

DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE ADMINISTRATIVO EN LA
FUNDACIÓN PREVENIR

JESUS FRANCISCO SOLIS RENGIFO

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
SAN JUAN DE PASTO
2015

DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE ADMINISTRATIVO EN LA
FUNDACIÓN PREVENIR

JESUS FRANCISCO SOLIS RENGIFO

Trabajo de grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas

HENRY LUIS RODRIGUEZ CARDENAS

UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERIA
PROGRAMA DE INGENIERIA DE SISTEMAS
SAN JUAN DE PASTO
2015

NOTA DE RESPONSABILIDAD

Las ideas y conclusiones aportadas en este Trabajo de Grado son Responsabilidad de los autores.

Artículo 1 del Acuerdo No. 324 de octubre 11 de 1966, emanado del honorable Concejo Directivo de la Universidad de Nariño.

“La Universidad de Nariño no se hace responsable de las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Artículo 13, Acuerdo N. 005 de 2010 emanado del Honorable Consejo Académico.

Nota de Aceptación

Firma del asesor

Firma del Jurado

Firma del Jurado

San Juan de Pasto, diciembre de 2015

RESUMEN

El ritmo de vida de los negocios se ha acelerado durante los últimos años, de tal manera que se buscan alcanzar mejoras incrementales en rendimiento. La única manera de igualar o superar la rapidez del cambio en el mercado actual es lograr avances decisivos. El panorama actual y la eminente globalización que se está viviendo dejan como resultado que las pequeñas, medianas y grandes empresas deban adaptarse a los nuevos cambios como es sistematizar los procesos internos y la información. Un sistema computarizado optimiza los procesos administrativos en costos y tiempos de operación, incrementando así la competitividad. Además permite tener acceso de manera instantánea a la información.

Toda organización se desarrolla en un ambiente lleno de cambios, políticas y normas que deben aplicarse para alcanzar sus objetivos de manera efectiva. A medida que la sociedad progresa la necesidad de realizar procesos de manera sistemática y ordenada se ha convertido en un principio absolutamente indispensable.

Por lo anteriormente mencionado, se desarrolló e implementó un software administrativo en la FUNDACIÓN PREVENIR, de acuerdo con las necesidades que presentaban lo cual logro agilizar los procesos de pagos por parte de la comunidad en lo referente a salud, pensión y riesgos profesionales, además de mejorar la gestión de la información.

ABSTRACT

The rhythm of businesses has increased during the last years with the purpose of achieving improvement in its performance. The current outlook and globalization bring the small, medium and big businesses to be adapted to changes such as systematization of internal processes and the information. A computerized system optimizes administrative processes in operating costs and time, increasing competitiveness. It also allows for instant access to information.

Any organization takes place in a room full of changes , rules to be applied to achieve their goals effectively

Taking into account the previous information, an administrative software was carried out and implemented in FUNDACION PREVENIR according to its needs. This software achieved to speed up management model of information and the process of payments for the services related to health, retirement pension and professional risks.

Contenido

INTRODUCCION	12
1 METODOLOGIAS TRADICIONALES DE DESARROLLO DE SOFTWARE.....	16
2. PLANTEAMIENTO DE LA SOLUCION	29
3. EJECUCION DEL PROYECTO	71
CONCLUSIONES	82
RECOMENDACIONES.....	83
BIBLIOGRAFIA.....	84

ANEXOS

ANEXO 1. MANUAL DE USUARIO	85
MANUAL DE USUARIO.....	85

LISTA DE TABLAS

Tabla 1. Diferencias entre metodología ágil y tradicional.....	18
Tabla 2. Cliente.....	32
Tabla 3. Afp.....	35
Tabla 4. Eps.....	36
Tabla 5. Novedad.....	37
Tabla 6. Entidad.....	38
Tabla 7. Ciudad.....	39
Tabla 8. Departamento.....	40
Tabla 9. Pago.....	41
Tabla 10. TarifaRiesgo.....	42
Tabla 11. Recibo.....	43
Tabla 12. Usuario.....	44
Tabla 13. Rol.....	44
Tabla 14. Detalle_Rol_Permission.....	45
Tabla 15. Permiso.....	46
Tabla 16. Egreso.....	47
Tabla 17. Cuentasxcobrar.....	48
Tabla 18. Banco.....	49
Tabla 19. TarifaPlanilla.....	50
Tabla 20. PlanillaConsignar.....	51
Tabla 21. Detalle_Pago.....	52
Tabla 22. Comparación entre Algoritmo Md5 y Sha1.....	62
Tabla 23. Equipo de trabajo.....	68
Tabla 24. Backlog del Producto.....	70
Tabla 25. Historia de usuario 1.....	73

