_la_tabla  
FOR EACH ROW
```

```
DEFINER={usuario|CURRENT_USER}
```

Indica al gestor de base de datos que usuario tiene privilegios en su cuenta para la invocación de los triggers cuando surjan los eventos DML; por omisión esta característica tiene el valor CURRENT_USER que hace referencia al usuario actual que está creando el trigger

Nombre_del_trigger

Se tiene una nomenclatura para nombrar al trigger, primero se pone el nombre de la tabla, después especificamos con la inicial de la operación DML finalmente se usa la inicial del momento de ejecución (AFTER o BEFORE). UPDATE, INSERT o DELETE; elegimos una de ellas, FOR EACH ROW , establece que el trigger se ejecute por cada fila en la tabla asociada, bloque_de_instrucciones define el bloque de sentencias que el trigger ejecutará al ser invocado. Si queremos relacionar el trigger con columnas específicas

Con los triggers podemos implementar varios casos de uso que mantengan la integridad de la base de datos como validar información, calcular atributos derivados, seguimiento de movimientos en la base de datos. Para generar un trigger mediante

XAMPP lo hacemos de la siguiente manera; primeramente necesitamos abrir la tabla sobre la cual se generará el trigger, enseguida buscamos la opción disparadores (trigger), ahora nuevo disparador, le damos un nombre al disparador, se define la tabla sobre la cual va a operar el disparador, se define el tiempo de operación (BEFORE, AFTER), el tipo de evento (INSERT, UPDATE, DELETE), se establece la definición y finalmente el definidor, como se muestra en la figura 4.2.8

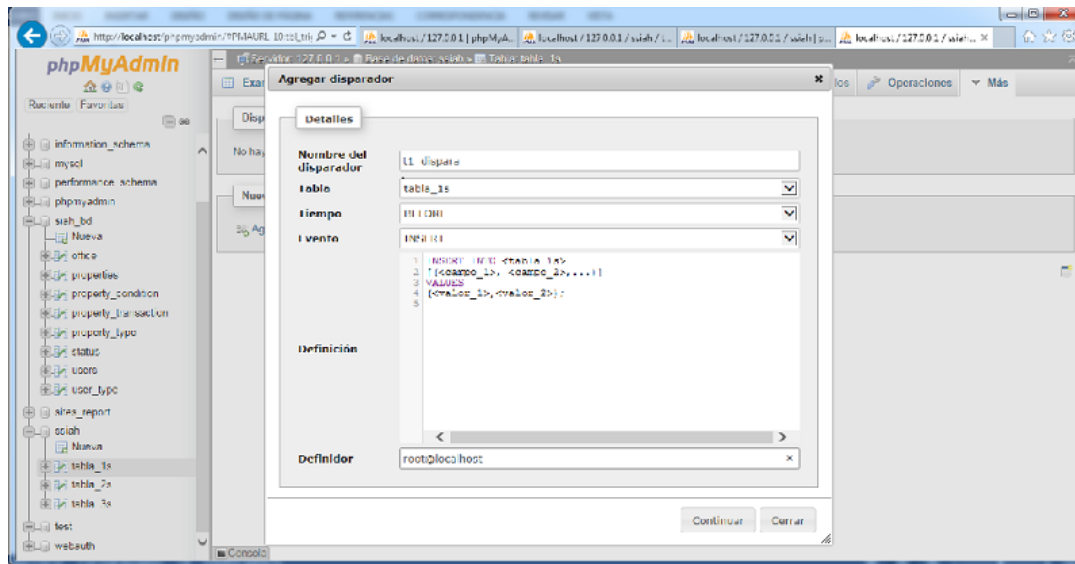


Figura 4.2.8 Definición de trigger mediante XAMPP

Un store procedure (procedimiento almacenado) es un procedimiento o programa almacenado físicamente en una base de datos, su implementación varía entre los diversos gestores de bases de datos, la ventaja es que al ser ejecutado en respuesta a una petición del usuario es ejecutado directamente en el motor de base de datos el cual usualmente corre en un servidor separado, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes. Cuando una base de datos es manipulada desde muchos programas externos, al incluir la lógica de la aplicación en la base de datos utilizando procedimientos almacenados, la necesidad de embeber la misma lógica en todos los



programas que acceden a los datos es reducida, lo cual puede simplificar la creación y particularmente el mantenimiento de los programas involucrados.

Desde MySQL 5 los procedimientos almacenados empezaron a ser soportados, las sentencias se ejecutan luego de escribir el signo punto y coma, por esta razón antes de escribir el procedimiento almacenado la función del punto y coma se asigna a otros caracteres usando la sentencia DELIMITER seguida de un carácter tal como |, de esta manera el procedimiento puede ser escrito usando los punto y coma sin que se ejecute mientras se escribe, después de escrito el procedimiento se escribe nuevamente la sentencia DELIMITER; para asignar al punto y coma su función habitual. En el PHPMyAdmin podemos crear el store procedure, primeramente se selecciona la tabla, se selecciona la opción de SQL, se escribe el código en la ventana de consulta, se especifica el delimitador, finalmente se ejecuta, como se muestra en la figura 4.2.9

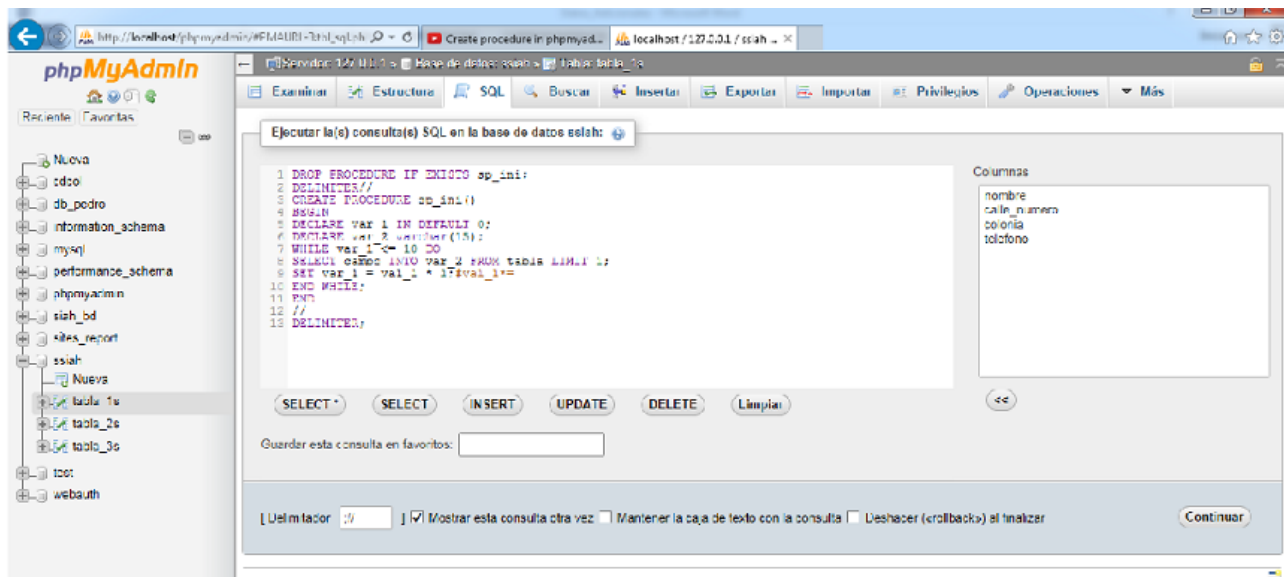


Figura 4.2.9 Creación de Store Procedure mediante XAMPP

Una consulta SQL es un tipo de consulta a una base de datos empleando lenguaje SQL, una consulta SQL básica puede constar con un máximo de seis cláusulas de las cuales solo dos son obligatorias, SELECT y FROM, las cláusulas se especifican en el siguiente orden:

- SELECT <lista de atributos>
- FROM <lista de tablas> (especifica todas las tablas que se necesitan en la consulta)
- WHERE <condición> (especifica las condiciones para seleccionar tuplas de esas relaciones, incluyendo las condiciones de reunión)
- GROUP BY <atributos de agrupación> (especifica atributos de agrupación)
- HAVING <condición de agrupación> (especifica una condición que deben de cumplir los grupos seleccionados, las funciones agregadas integrales COUNT, SUM, MIN, MAX Y AVG se usan junto con la agrupación)
- ORDER BY <lista de atributos> (especifican un orden para presentar el resultado de una consulta)

La instrucción que permite leer en una tabla es select , así se tiene:

```
SELECT * FROM persona
```

Dónde SELECT selecciona o lee

* Todos los campos

FROM de o en

Persona nombre de la tabla

Cuando usamos PHPMyAdmin podemos efectuar consultas de igual manera, para ello necesitamos seleccionar la tabla sobre la que haremos la operación, ahora seleccionamos SQL y ejecutamos, los datos serán mostrados en la ventana correspondiente, como se observa en la figura 4.2.10.

