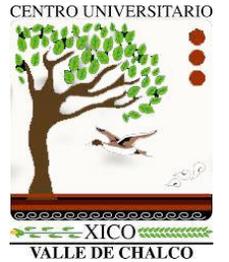




UAEM

Universidad Autónoma
del Estado de México

CENTRO UNIVERSITARIO VALLE DE CHALCO



CREACIÓN DE SOFTWARE DE ADMINISTRACIÓN DE BENEFICIARIOS DE PROGRAMAS SOCIALES EN EL MUNICIPIO DE CHALCO

MEMORIA DE EXPERIENCIA LABORAL

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A

MARIANO LEÓN NÁJERA

ASESORA:

DRA. ANABELEM SOBERANES MARTIN

Revisora: DRA. MAGALLY MARTÍNEZ REYES

Revisor: M. EN C. COM. RICARDO JAVIER BUCIO LÓPEZ

VALLE DE CHALCO SOLIDARIDAD, MÉXICO

ABRIL 2019.

CREACIÓN DE SOFTWARE DE ADMINISTRACIÓN DE
BENEFICIARIOS DE PROGRAMAS SOCIALES EN EL
MUNICIPIO DE CHALCO

ÍNDICE

I.	Resumen	8
II.	Importancia de la temática	10
III.	Descripción del puesto o empleo	15
IV.	Problemática identificada	20
V.	Informe detallado de las actividades	28
VI.	Solución desarrollada y sus alcances	31
VII.	Impacto de la experiencia laboral	77
VIII.	Referencias de consulta	79
IX.	Anexos	82

I. RESUMEN

El software desarrollado llamado Padrón Único de Beneficiarios, surge de la demanda de un sistema de información que pueda procesar datos de la dirección de desarrollo social del municipio de Chalco.

La cual puede resumirse como datos de los beneficiarios de los diferentes programas sociales que implementa la dirección de dichos programas que ha gestionado previamente ante dependencias de gobierno federal, estatal y asociaciones civiles.

El control de la información en la dirección de desarrollo social es precario, funcional todavía, pero precario al fin de cuentas, ya que para capturar, ordenar y procesar la información se realiza en hojas de cálculo de paquetería de software comercial, lo cual limita en mucho el poder procesar la información y obtener mejores resultados al momento de realizar un cruce de los objetivos planteados en las reglas de operación de los programas sociales, pues son éstas las que dicen si el programa que se está ejerciendo está implementado correctamente o no lo está.

El software permite ayudar a facilitar el proceso de la información con una mayor capacidad de representación al utilizar gráficos sobre mapas de Google los cuales permiten saber por ejemplo si se están atacando los sectores más vulnerables de la sociedad, o conocer donde hay mayor oportunidad de aplicación de un programa específico, saber trazar metas respecto al conocimiento de históricos, de programas sociales anteriores.

El software fue el resultado del conocimiento de una problemática que debía ser mitigada, abordándola desde mi experiencia en la carrera cursada de Ingeniería en Computación, utilizar esos conocimientos, y otros más que a raíz de esta necesidad fueron fortaleciéndose, siendo aplicados en algún momento de la investigación.

Al desarrollar el software de una manera correcta a través de un método y modelo recientes, permite ampliar las oportunidades de que pueda seguir siendo vigente en próximas generaciones, puesto que el sistema de información fue desarrollado de tal manera que pueda ser incremental, sacar nuevas versiones de este, con el firme propósito de que se mantenga vigente por un largo tiempo.

La innovación siempre será un impulsor a que éste pueda realizarse mejor y mayor eficiencia, con controles y comandos predictivos, que facilite la capacitación al momento de llevar la fase última fase de transición de un modelo precario a un modelo predictivo y muy intuitivo.

Este software me ayudo a retomar conocimientos que estaban en algún lugar de mi memoria, refrescarlos y lo más importante actualizarlos, la tecnología me ha superado y tengo que ponerme a la vanguardia, por lo tanto el seguimiento a este sistema de información será de vital importancia para poder llevarlo a un siguiente nivel.

II. IMPORTANCIA DE LA TEMÁTICA

En este apartado se expone la selección de la temática, se describe el lugar donde se detectó un área de oportunidad, el objetivo general de la dependencia gubernamental, su importancia o razón de ser de la dependencia y sus programas recurrentes. Con el fin de mejorar por medio de la creación e implantación de un software por internet para la correcta y eficiente manipulación de padrones de beneficiarios, pensando en: la captura, la búsqueda y la manipulación de ellos. Es simplificar el trabajo que se realiza en las oficinas de la Dirección Desarrollo Social del municipio de Chalco.

Hay que establecer una política organizacional donde el proyecto a desarrollar cumpla con el proceso de desarrollo, utilizando las fases, productos de trabajo y roles definidos, realizando los ajustes necesarios para realizar la correcta implementación del software.

El objetivo de la Dirección de Desarrollo Social en Chalco es: Promover, coordinar y establecer estrategias y programas que impacten en la disminución de carencias sociales y económicas del municipio teniendo un impacto mayor en el combate a la pobreza (Gaceta de Gobierno Número 65).

De ahí que la Dirección de Desarrollo Social es la entidad organizacional donde se aplicará el proyecto de software llamado “Padrón Único de Beneficiarios”, por lo tanto, es indispensable saber cómo es que funciona la Dirección de Desarrollo Social en Chalco, conocer su estructura, sus funciones, atribuciones, y los procesos que realiza.

En el proyecto se deben incluir los recursos planificados para la ejecución de cada fase del proyecto. En el plan de proyecto se deben mencionar los roles que desempeñan cada uno de los involucrados en el software, así como su responsabilidad y referenciar al documento de definición de roles y responsabilidades.

Debe quedar claro qué o quiénes deben quedar explícitamente definidos, las actividades en las cuales intervienen y cuál su nivel de participación en cada actividad. La dirección de Desarrollo Social se encarga de gestionar ante las dependencias de Gobierno Federal y Gobierno Estatal los programas a los cuales pueden ser acreedores respecto a las reglas de operación vigentes.

En la mayoría de ellos su eje transversal es similar, ya que las reglas de operación se enfocan para atender a los grupos vulnerable; los cuales se les podría definir como: La población entre los que se encuentran las niñas, los niños y jóvenes en situación de calle, los migrantes, las personas con discapacidad, los adultos mayores y la población indígena, que más allá de su pobreza, viven en situaciones de riesgo (grupos vulnerables).

Tomando como referencia lo mencionado, los programas y la implementación de estos tienen un amplio margen de selección, lo que implica que los beneficiarios serán de diferentes segmentos de la población; por ejemplo, va desde niñas, niños, jóvenes, adultos, mamás solteras, papás solteros, hasta adultos mayores, lo que deja muy abierto el margen de edad, lo que significa retomando de la página del número de habitantes que en Chalco, Estado de México, al 2015 existen 343,701 personas aproximadamente, lo que es equivalente al 2% de la población total del país.

Dentro del universo mencionado se atenderá a un segmento de la población, eso involucra una larga labor, que va desde la gestión, implementación del programa, búsqueda de aspirantes a beneficiarios, recopilación de papeles, evaluación de estos por parte de las dependencias de gobierno, conformación final de expedientes, entrega de recurso, y por último, el seguimiento al programa. Ya que se trata de desarrollar un servicio de acciones verticales ágil a la hora de adaptarse a los cambios de procesos y de cambios organizacionales.

Es de suma importancia estar monitoreando los procesos que se realizan en la dirección, para incluirlos dentro del proceso e ingresarlos como parte de los productos de trabajo desarrollados, durante el proceso se realizaran las desviaciones y las acciones correctivas correspondientes.

Para el Gobierno del Estado de México, de los 125 municipios que lo conforman, el municipio de Chalco siempre ha sido considerado uno de los 20 municipio de prioridad, por su ubicación geográfica y territorial, de la cual está conformada por: Una cabecera municipal, cuatro barrios, diecisiete colonias, trece pueblos, doce conjuntos urbanos y dos subdivisiones y condominios. Además de su desarrollo económico regional, ya que por generaciones Chalco se le conoce como un lugar de mercadeo.

En el Estado de México, Chalco es referente porque es la puerta a los municipios del Oriente del estado. Respecto a la nueva distribución de los distritos locales y federales en el 2018, es cabecera municipal del distrito local número 1 y de distrito federal número 33, en Chalco existen 71 secciones electorales, lo que representa más del 80% de la votación total para elegir funcionarios públicos por elección popular.

Lo que ocurra en Chalco es motivo de resonancia en los municipios cercanos, da pie que la asignación de los programas federales y estatales sea de mayor proporción en Chalco. De mi experiencia laboral en este municipio, los programas que se han gestionado estos últimos años, sin importar que administraciones federales lleguen y con ellas el cambio de nombre de los programas, los que se han implementado de manera recurrente son los publicados en la tabla 1 sobre Programas del Gobierno Federal y la tabla 2 respecto a Programas de Gobierno Estatal.

Tabla 1. Programas de Gobierno Federal.

N. P.	PROGRAMA	NOMBRE
1	PAL	Programa de apoyo Alimentario
2	PROSPERA	Antes llamado Oportunidades
3	65 Y +	Pensión para el Adulto Mayor
4	DICONSA	Programa de Abato Rural
5	LICONSA	Programa de Abasto de Leche
6	PET	Programa de Empleo Temporal
7	Rescate de espacios públicos	
8	Seguro para jefas madre de familias	
9	Comedores comunitarios	

Fuente: (DOF, 2018).

Tabla 2. Programas de Gobierno Estatal.

N. P.	PROGRAMA	NOMBRE
1	FISE,	Fondo de Infraestructura Social para Entidades
2	Por una infancia en grande	
3	De la mano con papá	
4	Seguridad alimentaria	

Fuente: (DOF, 2018).

Cada uno de los programas antes referidos, como ya se ha mencionado, cuenta con sus propias reglas de operación, los cuales los vuelve autónomos unos de otros, aquí es donde mi carrera de Ingeniería en Computación y los conocimientos adquiridos se aplican en una correcta manipulación de la información respecto a las bases de datos.

Se trata de desarrollar conceptos, políticas, lineamientos que constituyen el elemento de integración para software, tanto desde el punto de vista de aplicaciones que se constituyen como composición de otras bajo la filosofía de Servicios Web, como del de la automatización de procesos basados en reglas.

Debo comentar que las dependencias federales y estatales conservan la manipulación de sus beneficiarios de una manera confidencial y limitada, respecto a las leyes de Protección de Datos Personales. Por lo que los sistemas que puedan tener para la manipulación correcta de la información

obtenida y para su procesamiento queda completamente restringida para los municipios, incluyendo el municipio de Chalco.

Por lo que la oportunidad en la Dirección de Desarrollo Social surgió para implementar un software similar a los que tienen las dependencias federal y estatal, pero en servidores propios, en donde se tenga el control, la seguridad y el manejo adecuado de la información respecto a la Ley de Protección de Datos Personales, lo refiero mucho porque esta ley es nueva y la correcta aplicación de esta, nos alejará de sus consecuencias legales que pueda conllevar. A este software se le agregamos un toque de innovación que permita que, dentro de su ejecución, el software facilite el trabajo al mismo personal que labora, ahorrando procesos engorrosos que pueden ser solventados o pueden disminuir tiempos si al software lo administran de manera óptima.

El proyecto o software se basó en el modelo de desarrollo de software iterativo e incremental, repitiendo las etapas de cada ciclo, añadiendo funcionalidad al producto, comprimiendo al máximo el número de iteraciones, realizando entregas parciales en las cuales el producto o software es validado por él o los usuarios finales.

“Pero creo que todos sus grandes logros se deben a su capacidad para absorber mediante el sencillo arte de escuchar” (Brandson, 2016, 52). Esa es una virtud que durante toda la vida he aplicado y que si uno pone atención a lo que necesitan las demás personas podrá tener un gran crecimiento, pero el lugar donde mejor puede aplicarlo es en mi lugar laboral que es el siguiente capítulo en donde abordare la funciones que realizaba como los logros dentro de la dirección.

III. DESCRIPCIÓN DEL PUESTO O EMPLEO

Promotor de Desarrollo Social Chalco

La dirección de Desarrollo Social es aprobada por el Órgano de Gobierno Colegiado de nombre Ayuntamiento Constitucional, el cual está integrado por el Presidente Municipal, Síndica Municipal, Regidoras y Regidores. La asamblea deliberante es dirigida por la Secretaria del Ayuntamiento, la que por instrucciones del presidente municipal se somete a votación, y es aprobada ya sea por unanimidad o por mayoría de los integrantes de la asamblea.

Se hace de conocimiento en la publicación del Bando Municipal, en la Gaceta número 1, de fecha 05 de febrero de 2016, en el Título Quinto de la organización y funcionamiento del gobierno municipal, Capítulo IV de las dependencias. La figura I muestra el organigrama de la Administración Municipal Constitucional 2016-2018, se encuentra publicada en la Gaceta Número 62 del 23 de marzo de 2018.

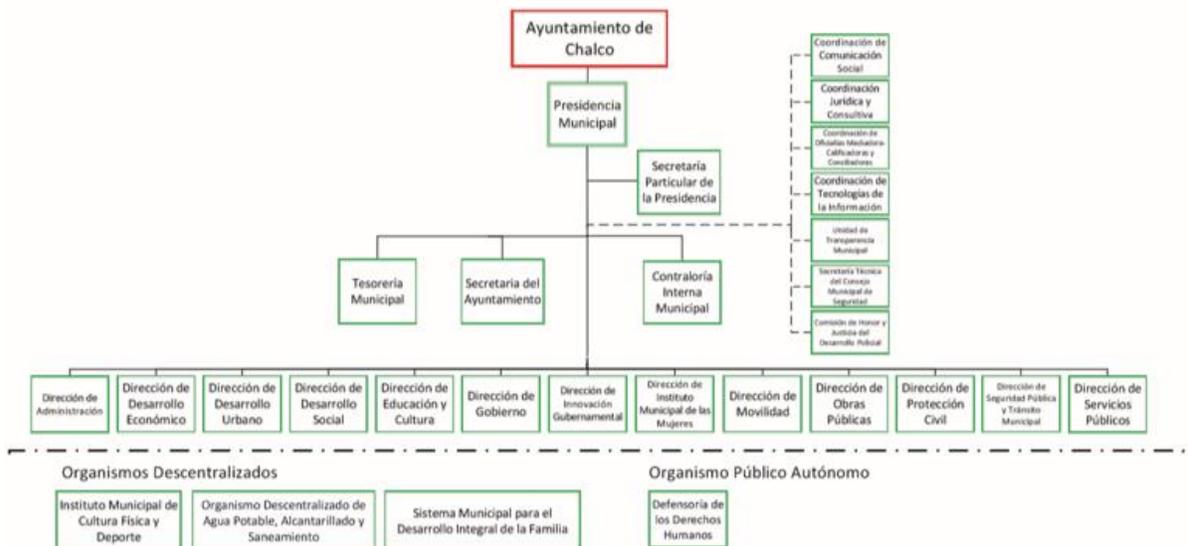


Figura I. Organigrama de la Administración Municipal Constitucional 2016-2018
Fuente: (Gaceta 62, 2018).

La Figura 2 muestra el organigrama interno de la Dirección de Desarrollo Social del municipio de Chalco, en orden jerárquico la estructura interna de la dirección comenzando con el director(a), subdirector(a), el departamento de planeación, departamento de centro de desarrollo comunitarios, departamento de comedores, lecherías y logística, departamento de programas sociales y grupos vulnerables y departamento de promotores.



Figura 2. Organigrama interno de la Dirección de Desarrollo Social del Municipio de Chalco
Fuente: (Gaceta 36, 2017).

La definición de promotor y el objetivo del área de promotores se encuentra publicada en la página web oficial en <https://gobiernodechalco.gob.mx/> y aprobada por el ayuntamiento constitucional 2016–2018, que queda vigente hasta que exista una derogación o actualización de la misma de parte de administraciones subsecuentes. La Gaceta de Gobierno es la Número 65 del día 4 de abril del 2018 que a la letra dice:

Promotor: Se aplica a una persona que promueve o promociona de forma profesional a una persona.

El objetivo: Informar a los grupos vulnerables del municipio de una forma práctica y eficiente los programas y acciones que realiza la Dirección de Desarrollo Social para mejorar su calidad de vida.

Hay que enfocar e integrar los datos a la estructura del software, la integración tiene que realizarse lo más simple posible partiendo de los formatos o estructuras establecidas en el área, esto con el fin de brindar funcionalidad en las interfaces del software, para lo cual se describirán las definiciones de las demás áreas involucradas como sus objetivos. Departamento de programas sociales y grupos vulnerables tiene como objetivo la responsabilidad y el desarrollo de sus actividades extraído de la Gaceta de Gobierno es la Número 65 del día 4 de abril del 2018 que a la letra dice:

Objetivo: Gestionar programas sociales aplicados de manera eficiente, contribuyendo a un desarrollo social sostenible, combatiendo las carencias sociales del municipio y ocupándose de aportar beneficios que cubran las necesidades básicas teniendo una población destinada a vivir en un equilibrio económico y social.

Responsabilidad:

- Gobierno federal será la autoridad competente de entregar la parte proporcional o total del apoyo otorgado.
- Gobierno estatal será la autoridad competente de entregar la parte proporcional o total del apoyo otorgado.
- El presidente o la presidenta municipal firmará los oficios de solicitudes a la incorporación de los programas.
- Es responsabilidad del director o la directora de desarrollo social establecer, controlar, distribuir y actualizar este procedimiento.
- Es responsabilidad de la persona a cargo del departamento de programas y grupos vulnerables la correcta aplicación de dicho procedimiento.
- La ciudadanía dará a conocer mediante peticiones la necesidad de ser incorporadas a algún programa social.

Desarrollo (anexo II diagrama de flujo):

1. El responsable o la responsable del departamento de programas sociales y grupos vulnerables realiza un diagnóstico de los censos realizados por INEGI (<http://www.inegi.org.mx/>), de la comisión nacional de población CONAPO (<http://www.gob.mx/conapo>), CONEVAL (<http://www.coneval.org.mx>) así mismo de las peticiones ciudadanas para detectar prioridades, evaluar y decidir qué programas sociales gestionar.
2. Ya realizado dicho diagnóstico el director o directora de desarrollo social determina las gestiones de los programas a aplicar durante el año.
3. El director o directora de desarrollo social envía oficios (firmados por el presidente o la presidenta municipal) de solicitud señalando la necesidad de incorporar al municipio dentro del programa social ante la instancia federal o estatal a la que corresponda dicho programa a gestionar.
4. Se espera oficio en donde se dé respuesta por parte de la instancia federal o estatal a la cual se le solicito la incorporación al programa.
5. En el caso que la respuesta sea negativa o tardía se solicitará por medio de un oficio a la instancia federal o estatal la reconsideración de incluirnos en el programa y solicitar las causas por las que no se da respuesta favorable a la incorporación del programa.
6. En el caso que la respuesta sea positiva la instancia federal o estatal solicita por medio de un oficio los datos del proyecto a realizar, suficiencia presupuestal en el caso que así sea y enlace para continuar con el trámite del programa.

7. El departamento de programas sociales y grupos vulnerables por medio de su responsable incorpora un expediente con los datos requeridos en tiempo y forma los cuales se presenta ante la institución estatal o federal por medio de un oficio firmado por el director o directora de desarrollo social.
8. La institución federal o estatal envía el oficio de aprobación con un número de expediente ya autorizado la incorporación al programa.
9. Servidores o servidoras públicos del departamento de programas sociales y grupos vulnerables asisten a las reuniones y cursos de capacitación que la instancia federal o estatal solicite para el conocimiento de los formatos y reglas de operación del programa a aplicar.
10. El departamento de programas sociales y grupos vulnerables da inicio la aplicación del programa.
11. El departamento de programas sociales y grupos vulnerables integra una carpeta con el nombre del programa incluyendo el año en el que se aplicó el programa incorporando todo el desarrollo del programa.
12. El departamento de programas sociales y grupos vulnerables archiva la carpeta del programa realizado.
13. Fin del proceso.

Aquellas partes de la dirección de desarrollo social que afectarán al software, y que serán afectadas por él, será el dominio de la aplicación. Es allí donde los usuarios observaran si el desarrollo del software ha cumplido su propósito, así como que se identifican los conceptos propios de la Dirección de Desarrollo Social que se encuentran en el entorno del software.

“Cuando descubras que hace que allá fuego en tu interior, aquello que te hace hervir cada vez más, abras encontrado tu talento” (Gomes, 2016, 38). Esa es la actitud que tome para abordar la problemática de la dirección.

IV. PROBLEMÁTICA IDENTIFICADA

Con los datos presentados en la Importancia de la temática, da pie que en este tercer apartado se planteé la problemática que se detectó; respecto al manejo de la información, no es que sea del todo mal la práctica que se lleva dentro el departamento responsable, pero comparado con la tecnología actual y todo el desarrollo que existe para tener un mayor control en: Seguridad, captura, manipulación de la información, se encuentra la dirección de Desarrollo Social del Municipio de Chalco en la oportunidad de mejorar y ser más eficiente en el control de sus padrones.

En las oficinas municipales de la Dirección de Desarrollo Social se captura, se procesa y se manipula en un simple archivo de hoja de cálculo, lo que deja expuesta la información y la vuelve vulnerable de muchas posibles amenazas, además de que su captura es altamente rudimentaria, así como su manipulación; por ejemplo, una simple muestra aleatoria estadística para un informe.

En los programas federales y estatales la dimensión de beneficiarios depende proporcionalmente del presupuesto asignado cada año y lo que a cada beneficiario le corresponde, se habla del programa Federal 65 y +, el cual en el 2018 tuvo un presupuesto asignado respecto al Diario Oficial de la Federación 2018 (DOF) de \$13,177,199,137.00 de pesos (publicado en www.dof.gob.mx/nota_detalle.php?codigo=5506080&fecha=29/11/2017 y que dan un apoyo mensual de \$580.00 por cada adulto mayor, además de las vertientes del mismo programa. La incorporación a este programa no está acotada por ninguna cifra en particular, ya que el ingreso y el tamaño de la población beneficiaria depende de múltiples factores, de los cuales los principales son: Cumplir con los requisitos publicados en las reglas de operación vigentes, tener todos los papeles actualizados, estar atentos a la apertura de ventanilla del mismo programa, de todas estas variables influye el

tamaño de beneficiarios de una programa, los beneficiarios del programa Pensión para el Adulto Mayor 65 y más en el Estado de México, según los datos que se muestran en la tabla 3 del Padrón Único de Beneficiarios.