Tabla 26. Historia de usuario 2.....75

LISTA DE FIGURAS

Figura 1. Modelo RUP	17
Figura 2. Ejemplo de impacto de riesgo en desarrollo cascada.....	20
Figura 3. Modelo de desarrollo Scrum	26
Figura 4. Modelo conceptual.....	32
Figura 5. Modelo relacional.....	34
Figura 6. SQLServer 2008	58
Figura 7. DBDesigner	63
Figura 8. Visual Studio Community	64
Figura 9. Arquitectura de tres niveles	70
Figura 10. Escala de importancia.....	77

INTRODUCCION

Para la elaboración de una herramienta software flexible y escalable, se hace necesario escoger una metodología de desarrollo de software, así como también herramientas que permitan la culminación del trabajo en el tiempo planificado y cumpliendo con todos los requerimientos.

Hoy en día la tecnología, las metodologías y las herramientas avanzan a una velocidad considerable, lo que ha provocado que el desarrollo de proyectos informáticos deba alcanzar la velocidad de los cambios ocasionados por esta aceleración. Es así, como la calidad, eficiencia, flexibilidad y rapidez en la entrega de un producto se ha vuelto prioridad. Pero, ¿Se adaptan bien las metodologías convencionales a las actuales características de dinamismo y variabilidad del mercado? ¿Se necesitan nuevas alternativas que se ajusten de un modo más incuestionable a dichas necesidades? Según Kent Beck, uno de los padres del modelo ágil, “Cada cosa en software cambia. Los requisitos cambian, el diseño cambia, el negocio cambia, la tecnología cambia, el equipo cambia y los miembros del equipo cambian. El problema, realmente, es nuestra incapacidad para hacer frente a los cambios”.

En este contexto las metodologías ágiles aparecen como una opción atractiva pues, en contraposición con las metodologías convencionales que actúan basadas en principios de estabilidad y control del contexto, las metodologías ágiles no se centran en habilidades de predicción, ni pretenden tener un sistema perfectamente definido como paso previo a su construcción, sino que reciben cada respuesta al cambio como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo software y, permitiendo de ese modo, una mejor adaptación en entornos turbulentos.

Las metodologías ágiles se orientan principalmente en proyectos que requieren soluciones a la medida, con una elevada simplificación en términos de tiempo y recursos, sin dejar de lado el aseguramiento de la calidad del producto.

DESCRIPCIÓN DEL PROBLEMA

PLANTEAMIENTO

Las empresas con sistemas manuales generan procesos lentos y poco confiables. Por tal razón, están obligadas a cambiar la manera como realizan sus procesos y de ese modo buscar una herramienta sistemática que fomente un cambio para dirigir las operaciones de la organización, buscando alcanzar eficiencia en los procesos, tal es el caso de la FUNDACIÓN PREVENIR, la cual presenta procesos internos y administrativos que requieren de un mejoramiento que permita lograr de manera eficiente las expectativas organizacionales fijadas por la misma. Siendo los pagos de salud, riesgo y pensión el servicio fundamental de la empresa, se tendría que pensar en mejorar la prestación de este servicio para lograr eficiencia y mayor productividad.

La FUNDACIÓN PREVENIR, no cuenta en la actualidad con una herramienta informática que le brinde ayuda y soporte a sus actividades diarias. La gestión de clientes, empleados, pagos por clientes, gastos y reportes, son realizados de manera manual, lo que trae como consecuencia procesos lentos, pérdida de información y menor rendimiento. Mostrando así la necesidad de adquirir un sistema que permita lograr optimización en los procesos internos y flujo de información.

FORMULACIÓN

¿Cómo mejorar el sistema administrativo en la FUNDACIÓN PREVENIR con el fin de optimizar los tiempos de operación y obtener información de manera fácil, eficiente y confiable?

SISTEMATIZACIÓN

¿Cómo se realizan actualmente los procesos de pagos y facturación en la empresa?

¿Es posible mejorar la situación actual de la empresa?

¿Cómo mejorar los tiempos de operación en la empresa?

¿Cómo validar el funcionamiento del sistema en un entorno real?

OBJETIVOS

OBJETIVO GENERAL

Diseñar e implementar un software administrativo en la FUNDACIÓN PREVENIR.