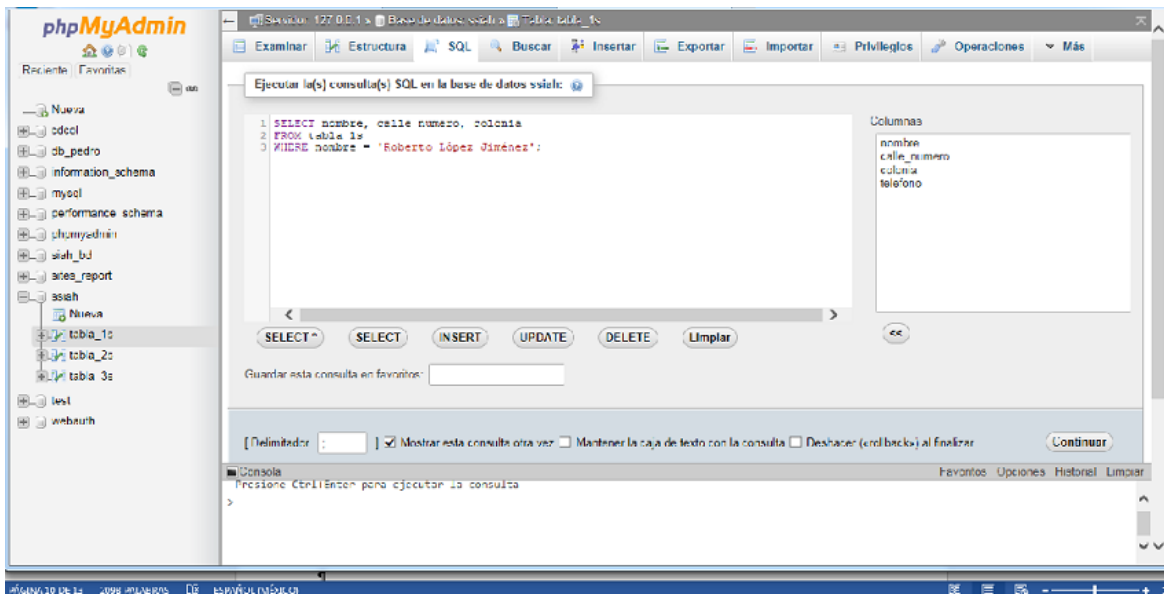


Figura 4.2.10 Consulta hecha en PHPMyAdmin

El resultado se muestra después de la ejecución, como se muestra en la figura 4.2.11



Figura 4.2.11 Resultado de la consulta hecha en la Figura 4.2.10.

4.3 Desarrollo del front-end del SIAH.

El diseño de interfaz de usuario o ingeniería de la interfaz es el diseño de computadoras, aplicaciones, máquinas, dispositivos de comunicación móvil, aplicaciones de software, y sitios web enfocado en la experiencia de usuario y la interacción.

Normalmente es una actividad multidisciplinar que involucra a varias ramas es decir al diseño y el conocimiento como el diseño gráfico, industrial, web, de software y la ergonomía; y está implicado en un amplio rango de proyectos, desde sistemas para computadoras, vehículos hasta aviones comerciales.

Su objetivo es que las aplicaciones o los objetos sean más atractivos y además, hacer que la interacción con el usuario sea lo más intuitiva posible, conocido como el diseño centrado en el usuario. En este sentido las disciplinas del diseño industrial y gráfico se encargan de que la actividad a desarrollar se comunique y aprenda lo más rápidamente, a través de recursos como la gráfica, los pictogramas, los estereotipos y la simbología, todo sin afectar el funcionamiento técnico eficiente.

La interfaz de usuario es el conjunto de pantallas que permiten la interacción del usuario con el sistema.

La facilidad de uso de una interfaz gráfica no depende de los elementos estéticos o modernos, la arquitectura de la interfaz se da cuando se cumple con las necesidades de las personas que la usaran, es por ello que para el diseño de la interfaz de usuario del SIAH, nos apegaremos a los requerimientos por parte del cliente, principalmente en la definición de las tareas que los usuarios finales deben realizar para el correcto desempeño de sus labores y el contenido que es necesario que se presente como parte de la interfaz.

El portal comienza con una página de autenticación como se muestra en la figura 4.3.1.

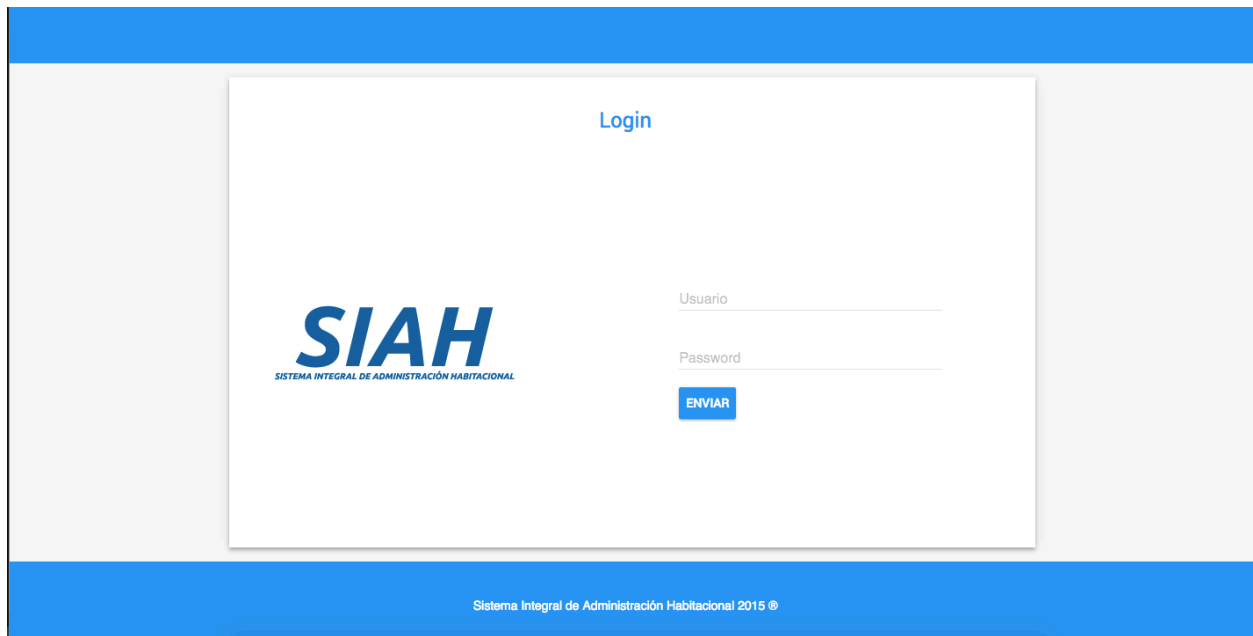


Figura 4.3.1 Autenticación del usuario

El sistema debe de validar la información ingresada y en caso de ser inválida deberá mandar un mensaje de error como se muestra en la figura 4.3.2.