Tabla 3. Padrón Único de Beneficiarios.

NO	PROGRAMA	SUB PROGRAMA	PERIODO	TOTAL INTERVENCIONES
1	E003	E003	abril-junio2018	523247
...
24	S176	S176	julio-agosto2018	528678
...
46	U011	0109	julio-septiembre2018	6

Fuente: (Programas de Desarrollo Social, 2019).

En donde 24 es el número progresivo dentro de la tabla, S176 es el programa, S176 el subprograma, julio – agosto 2018 es el periodo, y 528,678 es el total de intervenciones al padrón en el Estado de México de un total de 8,979 beneficiarios en el municipio de Chalco. De los cuales en esta dependencia solo se cuenta con simples hojas de cálculo para poder procesar la información.

El objetivo era dar el salto de un modelo de trabajo rudimentario a uno actualizado y de vanguardia, refiriéndome a la manera en cómo se venían realizando las cosas, si bien es cierto que se utilizaba software actualizado la paquetería más reciente de Microsoft office o de los más recientes. A utilizar las tecnologías actuales, en las que van encaminadas los sectores públicos y privados de vanguardia, utilizar la puerta que trajo el internet, y con él el contacto con todo el mundo, a cualquier hora y desde cualquier lugar. Ése es el objetivo de la solución planteada a este problema en particular.

Con base a los conocimientos proporcionados por mi carrera, llego a mi mente la propuesta de llevar a cabo esta solución simple que podría beneficiar, no solo a los que son directamente implicados, que son los capturan la información, sino en todos los niveles jerárquicos involucrados, también el que procesa la información capturada, el que coordina al personal involucrado, pero sobre todo el que dirige los trabajos, ya que al poder observar los datos

capturados y procesados puede tomar decisiones a partir de un análisis simple y para saber si está realizando un buen trabajo o se necesita reconducir los esfuerzos y buscar él o los objetivos esperados.

Después de analizar las deficiencias previamente ya mencionadas y contemplar las áreas de oportunidad que se tienen y que no se han tomado en cuenta, me llevo a redactarlos una y otra vez, estudiarlos y ser empático con mis compañeros de área y partir de ahí ver hacia donde se podría encaminar la propuesta, así que después de un tiempo logre concretar el rumbo y la dirección que tomo esta solución.

Si bien es cierto que son muchas las carencias en la dirección que demandan utilizan varias metodologías de desarrollo para aumentar la eficiencia en el trabajo, también es verdad que existen muchas oportunidades en el software que satisfacen las necesidades de manera adecuada.

Los objetivos que se pretenden alcanzar mediante la solución desarrollada son:

- Crear un ambiente de trabajo eficiente, moderno y eficaz entre las diferentes áreas y departamentos que tienen responsabilidad transversal con la información.
- Dar la importancia requerida al proceso de planeación, considerando tiempos, requerimientos y periodos de desarrollo, aplicación e implementación del software en el área administrativa.
- Fortalecer el canal de comunicación entre las áreas dependientes del manejo y proceso de la información.
- Mantener la información actualizada y homologada de todos los programas sociales existentes.
- Implementar de una manera eficiente la ley de protección de datos personales vigente al software desarrollado.

- Aplicar novedades tecnológicas que faciliten y simplifiquen el trabajo de las áreas y departamentos involucrados en el proceso de la información.

De acuerdo con los objetivos que se necesitan cumplir y para generar un mayor entendimiento he elaborado un esquema que nos ayuda a identificar como resuelve de una manera muy efectiva las causas (ver tabla 4) y consecuencias (ver tabla 5) identificadas en este trabajo capítulos anteriores.

Sin embargo, también es necesario que la solución presentada deba garantizar su vigencia dentro de la dirección en la cual cada vez más demandada la excelencia laboral, y a su vez aumentar la eficiencia, deben no solo sacar máximo provecho de las soluciones que la dirección tiene a su alcance, sino contar con soluciones de desarrollo que incluyan las metodologías de desarrollo de software que les permitan que la gestión de sus procesos gubernamentales sea más ágiles.

Tabla 4. Evaluación de Solución a Causas.

N. P.	CAUSAS	CREACIÓN DE SOFTWARE DE ADMINISTRACIÓN DE BENEFICIARIOS DE PROGRAMAS SOCIALES EN EL MUNICIPIO DE CHALCO
1	Deficiencias en sistemas de ejecución.	Efecto positivo: Pueden surgir al principio mientras el quipo logra adaptarse al sistema de trabajo, pero a largo plazo se logrará cierto grado de especialización.
2	Redundancia de datos.	Efecto positivo: Ya que se traga de datos que se someterán a la normalización de base de datos.
3	Deficiencia en los procesos y procedimientos de operaciones.	Efecto positivo: Esta nueva era servirá para vincular y agilizar los procesos y procedimientos.
4	Personal deficiente.	Efecto Negativo: La capacitación deberá ser extensiva y durante un lago periodo de tiempo.
5	Poco o nula seguridad	Efecto Positivo: Al tener cuentas de acceso los usuarios con el software permiten mantener la información blindada.
6	Acceso seguro limitado	Efecto Positivo: Sin importar el lugar donde se requiera de la información, esta estará disponible los 24/7/365 días del año.

Fuente: Elaboración propia.

De acuerdo con lo analizado anteriormente observamos que la propuesta de creación de software de administración de beneficiarios de programas sociales en el municipio de Chalco requiere de un mayor esfuerzo, los resultados a largo plazo superarán las causas. Sin embargo, la metodología destinada para el desarrollo de este software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializado.

Tabla 5. Evaluación de Solución a Consecuencias.

N. P.	CONSECUENCIAS	CREACIÓN DE SOFTWARE DE ADMINISTRACIÓN DE BENEFICIARIOS DE PROGRAMAS SOCIALES EN EL MUNICIPIO DE CHALCO
1	Personal suficiente y dispuesto	Efecto positivo: Al estar únicamente dirigidos a la captura se contaría con la participación y concentración de todos.
2	Eficiencia laboral	Efecto positivo: Ya que se trata de una herramienta nueva existe cierta motivación y entusiasmo, lo que ayudará a integrar al grupo.
3	Pérdida de tiempo	Efecto Positivo: Al inicio puede presentarse, pero ira disminuyendo conforme se adapten al software, finalmente se eliminará cuando el equipo alcance la especialización requerida.
4	Baja calidad en la forma de trabajo	Efecto Positivo: La calidad de trabajo en las áreas dependientes del software aumentara, debido a que cada área se enfocara solo en la manipulación del software.
5	Precisión en la toma de decisiones	Efecto Positivo: El software al dar informes en tiempo real amplia la asimilación de la información para compararlos con las metas establecidas y objetivos.

Fuente: Elaboración propia.

Como se puede observar en el esquema de consecuencias, los efectos positivos favorecen determinadamente la creación de una nueva herramienta. Además de analizar la solución de las causas y consecuencias donde la más importante se tomará como un máximo de la escala con la marca de 4X y la de menor importancia la marca de 1X, se debe tomar el riesgo que lleva la

propuesta de solución que se puede apreciar en la tabla 6, en base a los conceptos analizados en este trabajo que hablan acerca de las causas de las que es necesario realizar esta herramienta en la dirección. Esto nos ayuda a complementar nuestra elección de propuesta de solución.

Tabla 6. Conceptos de riesgo basados para esta solución.

N. P.	CRITERIOS	CREACIÓN DE SOFTWARE DE ADMINISTRACIÓN DE BENEFICIARIOS DE PROGRAMAS SOCIALES EN EL MUNICIPIO DE CHALCO
1	Costos	X
2	Dificultad de implementación	XX
3	Dificultad de aceptación por parte de los involucrados en el proceso	XXX
4	Dificultad de capacitación	XXX
5	Información ineficiente	XX
6	Falta de claridad en definición de papeles y responsabilidades	XXX
7	Seguridad poca o nula con la información	XXXX
8	Difícil acceso a la información.	XXX
9	Ambigüedad al tomar decisiones.	XXX

Fuente: Elaboración propia.

Se analizará punto por punto cada propuesta.

1. Costo: La implementación de esta propuesta sería relativamente económica, ya que el ayuntamiento cuenta con servidores propios, dominios y acceso a conexión a internet de gran velocidad.
2. Dificultad de implementación: Representa un reto ya que los conocimientos adquiridos en mi carrera de ingeniería en computación no han sido actualizados y retomar los conceptos nuevamente, implica tiempo, reaprendizaje y una correcta aplicación.
3. Dificultad de aceptación por parte de los involucrados en el proceso: No representa mayor esfuerzo ya que se necesita de un conocimiento de especialización que con el tiempo se profundizara.
4. Dificultad de capacitación: El estilo de vida en la actualidad permite que la mayoría cuente con un SmartPhone con aplicaciones predictibles, pero

sin dejar de ser sofisticadas, por lo que el tiempo de aprendizaje a un software nuevo no representa un reto mayor salvo, salvo que en un inicio habría poco personal capacitador.

5. Información ineficiente: Se resolvería este problema ya que con el software en funcionamiento la información se encuentra concentrada en un lugar seguro, procesada y disponible todo el tiempo.
6. Falta de claridad en definición de papeles y responsabilidades: Se terminaría con este conflicto de saber quién es responsable de cada tarea en el proceso, ya que el mismo sistema limita a tareas precisas
7. Seguridad poca o nula con la información: Al ya no concentrar la información en hojas de cálculo y tener un mismo archivo con muchas versiones, y en manos de todo mundo en la oficina, se terminará con ese problema, ya que la información estará disponible todo el tiempo y accesible basados en la ley de protección de datos personales. Este punto la seguridad se precisa en la seguridad de que la información no estará en manos inapropiadas, solo en aquellas que tiene una responsabilidad directa con la institución y el programa aplicado.
8. Difícil acceso a la información: Precizando en este punto me permito ampliar que la seguridad al salvaguardar la información en servidores permite que no todos tengan acceso a ella y los que la tienen, tendrán credenciales en el software que validara su acceso a ella desde cualquier lugar.
9. Ambigüedad al tomar decisiones: Con el modelo de trabajo rudimentario la información no estaba actualizada, en la aplicación del software la información está siempre comparada con las metas manteniéndose actualizada, vigente, estructurada y sobre todo útil para tomar decisiones acertadas en la dirección.

De acuerdo con el análisis de la propuesta de solución los esquemas antes presentados dejan ver claramente que la correcta implementación del software

traerá para la dirección la simplificación de las tareas realizadas en los departamentos involucrados en los programas sociales, puesto que reduce tiempos, estructura la información procesándola de manera eficiente y disponible para una oportuna toma de decisiones.

“Se requiere coraje para levantarse y hablar; también se requiere coraje para sentarse y escuchar” (Brandson, 2016, 46). El trabajo individual luce pero es ambiguo, ese trabajo individual es como ver ladrillos solos, pero unidos es ver una magnífica edificación, así el trabajo de todos, cuando se sabe escuchar se conoce que es lo que se puede hacer, de que sirve, se le da la importancia que merece y se sabe valorar lo que representa no solo para uno, si no de manera plural para la sociedad y su buen desarrollo.

V. INFORME DETALLADO DE LAS ACTIVIDADES

Como se mencionó la categoría en la cual estoy registrado en el ayuntamiento de Chalco pertenece a la de promotor. Así como hice referencia de la definición de promotor y el objetivo del área, algunas de las palabras reservadas que se emplean en el documento pueden definirse al consultar el Anexo I. De la misma forma cito de la Gaceta de Gobierno Número 65 que se encuentra publicada del día 4 de abril del 2018 y que permanece vigente hasta nueva modificación, que a la letra dice:

El alcance del departamento:

- Aplica al responsable o la responsable del área de promotores quien dirige la promoción de los programas sociales, a todos los coordinadores de zona y sus promotores respectivamente como también a los grupos vulnerables quienes serán los beneficiados, teniendo mayor impacto en el combate a la pobreza y una mejor calidad de vida en dichos grupos.

Responsabilidad:

- Es responsabilidad del director o de la directora de Desarrollo Social establecer, controlar, distribuir y actualizar este procedimiento.
- Es responsabilidad de la persona a cargo del Departamento de Promotores aplicar con disciplina y compromiso dicho procedimiento.
- Es responsabilidad de los coordinadores o coordinadoras de cada zona, llevar a cabo los pasos que señalan en este procedimiento.
- Es responsabilidad de cada promotor conocer este procedimiento para llevar a cabo actividades homogéneas y de resultado

eficientes en la promotoría de las acciones y programas de desarrollo social.

Desarrollo de actividades (Anexo III):

1. El responsable o la responsable del Departamento de Promotores asigna y organiza a cada coordinador de zona el o los programas sociales que se ejecutarán con el objetivo de que en cada zona se promocióne el beneficio y forma de participar en cada programa.
2. El coordinador o coordinadora de promoción de zona distribuirá a sus promotores o promotoras en las zonas de mayor relevancia donde se promocionará el apoyo.
3. El promotor o promotora dará al beneficiario o beneficiaria los requisitos para poder ser parte del programa.
4. El beneficiado o beneficiada asistirá a las oficinas de la Dirección de Desarrollo Social a dejar los documentos y llenar los formatos requeridos para participar en el programa.
5. El responsable o la responsable del Departamento de Promotores recibe la documentación, revisando que se cubra con los requerimientos del programa social a aplicar (reglas de operación del programa social).
6. El responsable o la responsable del Departamento de Promotores forma expedientes con los documentos de candidatos a participar en el programa y los entrega al Departamento de Programas Sociales y Grupos Vulnerables para enviarlos a la instancia federal o estatal correspondiente.
7. El responsable o la responsable del Departamento de Promotores integra una carpeta con los nombres de los beneficiados, en la cual se señala el nombre del programa y fecha en el que se incorporan al programa.

8. La instancia federal o estatal envía a la Dirección de Desarrollo Social el padrón de beneficiados ya autorizados.
9. El responsable o la responsable del Departamento de Promotores selecciona dicho padrón por zona para que cada coordinador o coordinadora por medio de sus promotores o promotoras visiten a los beneficiados o beneficiadas en sus domicilios notificándoles la integración al programa.
10. Fin del Proceso.

El software durante este proceso permitirá capturar una única ocasión al beneficiario final, esto se logra aplicando niveles de seguridad de autenticación, por lo que los promotores tendrán claves únicas de autenticación para tener acceso al software, en la interfaz de captura a un programa en específico.

La manipulación y modificación al beneficiario solo lo podrá hacer alguien que cuente con un nivel de seguridad mayor, lo que significa que también tenga un nivel jerárquico superior de la Dirección de Desarrollo Social, por lo que al departamento de programas sociales y grupos vulnerables le correspondería a partir de este momento la manipulación y modificación de los usuarios en la base de datos, es este departamento responsable de salvaguardar el información y tenerla en confidencialidad, así como los documentos en copia de todos y cada uno de los beneficiarios.

“El hecho sencillo es que si uno no goza lo que hace y no está a gusto con la gente que lo rodea, no hay manera de que produzca algo tan bueno como cuando disfruta lo que hace” (Brandson, 2016, 16). Trabajar en un buen ambiente de trabajo y contar con el apoyo necesario, ahí se pueden establecer objetivos cada vez más ambiciosos, metas más claras y de mayor impacto, así es como encontré la solución desarrollada y sus alcances.

VI. SOLUCIÓN DESARROLLADA Y SUS ALCANCES

Hoy en día hemos llegado a un punto en el que se ha logrado tener control de todo en la punta de los dedos. Esto se debe principalmente a que la tecnología se ha desarrollado más allá de la imaginación, pero también gracias a los avances en la industria del desarrollo de software. Hecho que no es de extrañar considerando que el mundo del software es ilimitado.

Una de las muchas soluciones a esta problemática identificada fue desarrollar un software web que pudiera concentrar la información, procesarla y organizarla, ponerla siempre disponible, al mantenerla accesible desde y en todo lugar, salvaguardando siempre la seguridad e integridad de la información.

Desarrollar el software significa hacer su construcción mediante la descripción o necesidad expresada a través de una especificación de requisitos. Por tanto, al proceso de ingeniería a realizar hay que tener el conjunto de métodos, herramientas y técnicas, hay que contar con las soluciones que permitan que la gestión de los procesos sea más ágil que se muestran en la figura 3.



Figura 3. Metodología de desarrollo de software.

Fuente: (http://aigroup.com.co/site/?page_id=18).

Hay que poner las metodologías de desarrollo enfocadas a satisfacer las necesidades de manera adecuada. Es por ello por lo que resulta importante contar con los conocimientos necesarios sobre los problemas que se quieren resolver optimizando sus procesos, además de también contar con la metodología adecuada que cumpla con los requerimientos para lograr que el software a desarrollar sea óptimo y funcional.

Siempre teniendo en cuenta que para lograr un desarrollo de software exitoso es necesario contar con un modelo que garantice su calidad de manera que los integrantes del equipo y todo aquel que pueda estar interesado en el producto final, tenga la misma visión y no ocurra cuando no se aplica un proceso de desarrollo que se muestran en la figura 4.

La metodología del software enfocada como una manera de interpretar la realidad utilizada para planificar y controlar el procedimiento de creación de un sistema de información. Permite hacer la incorporación de la experiencia adquirida en otras zonas, ponerlas en práctica al simplificar los procesos.

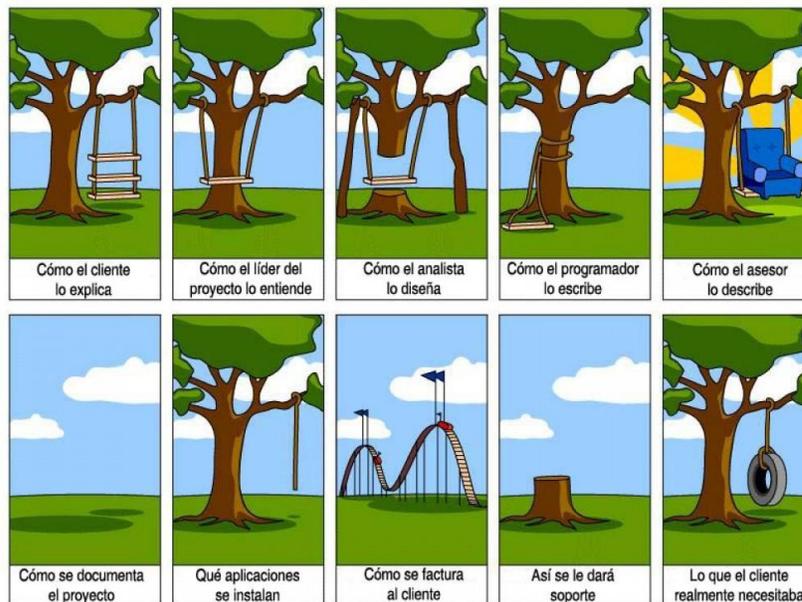


Figura 4. Problemas que conlleva una mala especificación.

Fuente: (<https://serprogramador.es/el-cliente-nunca-sabe-lo-que-quiere/>).

Utilizando el Proceso Unificado de Desarrollo de Software (RUP) como un proceso que de manera ordenada define las tareas y quién de los miembros del equipo de desarrollo las hará. El Proceso Unificado de Desarrollo es una guía para usar UML como se muestra en la figura 5.

“También conocido como UML por sus siglas en inglés (Unified Modeling Language), es un lenguaje estándar que permite escribir planos de software. Proporciona una notación gráfica que se puede usar para modelar sistemas de cómputo desarrollados con el uso de ingeniería de software orientado a objetos”. (Lilia, 2017).



Figura 5. Logotipo UML.

Fuente: (https://commons.wikimedia.org/wiki/File:UML_logo.svg).

CASOS DE USO

Los casos de uso se definen como: “Un Caso de Uso describe un servicio provisto por un sistema, es decir un modo específico de usarlo. El conjunto completo de Casos de Uso especifica todas las posibles maneras en las que el sistema puede ser usado, sin revelar cómo esto es implementado por el sistema. Esto hace a los Casos de Uso apropiados para definir requerimientos

funcionales en etapas tempranas del desarrollo del sistema, donde la estructura interna de éste aún no fue definida” (Giandini & Pons, 2000).

A partir de esta definición es como crea un esquema en borrador de la arquitectura, comenzando por la parte no específica de los casos de uso, pero con una comprensión general de los casos de uso fundamentales. En seguida se trabaja con un conjunto de casos de uso claves o fundamentales. Cada caso de uso es especificado en detalle y realizado en términos de subsistemas, clases, y componentes. Hay que contar con las soluciones de desarrollo de software que les permitan que la gestión de sus procesos sea más ágiles.

A medida que los casos de uso se especifican y maduran, es una buena opción para los proyectos ya que se descubre más de la arquitectura, y esto a su vez lleva a la maduración de más casos de uso donde todos los aspectos del desarrollo son conocidos de antemano así simplificando la arquitectura. La herramienta que sirvió de apoyo para generar los diagramas en UML y plantear los casos de uso de todos los escenarios posibles para el software StarUML versión 2.8.0 2014-2018 de MKLab que se muestra en la figura 6.

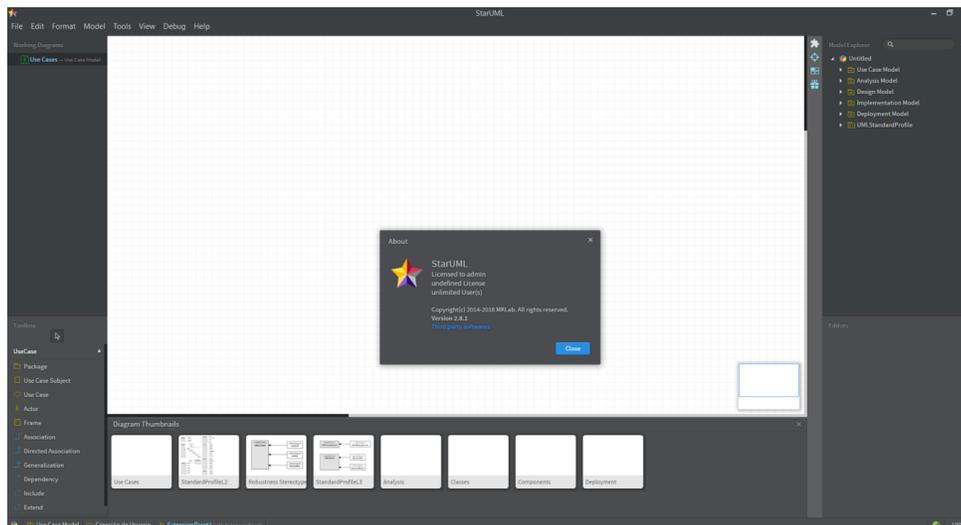


Figura 6. Software StarUML.