OBJETIVOS ESPECÍFICOS

- Identificar la situación actual de la FUNDACIÓN PREVENIR
- Apropiar el conocimiento acerca de la metodología de desarrollo ágil Scrum
- Identificar los requerimientos del sistema y construcción de modelo entidad-relación y relacional.
- Diseñar e implementar el software administrativo.
- Aplicación de pruebas y análisis de resultados.

JUSTIFICACIÓN

Los procesos internos y flujo de la información presentes en la FUNDACIÓN PREVENIR son el pilar fundamental de la organización que garantiza la estabilidad, eficacia y buen funcionamiento de la misma. Por ello, se debe brindar especial atención a éstos, dado que a medida que crecen las actividades en la entidad, se dan cambios importantes y se debe considerar todos los avances tecnológicos e informáticos que la mantengan a la par de los sistemas competitivos de globalización.

En la FUNDACIÓN PREVENIR, comprenden que en un mundo donde el cambio no es constante sino que se acelera a medida que el tiempo transcurre, la empresa que deja de innovar la manera en que se realizan las cosas, se quedará atrás.

Por tal motivo, resulta necesario diseñar un sistema que optimice el flujo de información, la gestión de clientes y empleados, pagos realizados por clientes, gastos y reportes, permitiendo así lograr excelentes resultados, mejor rendimiento y mayor eficiencia en los procesos internos y flujo de información.

ALCANCE Y DELIMITACIÓN

El software administrativo se encargará de capturar, procesar y administrar la información correspondiente a los pagos realizados por los clientes, generando además facturas y reportes que garanticen un eficiente funcionamiento de los procesos internos, así como del flujo de información. Los módulos a desarrollar serán los siguientes:

Módulo Clientes: Gestionará la información general de clientes, permitiendo registrar, consultar, modificar, y eliminar clientes del sistema.

Módulo Usuarios: Gestionará la información general de los usuarios que interactuarán con el sistema. Además, se asignará nombre de usuario, contraseña y el tipo de rol.

Módulo de Pagos: Registrará los pagos realizados por cliente.

Módulo de Seguridad del sistema: permitirá el manejo de usuarios y roles para garantizar el acceso, seguridad y veracidad de toda la información.

Módulo de Gastos: registrará los gastos o egresos ocasionados por servicios de internet, telefonía, energía u otros.

1 METODOLOGIAS TRADICIONALES DE DESARROLLO DE SOFTWARE

En un principio el desarrollo de software era realizado de una manera totalmente artesanal. El crecimiento espectacular de la demanda de sistemas cada vez más complejos asociado a la inmadurez del propio sector informático, provocó lo que se llamó la *crisis del software (1965-1985)*. Durante ese periodo, muchos proyectos informáticos superaban los presupuestos y fechas estimadas para su ejecución, lo que generó pérdidas que iban más allá de lo económico.

Fue así como los proyectos informáticos tomaron la importancia que merecían, dando lugar a mejorar el proceso para llevar a los proyectos informáticos a la meta deseada. Para resolver esta situación se hizo necesario adoptar al desarrollo de software metodologías existentes de otras áreas. Esto llevo a dividir el proceso de desarrollo de software en etapas secuenciales, lo que ayudó en formalizar los procesos de desarrollo. Obteniendo además mejoras en la gestión de proyectos en los procesos de diseño y metodologías a implementar.

Una solución definitiva al problema de planificación y aseguramiento de la calidad en el proceso de desarrollo de software, indudablemente está ligada a la correcta implementación y uso de herramientas, metodologías y tecnologías. Entre las principales metodologías de este tipo se puede mencionar RUP, que se centra en llevar una documentación exhaustiva de todo el proyecto y el cumplimiento estricto y rígido de un plan de proyecto, definido esto en la fase inicial del proyecto.

METODOLOGIA TRADICIONAL

RUP (rational unified process)

Esta metodología brinda una manera disciplinada para asignar tareas y responsabilidades dentro de una organización. Tiene como principal objetivo

asegurar la producción de software de calidad, que logre satisfacer los requerimientos de los usuarios. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso, centrado en la arquitectura y utiliza UML como lenguaje de notación. (Ver figura 1)