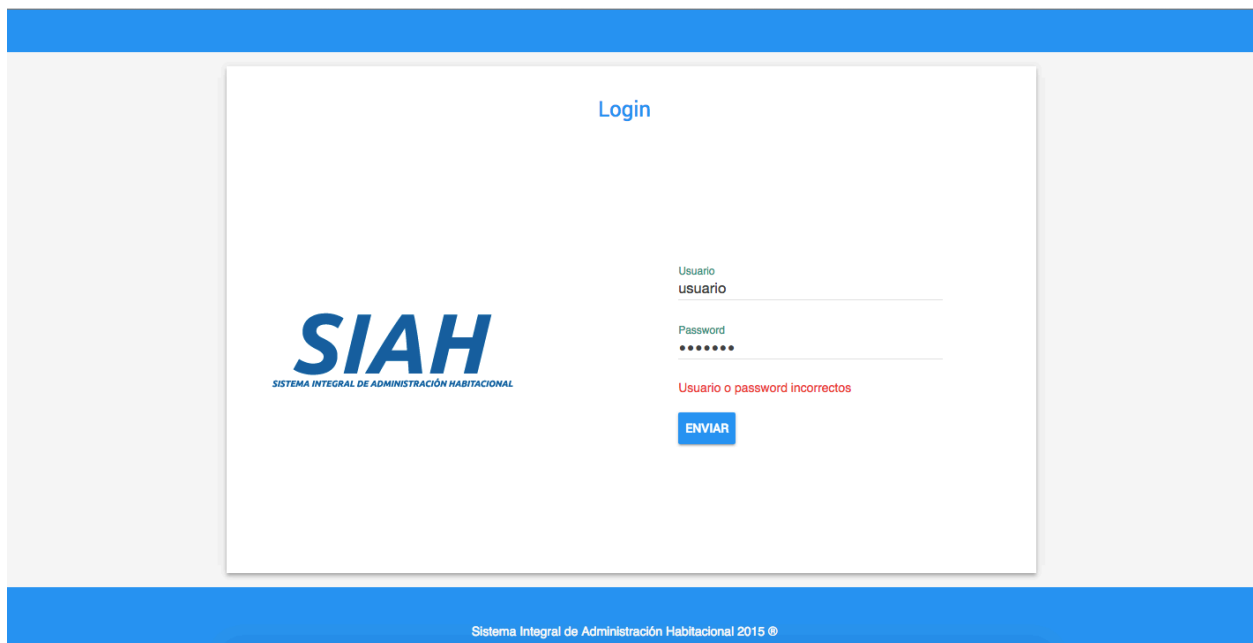


Figura 4.3.2 Error de autenticación del usuario

Una vez validado el usuario y la contraseña se tiene la pantalla principal del módulo donde se desplegará el menú de acciones y el área de trabajo como se muestra en la figura 4.3.3.



Figura 4.3.3 Página principal del sitio

El sistema se divide en 4 módulos que son:

- **Oficinas:** Aquí se podrán buscar, modificar y agregar las oficinas que tiene la inmobiliaria.
- **Propiedades:** Aquí se podrán buscar, modificar y agregar las propiedades que tiene la inmobiliaria.
- **Usuarios:** Aquí se podrán buscar, modificar y agregar los usuarios que harán uso de la aplicación.
- **Reportes:** Aquí se podrán generar los reportes sobre las rentas, ventas, oficinas y asesores.

La navegación se realiza en los botones del lado izquierdo de la página, el primer módulo que se tiene es el de Oficinas, como se muestra en la figura 4.3.4.

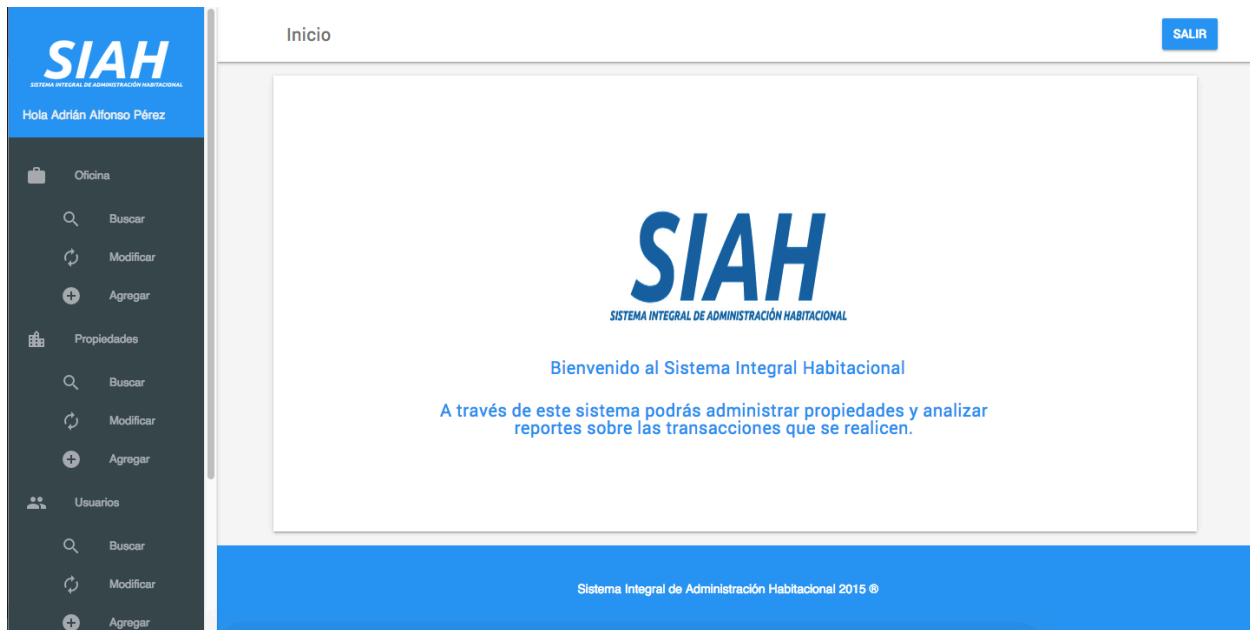


Figura 4.3.4 Submenú oficinas del sistema SIAH

Este submenú está compuesto de los siguientes apartados:

- **Buscar:** se despliega un formulario de búsqueda de oficinas colocando la información solicitada como es nombre, ciudad y estado como se muestra en la figura 4.3.5.

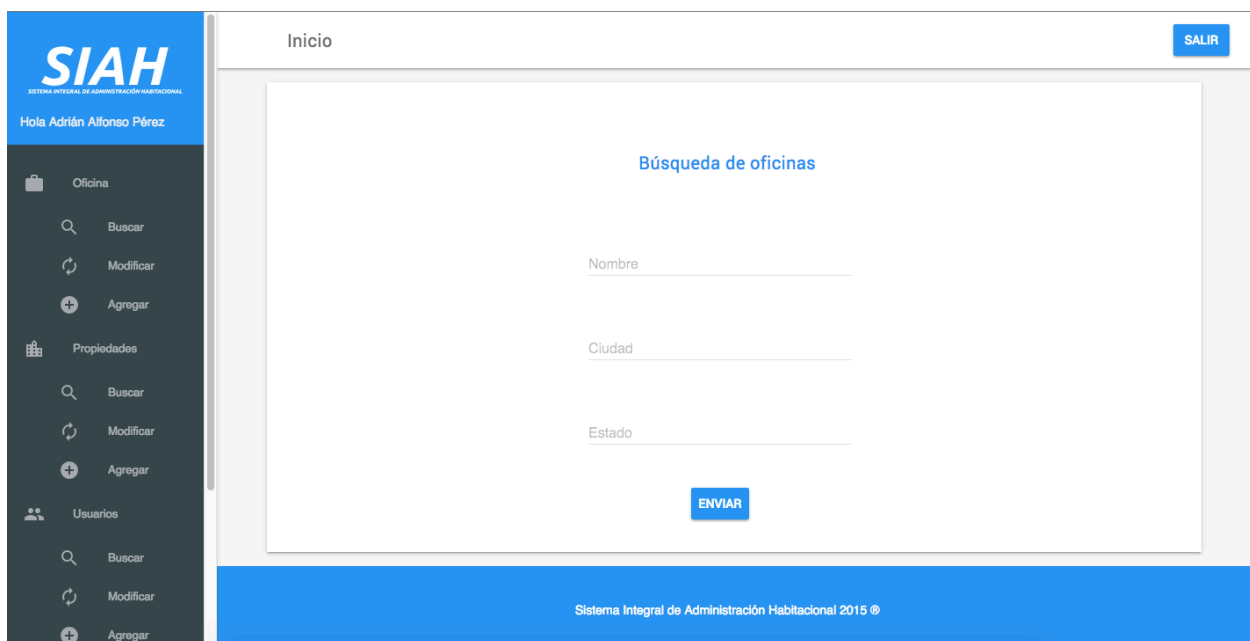


Figura 4.3.5 Apartado buscar del submenú oficinas del sistema SIAH

- **Modificar:** se despliegan las oficinas que se tienen dadas de alta como se muestra en la Figura 4.3.6 y al dar seleccionar el botón “modificar” da la opción de modificar los campos de esta de acuerdo a la información que ya se tiene guardada en la base de datos como se muestra en la figura 4.3.7.