Fuente: Elaboración propia.

De años de experiencia es donde se tiene que cubrir con aspectos importantes como la identificación de roles, conceptos y modelos consistentes, aspectos de la dirección y de origen técnicos. Proporcionar un entorno de lenguaje común, lo que se traduce como pasar las barreras de la comunicación, enfocarse en la mejora y en la idea global, con la visión de servicio de muy alta calidad.

Los diagramas de UML implementados al software desarrollado en el cual se muestran las especificaciones de casos de uso en el anexo IV, como parte fundamental del modelado se enumeran a continuación:

1. Creación de usuarios, menú y áreas administrativas (rol del usuario administrador).
2. Creación, modificación y/o actualización de la información (rol del usuario jefe de área).
3. Captura de beneficiarios (rol del usuario promotor).
4. Consultas y reportes.

Para el primer diagrama UML de nombre creación de usuarios, menú y áreas administrativas, también nombrado rol del administrador, este es el usuario de jerarquía superior, con funciones específicas de configurar, implementar y asegurar el correcto funcionamiento del sistema de información, además de documentar los aspectos esenciales del software, con la finalidad de que sin importar quien opere el sistema mientras sea un usuario con los conocimientos de administrador pueda asegurar las obligaciones y responsabilidades de usuario administrador.

El usuario administrador tiene la responsabilidad de dar de alta al personal de la administración municipal en los tres tipos de usuarios, los que tendrán obligaciones y responsabilidades específicas de operación en el software padrón único de beneficiarios que se pueden ver en la figura 7.

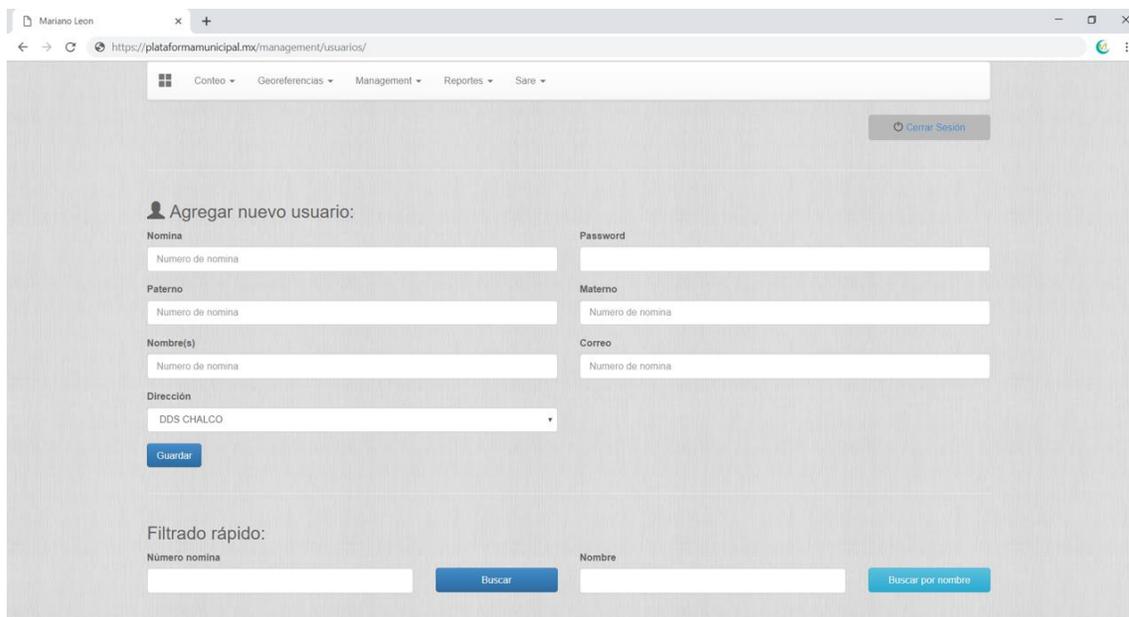


Figura 7. Creación de usuarios.

Fuente: Elaboración propia.

Los cuales se enlistan a continuación:

- 1.- Administrador.
- 2.- Titular de área.
- 3.- Capturista.

Al terminar el proceso tiene que darse la validación en la base de datos, con el fin de que cada administrador público de la Dirección de Desarrollo Social se le asigne un tipo de usuario y no pueda darse más de una vez, para el caso de que se quiera asignar a alguien externo a la dependencia administrativa no podrá realizarse dicha operación.

El usuario administrador tiene la responsabilidad de crear el menú de aplicaciones a las cuales tendrán acceso cada uno de los tres tipos de usuarios ya mencionados. El menú de aplicaciones les permitirá acotar el acceso al sistema que se muestran en la figura 8, esto se realiza al momento de realizar

un inicio de sesión al sistema, al momento se ha validado su identidad en la base de datos, esta arrojará el nivel del usuario, así dejando el ambiente de trabajo delimitado a solo realizar las funciones específicas del propio usuario, funciones que se retomaron de la documentación oficial de la Dirección de Desarrollo Social publicadas en la Gaceta de Gobierno es la Número 65 del día 4 de abril del 2018.

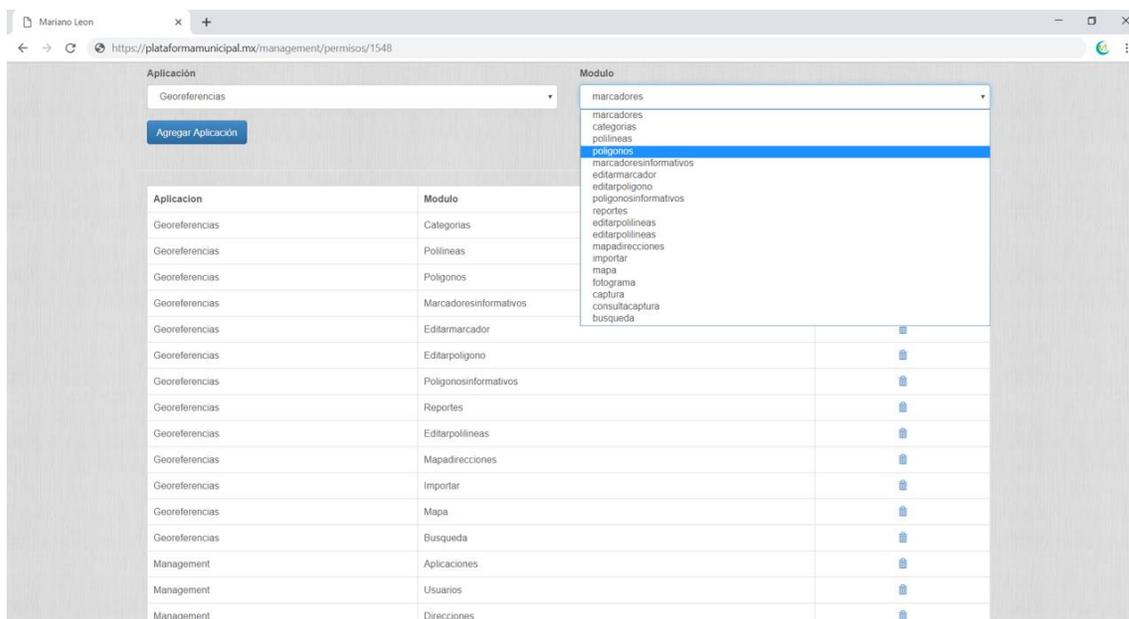


Figura 8. Creación de menú.

Fuente: Elaboración propia.

El usuario administrador tendrá que realizar la creación de áreas administrativas que se muestran en la figura 9, de las cuales dependerán los programas, ya que no solo se bajan de una dependencia nada más, sino que son de muchas dependencias de administraciones federales y estatales. Por lo que dejar señalado el área que se hará responsable del programa en específico es de vital importancia para el trabajo de la Dirección de Desarrollo Social, la validación en la base de datos permitirá que no se dupliquen áreas a las cuales se les asignen los programas gestionados.

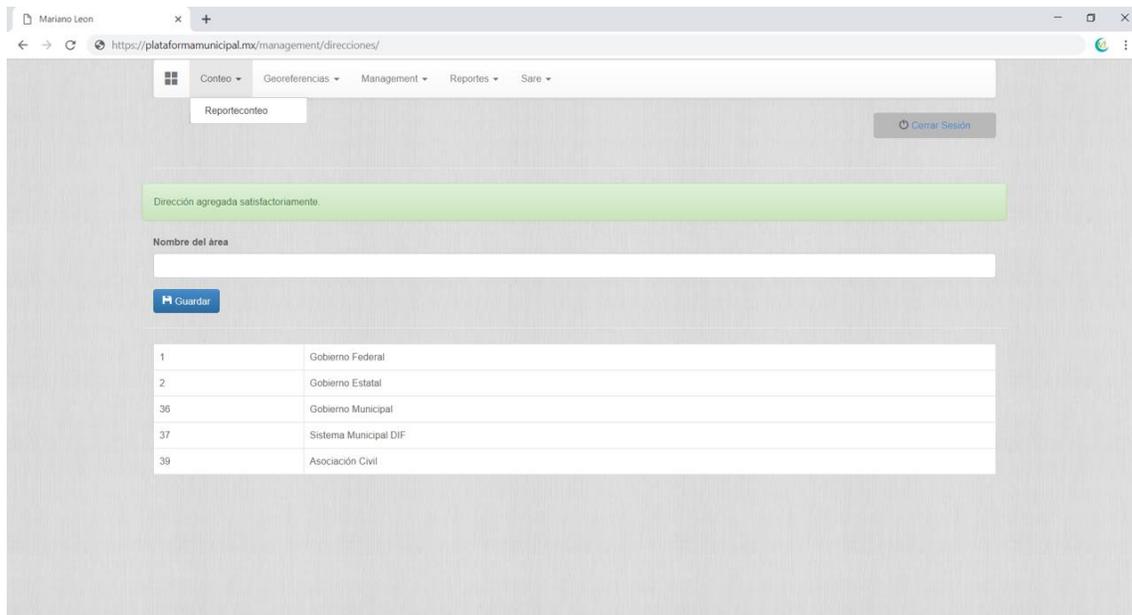


Figura 9. Creación de áreas administrativas.

Fuente: Elaboración propia.

Es así como el primer diagrama de UML de nombre creación de usuarios, menú y áreas administrativas, pertenecientes al rol del administrador se describen gráficamente en la figura 10. En la que intervienen dos actores, y cinco casos de uso, donde uno de ellos tiene puntos de extensión.

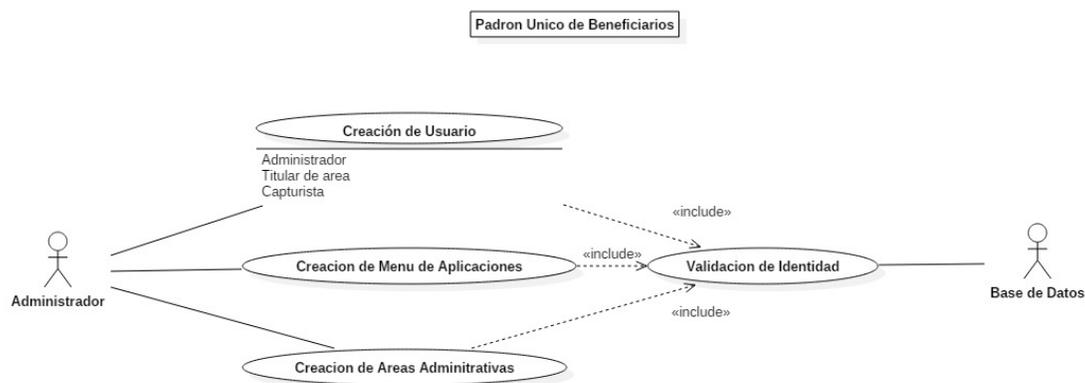


Figura 10. Diagrama de casos de uso de captura de creación de usuarios, menú y áreas administrativas.

Fuente: Elaboración propia.

Para el segundo diagrama de UML de nombre creación, modificación y actualización de la información o también nombrado rol del usuario jefe de área, en este usuario recae más la responsabilidad del correcto funcionamiento del sistema, lo eficiente, coherente y ordenada de la información.

El diagrama que se muestra en la figura 11 muestra lo complicado de la labor del mismo usuario, en la que intervienen dos actores, y nueve casos de uso, donde todos convergen en uno de ellos, la validación, proceso sumamente importante.

De igual manera que la responsabilidad que sobre el usuario jefe de área recae, en este usuario de nivel medio puede crear y modificar información que a diferencia del usuario anterior que lo hacía sobre el sistema o software, este lo hace sobre los programas que la Dirección de Desarrollo Social ejecuta, esto tiene que ver con:

- 1.- Crear/modificar programas sociales.
- 2.- Crear/modificar Polilineas.
- 3.- Crear/modificar polígonos.
- 4.- Modificar beneficiarios.
- 5.- Modificar geolocalización.

El usuario jefe de área tiene la responsabilidad de planificar, coordinar y controlar al personal que depende de sus áreas, como el trabajo que les permite nutrir al software de información. Al igual que todos los usuarios que interactúan con el software tiene que primeramente autenticarse al iniciar sesión en el sistema. Este regresara con el nivel de usuario y sus privilegios dentro del software y puede desarrollar las funciones que le correspondan, de igual manera funciones que se retomaron de la documentación oficial de la Dirección

de Desarrollo Social publicadas en la Gaceta de Gobierno es la Número 65 del día 4 de abril del 2018.

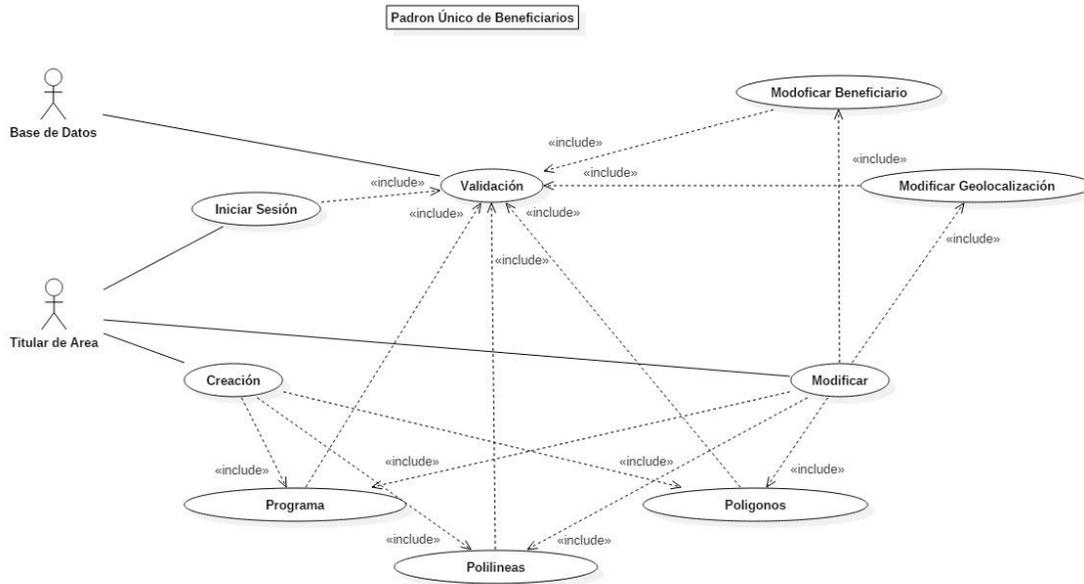


Figura 11. Diagrama de casos de uso de creación, modificación y/o actualización de la información.

Fuente: Elaboración propia.

El usuario jefe de área tiene la responsabilidad de crear/modificar programas sociales los cuales son directamente proporcionales a los que se ejecutaran en la dirección para ejemplificar se muestra la figura 12, la creación de los programas en la dirección tiene una vinculación directa con los beneficiarios, conformado el padrón de cada uno de ellos.

La modificación de los programas dependerá de la ejecución y la gestión anual de ellos, ya que existen programas sociales que se gestionan, se implementan y se terminan en un periodo de duración demasiado corto, pero sin importar la duración o el tamaño de beneficiarios, es necesario que todos estos existan en la base de datos, ya que además de llevar un control ordenado y progresivo de ellos, servirá para en un caso de auditoria presentar dicha información como evidencia de la ejecución acertada del programa.

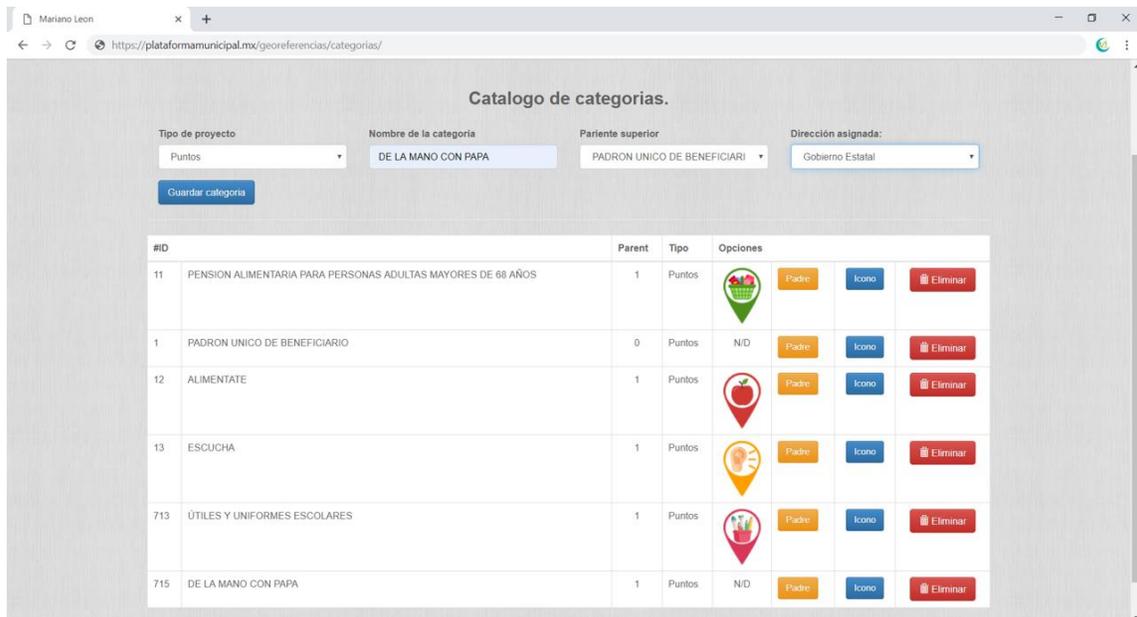


Figura 12. Creación de programas sociales.

Fuente: Elaboración propia.

El usuario jefe de área el software le permite generar polilíneas que es una línea delimitando un área en una superficie de territorio, en este caso las polilíneas servirán para señalar a un beneficiario, programa u obra municipal sobre la superficie de los mapas de Google y para ejemplificar lo antes mencionado se muestra la figura 13.

La planeación y ejecución del programa es de vital importancia, ya que de los objetivos y metas dependerá la organización y conformación en el sistema. La flexibilidad del sistema permite la modificación de la polilínea, que en un sentido estricto no es recomendada porque demandara de diferentes recursos para su modificación, iniciando con una nueva planeación, objetivos, metas, además de nuevamente vincular en el sistema a los usuarios con la nueva información.

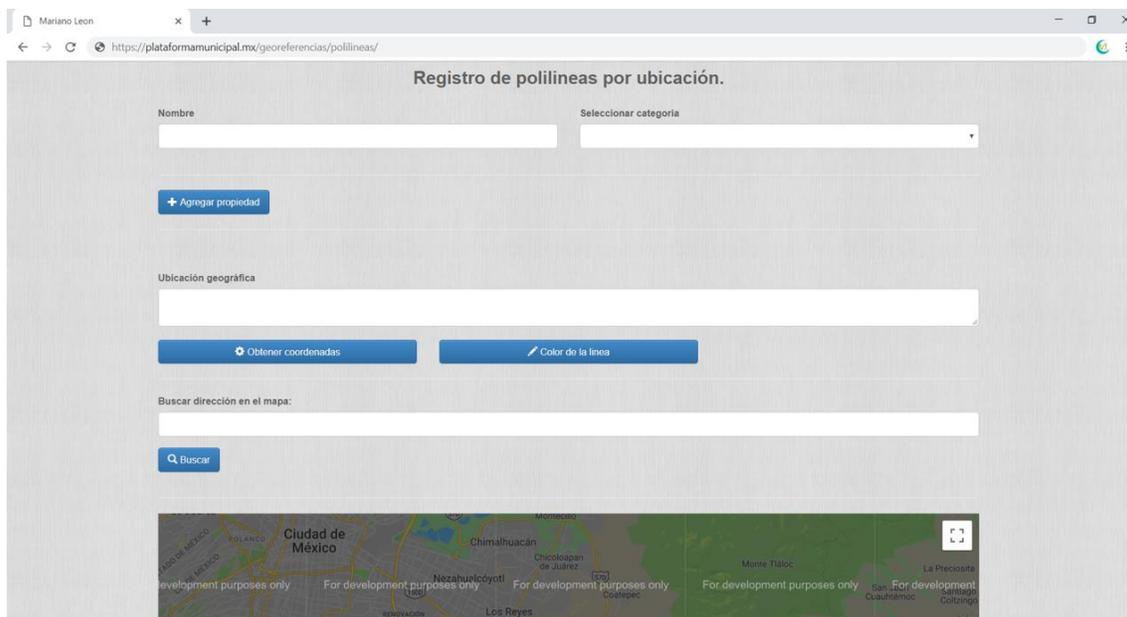


Figura 13. Creación de polilínea.

Fuente: Elaboración propia.

El usuario jefe de área como en el caso anterior también podrá crear/modificar polígonos, estos son formas a mórficas que delimitan un área en específico que puede representar desde una comunidad, un pueblo, barrio, colonia, unidad habitacional, una manzana, una sección electoral, una casa o un beneficiario, en la figura 14 se muestra la interfaz del sistema.