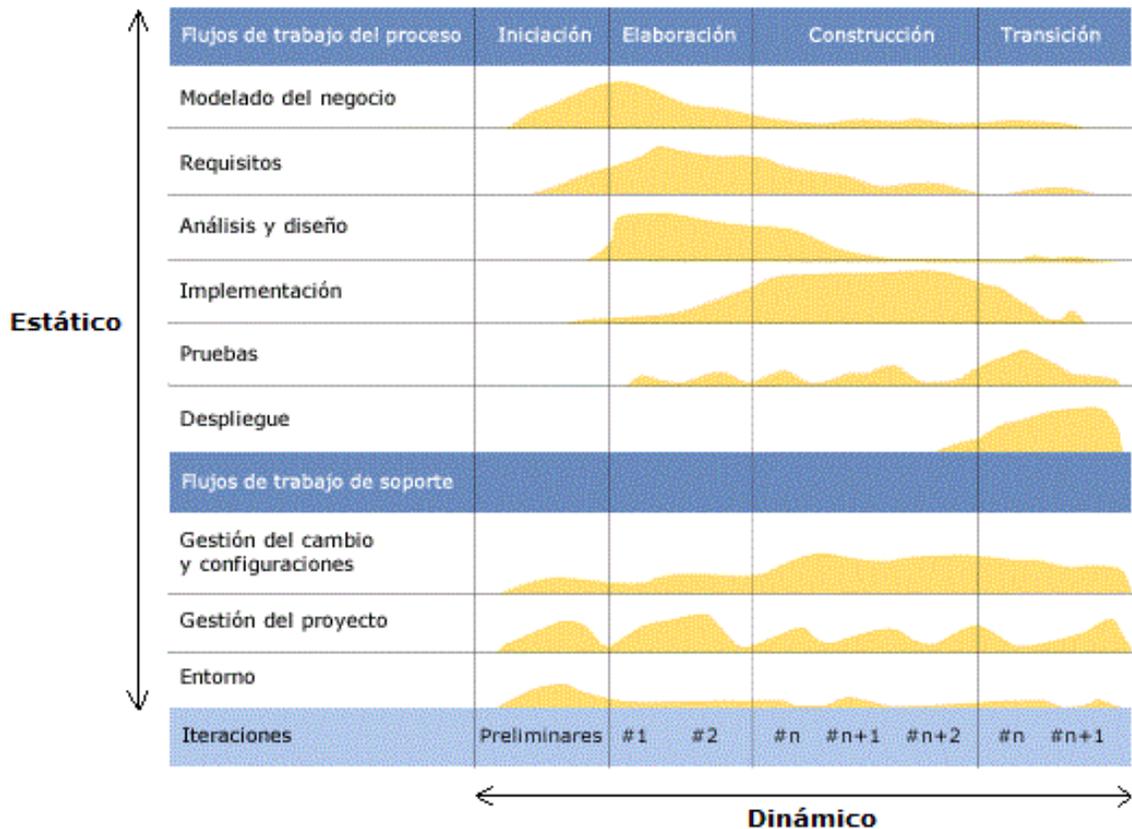


Figura 1. Modelo RUP

METODOLOGÍA ÁGIL

Las metodologías ágiles son sin duda uno de los temas recientes en ingeniería de desarrollo que están acaparando gran interés y controversia. A mediados de los años 90's comenzó a forjarse una definición moderna de desarrollo ágil de software como una reacción contra las metodologías tradicionales utilizadas hasta

el momento, consideradas excesivamente pesadas y rígidas por su carácter normativo y fuerte dependencia planificaciones detalladas previas al desarrollo.

En febrero de 2001, diecisiete miembros destacados de la comunidad de software, incluyendo algunos de los creadores o impulsores de las metodologías en software, se reunieron en Utah, EEUU y adoptaron el nombre de “Metodologías ágiles” para denominar a esta nueva corriente de desarrollo software. Su objetivo fue proyectar algunos principios y valores que deberían permitir a los equipos de desarrollar software de manera rápida responder a los cambios que pudieran surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó *The Agile Alliance*, organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el manifiesto ágil, documento que resume la filosofía ágil.

¿PORQUE LAS METODOLOGÍAS ÁGILES?

Tradicionalmente se ha desarrollado software a través de metodologías tradicionales que cada vez más, se muestran erróneas en las actuales características de dinamismo y variabilidad del mercado software. En la actualidad se busca el desarrollo rápido de aplicaciones incrementando la productividad y satisfaciendo las variantes necesidades del cliente en el menor tiempo posible.