Figura 4.3.6 Apartado modificar del submenú oficinas del sistema SIAH

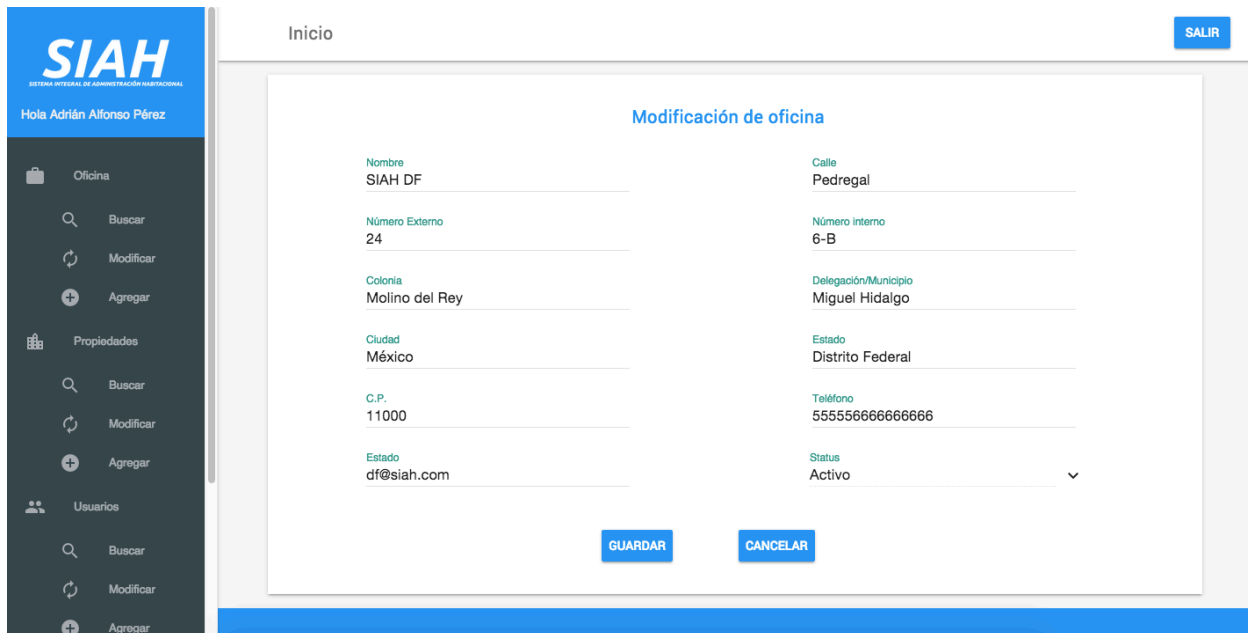


Figura 4.3.7 Apartado modificar del submenú oficinas del sistema SIAH



- Agregar: se despliega la opción de ingresar nuevas oficinas a través de un formulario en el cual se ingresarán los datos de la nueva oficina como se muestra en la figura 4.3.8.

Figura 4.3.8 Apartado agregar del submenú oficinas del sistema SIAH

Esto es igual para el submenú propiedades y el submenú usuarios.

Nuestro principal submenú es el de reportes, este es de gran relevancia, ya que aquí se despliega toda la información en gráficos, esta información es obtenida de acuerdo a las ventas o rentas que los asesores han obtenido en cada oficina.

El submenú reportes está compuesto de los siguientes apartados como se muestra en la Figura 4.3.9:

- Ventas: despliega gráficas de las ventas obtenidas.
- Rentas: despliega gráficas de acuerdo a las rentas obtenidas.

- Oficinas: despliega gráficas de acuerdo a los resultados obtenidos en cada oficina.
- Asesores: muestra reportes de los resultados de cada asesor.

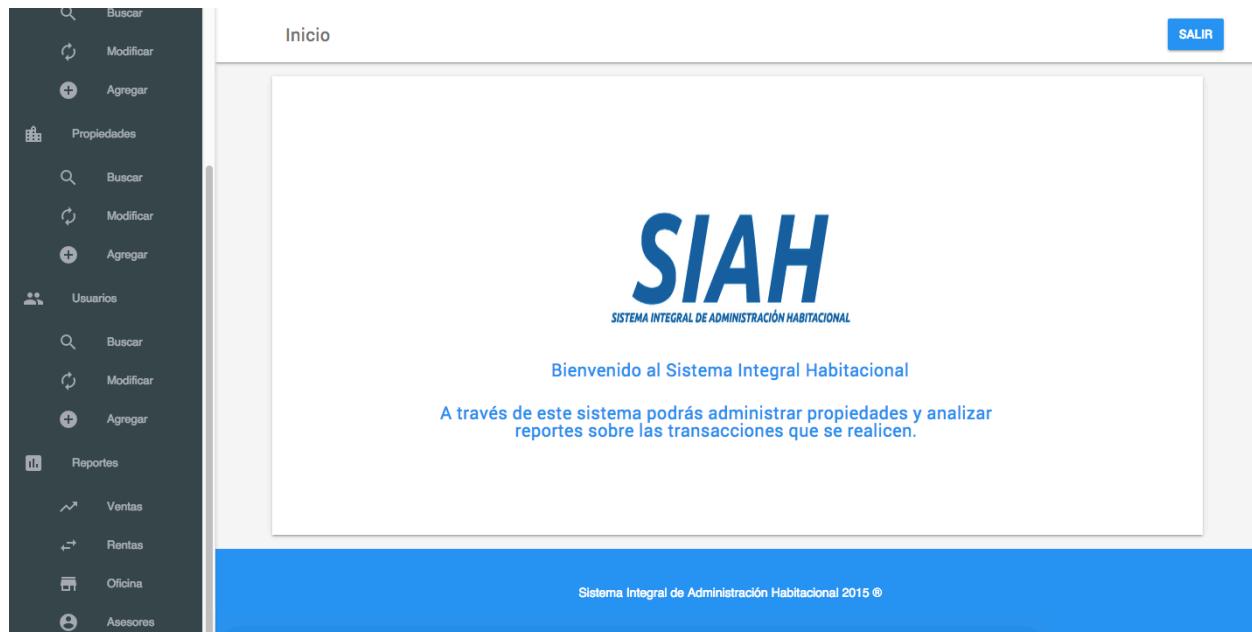


Figura 4.3.9 Submenú reportes del sistema SIAH



4.4 Aplicación de pruebas y soporte.

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general finitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles errores de implementación, calidad o usabilidad de un programa; probando el comportamiento del mismo.

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

La prueba de software es un elemento crítico para la garantía del correcto funcionamiento del software. Entre sus objetivos están:

- Detectar defectos en el software.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- Diseñar casos de prueba que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Las pruebas se rigen por una serie de principios, a continuación se citan:

- La prueba puede ser usada para mostrar la presencia de errores, pero nunca su ausencia.
- La principal dificultad del proceso de prueba es decidir cuándo parar.
- Evitar casos de pruebas no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.

- Una parte necesaria de un caso de prueba es la definición del resultado esperado.
- Los casos de pruebas tienen que ser escritos no sólo para condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.
- El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.