De igual manera que el caso anterior el trazado de estos polígonos serán realizados por este usuario sobre los mapas de Google en el territorio marcado como municipio de Chalco. Sin el ánimo de ser redundante, y con la finalidad de dejar en claro se retoma la explicación de que el sistema permitirá la modificación del polígono, siendo flexible con las circunstancias que pueden presentarse para que pudiera realizarse dicha modificación; sin embargo, tendría que realizar nuevamente el trabajo iniciando con una nueva planeación, objetivos, metas, además de nuevamente vincular en el sistema a los usuarios con la nueva información del polígono ya modificado, realizar el mismo

procedimiento hasta cumplir con todos los beneficiarios que tengan que ser dados de alta en el propio programa social.

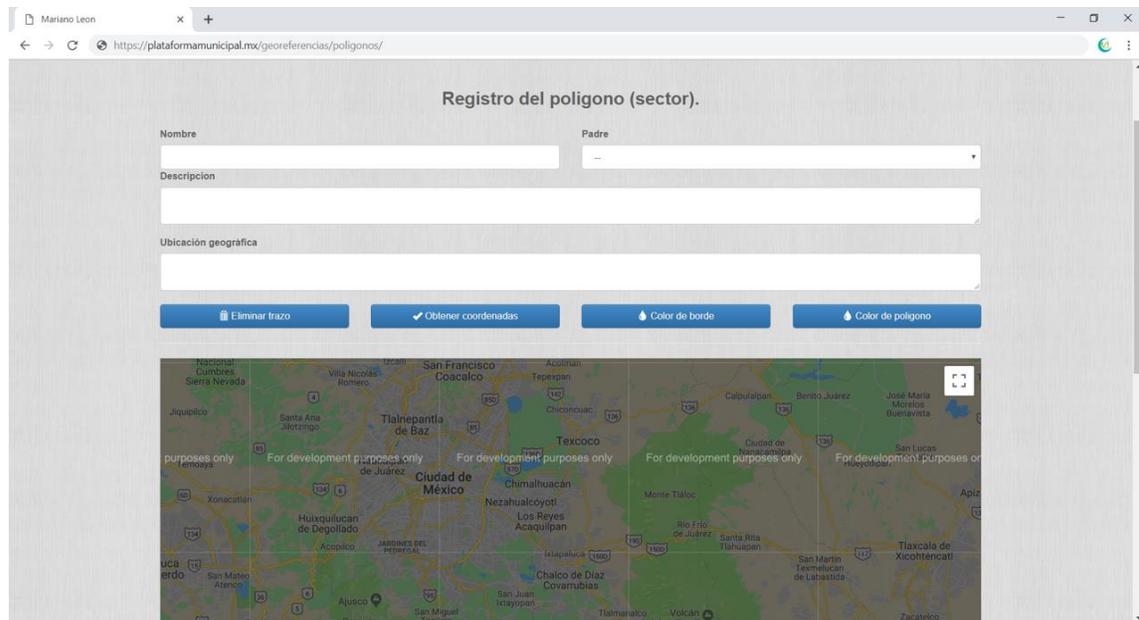


Figura 14. Creación de polígono.

Fuente: Elaboración propia.

El usuario jefe de área podrá modificar los usuarios que se hayan dado de alta dentro del software, si bien es cierto que este usuario no da de alta los beneficiarios de los diferentes programas ejecutados por la Dirección de Desarrollo Social, propiamente tiene la jerarquía de poder modificarlos, esto por alguna situación en que los promotores capturistas no hayan podido realizar debidamente sus actividades y exista una discrepancia en los datos almacenados en la base de datos generada por el sistema, esto en el caso de que se trate de pocos usuarios los cuales puedan ser modificados de manera que los tiempos y capacidades del servidor público jefe de área se lo permita, de caso contrario tendrá que realizarse nuevamente el trabajo realizando todos los pasos desde una nueva planeación, objetivos, metas, entre otras actividades.

El usuario jefe de área podrá modificar a su vez la geolocalización con la que fueron dados de alta los usuarios en un principio por los capturistas o promotores. De misma forma que en el caso anterior este podrá ser dependiendo de los tiempos y capacidades del servidor público jefe de área se le permita, de caso contrario tendrá que realizarse nuevamente el trabajo realizando todos los pasos desde una nueva planeación, objetivos, metas, entre otras actividades.

Un factor que es diferente al caso anterior es que se depende de un sistema externo al que no se tiene el control que son los mapas de Google, en el caso de que existiera un problema con ellos, se quedará dependiente del tiempo y de la solución que le den los desarrolladores de esta aplicación, esta situación tendrá que ser contemplada y documentada como una debilidad del propio software.

Por último, en todos procesos que tenga que realizar este usuario desde el inicio de sesión, creación, modificación tendrán que pasar por el proceso de validación en la base de datos para hacer cumplir la normalización de bases de datos. En el tercer diagrama UML de nombre captura de beneficiarios o nombrado también rol del usuario promotor, podría decirse que es el más importante, aunque a consideración de su servidor puedo decir que es la de más labor, puesto que es aquí donde solo se dedican a alimentar a la base de datos a treves de software web, puesto que dado el trabajo de planeación, objetivos, metas fue realizado previamente, ya con una dirección y metas claras ahora solo los promotores pueden nutrir el sistema de información.

Al igual que todos los usuarios que interactúan con el software tiene que primeramente autenticarse al iniciar sesión en el sistema. Este regresara con el nivel de usuario y sus privilegios dentro del software y puede desarrollar las funciones que le correspondan mostrados en la figura 15, de igual manera funciones que se retomaron de la documentación oficial de la Dirección de

Desarrollo Social publicadas en la Gaceta de Gobierno es la Número 65 del día 4 de abril del 2018.



Figura 15. Agregar beneficiario.

Fuente: Elaboración propia.

Aunque dentro del sistema el jefe de área en este apartado solo tiene injerencia o participación al modificar un beneficiario, también en el trabajo administrativo dependerá de dar buenas instrucciones a sus subordinados para amplificar el contexto cito el libro el arte de la guerra que a la letra dice “si las instrucciones no son claras y las ordenes no han sido explicadas, el comandante tiene la culpa. Pero si las instrucciones han sido explicadas y las ordenes no son ejecutadas de acuerdo con la ley militar, los oficiales son los culpables” (Sun tzu, 2008, 14). Cuando existe planeación y dirección lo único que se puede esperar como resultado es una serie de sucesos satisfactorios para bien del área administrativa.

Es así como el tercer diagrama de UML de nombre captura de beneficiarios o nombrado también rol del usuario promotor se describen gráficamente en la figura 16. En la que intervienen tres actores, y cuatro casos

de uso. Al ingresar la geolocalización al beneficiario al programa que le corresponda dependerá de un actor externo el cual nos deja fuera de nuestro alcance, esto en el caso de que existiera una falla, modificación o actualización del este, el desarrollador del software tendrá que realizar a la actualización al propio sistema de padrón único de beneficiarios para que pueda implementarse de manera correcta.

Uno de los casos de uso que tiene que realizar este usuario desde el inicio de sesión, agregar usuario, agregar geolocalización tendrán que pasar por el proceso de validación en la base de datos, con el fin de pasar cumplir con las normas propias del software.

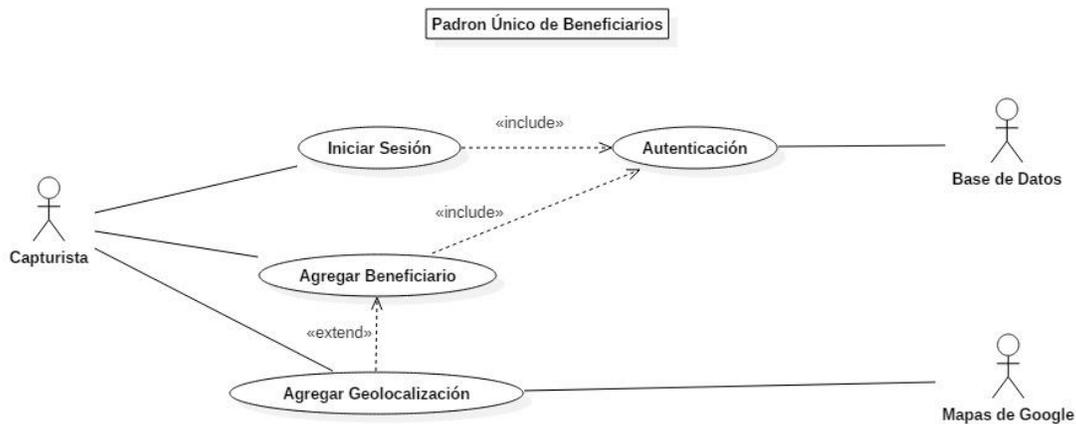


Figura 16. Diagrama de casos de uso de captura de beneficiarios.

Fuente: Elaboración propia.

Por último, se tiene el cuarto diagrama de UML llamado consultas y reportes. El cual no puede faltar en todo sistema de información, puesto que es aquí donde los datos o información ingresados adquiere su fin, porque de que serviría todo esto sino para tomar decisiones, la información sirve de parámetro y de medición, que mostrará si el rumbo por el que vamos es el correcto a través de una proyección en base a los datos obtenidos, así conocer si es correcto o de

ser caso contrario tiene que implementar un proyecto de cambio fijando nuevos objetivos y metas que cumplan con los establecidos en las reglas de operación de los programas sociales gestionados.

Como puede apreciarse en la figura 17 nuevamente el jefe de área es quien tiene la jerarquía y los privilegios de poder solicitar al sistema reportes del trabajo realizado en el mismo software respecto a los programas sociales ejecutados en la Dirección de Desarrollo Social.

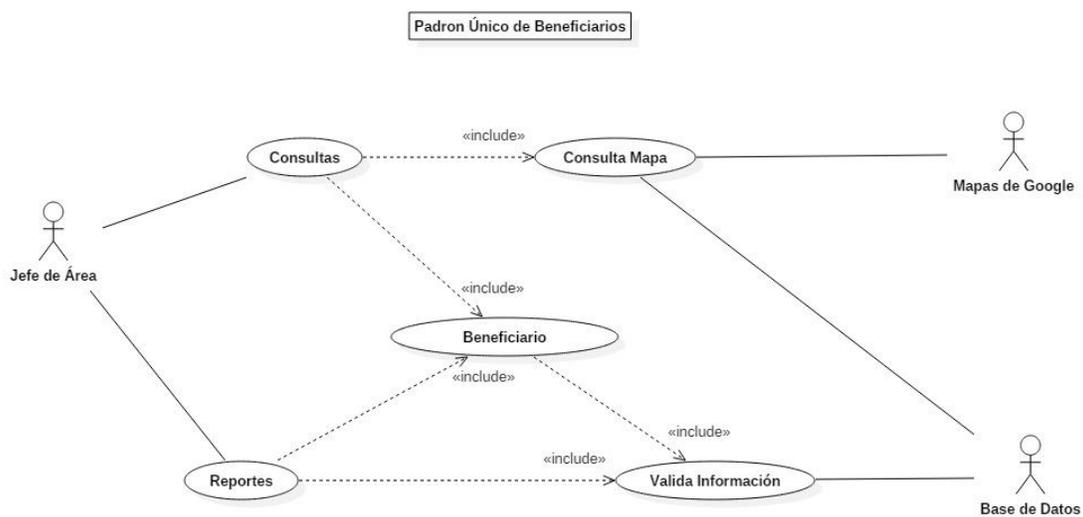


Figura 17. Diagrama de casos de uso de consultas y reportes.

Fuente: Elaboración propia.

Las consultas son solicitudes al sistema que permiten ubicar en el mapa de Google un grupo de beneficiarios enmarcados dentro del área de un polígono o una polilínea y ser agrupados por programa, las consultas ya están previamente establecidas por área administrativa y por programa así que solo pueden ser llamadas y estas serán mostradas en los mapas.

Los reportes son documentos que realiza el sistema en formatos de hoja de cálculo que provienen de la propia información almacenada en la base de datos, es similar al caso de uso de consultas al sistema o software, la diferencia

radica en que las consultas se realizan en una interfaz del propio sistema y un reporte es un documento que genera el software en hoja de cálculo para ser procesado de manera independiente sin que se afecte la base de datos, manteniendo íntegra la información.

En ambos casos la validación en la base de datos es fundamental ya que es de este actor donde se extrae la información realizando consultas previamente establecidas por el desarrollador del software que se muestran en la figura 18.

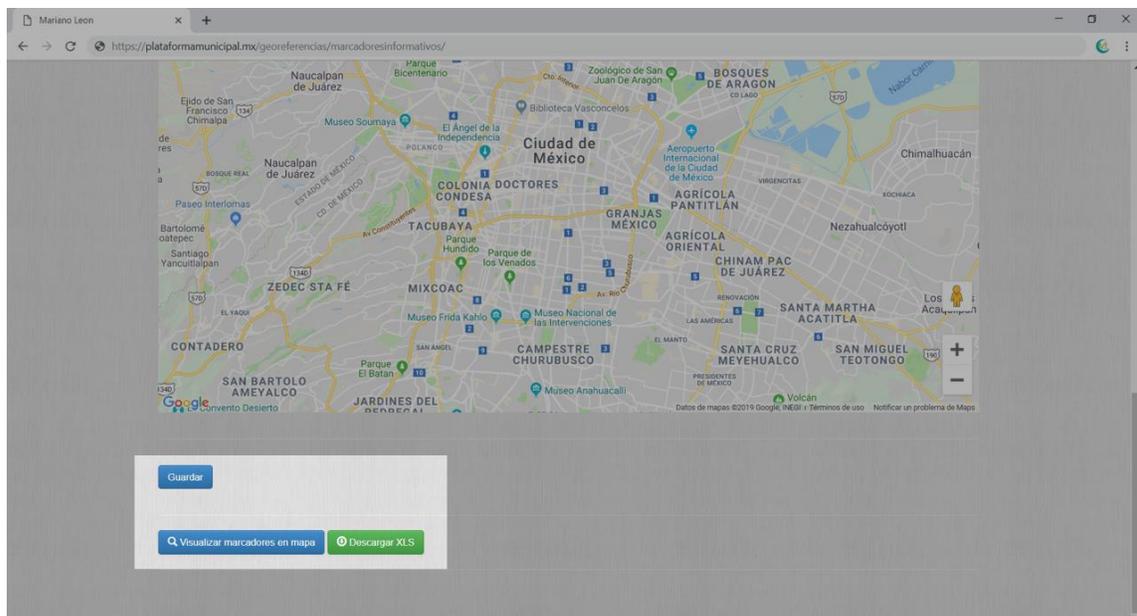


Figura 18. Consultas y reportes.

Fuente: Elaboración propia.

ARQUITECTURA

Los casos de uso y la arquitectura están profundamente relacionados. Los casos de uso deben encajar en la arquitectura, y a su vez la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos, actualmente y a futuro. Contemplando el conjunto de decisiones significativas acerca de la

organización de un sistema software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, las interfaces entre ellos, su comportamiento, sus colaboraciones, y su composición. El modelo de arquitectura de Software es lo esencial y se expresa mediante modelos los diagramas de UML que ya se han expuesto y que se muestran en la figura 19.

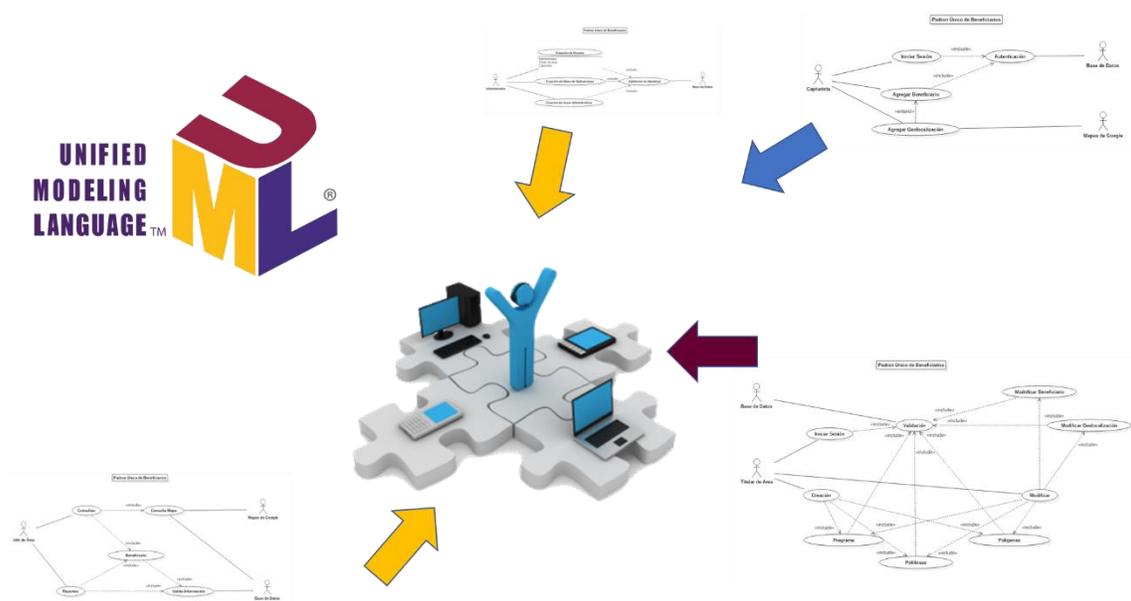


Figura 19. Arquitectura basada en diagramas UML.

Fuente: Elaboración propia.

En la etapa de arquitectura se muestra la visión común del sistema de información completo en la que los usuarios finales como los jefes de área, los promotores y los demás servidores públicos de la Dirección de Desarrollo Social que participan ya que deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. El desarrollador del software y de la arquitectura del sistema de información dirige y coordina el proceso de modelamiento del sistema, él decide cuáles son

actores y los procesos del software y las relaciones entre ellos y cuáles son las reglas debe tener en cuenta.

La base de datos como un contenedor en el que almacenamos la información recopilada de los diferentes programas sociales de la Dirección de Desarrollo Social se relaciona y se organiza, se realizó además una serie de estructuras en tablas con los campos necesarios para almacenar la información. Antes de que antes de que se diseñara las tablas y su relación entre sí, fue preciso que crear ese contenedor de mismo nombre del software padrón único de beneficiarios.

El software que se utilizó para albergar la información fue el que está instalado por default en los servidores más comunes llamado MySQL que se aprecia en la figura 20, de un poco de historia rescatamos que “MySQL es muy utilizado en aplicaciones web, como Joomla, WordPress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL”. (Solano L, 2019)

Los cimientos del sistema que son necesarios como base para comprenderlo, transmitirlo, desarrollarlo y producirlo. El modelo de arquitectura es donde se formulan que pruebas al software se le puedan realizar para que se aborden, se corrijan con la finalidad de que este bien documentado sea fácil de operar, de mantener, de capacitar y que sea iterativo e incremental.

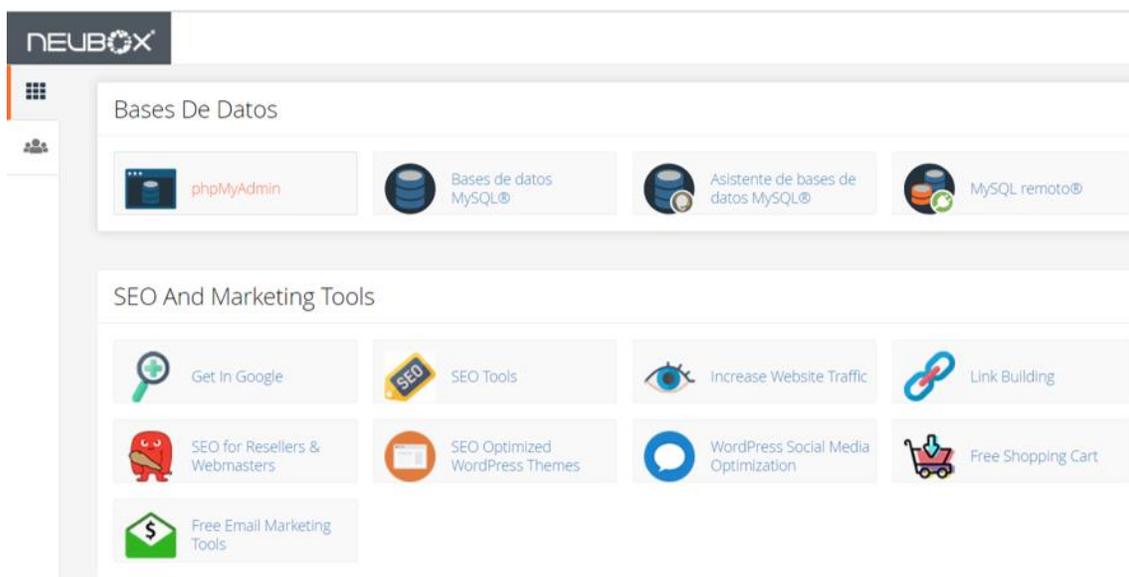


Figura 20. Gestor de base de datos MySQL.

Fuente: Elaboración propia.

Ahora que se ha creado la base de datos, que ya existe en nuestro servidor de bases de datos MySQL. Tendremos todo listo para comenzar a trabajar sobre ella, comenzaremos por nombrar nuestras tablas que al igual que en un editor de textos simple como Word una tabla es una lista de datos, organizados por columnas y filas. Dentro del sistema gestor de bases de datos, la información se puede organizar dentro de las tablas, en las que solo tendremos que decidir qué columnas queremos que tenga y de que tipo va a ser la información que tendrán, mostradas en la tabla 7.

Tabla 7. Tablas de la base de datos.

N. P:	NOMBRE	DESCRIPCIÓN
1	Aplicaciones	Almacena las aplicaciones que se encuentran en el menú, delimita el ambiente en que cada usuario puede interactuar.
2	Direcciones	Almacena la información de dónde proviene el programa que se encuentra en ejecución.
3	Geo_catalogo	En esta tabla se cargará el catálogo de programas que la Dirección de Desarrollo Social de Chalco aplica durante el año, al momento de que se libera la aplicación de un programa se

		cargará enseguida en la tabla para que pueda posteriormente vincularse a sus beneficiarios.
4	Geo_coordenadas	Aquí es donde se almacenan los beneficiarios de los programas ya precargados, es la tabla que mayor información almacenara.
5	Usermenu	Esta es una tabla puente entre las aplicaciones y los usuarios.
6	Usuarios	Almacenara al personal de la Dirección de Desarrollo Social de Chalco que tienen una responsabilidad en el proceso, de los cuales se encuentran; Administrador, jefe de área y capturista.