Las metodologías tradicionales presentan diversos problemas a la hora de abordar un amplio rango de proyectos en un ambiente variante. Entre los problemas se puede destacar los siguientes:

- Perciben la captura de requisitos del trabajo como una fase previa al desarrollo del mismo, que una vez completada, debe proporcionar una

fotografía exacta de qué desea el cliente. Se trata de evitar que se produzcan cambios en el conjunto de requisitos iniciales, puesto que a medida que avanza el proyecto resulta más costoso solucionar los errores detectados o introducir modificaciones, y se delega toda la responsabilidad económica al cliente en caso de que estos cambios de requisitos produzcan. Sin embargo, el esfuerzo, tanto en costos como en tiempo, que supone hacer una captura detallada de todos los requisitos al comienzo de un proyecto es enorme. Además, en muchas ocasiones el cliente no conoce sus propias necesidades con la profundidad suficiente como para definir las de manera exacta en el momento de establecer los requisitos iniciales, y a menudo, estas necesidades y sus prioridades cambian durante la vida del proyecto. Establecer mecanismos de control es una de las opciones existentes para protegerse de estos cambios, aunque frecuentemente provocan insatisfacción por parte de los clientes, que perciben el cambio como algo inflexible que no se adapta a sus necesidades y que si lo hacen afecta negativamente en costos añadidos al presupuesto del proyecto.

- El proceso de desarrollo convencional está oprimido por excesiva documentación no siempre útil. Un porcentaje elevado del tiempo de desarrollo de un producto software se malgasta (desde el punto de vista de las metodologías ágiles) en crear documentación que finalmente no se utiliza.
- Lentitud del proceso de desarrollo. Se hace difícil para los desarrolladores entender un sistema complejo en su totalidad, lo que provoca que las diferentes etapas del ciclo de vida convencional transcurra lentamente. Dividir el trabajo en módulos ayuda a minimizar los fallos y por lo tanto el costo de desarrollo. Además, permite el incremento de la funcionalidad.
- Las metodologías convencionales tienden a acumular riesgos y dificultades que surgen en el desarrollo del producto final ocasionando retrasos en la entrega o influyendo en la incorrecta ejecución de las fases siguientes del ciclo de desarrollo, como puede apreciarse en la tabla. (Ver Figura 2)

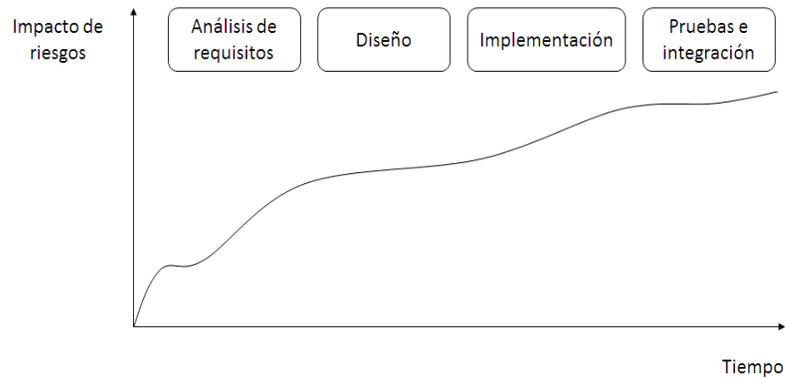


Figura 2. Ejemplo de impacto de riesgo en desarrollo cascada

En la figura 2, aparecen las metodologías ágiles como una alternativa para el desarrollo de software cuando los requisitos son variantes y cambian rápidamente. De este modo, presenta diversas ventajas que se mencionan a continuación:

- Capacidad de respuesta a cambios a lo largo del desarrollo ya que no se perciben como un obstáculo sino como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto característico del propio proceso de desarrollo de software.
- Entrega continua y en plazos cortos del producto software funcional, lo que permite al cliente verificar el desarrollo del proyecto e ir disfrutando de la funcionalidad del producto progresivamente y comprobando si satisface sus necesidades, mejorando de esta manera su satisfacción.
- Comunicación directa entre el cliente y el equipo de desarrollo disminuyendo los malentendidos, que constituyen una de las principales fuentes de errores en los productos software, y el exceso de documentación improductiva.

La siguiente tabla (Tabla N. 1), refleja algunas diferencias de las metodologías ágiles respecto a las tradicionales (no ágiles).

Tabla 1. Diferencias entre metodología ágil y tradicional

Metodologías ágiles	Metodologías tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo)	Impuestas externamente.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

MANIFIESTO AGIL

El desarrollo ágil es una filosofía de desarrollo software. El punto de partida se establece en las ideas procedentes del Manifiesto Ágil.