Se cuentan con diversos métodos de prueba, de los cuales se mencionan los más significativos a continuación:

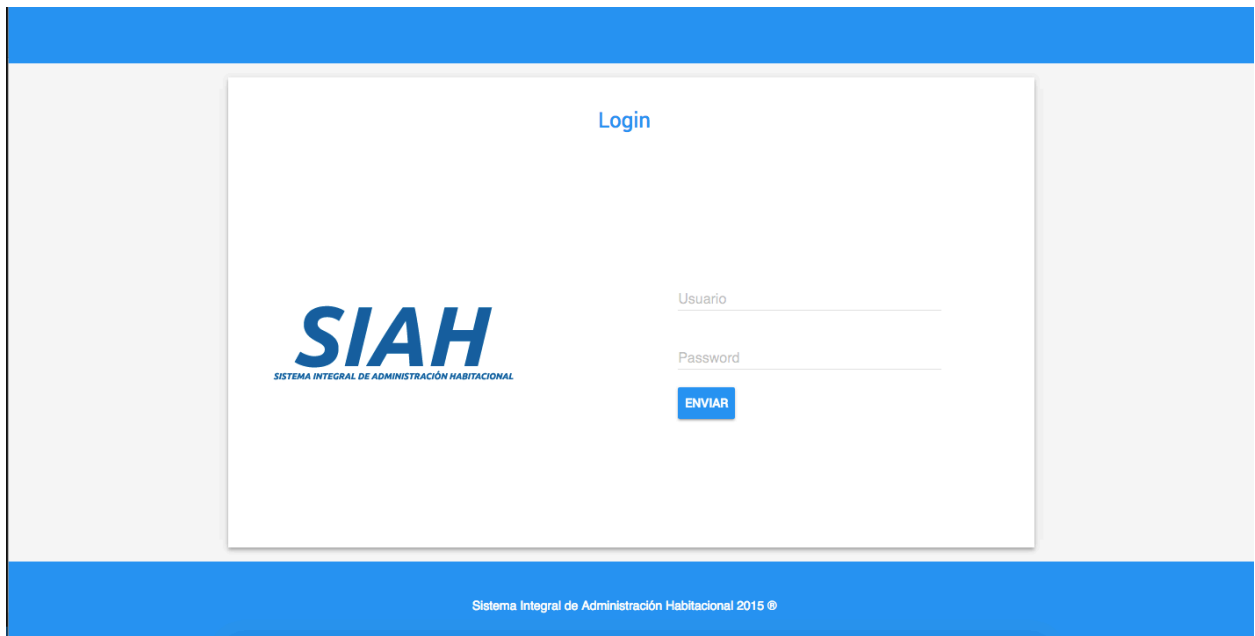
- Prueba de la caja negra. Se enfoca directamente en la función del programa, sin importar la codificación. Se aplica a los módulos para verificar que realizan la función para la cual fueron programados. El método de la caja negra se centra en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del programa.
- Prueba de la caja blanca. Las pruebas de caja blanca se centran en los detalles procedurales del software, por lo que su diseño está fuertemente ligado al código fuente. Se eligen distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa y verificar que se obtienen los valores de salida adecuados. Al total de pruebas de caja blanca se le llama cobertura. La cobertura es un número porcentual que indica cuánto código del programa se ha probado.
- Prueba de integración. Se lleva a cabo durante el proceso de construcción del sistema, involucra un número creciente de módulos lo que permite probar el sistema como conjunto. El objetivo de las pruebas de integración es verificar el correcto ensamblaje entre los distintos componentes una vez que han sido probados unitariamente con el fin de comprobar que interactúan correctamente a través de sus interfaces, tanto internas como externas.



- Prueba de regresión. A través de esta prueba se buscan las causas de nuevos errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software inducidos por cambios recientemente realizados en partes de la aplicación el cual se genera de manera inesperada.
- Prueba de validación. Es el proceso de revisión que el sistema de software generado cumple con las especificaciones y cometido. Utiliza técnicas tales como evaluaciones, inspecciones, y tutoriales. La validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales de comportamiento y rendimiento.
- Prueba de estrés. Estas pruebas se ejecutan en el sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales de información. Tiene como finalidad someter la aplicación a situaciones irregulares o poco comunes.
- Prueba alfa. Las pruebas alfa consisten en la utilización del sistema por parte de usuarios que no formen parte del equipo de desarrollo. Se trabaja en un entorno controlado que permite la utilización del sistema y generación de resultados para análisis posterior.
- Prueba beta. Las pruebas betas se realizan en ambientes más cercanos al uso final del sistema. A diferencia de la prueba alfa, el desarrollador no está presente durante la realización de la prueba. El usuario debe registrar los problemas que detecte durante la realización de la prueba.

A continuación se describen la pruebas que se realizaron al sistema para verificar su correcto funcionamiento:

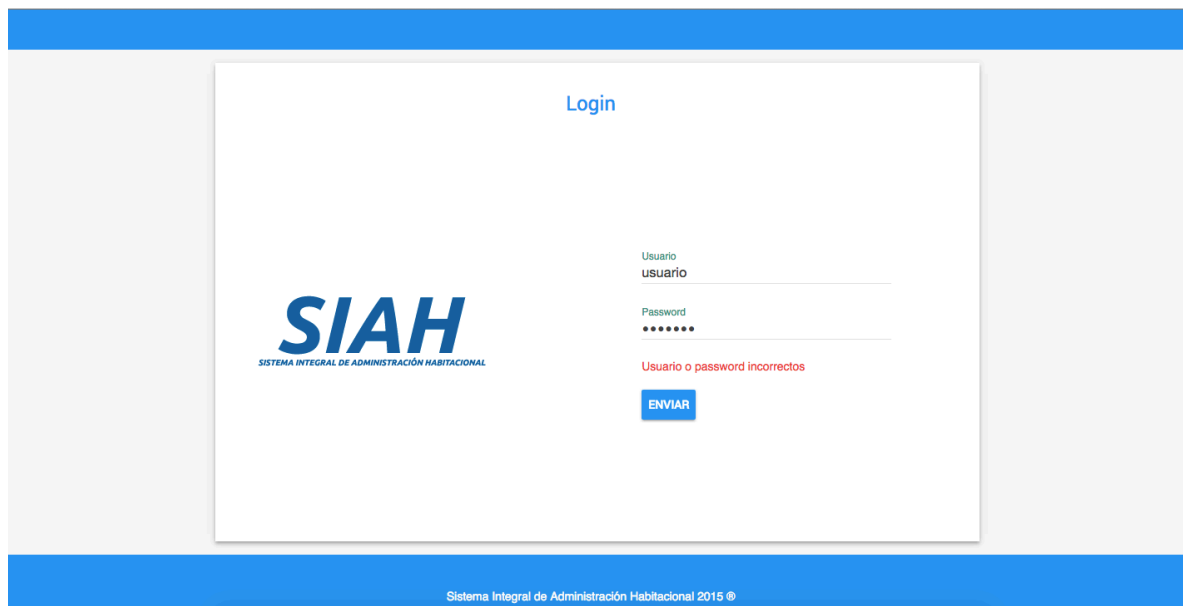
- Prueba de caja negra. Para ingresar a la aplicación, el usuario debe completar los campos de nombre y contraseña como se muestra en la figura 4.4.1



The screenshot shows the login interface for the SIAH system. At the top, the word "Login" is displayed in blue. On the left side, the SIAH logo is prominent, consisting of the letters "SIAH" in a large, bold, blue font, with the full name "SISTEMA INTEGRAL DE ADMINISTRACIÓN HABITACIONAL" in a smaller font below it. To the right of the logo, there are two input fields: "Usuario" and "Password", each with a horizontal line for text entry. Below these fields is a blue button labeled "ENVIAR". At the bottom of the page, a blue footer bar contains the text "Sistema Integral de Administración Habitacional 2015 ©".

Figura 4.4.1 Login

Si el usuario ingresa datos inválidos se espera que el sistema indique que ocurrió algún error en usuario o contraseña como se muestra en la figura 4.4.2.



This screenshot shows the same login interface as Figure 4.4.1, but with an error message. The "Usuario" field now contains the text "usuario" and the "Password" field contains a series of dots. Below the password field, a red error message reads "Usuario o password incorrectos". The "ENVIAR" button remains visible. The footer bar at the bottom still displays "Sistema Integral de Administración Habitacional 2015 ©".

Figura 4.4.2 Página de error de inicio de sesión.