Fuente: Elaboración propia.

Las tablas que se muestran en la figura anterior tienen estructura propia estructurada para cumplir con su propósito y fin. Se describe la primer tabla llamada aplicaciones, que muestra su estructura en la tabla 8 a la tabla 13.

Tabla 8. Diccionario de datos de tabla aplicaciones.

aplicaciones						
#	Nombre	Tipo	Cotejamiento	Nulo	Predeterminado	Extra
1	id	int(11)		No	<i>Ninguna</i>	AUTO_INCREMENT
2	aplicacion	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
3	modulo	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	

Fuente: Elaboración propia.

Tabla 9. Diccionario de datos de tabla direcciones.

direcciones						
#	Nombre	Tipo	Cotejamiento	Nulo	Predeterminado	Extra
1	id_area	int(11)		No	<i>Ninguna</i>	AUTO_INCREMENT
2	nombre_area	varchar(255)	latin1_swedish_ci	No	<i>Ninguna</i>	
3	archivo	varchar(255)	latin1_swedish_ci	Sí	<i>NULL</i>	
4	tipo	varchar(10)	latin1_swedish_ci	Sí	<i>NULL</i>	

Fuente: Elaboración propia.

Tabla 10. Diccionario de datos de tabla geo_catalogo.

geo_catalogo						
#	Nombre	Tipo	Cotejamiento	Nulo	Predeterminado	Extra
1	id	int(11)		No	<i>Ninguna</i>	AUTO_INCREMENT
2	categoria	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
3	cat_tipo	varchar(100)	utf8_general_ci	Sí	<i>NULL</i>	
4	cat_geo	varchar(50)	utf8_general_ci	No	<i>Ninguna</i>	
5	cat_img	varchar(100)	utf8_general_ci	Sí	<i>NULL</i>	
6	direccion	int(3)		No	<i>Ninguna</i>	
7	cat_parent	int(3)		No	<i>Ninguna</i>	

Fuente: Elaboración propia.

Tabla 11. Diccionario de datos de tabla geo_coordenadas.

geo_coordenadas						
#	Nombre	Tipo	Cotejamiento	Nulo	Predeterminado	Extra
1	id	int(11)		No	<i>Ninguna</i>	AUTO_INCREMENT
2	usuario	varchar(400)	utf8_general_ci	No	<i>Ninguna</i>	
3	fechahora	datetime		No	<i>Ninguna</i>	
4	tipo	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
5	categoria	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
6	subcategoria	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
7	coordenadas	text	utf8_general_ci	No	<i>Ninguna</i>	
8	direccion	int(3)		No	<i>Ninguna</i>	
9	nombre	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
10	apaterno	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
11	amaterno	varchar(100)	utf8_general_ci	No	<i>Ninguna</i>	
12	seccion	int(11)		No	<i>Ninguna</i>	
13	calle	varchar(500)	utf8_general_ci	No	<i>Ninguna</i>	
14	colonia	varchar(500)	utf8_general_ci	No	<i>Ninguna</i>	
15	delegacion	varchar(200)	utf8_general_ci	No	<i>Ninguna</i>	
16	clavee	varchar(50)	utf8_general_ci	No	<i>Ninguna</i>	
17	informacion	text	utf8_general_ci	Sí	<i>NULL</i>	
18	propiedades	text	utf8_general_ci	Sí	<i>NULL</i>	

Fuente: Elaboración propia.

Tabla 12. Diccionario de datos de tabla usermenu.

usermenu						
#	Nombre	Tipo	Cotejamiento	Nulo	Predeterminado	Extra
1	id	int(11)		No	<i>Ninguna</i>	AUTO_INCREMENT
2	iduser	int(11)		No	<i>Ninguna</i>	
3	application	varchar(30)	utf8_general_ci	No	<i>Ninguna</i>	
4	module	varchar(30)	utf8_general_ci	No	<i>Ninguna</i>	

Fuente: Elaboración propia.

Tabla 13. Diccionario de datos de tabla usuarios.

usuarios						
#	Nombre	Tipo	Cotejamiento	Nulo	Predeterminado	Extra
1	id	int(11)		No	<i>Ninguna</i>	AUTO_INCREMENT
2	nomina	int(4)		No	<i>Ninguna</i>	
3	contrasena	varchar(50)	latin1_swedish_ci	Sí	<i>NULL</i>	
4	nombre	varchar(50)	latin1_swedish_ci	No	<i>Ninguna</i>	
5	paterno	varchar(50)	latin1_swedish_ci	No	<i>Ninguna</i>	
6	materno	varchar(50)	latin1_swedish_ci	No	<i>Ninguna</i>	
7	direccion	int(4)		No	<i>Ninguna</i>	
8	correo_electronico	varchar(100)	latin1_swedish_ci	No	<i>Ninguna</i>	
9	fecha_alta	datetime		Sí	<i>NULL</i>	

Fuente: Elaboración propia.

Derivado de la estructura de las tablas mostradas en las tabas previas se generó un diagrama entidad-relación, también conocido como modelo entidad relación, en un tipo de diagrama de flujo que ilustra cómo las entidades, como personas, objetos o conceptos, se relacionan entre sí dentro del sistema. El diagrama entidad relación se usa para diseñar o depurar la base de datos relacional.

En el modelo diagrama de clases se plasma la relación del sistema de información padrón único de beneficiarios, emplean un conjunto definido de símbolos como las líneas de conexión para representar la interconexión de entidades, relaciones y sus atributos. El diagrama entidad relación presentado

es un reflejo de la estructura gramatical que se empleó en las entidades como sustantivos y relaciones como verbos para desarrollar el sistema ver figura 21.

PADRÓN ÚNICO DE BENEFICIARIOS

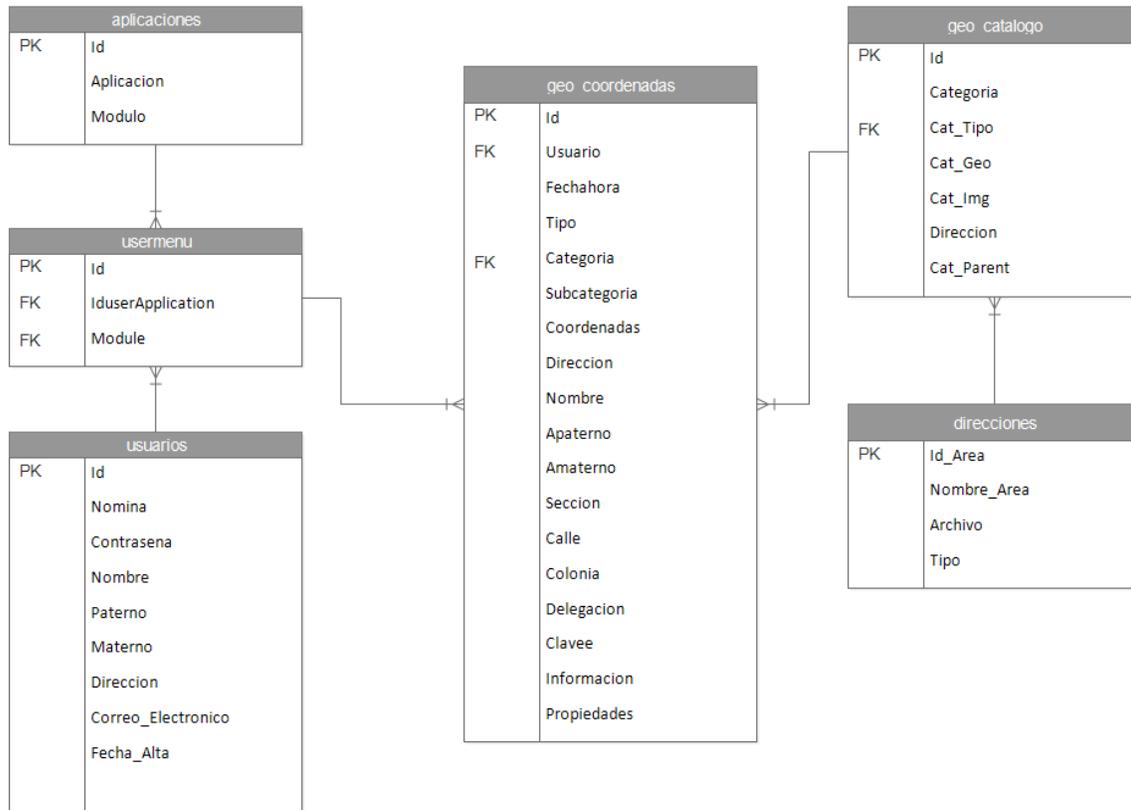


Figura 21. Diagrama de clases.

Fuente: Elaboración propia.

El diagrama entidad relación se utilizó para modelar y diseñar la base de datos, bajo la implementación y en términos de reglas de negocio y lógicas y en términos de la tecnología específica que se implementará en el modelo de datos físicos. En ingeniería de software utilizada, el diagrama entidad relación a menudo

Los diagramas de entidad relación se relacionan con los diagramas de estructura de datos generados en los diagramas de UML, que se centran en las

relaciones de los elementos dentro de las entidades y en los casos de uso, en lugar de las relaciones entre las entidades mismas. Los diagramas entidad relación a menudo se combinan con los diagramas de flujo de datos de UML, que trazan el flujo de la información para procesos o sistemas, comprende que el propósito es solo mostrar las relaciones. Los diagramas entidad relación muestran únicamente la estructura relacional.

El modelado de las máscaras <Ade captura se realizó en la versión 5.5.38 de PHP procesador de Hipertexto con el fin de ser utilizado en la mayoría de los servidores web, en sistemas operativos y plataformas sin ningún costo, siendo en los últimos años reducido progresivamente por la aparición de otras tecnologías como Node.JS, Golang, ASP.NET, entre otras.

La arquitectura se basó en el modelo vista controlador, si hay algo que me gusta de la programación web es que podemos encontrar un sin fin de opciones para desarrollar nuestras aplicaciones web, lenguajes de programación, APIs, frameworks y ciertas herramientas que nos agiliza el proceso de desarrollo.

Al momento de desarrollar el sistema de información es el de tener buenas prácticas, en concreto del lenguaje de programación PHP nos da bastante libertad y fácilmente podemos descuidar el desarrollo de nuestro código obteniendo así uno desordenado y muchas veces mal aplicado. Para solventar este tipo de incidentes existen modelos de desarrollo que nos ayudan a mantener un código mejor estructurado para ello el software está basado en el modelo vista controlador.

El lenguaje de PHP es uno de los lenguajes que sirven para la programación de scripts que se ejecutan del lado del servidor, por lo que las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verán los usuarios finales de la dirección.

Para poder programar en PHP se requiere de un servidor preparado para ello, no tanto programar, sino, que el servidor este habilitado con el lenguaje de programación php, o de ser posible que sea un servidor multiplataforma, el servidor principal web nos servirá para ello, PHP propone un sinfín de funciones para la gestión de archivos que van desde las más elementales de apertura, lectura y cierre a otras más rebuscadas como el cálculo de espacio en el disco duro, tamaño del archivo, gestión de derechos de acceso, entre otras más que siguen facilitando la vida del programador.

Si se buscara referencia de lenguaje de programación PHP, se encontrará que presenta una filosofía de código abierto o libre, a la fecha existen multitud de herramientas que complementan y fortalecen el desarrollo de proyectos gracias a sus; librerías, frameworks, apis gratuitas que llevan PHP a un nuevo nivel y la tendencia seguirá en desarrollo por muchos años más. Además, el propio núcleo del lenguaje tiene una de las más nutridas cantidades de funciones para hacer todo tipo de operaciones, no se necesita invertir nada, para disponer de un lenguaje poderoso y los mejores complementos para acelerar los proyectos basados en este lenguaje acoplados en este proyecto.

Por otra parte, las posibilidades que estas operaciones nos pueden ofrecer conjuntamente son realmente sorprendentes. Sin ir más lejos, y guardando las distancias, escribir y leer archivos puede en cierta manera sustituir muy primariamente a una base de datos. Otra de sus mejoras en cuanto a rendimiento es muy notables y ha situado de nuevo el lenguaje entre los más poderosos. El lenguaje de php en la actualidad está en gran cantidad de servidores, pero sin embargo su adopción todavía no es total.

En la imagen 22 se muestra un fragmento del código utilizado para la máscara de captura o interfaz de usuario para dar de alta un administrador, un jefe de área, un capturista o promotor, la estructura es de un código simple de HTML y de estilos de CCS que se encuentran ya en internet donde se aprecia

que se colocan unas cajas de texto, y una de ellas es de contraseña, un apartado de php donde se realiza la lista, la cual se manda a llamar desde la base de datos las opciones, de resultado las direcciones y poder asignárselas a un usuario al momento de darlos de alta.

```

27 $currentTime = date("H:i");
28 if($currentTime > "16:00"){
29     <!--(Registro habilitado hasta las 4:00pm)-->
30 }
31
32
33 <div class="col-md-12">
34
35 <form method="post" action="">
36 <table class="table-bordered table-striped">
37 <tr>
38 <td>Numero de nominaci</td>
39 <td><input type="text" name="nomina" placeholder="Numero de nomina" class="form-control"></td>
40 </tr>
41 <tr>
42 <td>Contraseña</td>
43 <td><input type="password" name="clave" class="form-control" name="password"></td>
44 </tr>
45 <tr>
46 <td>Nombre</td>
47 <td><input type="text" name="nombre" placeholder="Nombre" class="form-control"></td>
48 </tr>
49 <tr>
50 <td>Paterno</td>
51 <td><input type="text" name="paterno" placeholder="Paterno" class="form-control"></td>
52 </tr>
53 <tr>
54 <td>Materno</td>
55 <td><input type="text" name="materno" placeholder="Materno" class="form-control"></td>
56 </tr>
57 <tr>
58 <td>Correo electrónico</td>
59 <td><input type="text" name="correo_electronico" placeholder="Correo electronico" class="form-control"></td>
60 </tr>
61 <tr>
62 <td>Dirección</td>
63 <td>
64 <select name="direccion" class="validate[required]">
65 <option value="">
66 <!--(Selecciona una dirección)-->
67 <!--(Selecciona una dirección)-->
68 <!--(Selecciona una dirección)-->
69 <!--(Selecciona una dirección)-->
70 <!--(Selecciona una dirección)-->
71 <!--(Selecciona una dirección)-->
72 <!--(Selecciona una dirección)-->
73 </select>
74 </td>
75 </tr>
76 <tr>
77 <td colspan="2"><input type="submit" value="Guardar" class="btn btn-primary" /></td>
78 </tr>
79 </table>
80 </form>
81 </div>
82

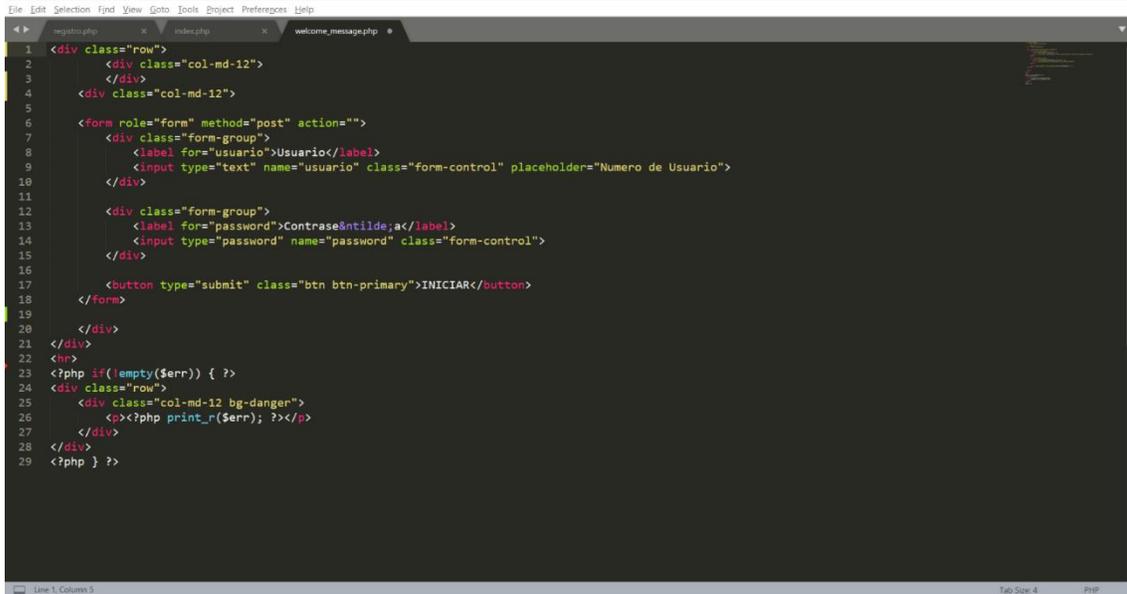
```

Figura 22. Agregar usuario.

Fuente: Elaboración propia.

Para poder tener acceso a cualquiera de las máscaras de captura o interfaz de usuario se debe de contar con privilegios, ingresar con un usuario y contraseña válidos u oficiales para el sistema de información, ya que sin ellos simplemente no mostrara nada, el sistema negara toda interacción, mismo procedimiento en todas y cada una de las carpetas que contengan información de este. El procedimiento de validación se muestra en la figura 23 y la figura 24. De tal forma no cualquier persona puede ingresar al sistema, esto se consideraría parte de la seguridad del mismo sistema y de la información. La información en todo momento el objetivo es tratar de salvarla; sin embargo, en el mundo de internet, no todos los que navegan en la internet son para construir, los crackers siempre estarán al asecho de encontrar una pequeña puerta para

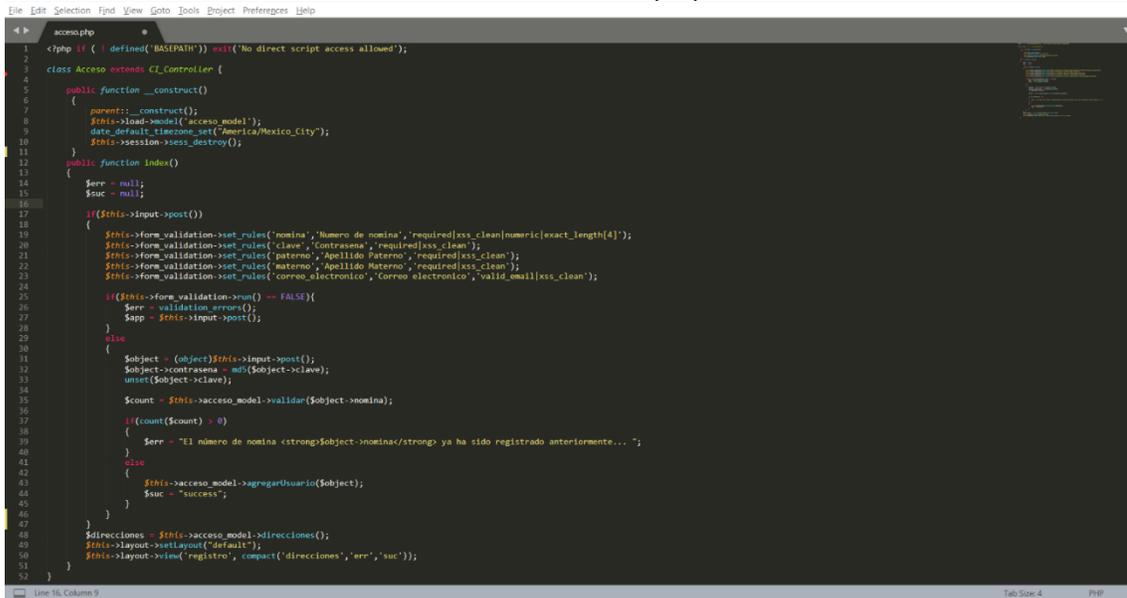
poder destruir, para evitar una catástrofe para ello la información se respalda como mínimo cada 8 días.



```
1 <div class="row">
2   <div class="col-md-12">
3   </div>
4   <div class="col-md-12">
5
6   <form role="form" method="post" action="">
7     <div class="form-group">
8       <label for="usuario">Usuario</label>
9       <input type="text" name="usuario" class="form-control" placeholder="Numero de Usuario">
10    </div>
11
12    <div class="form-group">
13      <label for="password">Contraseña</label>
14      <input type="password" name="password" class="form-control">
15    </div>
16
17    <button type="submit" class="btn btn-primary">INICIAR</button>
18  </form>
19
20 </div>
21 </div>
22 <hr>
23 <?php if(!empty($err)) { ?>
24 <div class="row">
25   <div class="col-md-12 bg-danger">
26     <?php print_r($err); ?></div>
27 </div>
28 </div>
29 <?php ?>
```

Figura 23. Inicio de sesión.

Fuente: Elaboración propia.