Según expresa el manifiesto, se valora:

- a) Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. Las personas son el componente principal para que un

proyecto de software tenga éxito. Es más importante construir un buen equipo que construir un buen entorno de desarrollo. En muchas ocasiones se comete el error de construir primero el entorno y esperar a que el equipo se adapte, cuando es mejor crear el equipo y que éste cree su propio entorno de desarrollo en base a sus necesidades.

- b) Desarrollar software que funciona, más que conseguir una buena documentación. La regla a seguir es “No producir documentos a menos que sean necesarios de forma inmediata para tomar decisiones importantes”. Estos documentos deben ser cortos y centrarse en lo importante.
- c) La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- d) Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan aparecer a lo largo del ciclo de desarrollo también determinan el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta, sino flexible.

Los ítems mencionados inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto a metas a seguir y la organización del mismo. Los principios, son:

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III. Entregar frecuentemente software que funciones desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre las entregas.
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.

- VII. El software que funciona es la medida principal de progreso.
- VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. La simplicidad es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

CÓMO SER ÁGIL. BUENAS PRÁCTICAS

A pesar que la metodología ágil está sustentada por principios y valores para el desarrollo de productos software, la mayoría de estas metodologías tienen asociadas un conjunto de buenas prácticas que buscan la agilidad en el desarrollo. Entre las cuales se mencionan las siguientes:

PLANIFICACIÓN

Mientras las metodologías tradicionales centran la ingeniería de requisitos en habilidades de predicción basándose en planificaciones previas al desarrollo, las metodologías ágiles perciben la gestión de cambios como un aspecto esencial al desarrollo de software, considerando planificaciones continuas, flexibles y en cortos plazos, que respondan mejor a los cambios en las necesidades del cliente.

Involucrar al cliente a lo largo de todo el proceso de desarrollo, de manera que exista una comunicación directa y se realice frecuente retroalimentación son las prácticas más importantes para la planificación y especificación de requisitos en estas metodologías.

PROGRAMACIÓN EN PAREJAS (PAIR PROGRAMMING)

Una de las prácticas más utilizadas en metodologías ágiles y en sistemas de alta complejidad, es la programación en parejas, que dice que el desarrollo de software se realice en pareja de programadores. El utilizar Pair Programming en un proyecto no quiere decir que todas las líneas de código serán escritas en pareja, ya que mientras un programador está escribiendo el código, el otro puede estar verificando errores, proponiendo mejores alternativas, identificando casos de prueba y realizando mejoras en el diseño. El objetivo principal de esta técnica es disminuir la tasa de errores. No obstante, se trata de una técnica intensa y estresante para los propios programadores (ejemplo: diferencias en el estilo de codificación), aunque al final se concluye que acelera el proceso de desarrollo.

INTEGRACIÓN CONTINUA

El objetivo es integrar cada cambio introducido, de tal manera que pequeñas piezas de código sean integradas de forma continua, ya que se parte de la base de que cuando más se espere para integrar más costoso e impredecible se vuelve el proceso. Para que esta práctica sea viable es importante disponer de casos de prueba, preferiblemente automatizados, de tal manera, que una vez el nuevo código este integrado con el resto del sistema se pueda verificar si este cumple o no con la funcionalidad esperada.

REFACTORIZACIÓN

Esta actividad es constante en los desarrollos ágiles cuyo objetivo es mejorar el diseño de un sistema sin influir en su funcionalidad. Se pretende reestructurar el código para eliminar duplicaciones, hacerlo más legible, simplificarlo y volverlo más flexible, de manera que se faciliten posteriormente los cambios. Se recuerda que el código que funciona es el principal aporte de valor para las metodologías ágiles, por encima de la documentación.

DESARROLLO DIRIGIDO POR PRUEBAS

En esta práctica la producción de código está dirigida por las pruebas. Se trata de escribir las pruebas que validarán el sistema antes de comenzar a construirlo. Esta práctica garantiza la calidad del producto y su correcta funcionalidad. Entre las ventajas que ofrece el desarrollo dirigido por pruebas se puede destacar que disminuye el tiempo dedicado a solucionar errores y genera un sentimiento de confianza en el éxito del proyecto.

METODOLOGÍA SCRUM

Scrum es un término que define una manera para desarrollar productos software iniciada en Japón. Este modelo fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi en 1987 al observar una estrategia utilizada en rugby en la que todos los integrantes del equipo actuaban juntos para avanzar la pelota y ganar el partido. Escogieron este nombre por las similitudes encontradas entre el juego del rugby y el tipo de proceso que proponían (adaptable, ágil, rápido, auto-organizable).