Si el usuario proporciona datos válidos, el sistema permitirá el acceso a la parte inicial de la aplicación así como al menú de selección como se muestra en la figura 4.4.3

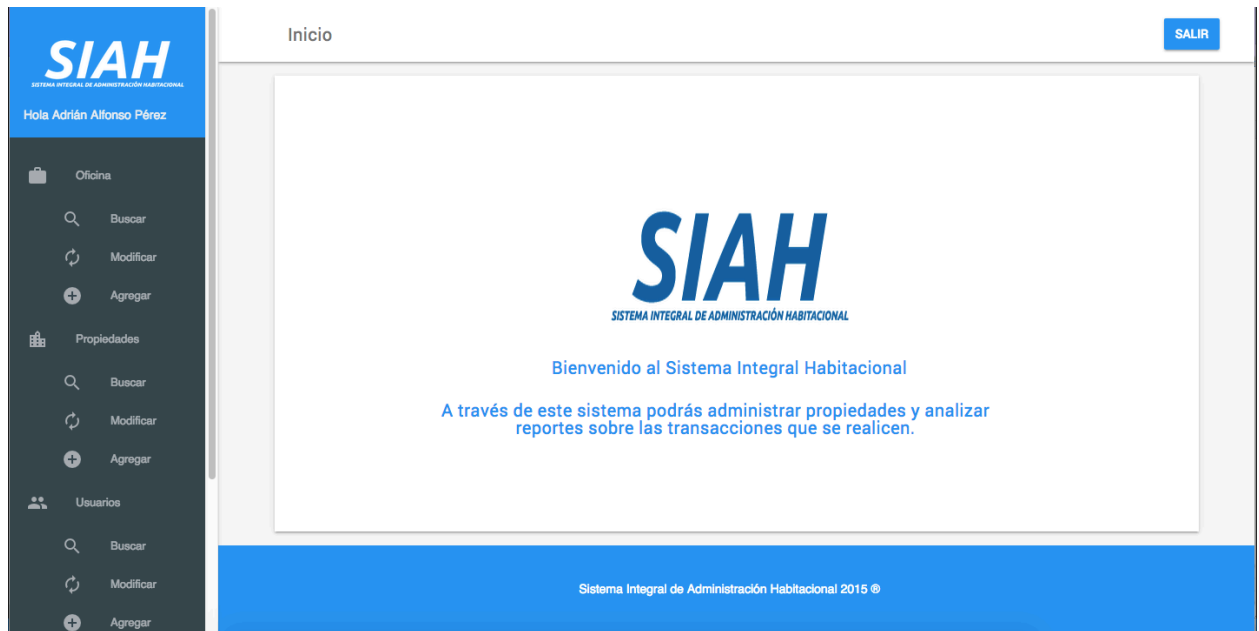


Figura 4.4.3 Página de inicio.

- Prueba de la caja blanca. Se realizarán pruebas al código encargado de realizar la validación de ingreso de usuario. Como se muestra en la figura 4.4.4.

```

$user = sanitize($_POST['username']);
$password = sanitize($_POST['password']);
if(isset($user) && isset($password)){
    $encrypted=md5($password);
    $query = "SELECT u_id,u_name,pass,type,name,last_name_1 FROM users
    WHERE u_name='". $user . "' AND pass='". $encrypted . "'";
    $result = $db->executeQuery($query);
    if($result && mysql_num_rows($result) > 0){
        $row = mysql_fetch_array($result);
    }
}
    
```

```
session_start();

$_SESSION['user'] = $row['u_name'];

$_SESSION['name'] = $row['name'] . " " . $row['last_name_1'];

session_cache_expire(30);

if($row [1] == 0)   echo "1";

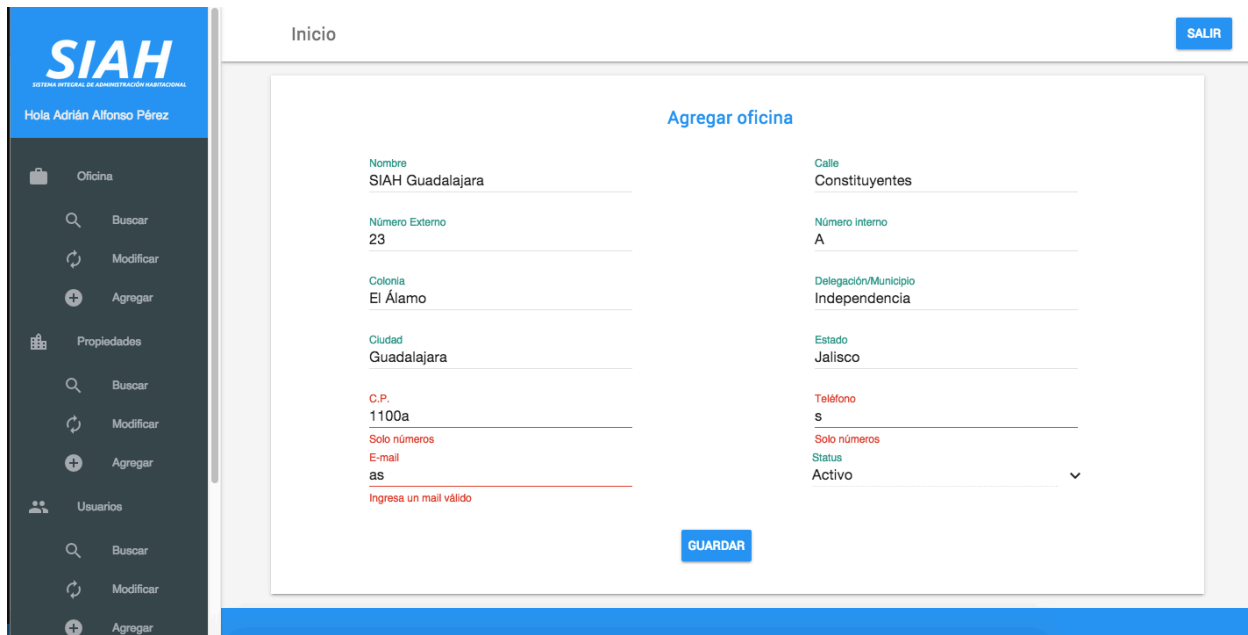
else   echo "0";

}   else   echo "0";

}   else   echo "0";
```

Figura 4.4.4 Código ingreso

- Prueba de validación. Para la prueba de validación se utilizó el módulo de ingreso de datos de oficinas. El sistema es capaz de enviar mensajes al usuario cuando no ha completado todos los campos de manera correcta como se muestra en la figura 4.4.5 y 4.4.6



The screenshot shows the 'Agregar oficina' form in the SIAH application. The form is titled 'Agregar oficina' and is located on the 'Inicio' page. The form contains the following fields and values:

Field	Value	Validation
Nombre	SIAH Guadalajara	
Calle	Constituyentes	
Número Externo	23	
Número interno	A	
Colonia	El Álamo	
Delegación/Municipio	Independencia	
Ciudad	Guadalajara	
Estado	Jalisco	
C.P.	1100a	
Teléfono	s	
E-mail	as	Ingresar un mail válido
Status	Activo	

The 'GUARDAR' button is visible at the bottom of the form.

Figura 4.4.5 Ingresar oficina

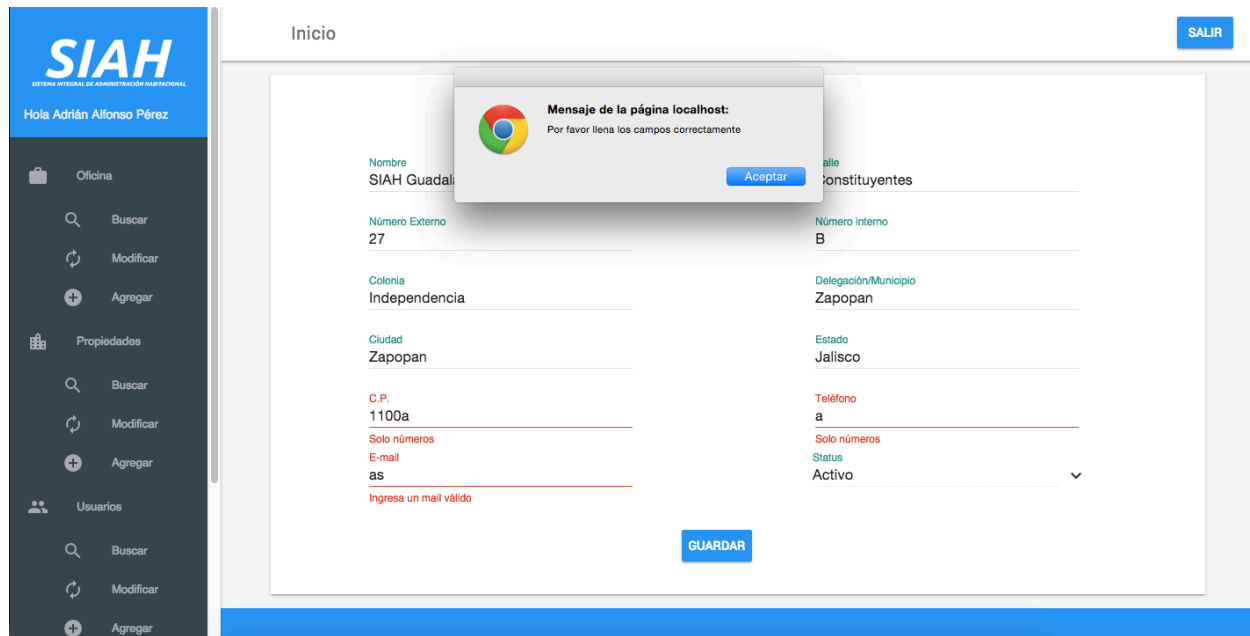


Figura 4.4.6 Mensaje de error

El soporte y mantenimiento de un sistema es el proceso de mejora y optimización de la aplicación después de su entrega al usuario final. Incluye corrección y prevención de defectos. El mantenimiento de software es una de las fases en el ciclo de vida de desarrollo de sistemas. La fase de mantenimiento se aplica después de la implementación del software en el ambiente real de trabajo.