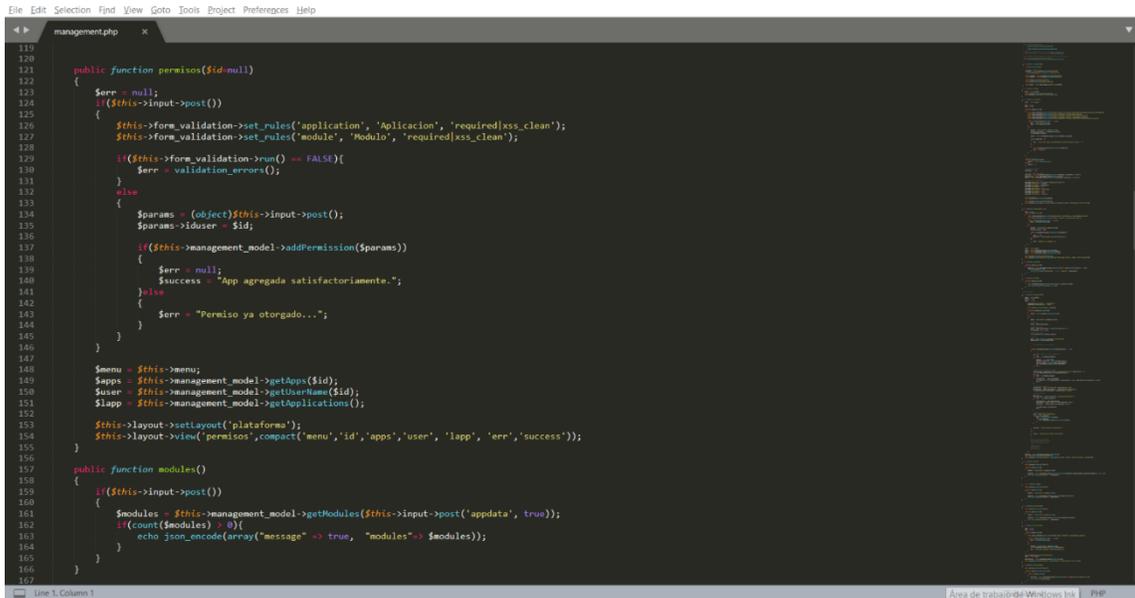


```
1 <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2
3 class Acceso extends CI_Controller {
4
5   public function __construct()
6   {
7     parent::__construct();
8     $this->load->model('acceso_model');
9     date_default_timezone_set('America/Mexico_City');
10    $this->session->sess_destroy();
11  }
12
13   public function index()
14   {
15     $err = null;
16     $suc = null;
17
18     if($this->input->post())
19     {
20       $this->form_validation->set_rules('nomina', 'Numero de nomina', 'required|is_clean|numeric|exact_length[4]');
21       $this->form_validation->set_rules('clave', 'Contraseña', 'required|is_clean');
22       $this->form_validation->set_rules('paterno', 'Apellido Paterno', 'required|is_clean');
23       $this->form_validation->set_rules('materno', 'Apellido Materno', 'required|is_clean');
24       $this->form_validation->set_rules('correo_electronico', 'Correo electronico', 'valid_email|is_clean');
25
26       if($this->form_validation->run() == FALSE){
27         $err = validation_errors();
28         $app = $this->input->post();
29       }
30       else
31       {
32         $subject = (object)$this->input->post();
33         $subject->nonpasena = md5($subject->clave);
34         unset($subject->clave);
35
36         $count = $this->acceso_model->validar($subject->nomina);
37         if(count($count) > 0)
38         {
39           $err = "El número de nomina <strong>{$subject->nomina</strong> ya ha sido registrado anteriormente... ";
40         }
41         else
42         {
43           $this->acceso_model->agregarUsuario($subject);
44           $suc = "success";
45         }
46       }
47
48       $direcciones = $this->acceso_model->direcciones();
49       $this->layout->setlayout('default');
50       $this->layout->view('registro', compact('direcciones', 'err', 'suc'));
51     }
52  }
53 }
```

Figura 24. Control de acceso a máscaras de interfaz.

Fuente: Elaboración propia.

Antes de que la magia funcione mostrare que en la imagen 25 se encuentra fragmento del código que permite asignarle a todos y cada uno de los usuarios previos a darlos de alta los privilegios de los que cada uno va a gozar o más bien la responsabilidad que está a punto de heredarse, el usuario administrador, el usuario jefe de área, y el usuario capturista o promotor.



```
119
120
121 public function permisos($id=null)
122 {
123     $serr = null;
124     if($this->input->post())
125     {
126         $this->form_validation->set_rules('application', 'Aplicacion', 'required|ss_clean');
127         $this->form_validation->set_rules('module', 'Modulo', 'required|ss_clean');
128
129         if($this->form_validation->run() == FALSE){
130             $serr = validation_errors();
131         }
132         else
133         {
134             $params = (object)$this->input->post();
135             $params->iduser = $id;
136
137             if($this->management_model->addPermission($params))
138             {
139                 $serr = null;
140                 $success = "App agregada satisfactoriamente.";
141             }
142             else
143             {
144                 $serr = "Permiso ya otorgado...";
145             }
146         }
147
148         $menu = $this->menu;
149         $apps = $this->management_model->getApps($id);
150         $user = $this->management_model->getUserName($id);
151         $lapp = $this->management_model->getApplications();
152
153         $this->layout->setLayout('plataforma');
154         $this->layout->view('permisos',compact('menu','id','apps','user','lapp','err','success'));
155     }
156
157 public function modules()
158 {
159     if($this->input->post())
160     {
161         $modules = $this->management_model->getModules($this->input->post('appidata', true));
162         if(count($modules) > 0){
163             echo json_encode(array("message" => true, "modules"=> $modules));
164         }
165     }
166 }
167
```

Figura 25. Función permisos.

Fuente: Elaboración propia.

Una de las más importantes operaciones que realiza el software es la asignación del menú a cada uno de los usuarios por tal motivo en la figura 26 se muestra la función que asigna el menú para que pueda observarlo el usuario final y sobre todo que delimita el acceso a solo las tareas que le son correspondientes. El software siendo flexible permite que el administrador pueda de manera temporal o de manera permanente asignarle mayor responsabilidad y modificar el menú y con ello su área de trabajo. Podría no parecer tan importante pero la verdad es que, imaginemos un trabajo extraordinario en el cual todos tuvieran que participar, esta opción es excelente, por ejemplo: Realizar una encuesta la cual todos tienen que capturar, procesar y entregar resultados. Simplemente se crea la aplicación o la interfaz de usuario

y se carga en el software como se muestra en la imagen 27, se asignan privilegios y fácilmente todos podrían contribuir a ayudar sin la necesidad de crear un sistema nuevo, y de esa manera que se crea, puede destruirse.

```

File Edit Selection Find View Goto Tools Project Preferences Help
menu_model.php
4
5 class Menu_model extends CI_Model {
6
7     function __construct()
8     {
9         parent::__construct();
10    }
11
12    public function get($userid)
13    {
14        $applications = $this->get_applications($userid);
15        $menu = array();
16        $i = 0;
17        foreach($applications as $items)
18        {
19            $content = new stdClass();
20            $content->application = $items->application;
21            $content->modules = $this->get_modules($items->application, $userid);
22            $menu[$i] = $content;
23            $i++;
24        }
25        return $menu;
26    }
27
28    public function get_applications($userid)
29    {
30        $this->db->select('application')->group_by('application');
31        $response = $this->db->get_where('usermenu', array('iduser' => $userid));
32        return $response->result();
33    }
34
35    public function get_modules($application, $userid)
36    {
37        $this->db->select('module');
38        $response = $this->db->get_where('usermenu', array('application' => $application, 'iduser' => $userid));
39        return $response->result();
40    }
41
42

```

Figura 26. Estructuración del menú.

Fuente: Elaboración propia.

```

File Edit Selection Find View Goto Tools Project Preferences Help
permisos.php
72 <div class="row">
73 <div>
74 </div>
75
76 <div class="row">
77 <div class="col-md-12">
78 <table class="table table-bordered table-striped">
79 <tr>
80 <th>Application</th>
81 <th>Module</th>
82 <th>Eliminar</th>
83 </tr>
84 <tbody>
85 <tr>
86 <td>
87 <tr id="row" $items->id "$">
88 <td>
89 <td>
90 <td class="text-center">
91 <td class="text-center">
92 </tr>
93 </tr>
94 </tbody>
95 </table>
96 </div>
97 </div>
98
99 <script type="text/javascript">
100 $(document).ready(function(){
101     $('#deleteapp').click(function(){
102         e.preventDefault();
103         var target = $(this).attr('data-target');
104         var userid = $('#userid').val();
105         $.post('<php echo base_url(); >management/deleteapp', {appid:target, appuser: userid}, function(response){
106             if(response.response == true)
107             {
108                 $('#row-'+target).fadeOut('slow');
109             }
110         }, 'json');
111     });
112
113     $('#appolist').bind('change', function(){
114         $('#moduleslist').html('');
115         var app = $(this).val();
116         $.post('<? base_url(); >management/modules', {appid: app}, function(data){
117             if(data.message == true)
118             {
119                 $(data.modules).each(function(i, v){
120                     $('#moduleslist').append('option' + v.module + '</option>');
121                 });
122             }
123         }, 'json');
124     });
125 });
126 </script>
127

```

Figura 27. Agregar o eliminar aplicaciones.

Fuente: Elaboración propia.

El modelo vista controlador es un patrón de arquitectura de software que nos aporta separando los datos de nuestra aplicación de la lógica y de la interfaz de usuario, obteniendo así tres componentes relacionados entre sí. El patrón Modelo-Vista-Controlador para el diseño de aplicaciones Web es un estándar de la industria en el mundo, el modelo MVC viene de palabras del idioma ingles: Model, View, Controller, o bien: Modelo, Vista y Controlador. La idea básica de este patrón es separar nuestros sistemas en tres capas, el Modelo, la Vista y el Controlador, los cuales en el sistema se encuentra en la carpeta application en el servidor que se muestran en la figura 28.

Name	Size
barcode	4 KB
cache	4 KB
config	4 KB
controllers	Controlador 4 KB
core	4 KB
errors	4 KB
helpers	4 KB
hooks	4 KB
language	4 KB
libraries	4 KB
logs	4 KB
models	Modelo 4 KB
third_party	4 KB
views	Vista 4 KB
.htaccess	13 by
index.html	114 t

Figura 28. Contenido de carpeta application.

Fuente: Elaboración propia.

En la capa de modelo o modelado se encarga de todo lo que tiene que ver con la persistencia de los datos, la consistencia de la información que va a ser ingresada a la base de datos del servidor es la capa que valida toda la información que recibe de todas las máscaras de captura o interfaces, aquí es

donde valida la información contenida a través de funciones y métodos estructurados con la simplificación de código.

La capa de vista presenta la información obtenida con el modelo de manera que el usuario la pueda visualizar, en esta capa es donde se realiza la programación de las máscaras de captura, todas las interfaces con las que interactúan todos los usuarios, el administrador, el jefe de área y los capturistas, en capítulos anteriores se han presentado capturas de pantallas de las diferentes interfaces.

El Controlador, dependiendo de la acción solicitada por el usuario, es el que pide al modelo la información necesaria e invoca a la plantilla de la vista que corresponda para que la información sea presentada. En los diferentes archivos php contenidos en esta carpeta se tienen almacenadas las funciones las cuales interactúan con el sistema recibe valores de entrada, parámetros y el cual retorna un valor resultado.

Las funciones que son más recurrentes se pueden invocar o ejecutar desde cualquier parte del programa, es decir, desde otra función, desde la misma función, cuantas veces sea necesario, funciones como la que valida que menú le corresponde a cada usuario, la que valida o autentica el inicio de sesión de todos y cada uno de los empleados de la dirección, entre otras.

En el servidor se creó una estructura de directorios donde se encuentran separadas cada una de las capas; modelado, vistas y controladores para que de esa manera conforme crezca el sistema se sabrá donde se encuentra ubicado cada uno de los archivos. Las funciones que serán llamadas continuamente se creó una clase padre para que todas sus clases hijos hereden de ella, de esa forma optimizar más el código. Para comprender mejor el modelado MVC se muestra la figura 29, si queremos en algún momento reinventar o escalar el sistema tengamos que afrontar el proyecto trabajar con un MVC es una buena opción.

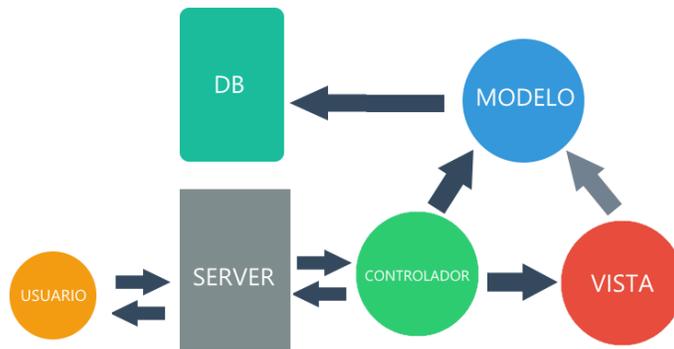


Figura 29. Modelado MVC.
Fuente: Elaboración propia.

ITERATIVO E INCREMENTAL

Es práctico dividir el esfuerzo de desarrollo del software en partes más pequeñas, como dividirlo en mini proyectos. Para tal motivo se generaron los diagramas de UML para muestran estas pequeñas etapas del programa y facilita su creación y una mayor comprensión al momento de programar. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos a crecimientos en el producto ejemplificado en la figura 30, de un modelo de proceso incremental.

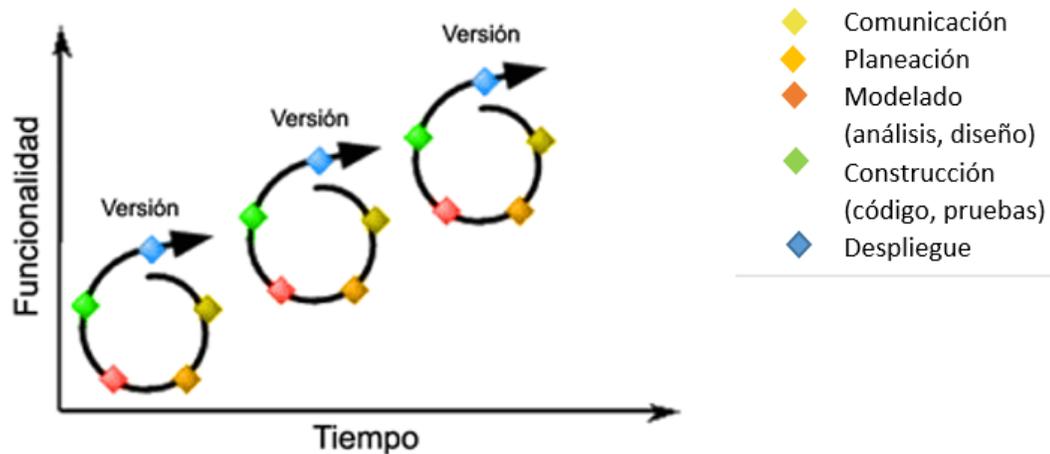


Figura 30. Modelado de proceso incremental.

Fuente: (<https://ingenieriaensoftware.nathalyalava.wordpress.com/2015/04/25/modelos-de-procesos-prescriptivos/>)

Las iteraciones fueron controladas. En donde cada mini proyecto es una iteración que resulta en un incremento. Esto significa que se seleccionaron y ejecutaron de una forma planificada. El desarrollador del software se basó en la selección de lo que implementará en cada iteración en dos cosas: el conjunto de casos de uso que amplían la funcionalidad, y en los riesgos más importantes que deben mitigarse.

Cuando se utiliza un modelo incremental, el primer incremento o hito es a menudo un producto esencial, rudimentario, sólo con los requisitos básicos. Centrándose en la entrega del producto, software o sistema de información cada incremento. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan a los usuarios la funcionalidad que precisa y también una plataforma para la evaluación.

En cada iteración se identificó y especifico los casos de uso relevantes se tomará como un máximo de la escala con la marca de 4X y la de menor importancia la marca de 1X, creando un diseño utilizando la arquitectura seleccionada como guía, para implementar dichos casos de uso. Si la iteración cumple sus objetivos, se continúa con la próxima mostrado en la tabla 14. Sino deben revisarse las decisiones previas y probar un nuevo enfoque, tener que replantear la dirección del software para cumplir con las metas y objetivos que la dirección de Desarrollo Social busca con la implementación de este sistema de información.

Tabla 14. Beneficios del enfoque iterativo.

N. P.	BENEFICIO	IMPORTANCIA
1	La iteración controlada reduce el riesgo a los costos de un solo incremento.	XXXX
2	Reduce el riesgo de retrasos en el calendario atacando los riesgos más importantes primero.	XXX
3	Acelera el desarrollo. Los trabajos se incrementan, pero son más eficientes al obtener resultados a corto plazo	XXX

4	Tiene un enfoque más realista al reconocer que los requisitos no pueden definirse completamente al principio	XX
---	--	----

Fuente: Elaboración propia.

Si algo es cierto y uno debe tomar en cuenta que nunca un software va a estar terminado, ya que las modificaciones físicas y lógicas nunca están estáticas, sino que están en constante cambio, en innovación permanente. El modelo incremental tiene sus desventajas, no es recomendable para casos de sistemas de tiempo real, en donde el alto nivel de seguridad es primordial, de procesamiento distribuido de alto índice de riesgos.

Unas de las preguntas que se puede un desarrollador hacer para saber si el software o sistema puede seguir escalando son: ¿qué le hace falta?, ¿en qué se puede mejorar?, ¿con quién puede trabajar o fusionar su código para complementar o complementarse?, ¿hacia dónde va la tendencia o donde estará la demanda el día de mañana? Preguntas como estas en la que la respuesta se encuentra reestructurando el código, los diagramas, inventar y renovar la información la hace que siempre este un en permanente cambio en un estado iterativo e incremental.

EL CICLO DE VIDA

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo constituye una versión del sistema. Se espera que desarrollar un componente de aplicaciones verticales ágil a la hora de adaptarse a los cambios de procesos y cambios organizacionales permitiendo ser flexible para la Dirección de Desarrollo Social respecto al personal que pueda operar en él.

El ciclo de vida sirve como un bloque de construcción para los demás modelos de ciclo de vida esto lo hace iterativo que puedan desarrollarse hasta

llegar al previamente definitivo, aquí es donde se vuelve incremental. Cada fase tiene un conjunto de metas bien definidas respecto a las reglas de operación vigentes de los programas sociales, que parte de las objetivos, metas y actividades que ahí se encuentran, se desarrollan conceptos, políticas, lineamientos que constituyen el elemento de integración, desde el punto de vista de la Dirección de Desarrollo Social que se construyen como composición de la automatización de procesos basados en reglas.

Durante el desarrollo del software podemos caer en el flujo en cascada, que es un modelo diferente al que estamos utilizando ya que organiza los proyectos en etapas que se ejecutan secuencialmente, se ejecutan varias iteraciones, mientras que cada iteración ejecuta cada una de las etapas definidas una sola vez, lo que resulta un tanto arriesgado ya que hasta que no finaliza con éxito una etapa no se pasa a la siguiente, de manera que los que interactúan con el software se tienen que realizar proyecciones para planificar y redirigir el trayecto del desarrollo.

FASES

La correcta implementación del modelo está en repetir las etapas de cada ciclo, se va añadiendo funcionalidad al producto y se comprime al máximo el tiempo de las iteraciones, iteraciones que al final se van recortando conforme se presenta el proyecto, se hacen entregas parciales del producto para probar y validar con los usuarios que el producto cumple los requisitos primordiales que solicitan en las reglas de operación de los programas sociales vigentes. Los procesos de aseguramiento de la calidad aplicados a nivel del proyecto constan de cuatro fases:

1. Inicio: Es definir los alcances del proyecto, objetivos.
2. Elaboración: Planificar el proyecto elaborar una estructura base.
3. Construcción: Construir el sistema de información o software.
4. Transición: transición a los usuarios a su implementación.

Serán actividades verticales pudiéndose ejecutar en cualquier fase del proyecto ver en la figura 31 la aplicación en un modelo de estado ideal.

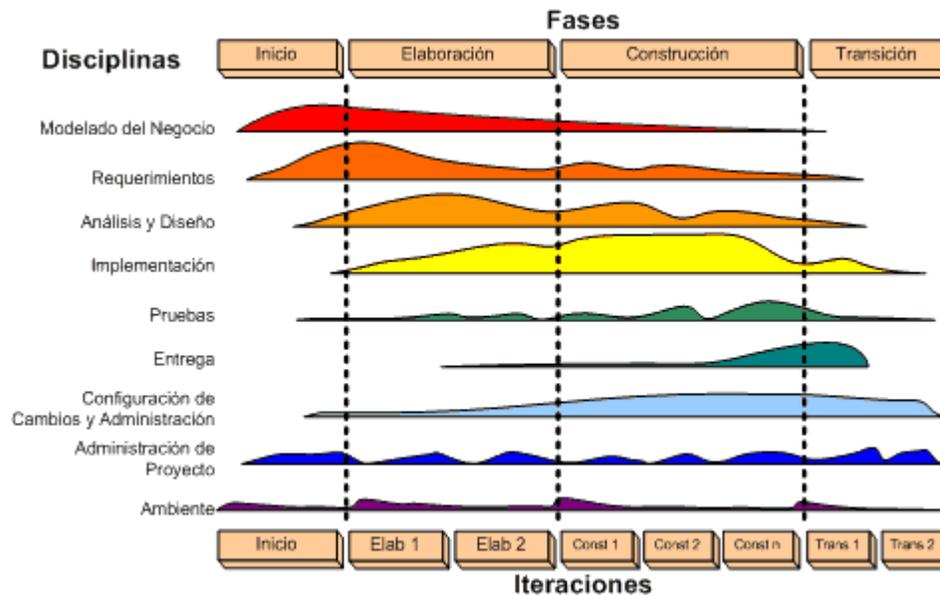


Figura 31. Ciclo de vida modelo RUP.

Fuente: (<http://proyectosociotecnologicodesailinperez.blogspot.com>).

DISCIPLINAS

Cada disciplina es considerada como un conjunto de actividades relacionadas, vinculadas a un área específica dentro del proyecto de desarrollo del software. La disciplina de implementación tiene como propósito albergar este marco de trabajo general para cualquier lenguaje orientado a objetos. Por tanto, es necesario especializarse en la disciplina de implementación, apegarse al desarrollo del sistema de información para explotar al máximo las posibilidades y manipular las limitaciones de cada lenguaje de programación.

Las actividades relacionadas más importantes son:

1. Requerimientos.
2. Análisis y diseño.

3. Codificación.
4. Prueba.

El agrupamiento de actividades en disciplinas es principalmente una ayuda para comprender el proyecto desde la visión tradicional en cascada para ejemplificar ver la tabla 15.

Tabla 15. Fases e iteraciones y disciplinas.

N. P.	FASES		ITERACIONES	DISCIPLINAS
1	Inicio	Alcances y objetivos	Iteración 1	Requerimientos
2	Elaboración	Arquitectura	Iteración 2	Análisis y diseño
3	Construcción	Versiones BETA	Iteración ...	Codificación
4	Transición	Versión Final	Iteración n	Prueba

Fuente: Elaboración propia.

El modelo de Proceso Unificado RUP, consiste en una serie de disciplinas que van desde los requisitos hasta las pruebas. Cada disciplina está asociada con un conjunto de modelos que ayudan y que se desarrollan, están compuestos por instrumentos, entre los más importantes están los flujos de trabajo que desarrollan desde el modelo de casos de uso, hasta el modelo de pruebas que se muestran en la tabla 16.