Scrum es un modelo para la gestión y control del producto software, que busca eliminar la complejidad para centrarse en la elaboración del producto para satisfacer las necesidades del negocio. Es simple y escalable, ya que no establece prácticas de ingeniería de software sino que aplica o combina, otras prácticas de ingeniería, metodologías de desarrollo o estándares ya existentes.

Scrum tiene como objetivo lograr que los miembros del equipo de desarrollo trabajen juntos y de manera eficiente logren obtener productos complejos y sofisticados. Pretende que no haya problemas ocultos u obstáculos que puedan poner en riesgo el desarrollo del proyecto. La siguiente figura muestra el ciclo de vida del desarrollo Scrum para un producto software. (Ver Figura 3)

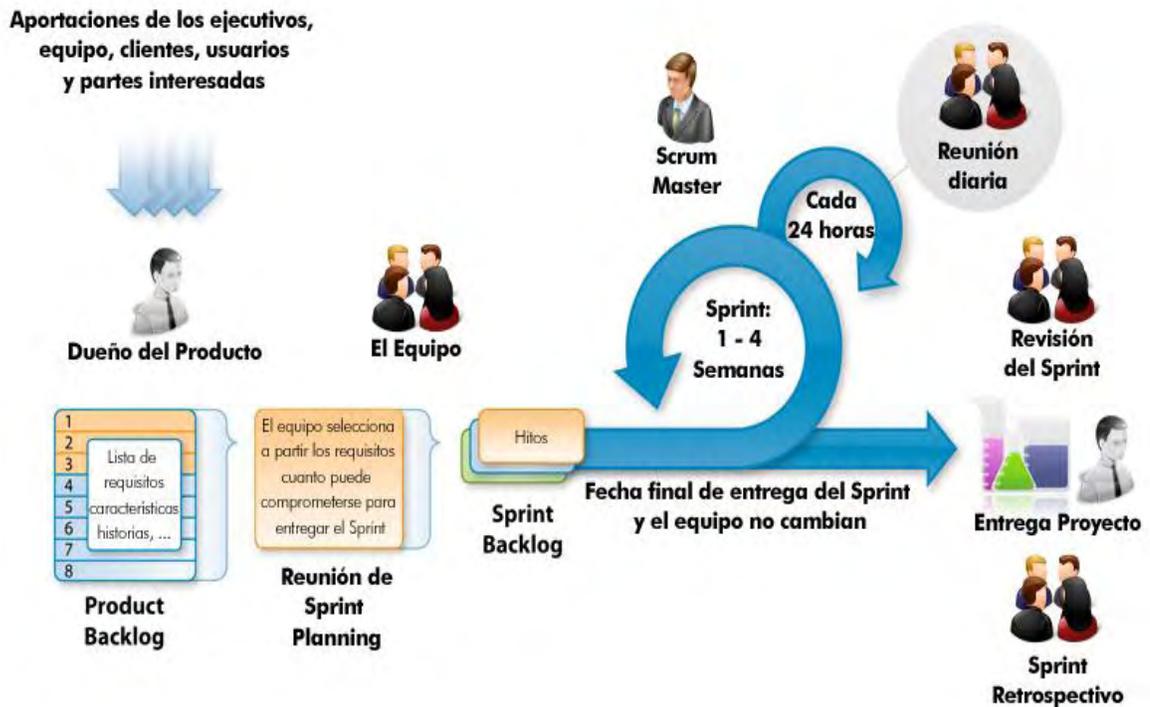


Figura 3. Modelo de desarrollo Scrum

Independiente de la metodología, el desarrollo de un producto software parte siempre del mismo problema, las necesidades del cliente. La metodología Scrum, así como las otras metodologías ágiles, buscan no centrar las etapas de desarrollo en los requisitos definidos formalmente en la etapa inicial sino que defiende la incorporación del cliente como parte del equipo de desarrollo. De esta manera, la etapa de definición de requisitos no se considera que pasará una sola vez durante el desarrollo del proyecto, sino que los requisitos aparecerán implícitamente dentro de las denominadas historias de usuario.

HISTORIAS DE USUARIO

Las historias de usuarios son la base de Scrum para describir las características que el usuario desea obtener del producto que se va a desarrollar. Se puede

incorporar puntos relacionados con las funciones, restricciones o rendimiento del sistema.