Se cuenta con diversos métodos de soporte y mantenimiento los cuales se mencionan a continuación:

- Soporte preventivo. Se trata de la modificación del software para mejorar sus funciones sin alterar sus especificaciones iniciales, se deben incluir sentencias que comprueben la validez de los datos de entrada, reestructuración de los programas para aumentar su legibilidad o incluir nuevos comentarios. Este soporte tiene lugar antes de que ocurra una falla, se efectúa bajo condiciones controladas sin la existencia de algún error en el sistema.

- Soporte correctivo. Tiene por objetivo localizar y eliminar los posibles defectos de los programas. Un defecto en un sistema es una característica del sistema con el potencial de provocar un fallo. Un fallo se produce cuando el comportamiento de un sistema difiere con respecto al comportamiento definido en la especificación. Este soporte se efectúa cuando ocurre una falla y solo actuará cuando se presenta un error en el sistema.
- Soporte perfectivo. Este tipo de mantenimiento se da cuando el sistema se encuentra estable y es utilizado a su completa capacidad. Son un conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.
- Soporte vertical. Es el proceso por el cual se mejora y optimiza el software que se ha instalado, además se realiza la prevención de posibles problemas que pueden llegar a surgir a medida que se utiliza. Actualmente este mantenimiento se realiza en busca de fallas que pueden causar la inestabilidad del sistema, bajas en el rendimiento, pérdida de productividad, cortes en los sistemas y probables errores en el mismo.
- Soporte horizontal. Este tipo de mantenimiento se enfoca al hardware, es decir a todos los componentes físicos y es necesario llevarlo a cabo para que el usuario, haga un mejor uso de los equipos de cómputo.



4.5 Obtención de reportes y estadísticas

Nuestro sistema permite obtener reportes y estadísticas que puedan apoyar a los gerentes y dueños de oficinas a evaluar el desempeño de las ventas y rentas efectuadas, así como el desempeño individual de cada uno de los asesores dentro de la oficina como se muestra en la figura 4.5.1

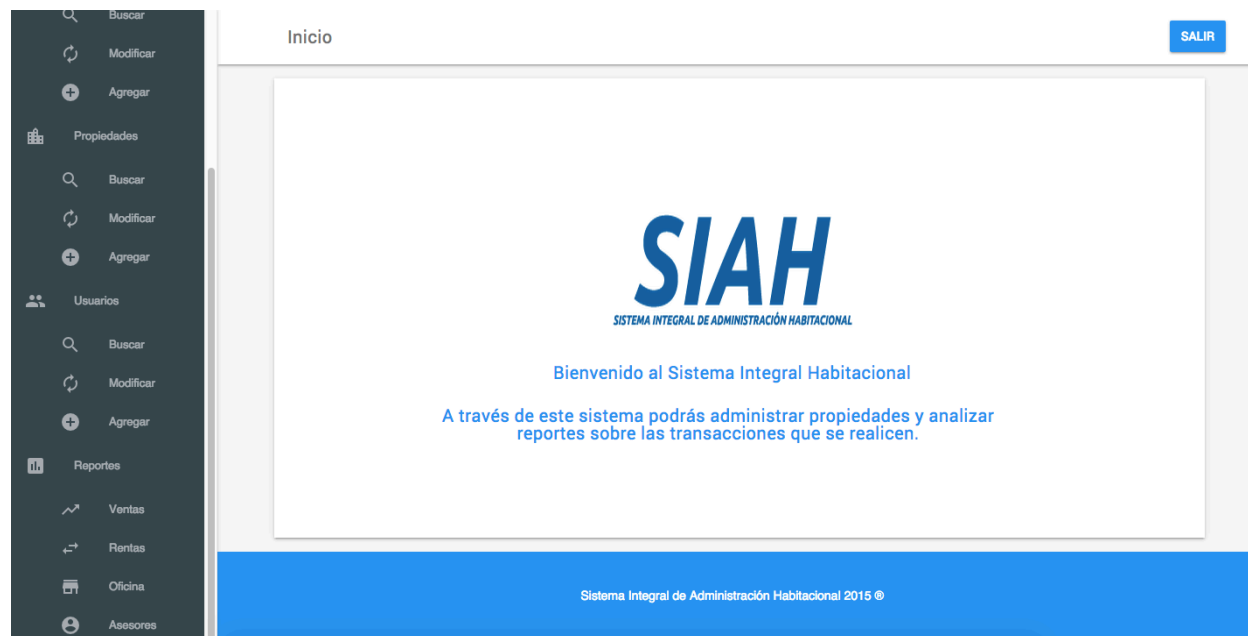


Figura 4.5.1 Reportes

Los administradores tienen la posibilidad de analizar el desempeño de todas las oficinas y usuarios en conjunto.

Los reportes de ventas y rentas incluyen un desglose de precio, años de las propiedades, tamaño de la propiedad, metros construidos, niveles, cuartos, baños, estacionamientos, condición y tipo del inmueble como se muestra en la figura 4.5.2



Figura 4.5.2 Ventas

Los reportes por oficina incluyen los datos en conjunto de todos los inmuebles y asesores como se muestra en las figuras 4.5.3 y 4.5.4

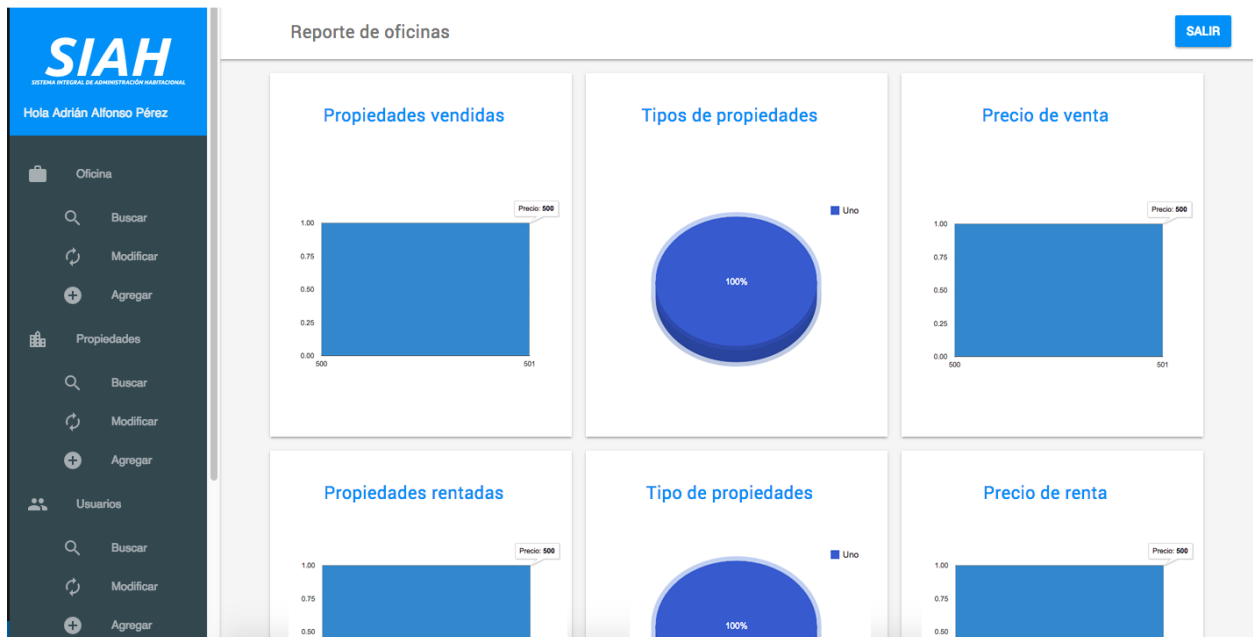


Figura 4.5.3 Oficinas



Figura 4.5.4 Oficinas

Los reportes por usuario incluyen el desglose de las ventas y rentas realizadas por los asesores de manera individual como se muestra en la figura 4.5.5



Figura 4.5.4 Usuarios

Conclusiones

- El desarrollo de nuestro SIAH como un sistema web nos ofrece muchas ventajas, ya que además de desarrollar aplicaciones complejas que cumplan con las necesidades que requiere el cliente, también cumple con los requerimientos de seguridad, rapidez y facilidad de uso.