Tabla 16. Modelos de pruebas.

N. P.	Disciplina	Modelo
1	Requisitos	Modelo de casos de uso.
2	Análisis	Modelo de análisis.
3	Diseño	Modelo de despliegue.
4	Implementación	Modelo de implementación.
5	Prueba	Modelo de prueba.

Fuente: Elaboración propia.

HITOS

Los hitos son logros que se van alcanzando conforme se van superando cada una de las fases y cada fase finaliza con un hito. Cada hito se determina por la disponibilidad de un conjunto de instrumentos, documentos que han sido desarrollados hasta alcanzar un estado predefinido.

Los hitos tienen objetivos clave que se han sido el resultado del análisis y estudio de toda la metodología de desarrollo de software. El más crítico de los hitos es en el que el desarrollador debe tomar ciertas decisiones antes de que el trabajo continúe con la siguiente fase, el desarrollador tendrá que ser ayudado de los usuarios finales, en este caso servidores públicos y todo aquel que tenga una participación en el sistema de información.

Los hitos también permiten controlar la dirección y progreso del trabajo, pues son estos logros que se deben alcanzar, son objetivos que se deben de cumplir. Al final se obtiene un conjunto de datos a partir del seguimiento del tiempo y esfuerzo consumidos en cada fase concluida satisfactoriamente del software. Estos datos son útiles para la estimación en futuros proyectos o la incrementación de este, futuras versiones.

FASE DE INICIO

Durante esta fase de inicio se desarrolla una proyección o descripción del producto final tomando como base tiempos, recursos, y se presenta el análisis del sistema de información. El saber cuáles son las principales funciones del sistema, y que podemos priorizar como de vital importancia para los usuarios de un nivel jerárquico superior, como la captura de beneficiarios, las consultas y reportes y estos se encuentren listos en la figura 32, es señal de que se están cumpliendo los hitos.

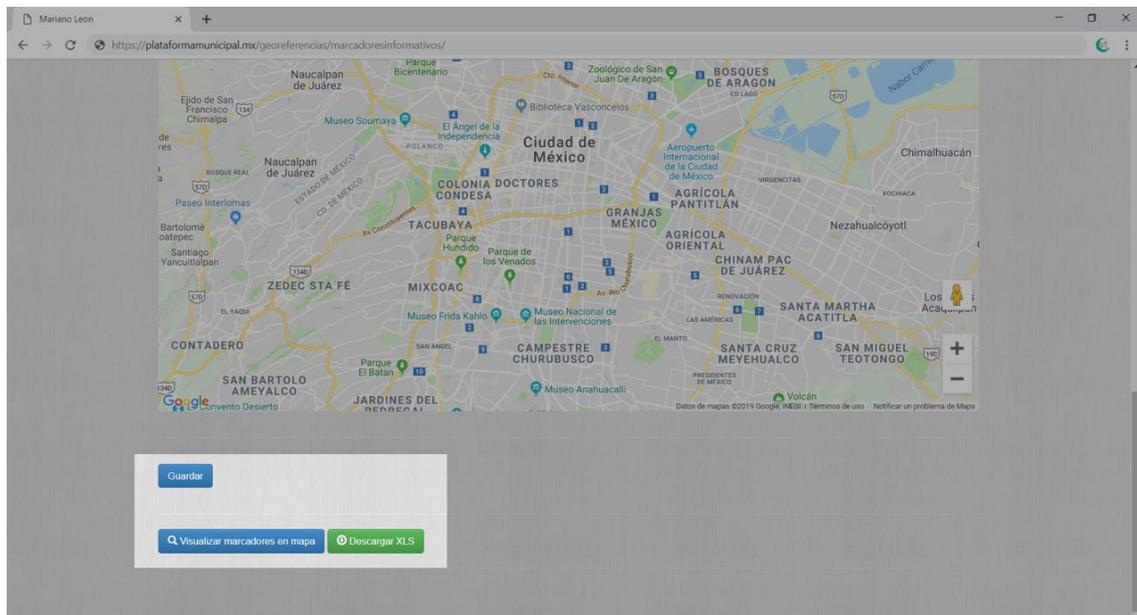


Figura 32. Consultas y reportes.

Fuente: Elaboración propia.

Identifican y priorizan los riesgos más importantes, porque es en la planeación de aplicación de los programas sociales que se determina en la dirección, puesto que como ya lo he mencionado si este no cuenta con una buena planeación así el sistema o software esté funcionando en óptimas condiciones la información no será útil y se tendrá que rehacer el trabajo para cumplir con lo en marcado en las reglas de operación del programa.

El objetivo de esta fase es ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto. Las iteraciones exploran diferentes soluciones posibles, y diferentes arquitecturas posibles. Puede que todo el trabajo físico realizado en esta fase sea descartado. Lo único que normalmente sobrevive a la fase de inicio es el incremento del conocimiento en el sistema de información.

Los artefactos que típicamente sobreviven a esta fase son:

- Los casos de uso.

- Un boceto inicial de la arquitectura.
- Una descripción de los objetivos del proyecto.
- Una versión muy preliminar del plan del proyecto.
- Un modelo del negocio.

Determinar si es posible obtener una solución que satisfaga a los requerimientos significativos de la arquitectura con el conjunto de necesidades de la Dirección de Desarrollo Social, y que conjunto de funciones satisfacen estas necesidades.

Porque es muy fácil dejarse llevar por lo que se ha determinado que va a estar dentro del sistema y que fuera del sistema, la línea es muy delgada, podría ser imperceptible, pero es de suma importancia ya que, al no llegar a un acuerdo con todas las personas involucradas sobre los requisitos funcionales del sistema o software, puede ser no funcional. Por tal motivo la arquitectura que pueda soportar estas características no sea la adecuada. De nuevo recalco que es de suma importancia identificar los riesgos críticos, conocerlos, para encontrar la manera de mitigarlos.

FASE DE ELABORACIÓN

En esta fase de elaboración se comienza con un análisis completo de los elementos significativos de la arquitectura del software como parte de la 1ra iteración o el primer intento de desarrollo, y en las siguientes iteraciones se refina la arquitectura hasta diseñar todos sus elementos.

Durante la fase de elaboración se especifican en detalle la mayoría de los casos de uso del producto en formatos que recaban información relevante al software y se diseña la arquitectura podemos verlos en el anexo VIII.

Las iteraciones en la fase de elaboración:

- Establecen una firme comprensión del problema a solucionar, basados en la metodología empleada de diagramas de UML.
- Establece la fundación arquitectural para el software, el modelado como un boceto que madurara en las diferentes iteraciones del sistema llamadas también versiones de prueba que se muestra en la figura 33.
- Establece un plan detallado para las siguientes iteraciones, identificando que es lo que se debe superar respecto a las iteraciones anteriores.
- Elimina los mayores riesgos, al repasar los diagramas de UML, los casos de uso, los casos de prueba, los modelos de estructura, afinar y refinar los procesos de los algoritmos, hasta volverlos estables y escalables.

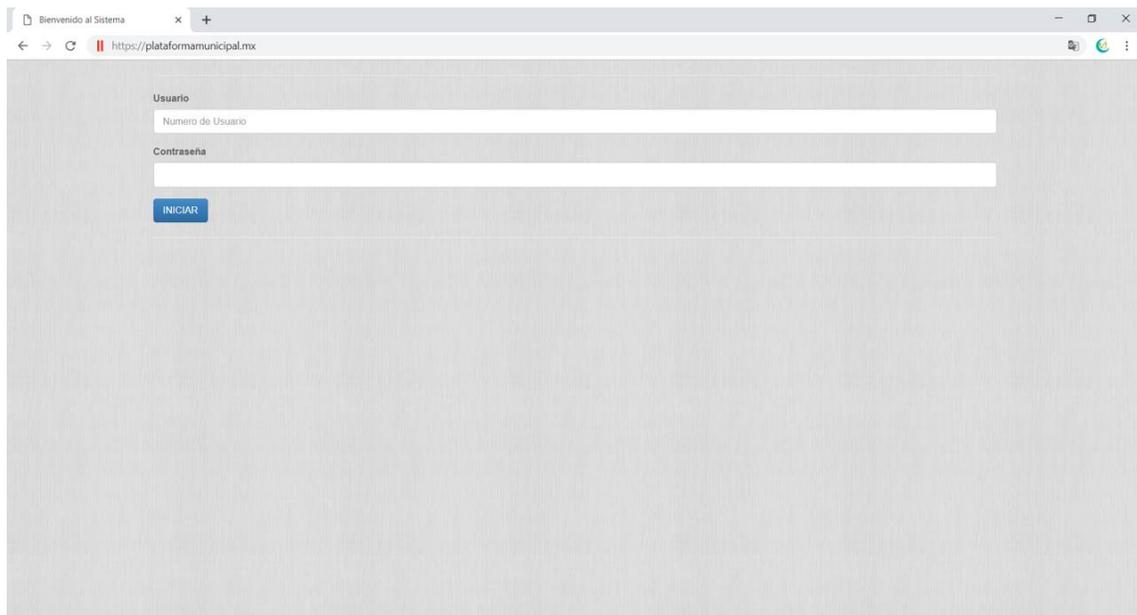


Figura 33. Versión de prueba.

Fuente: Elaboración propia.

La fase de elaboración finaliza con el hito de la Arquitectura del Ciclo de Vida, alcanzar este hito debe poder crear una línea base de la arquitectura adaptable y robusta, que deba evolucionar. haber desarrollado un plan del proyecto hasta el nivel necesario para respaldar una agenda, costes, y calidades realistas, además de proporciona un proyecto, de una adecuada recuperación de la inversión, no financiera precisamente, pero que se noten los resultados de la implementación del software.

FASE DE CONSTRUCCIÓN

En esta fase de construcción se crea el producto. La línea base de la arquitectura ya planteada anteriormente, crece hasta convertirse en el sistema completo, funcional es escalable. Al final de esta fase, el producto contiene todos los casos de uso implementados, sin embargo, puede que no esté libre de defectos, siempre se encontrara la forma de mejorar un producto, de innovarlo, de ampliarlo, por eso siempre se debe dejar la puerta abierta a través de los instrumentos producidos durante esta fase los cuales son:

1. El sistema software
2. Los casos de prueba
3. Los manuales de usuario

La fase de construcción finaliza con el hito de Capacidad Operativa Inicial como se muestra en la tabla 17.

Tabla 17. Fase de Construcción.

FASE	OBJETIVOS	MODELO FUENTE	MODELO PRODUCTO DE TRABAJO
CONSTRUCCIÓN	Desarrollar productos a ser liberados.	Arquitectura ejecutable.	Modelo de despliegue.

	Hacer la integración de subsistemas.	Modelo de componentes.	Programa de software de la solución.
	Realizar pruebas de unidad.	Esquema de la base de datos.	Resultado de pruebas de unidad.
	Realizar pruebas de integración.		

Fuente: (<https://es.slideshare.net/mstabare/ingeniera-de-software-ii-parte-2>).

FASE DE TRANSICIÓN

La fase de transición cubre el período durante el cual el producto se convierte en la versión beta o de prueba final. Las iteraciones en esta fase continúan agregando características al software. Sin embargo, las características se agregan a un sistema que el usuario se encuentra utilizando activamente puede apreciarse en la tabla 18.

Los instrumentos construidos en esta fase son los mismos que en la fase de construcción. El equipo de trabajo de la Dirección de Desarrollo Social se encuentra ocupado fundamentalmente probar, señalar, para que el desarrollador se enfoque en corregir y extender la funcionalidad del sistema desarrollado en la fase anterior.

Tabla 18. Fase de transición.

FASE	OBJETIVOS	MODELO FUENTE	MODELO PRODUCTO DE TRABAJO
TRANSICIÓN	Ejecutar pruebas operativas del sistema.	Modelo de despliegue Modelo de componentes.	Resultado de pruebas funcionales y de la capacidad operativa del sistema.
	Corregir errores de	Esquema de la base	

	construcción.	de datos.	Manuales de Usuario.
	Hacer pruebas para la liberación de productos de trabajo.	Programación de software a ser liberado.	Documento con plan de implementación.

Fuente: Elaboración propia.

La fase de transición finaliza con el hito o logro de Lanzamiento del Producto Final.

VII. IMPACTO DE LA EXPERIENCIA LABORAL

Al laborar en este lugar por un tiempo relativamente largo alrededor de nueve años pude observar que las administraciones municipales varían su organización, su economía y su plantilla laboral. He conocido ayuntamientos que se encuentran en condiciones deplorables donde no cuentan con la infraestructura para que los empleados puedan desarrollar óptimamente su jornada laboral, tienen un rezago en infraestructura, en instalaciones, en mobiliario, equipos de cómputo, sin sistemas de información que hacen que un trámite se prolongue por mucho, pero mucho tiempo.

Caso contrario a los ayuntamientos modernos con instalaciones de primer nivel, equipados en mobiliario, instalaciones, parque vehicular, equipos de cómputo y sistemas de información que pueden hacer y simplificar muchos de los tramites que solían hacerse manera rudimentaria en paquetería ordinaria como procesadores de texto, hojas de cálculo entre otros, que retrasan el fluir de la información. Con los sistemas nuevos a través de aplicaciones web, o aplicaciones móviles por medio de dispositivos inteligentes como SmartPhones, hacen que la vida y los trabajos en ellas puedan simplificar la vida.

El municipio de Chalco se encuentra en el rango de los 20 municipios prioritarios del Estado de México, al cual tanto la federación como el gobierno del estado le dan un presupuesto considerable, lo que permite que tenga un mayor desarrollo y pueda colocarse dentro de los mejores municipios y de vanguardia.

Lamentablemente no cuenta con muchos sistemas de información, son pocas las áreas donde existen y pueden utilizarse, dando la oportunidad de que puedan generarse más sistemas de información similar a este software de control de Padrón Único de Beneficiarios La aplicación de este software fue a finales de una administración por lo que su implementación no fue la óptima para poder almacenar todos los beneficiarios de programas ejecutados en

diferentes años, eso hubiese fortalecido en mucho ya podría darse la oportunidad de poder solventar todos los posibles bugs o errores de sistema.

Además el sistema de información tal y como se encuentra estructurado en la capa de control en la interfaz de aplicaciones exista la posibilidad de crear y seguir enriqueciendo aún más el sistema de información, puesto de la Dirección de Desarrollo Social es solo un brazo o una extensión que conforma toda la estructura administrativa, así pues de otras direcciones manejan sus propios programas sociales, como ejemplo mencionar la Dirección de Desarrollo Económico, la Dirección de Educación o el mismo Sistema Municipal DIF. Hay mucho que hacer todavía, por eso se deja abierta la puerta al desarrollo de otras aplicaciones que puedan ayudar a facilitar el buen manejo de la información para bien de un buen ejercicio de poder y ejercicio de administrar de una manera correcta los erarios. Tal vez aplicando nuevos modelos de interfases de captura.

VIII. REFERENCIAS DE CONSULTA

Branzon, R. (2016). *El estilo virgin. Escuchar, apender, reír y lliderear*. Ciudad de México: Editorial Planeta Colombiana.

Cámara de Diputados del H. Congreso de la Unión. (2018). *Constitución Política de los Estados Unidos Mexicanos*. Recuperado de:

http://www.diputados.gob.mx/LeyesBiblio/pdf/1_270818.pdf

Cámara de Diputados del H. Congreso de la Unión. (2018). *Grupos Vulnerables*. Recuperado de:

http://archivos.diputados.gob.mx/Centros_Estudio/Cesop/Eje_tematico_ol_d_14062011/9_gvulnerables_archivos/G_vulnerables/d_gvulnerables.htm

Congreso del Estado de México. (2018). *Constitución Política del Estado Libre y Soberano de México*. Recuperado de:

<http://legislacion.edomex.gob.mx/sites/legislacion.edomex.gob.mx/files/files/pdf/ley/vig/leyvig001.pdf>

Congreso de la Unión. (2017). *Presupuesto de Egresos de la Federación para el ejercicio fiscal 2018*. Recuperado de:

http://www.dof.gob.mx/nota_detalle.php?codigo=5506080&fecha=29/11/2017

Diario Oficial de la Federación. (2017). *Presupuesto de Egresos de la Federación para el Ejercicio Fiscal 2018*. Recuperado de:

http://www.dof.gob.mx/nota_detalle.php?codigo=5506080&fecha=29/11/2017

Giandini, R. S., & Pons, C. (2000). *Relaciones entre Casos de Uso en el Unified Modeling Language*. *Revista Colombiana De Computacion V Colombian Journal of Computation*, 1(1), 73-90. Recuperado el 2 de 4 de 2019, de:

http://exa.unicen.edu.ar/catedras/modysim/teoria/casos_de_uso_b.pdf

Gómez, J. D. (2016). *Hábitos de ricos, nuevas ideas para alcanzar la libertad financiera*. Ciudad de México: Ediciones Culturales Paidós.

Instituto Nacional de Estadística y Geografía INEGI. (2019). *Número de habitantes*. Recuperado de:

<http://cuentame.inegi.org.mx/monografias/informacion/mex/poblacion/>

Lilia, C. S. (2017). *Elementos básicos en el lenguaje unificado de modelado para análisis de sistemas*. Recuperado el 1 de 4 de 2019, de:

<http://ri.uaemex.mx/handle/20.500.11799/70442>

Municipio de Chalco. (2017). *Plan de Desarrollo Municipal*. Recuperado de:

<https://gobiernodechalco.gob.mx/reglamentacion/standard-post-format/PlandeDesarrollo2017.pdf>

Municipio de Chalco. (2017). *Gaceta 36*. Recuperado de:

<https://gobiernodechalco.gob.mx/reglamentacion/standard-post-format/gaceta36.pdf>

Municipio de Chalco. (2018). *Gaceta 62*. Recuperado de:

<https://gobiernodechalco.gob.mx/reglamentacion/standard-post-format/gaceta62.pdf>

SEDESOL. (2019). *Padrón único de beneficiarios*. Recuperado de:
<https://pub.sedesol.gob.mx/spp/resumenes/padronunico.jsp>

SEDESOL. (2019). *Padrón único de beneficiarios*. Recuperado de:
<https://pub.sedesol.gob.mx/spp/resumenes/padronunico.jsp>

Solano, L (2019). *Historia de MySQL*. Recuperado de:
<https://basededatosutp26.wordpress.com/mysql/>

Sun, Tzu. (2008). *El arte de la guerra*. México Distrito Federal: Grupo Editorial Tomo.

IX. ANEXO I DEFINICIONES

Como anexos se agregan definiciones que dentro del documento que emplean, ya que son términos o palabras reservadas utilizadas de manera permanente en la Dirección de Desarrollo Social.

Beneficiado o beneficiada. Ciudadanos quienes solicitan la integración a un programa social.

Carpeta. Objeto para archivar información.

Coordinador o Coordinadora. Persona que organiza a personas para lograr un objetivo común.

Expediente. Conjunto de todos los documentos relativos a un asunto.

Formatos. Registro de datos para obtener un beneficio de algún programa.

Grupo Vulnerable. Grupo social clasificado por su capacidad para prevenir, resistir y sobreponerse de un impacto.

Instancias. Institución o grupo de poder.

Padrón. Documento oficial con la lista de los habitantes de una población.

Programa Social. Iniciativa a mejorar las condiciones de vida de una población.

Promotor. Se aplica a una persona que promueve o promociona de forma profesional a una persona.

Reglas de Operación. Lineamientos de la Instancia Federal o Estatal para que pueda ser aplicado el beneficio a otorgar.

Gestión. Acción que se realiza por conseguir una cosa.

Oficio. Documento Oficial que sirve para comunicar, solicitar, informar, acordar, etc.

PbRM. Presupuesto basado en Resultados Municipales.

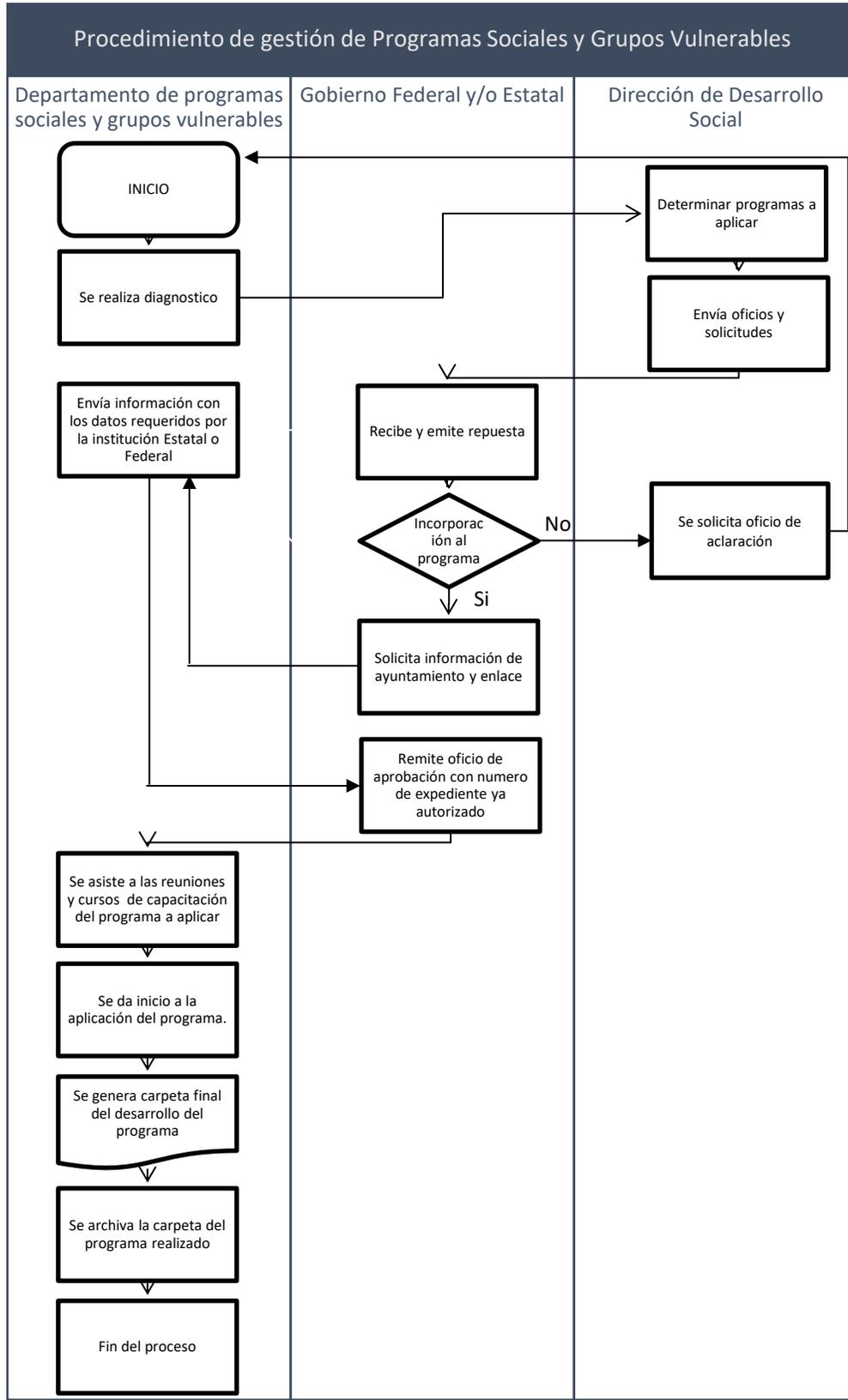
Procedimientos. Pasos predeterminados para desarrollar, una labor de manera eficaz.

Proyecto. Conjunto de actividades que desarrolla una persona o una entidad para alcanzar un determinado objetivo.

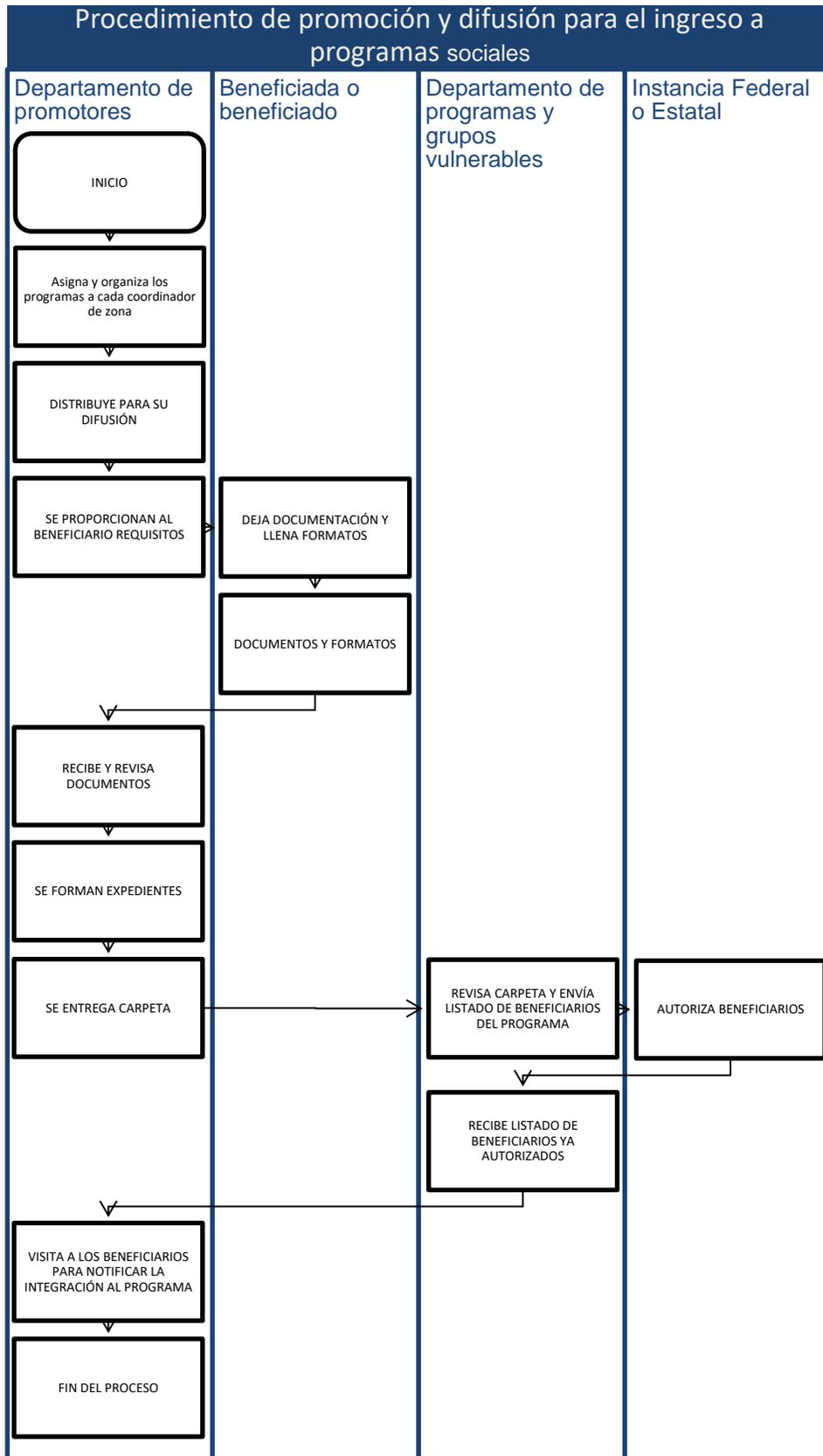
Suficiencia presupuestal. Capacidad de recursos financieros que tienen una partida con la posibilidad de ser efectuada.

Trámite. Cada una de las gestiones que hay que recorrer en la resolución de un asunto.

X. ANEXO II DIAGRAMA DE FLUJO



XI. ANEXO III DIAGRAMA DE FLUJO



XII. ANEXO IV ESPECIFICACIONES DE CASOS DE USO

Nombre del caso de uso del software:	Creación de usuarios, menú y áreas administrativas
Actores del software:	Administrador y Bases de datos
Propósito:	Generar las áreas administrativas, el menú y los usuarios para que de pie a poder el software entrar en operación.
Resumen:	
<p>Crear las áreas administrativas en el software puede dar pie a crear los menús que cada uno de los usuarios que existen en el sistema, crear los usuarios y tener listos los menús es hora de poner en marcha el software dentro de la Dirección de Desarrollo Social.</p>	
Casos de uso asociados:	<p>Creación de usuario.</p> <p>Creación de menú o aplicaciones.</p> <p>Creación de áreas administrativas.</p> <p>Validación de identidad.</p>
Flujo de trabajo:	
Acción del actor:	Respuesta del negocio:
<p>Administrador: es quien precargara el sistema con información útil para dar marcha al software.</p> <p>Base de datos: aquí es donde se valida en la base de datos que no exista duplicidad y</p>	<p>Al crear las áreas administrativas en el software puede dar pie a crear los programas que emanan de estas dependencias administrativas, así sucesivamente asignar polilíneas, polígonos y al final los beneficiarios.</p> <p>Al crear los menús que cada uno de los usuarios que existen en el sistema, puede delimitar las acciones que pueden realizar.</p> <p>Al crear los usuarios y tener listos los menús es hora de poner en marcha el software dentro de la Dirección</p>

redundancia en los datos.	de Desarrollo Social.
Prioridad:	Muy alta
Mejoras:	Mejor manipulación de la información.
Cursos alternos:	--

Nombre del caso de uso del software:	Creación, modificación y/o actualización de la información
Actores del software:	Titular de área y base de datos.
Propósito:	Crear los programas, polilíneas y polígonos, además de que él podrá modificar usuarios y la geolocalización en el sistema.
<p>Resumen:</p> <p>Una vez que se han creado los usuarios, y delimitado los menús, lo siguiente es crear los programas, las polilíneas y polígonos para en ellos asignar los beneficiarios. Una vez que se agrega un beneficiario se asigna también su geolocalización.</p>	
Casos de uso asociados:	<p>Primeramente, se tiene que iniciar sesión en el sistema.</p> <p>Creación de programas, polilíneas y polígonos.</p> <p>Modificación de programas, polilíneas y polígonos.</p>

	Modificar beneficiario y geolocalización.
Flujo de trabajo:	
Acción del actor:	Respuesta del negocio:
El jefe de área es quien tiene toda la responsabilidad de habilitar los programas, polilíneas y polígonos,	<p>El jefe de área tendrá que iniciar sesión para que pueda ser validado como un usuario legitimo para el software.</p> <p>El jefe de área crea los programas y los asigna a un área administrativa previamente dada de alta por el administrador, las polilíneas se cran de manera independiente al igual que los polígonos para posteriormente ser asignados a los usuarios y programas sociales.</p> <p>El jefe de área tiene la opción de poder hacer modificaciones a los registros previamente capturados en el sistema con el fin de que si existiera un error dejar la información corregida.</p>
Prioridad:	Alta
Mejoras:	Existirán programas que surjan de manera muy rápida, de rápida ejecución y de pocos beneficiarios, pero sin importan que tan grande o pequeños puedan ser es indispensable que tengan que realizarse.
Cursos alternos:	En caso de que se tenga que hacer la modificación de una geolocalización y al no tener control sobre el esperar a que el desarrollador de la aplicación resuelva el mismo.

Nombre del caso de uso del software:	Captura de beneficiarios.
Actores del software:	Capturista. Base de datos. Mapas de Google.
Propósito:	Ingresar todos y cada uno de los beneficiarios a los programas que ya ha sido precargados previamente por el usuario jefe de área.
Resumen: Es la parte más laboriosa ya que el o los capturistas tendrán que dar de alta a todos y cada uno de los beneficiarios que hay sido contemplados para ser atendidos por el programa ejecutado en la Dirección de Desarrollo Social.	
Casos de uso asociados:	Iniciar sesión. Agregar beneficiario. Agregar geolocalización. Autenticación.
Flujo de trabajo:	
Acción del actor:	Respuesta del negocio:
Los capturistas o promotores tendrán que dar de alta todos y cada uno de los beneficiarios en el sistema.	Primeramente, tendrá el promotor o capturista iniciar sesión en el sistema para validar su usuario. Tendrá que capturar todos los beneficiarios en el software para nutrir el sistema, asignándolo a los programas que ya el jefe de área ha precargado en el sistema.
Prioridad:	Normal.
Mejoras:	En un futuro poder implementar OCR aplicado a

	detectar en imágenes los textos.
Cursos alternos:	---

Nombre del caso de uso del software:	Consultas y reportes.
Actores del software:	Jefe de área. Base de datos. Mapas de Google.
Propósito:	De los datos que se han recopilado y organizado se tomaran para poder generar consultas y reportes que sirvan para tomar decisiones oportunas.
Resumen: Los datos que se han ingresado al sistema de información tendrán que ser de utilidad y medidos para conocer si la aplicación de esta es correcta respecto a las reglas de operación, en caso de ser contrario, tomar decisiones oportunas para redirigir y cumplir con las metas.	
Casos de uso asociados:	Consultas. Reportes. Beneficiario. Consulta mapa. Validación de la información.
Flujo de trabajo:	
Acción del actor:	Respuesta del negocio:

Nuevamente el jefe de área es quien tiene acceso para poder generar dichos reportes y consultas.	El usuario una vez dentro del sistema tendrá la opción de generar esas consultas dentro del sistema, y los reportes genera el software un archivo en una hoja de cálculo con la información solicitada para poder manipularla sin afectar directamente a la base de datos.
Prioridad:	Alta
Mejoras:	Poder generar graficas que simplifiquen aun mas los datos.
Cursos alternos:	-----

XIII. ANEXO V RELACIÓN DE TABLAS Y FIGURAS

NUMERO DE TABLA	NOMBRE	FUENTE	PAGINA
1	Programas de Gobierno Federal	DOF, 2018	13
2	Programas de Gobierno Estatal.	DOF, 2018	13
3	Padrón Único de Beneficiarios	Programas de Desarrollo Social, 2019	21
4	Evaluación de Solución a Causas	Elaboración propia	23
5	Evaluación de Solución a Consecuencias	Elaboración propia	24
6	Conceptos de riesgo basados para esta solución	Elaboración propia	25
7	Tablas de la base de datos	Elaboración propia	51
8	Diccionario de datos de tabla aplicaciones	Elaboración propia	52
9	Diccionario de datos de tabla direcciones	Elaboración propia	52
10	Diccionario de datos de tabla geo_catalogo	Elaboración propia	53
11	Diccionario de datos de tabla geo_coordenadas	Elaboración propia	53
12	Diccionario de datos de tabla usermenu	Elaboración propia	54
13	Diccionario de datos de tabla usuarios	Elaboración propia	54
14	Beneficios del enfoque iterativo	Elaboración propia	60
15	Fases e iteraciones y disciplinas	Elaboración propia	64
16	Modelos de pruebas	Elaboración propia	64
17	Fase de Construcción	https://es.slideshare.net/mstabar e/ingeniera-de-software-ii-parte-2	69
18	Fase de transición	Elaboración propia	70

NUMERO DE FIGURA	NOMBRE	FUENTE	PAGINA
1	Organigrama de la Administración Municipal Constitucional 2016-2018	Gaceta 62, 2018	15
2	Organigrama interno de la Dirección de Desarrollo Social del Municipio de Chalco	Gaceta 36, 2017	16
3	Metodología de desarrollo de software	http://aigroup.com.co/site/?page_id=18	31
4	Problemas que conlleva una mala especificación	https://serprogramador.es/el-cliente-nunca-sabe-lo-que-quiere/	32
5	Logotipo UML	https://commons.wikimedia.org/	33

		wiki/File:UML_logo.svg	
6	Software StarUML	Elaboración propia	34
7	Creación de usuarios	Elaboración propia	36
8	Creación de menú	Elaboración propia	37
9	Creación de áreas administrativas	Elaboración propia	38
10	Diagrama de casos de uso de captura de creación de usuarios, menú y áreas administrativas	Elaboración propia	38
11	Diagrama de casos de uso de creación, modificación y/o actualización de la información	Elaboración propia	40
12	Creación de programas sociales	Elaboración propia	41
13	Creación de polilínea	Elaboración propia	42
14	Creación de polígono	Elaboración propia	43
15	Agregar beneficiario	Elaboración propia	45
16	Diagrama de casos de uso de captura de beneficiarios	Elaboración propia	46
17	Diagrama de casos de uso de consultas y reportes	Elaboración propia	47
18	Consultas y reportes	Elaboración propia	48
19	Arquitectura basada en diagramas UML	Elaboración propia	49
20	Gestor de base de datos MySQL	Elaboración propia	51
21	Diagrama de clases	Elaboración propia	55
22	Agregar usuario	Elaboración propia	58
23	Inicio de sesión	Elaboración propia	59
24	Control de acceso a máscaras de interfaz	Elaboración propia	59
25	Función permisos	Elaboración propia	60
26	Estructuración del menú	Elaboración propia	61
27	Agregar o eliminar aplicaciones	Elaboración propia	61
28	Contenido de carpeta application	Elaboración propia	62
29	Modelado MVC	Elaboración propia	64
30	Modelado de proceso incremental	https://ingenieriaensoftware.nathalyalava.wordpress.com/2015/04/25/modelos-de-procesos-prescriptivos/	64
31	Ciclo de vida modelo RUP	http://proyectosociotecnologicosailinperez.blogspot.com	68
32	Consultas y reportes	Elaboración propia	71
33	Versión de prueba	Elaboración propia	73

XIV. ANEXO VI FRAGMENTO DE CÓDIGO

AGREGAR USUARIO

```
<div class="row">
<!-- Restriccion de tiempo -->
<?php
$currtime = date("H:i");
if($currtime > "16:00"){
die("Registro habilitado hasta las 4:00pm ");
}
?>
<div class="col-md-12">
<form method="post" action="">
<table class="table table-bordered table-striped">
<tr>
<td>Numero de nomina</td>
<td><input type="text" name="nomina" placeholder="Numero de nomina" class="form-control"></td>
</tr><tr>
<td>Contrase&ntilde;a</td>
<td><input type="password" name="clave" class="form-control" name="password"></td>
</tr><tr>
<td>Nombre</td>
<td><input type="text" name="nombre" placeholder="Nombre" class="form-control"></td>
</tr><tr>
<td>Paterno</td>
<td><input type="text" name="paterno" placeholder="Paterno" class="form-control"></td>
```

```

</tr><tr>
<td>Materno</td>
<td><input type="text" name="materno" placeholder="Materno" class="form-control"></td>
</tr><tr>
<td>Correo electrónico</td>
<td><input type="text" name="correo_electronico" placeholder="Correo electronico"
class="form-control"></td>
</tr><tr>
<td>Dirección</td>
<td>
<select name="direccion" class="validate[required]">
<?php
print_r($direcciones);
foreach($direcciones as $lista){
echo "<option value='". $lista->id_area .">". $lista->nombre_area."</option>";
}
?>
</select>
</td></tr><tr>
<td colspan="2"> <input type="submit" value="Guardar" class="btn btn-primary" /> </td>
</tr></table>
</form>
</div></div>

```

INICIO DE SESIÓN

```

<div class="row">
<div class="col-md-12">

```

```

</div>

<div class="col-md-12">

<form role="form" method="post" action="">

<div class="form-group">

<label for="usuario">Usuario</label>

<input type="text" name="usuario" class="form-control" placeholder="Numero de Usuario">

</div>

<div class="form-group">

<label for="password">Contrase&ntilde;a</label>

<input type="password" name="password" class="form-control">

</div>

<button type="submit" class="btn btn-primary">INICIAR</button>

</form></div></div>

<hr>

<?php if(!empty($err)) { ?>

<div class="row">

<div class="col-md-12 bg-danger">

<p><?php print_r($err); ?></p>

</div></div><?php } ?>

```

CONTROL DE ACCESO A MÁSCARAS DE INTERFAZ

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Acceso extends CI_Controller {

public function __construct()

{

parent::__construct();

```

```

$this->load->model('acceso_model');

date_default_timezone_set("America/Mexico_City");

$this->session->sess_destroy();

}

public function index()

{

    $err = null;

    $suc = null;

    if($this->input->post())

    {

        $this->form_validation->set_rules('nomina','Numero de
nomina','required|xss_clean|numeric|exact_length[4]');

        $this->form_validation->set_rules('clave','Contrasena','required|xss_clean');

        $this->form_validation->set_rules('paterno','Apellido Paterno','required|xss_clean');

        $this->form_validation->set_rules('materno','Apellido Materno','required|xss_clean');

        $this->form_validation->set_rules('correo_electronico','Correo
electronico','valid_email|xss_clean');

        if($this->form_validation->run() == FALSE){

            $err = validation_errors();

            $app = $this->input->post();

        }

        else

        {

            $object = (object)$this->input->post();

            $object->contrasena = md5($object->clave);

            unset($object->clave);

            $count = $this->acceso_model->validar($object->nomina);

```

```

if(count($count) > 0)
{
    $err = "El número de nomina <strong>$object->nomina</strong> ya ha sido registrado
    anteriormente... ";
}
else
{
    $this->acceso_model->agregarUsuario($object);
    $suc = "success";
    }}}
    $direcciones = $this->acceso_model->direcciones();
    $this->layout->setLayout("default");
    $this->layout->view('registro', compact('direcciones','err','suc'));
}
}

```

FUNCIÓN PERMISOS

```

public function permisos($id=null)
{
    $err = null;
    if($this->input->post())
    {
        $this->form_validation->set_rules('application', 'Aplicacion', 'required|xss_clean');
        $this->form_validation->set_rules('module', 'Modulo', 'required|xss_clean');
        if($this->form_validation->run() == FALSE){
            $err = validation_errors();
        }
    }
    else

```

```

{
$params = (object)$this->input->post();

$params->iduser = $id;

if($this->management_model->addPermission($params))

{
    $err = null;

    $success = "App agregada satisfactoriamente.";

}else

{
    $err = "Permiso ya otorgado...";

}}

$menu = $this->menu;

$app = $this->management_model->getApps($id);

$user = $this->management_model->getUserName($id);

$lapp = $this->management_model->getApplications();

$this->layout->setLayout('plataforma');

$this->layout->view('permisos',compact('menu','id','apps','user', 'lapp', 'err','success'));

}

public function modules()

{

if($this->input->post())

{

    $modules = $this->management_model->getModules($this->input->post('appdata', true));

    if(count($modules) > 0){

        echo json_encode(array("message" => true, "modules"=> $modules));

    }}

}

```

ESTRUCTURACIÓN DEL MENÚ

```
class Menu_model extends CI_Model {

function __construct()
{
parent::__construct();
}

public function get($userid)
{
$applications = $this->get_applications($userid);
$menu = array();
$i = 0;
foreach($applications as $items)
{
$content = new stdClass();
$content->application = $items->application;
$content->modules = $this->get_modules($items->application, $userid);
$menu[$i] = $content;
$i++;
}
return $menu;
}

public function get_applications($userid)
{
$this->db->select('application')->group_by('application');
$response = $this->db->get_where('usermenu', array('iduser'=> $userid));
return $response->result();
}
```

```

}

public function get_modules($application, $userid)
{
$this->db->select('module');

$response = $this->db->get_where('usermenu', array('application' => $application, 'iduser' =>
$userid ));

return $response->result();
}}

```

AGREGAR O ELIMINAR APLICACIONES

```

<div class="row">

<div class="col-md-12">

<table class="table table-bordered table-striped">

<tr>

<th>Aplicacion</th>

<th>Modulo</th>

<th>Eliminar</th>

</tr>

<?php

foreach($apps as $items){

echo "<tr id='row-".$items->id."'>";

echo "<td>". ucfirst($items->application) . "</td>";

echo "<td>". ucfirst($items->module) . "</td>";

echo "<td class='text-center'><a href='#' class='deleteapp' data-target='".$items->id."'>";

echo "<i class='glyphicon glyphicon-trash'></i></a></td>";

echo "</tr>";}

?>

```

```

</table>

</div></div>

<script type="text/javascript">

$(document).ready(function(){

$(".deleteapp").click(function(e){

e.preventDefault();

var target = $(this).attr('data-target');

var userid = <?php echo $id; ?>;

$.post('<?php echo base_url(); ?>management/delapp/', {appdat:target, appusr: userid},
function(response){

if(response.response == true)

{

$("#row-"+target).fadeOut('slow');

}},'json'));

$("#applist").bind('change', function(){

$("#moduleslist").html("");

var app = $(this).val();

$.post('<?php echo base_url(); ?>management/modules/',{ appdata: app}, function(data){

if(data.message == true)

{

$(data.modules).each(function(i, v){

$("#moduleslist").append('<option>'+ v.modulo +'</option>');

}}

},'json');

});});

</script>

```