- El SIAH es una herramienta bastante útil en el momento de una toma de decisiones, ya que permite conocer dónde se pueden aplicar las mejoras, ya sea en una oficina o directamente a un asesor. El sistema ayuda a que la inmobiliaria tome decisiones con base a los reportes generados, por ejemplo, se puede llegar a mover a un asesor para subir el potencial de una oficina.
- Con el desarrollo de esta aplicación web observamos que las tecnologías emergentes abren una nueva ventana de posibilidades y que actualmente en el campo laboral ya existen bastantes ejemplos de aplicaciones web complejas, tales como las instituciones bancarias, algunas dependencias gubernamentales y la infinidad de tiendas que cuentan con una infraestructura muy desarrollada de aplicaciones y servicios a través de la web.
- La anterior conclusión nos lleva a otra, ya que hoy en día toda pequeña, mediana o gran empresa, busca ofrecer tanto a sus clientes como a sus empleados o trabajadores un mejor servicio, así de una mejor manera, administrar sus recursos y los procesos que se deban cumplir. Nuestra aplicación ayuda a tener una mejor administración de las oficinas, usuarios y propiedades de la inmobiliaria para después ayudar a la empresa a generar reportes que logren el mejor desempeño de cada oficina y de sus empleados.
- El sistema cumplió de manera satisfactoria con los requerimientos solicitados por el cliente, aprobó correctamente las pruebas realizadas a cada una de las



partes que compone el SIAH, tales como el front-end y cada uno de sus módulos, así como las pruebas a la base de datos.

- Aplicamos nuestros conocimientos teóricos que adquirimos a lo largo de la carrera en el desarrollo de esta aplicación web, así nos damos cuenta que la Facultad de Ingeniería nos da una formación lo más integral posible.
- Como ingenieros de esta facultad, cada uno se pudo dar cuenta que el plan de estudios y las materias impartidas son las correctas para dar al alumno las herramientas necesarias para desarrollar un trabajo de tesis de este tipo.
- Trabajar en equipo para resolver el problema que nos presentaba el desarrollo de este Sistema Integral nos ayudó para pensar en cómo obtener la solución óptima, la cual es posible hallar mediante un buen análisis, metodología y diseño en la construcción de la solución.
- Reforzamos y aumentamos nuestros conocimientos en el área de desarrollo de software, bases de datos, diseño y programación ya que conocimos nuevas tecnologías como Google Charts, Polymer y conceptos de ingeniería más allá de los aprendidos en clases.
- El uso de las buenas prácticas en cuanto a programación, el aplicar las formas normales en nuestra base de datos y el uso de la metodología iterativa-incremental nos ayudó a obtener una solución adecuada en los tiempos estipulados y detectando a tiempo los errores que se presentaron.
- Trabajar de manera conjunta con el cliente ayuda a limitar de manera correcta el alcance de nuestro proyecto, así ambas partes logramos un acuerdo, que la aplicación solucione los requerimientos que el cliente quiere abarcar y nosotros como desarrolladores del proyecto elegimos las tecnologías y herramientas necesarias y suficientes para desarrollar la solución.

- Como resultado del desarrollo de este proyecto se obtuvo la implementación del sitio web para el SIAH, el cual permite realizar la actualización dinámica de información utilizando la tecnología PHP y Bases de Datos MySQL, facilitando la actualización de la información relevante y generando reportes.
- En un mundo tecnológico y con descubrimientos e investigación constante, visualizamos un futuro donde la tecnología se acerque a la sociedad, en especial a los que menos han podido aprovecharla y disfrutarla. Por lo que nuestro objetivo como ingenieros es seguir solucionando problemas y desarrollando aplicaciones que permitan ayudar a la población en distintos ámbitos.
- Con el sistema SIAH, cumple el objetivo de satisfacer las necesidades operativas de la administración, dirección y asesores de la inmobiliaria en el otorgamiento de reportes de las transacciones que realizaron.
- El éxito obtenido al desarrollar este sistema nos ayuda a mejorar la calidad de nuestro trabajo y nos sirve como experiencia para aplicar nuestro conocimiento y habilidades adquiridas en futuros proyectos.



Bibliografía

Título: Manual de prácticas de laboratorio de bases de datos

Autor: Lucila Patricia Arellano Mendoza, Luciralia Hernández Hernández

Editorial: Facultad de Ingeniería

Año: 2010

Título: Manual del Programador, PHP 5, Evolución y Madurez

Autor: Francisco José Minera

Editorial: Colección Manuales Users

Año: 2006

Sitios Web

1. <http://es.scribd.com/doc/464379/Crows-Foot-Notation-ER-Diagrams> (2 de mayo 2015)
2. <http://geektheplanet.net/7836/polymer-el-nuevo-proyecto-de-google-para-web-components.shtml> (5 de mayo 2015)
3. <http://mundoerp.com/blog/metodologia-iterativa-o-incremental-gestion-proyectos/> (27 de abril 2015)
4. <http://php.net/> (5 de mayo 2015)
5. <http://postgresql-dbms.blogspot.mx/p/limitaciones-puntos-de-recuperacion.html> (3 de mayo 2015)
6. http://programa-con-google.blogspot.mx/2015_03_01_archive.html (1 de mayo 2015)
7. <http://www.diputados.gob.mx/cronica57/contenido/cont6/leycondf.htm> (25 de abril 2015)

8. http://www.firebird.com.mx/descargas/documentos/tema_2-caracteristicas_basicas.pdf (3 de mayo 2015)
9. <http://www.giv.com.mx/GuiadelCondomino.php> (23 de abril 2015)
10. http://www.minvu.cl/opensite_20070212165816.aspx (25 de abril 2015)
11. <http://www.php.net/downloads.php> (1 de mayo 2015)
12. http://www.poderjudicialdf.gob.mx/work/models/PJDF/Transparencia/IPO/Art14/Fr01/02Reglamentos/ReglConstruccionesDF_20040129.pdf (26 de abril 2015)
13. <http://www.proyectosagiles.org/desarrollo-iterativo-incremental> (27 de abril 2015)
14. <http://www.soyentrepreneur.com/iniciate-en-la-administracion-de-condominios.html> (22 de abril 2015)
15. <https://dev.mysql.com/> (4 de mayo 2015)
16. <https://developers.google.com/chart/> (1 de mayo 2015)
17. <https://mysticalpotato.wordpress.com/2014/11/15/charla-polymer-en-3er-betabeers-guadalajara/> (1 de mayo 2015)
18. <https://polymer-designer.appspot.com/> (5 de mayo 2015)
19. https://portal.infonavit.org.mx/wps/wcm/connect/56e399ea-bd81-4085-a219-664a9a12a0e3/01_manual_para_el_participante.pdf?MOD=AJPERES (20 de abril 2015)
20. https://techfest.uc3m.es/files/uploads/2015/01/T3F_Polymer.pdf (3 de mayo 2015)
21. <https://www.polymer-project.org/0.5/> (4 de mayo 2015)
22. www.apachefriends.org (3 de mayo 2015)
23. www.mysql.com (1 de mayo 2015)