El proceso se inicia con la definición de los requisitos iniciales, en la que de manera conjunta con el cliente se realiza una definición clara y sencilla de las características que debe tener el sistema a desarrollar, definiendo las historias de usuario que van a guiar el proceso de desarrollo. Posteriormente se identifican las tareas necesarias para llevar a cabo el desarrollo de las historias de usuario. El resultado de este proceso se denomina en Scrum, Product Backlog.

PRODUCT BACKLOG

El Product Backlog contiene una lista de todas las historias de usuarios, ordenadas por prioridad, así como las tareas que se deben llevar a cabo para completar el desarrollo. Una vez se tiene identificada y priorizadas las historias de usuario, se separan en etapas de corta duración (máximo 30 días) denominadas Sprints.

SPRINT

Para cada sprint se realiza una reunión de planificación de las actividades a desarrollar en el Sprint y se establece la fecha de finalización. El objetivo es mover las historias de usuario con mayor prioridad al llamado Sprint Backlog, que contiene el conjunto de tareas a desarrollar para ese Sprint, incluyendo análisis, diseño, desarrollo y pruebas.

Una vez terminado el Sprint se deberá obtener parte del producto, es decir un entregable que muestre los avances realizados en ese Sprint. En ese instante se procede a una fase de revisión del proyecto para determinar si se incorpora al producto. Después de terminar cada Sprint se realiza una reunión del equipo de desarrollo que permite aprender de las experiencias aprendidas durante el desarrollo de dicho Sprint y mejorar el proceso de desarrollo.

Para dirigir el proceso Scrum define distintos actores con diferentes roles dentro del proceso. De manera general se tiene: Product Owner (Propietario del producto), Scrum Master (Maestro de Scrum), Scrum Team (Equipo de desarrollo) y el cliente.

PRODUCT OWNER

Es la persona encargada de la dirección del Product Backlog, es decir, de las historias de usuarios que tienen los requerimientos que debe cumplir el sistema. Es una persona física y no, una organización o comité. Por lo tanto es oficialmente responsable de tomar las decisiones que asegurarán el desarrollo exitoso del producto. Esta persona asistirá a las reuniones de planificación y de revisión de cada Sprint para asegurar su desarrollo.

SCRUM MASTER

Es la persona encargada de verificar que la metodología Scrum se esté aplicando en el desarrollo del producto. La labor del Scrum master requiere de una fuerte personalidad y ser capaz de hacer cualquier cosa por ayudar al equipo de desarrollo. Debe ser una persona poco conformista y con buena iniciativa.

SCRUM TEAM

Son aquellas personas responsables de implementar los requerimientos del sistema. El equipo tiene autoridad para tomar decisiones que consideren adecuadas para el cumplimiento eficiente del proyecto.

CLIENTE

El o los clientes son aquellos beneficiarios del producto final. Su participación es imprescindible en esta metodología ya que pueden aportar ideas y sugerencias.

2. PLANTEAMIENTO DE LA SOLUCION

Con base en la descripción del problema y en los objetivos que se desean cubrir, se pueden presentar diferentes soluciones para mejorar el funcionamiento de la empresa. La implementación de una base de datos hace parte de la solución propuesta por las siguientes razones:

- Proporciona una solución factible, ya que permite el acceso rápido a la información.
- Permite realizar consultas de clientes, pagos realizados, ingresos diarios o mensuales, egresos, entre otros, de manera eficiente y organizada.
- Permite la expansión del modelo de negocio que presenta actualmente la empresa.

- El software requerido para la implementación de la base de datos se puede adquirir fácilmente, ya que existen muchas distribuciones libres que permiten diversos tipos de instalación, dependiendo de los requerimientos del usuario.
- Se puede realizar respaldos periódicos de la información que se va acumulando en el sistema.
- Se elimina la redundancia de datos, ya que de esta manera es menos probable que lleguen a existir registros duplicados.
- Si se realiza una registro, modificación, actualización o eliminación de datos, esta es visible para todos los usuarios del sistema, ya que los datos de comparten.

Finalmente, la implementación de una base de datos permitirá una adecuada administración de la información que se maneja actualmente en hojas de Excel y en papel.

MODELO DE DATOS

Los modelos de datos son una representación abstracta de los datos en una organización y las relaciones entre ellos. El modelo de datos tiene como propósito representar los datos y ser comprensibles para el usuario.

Los modelos de datos son usados para escribir las bases de datos que describen la realidad que está reflejada en las historias de usuarios obtenidas a partir de los requisitos iniciales.

DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos es un factor muy importante ya que se debe asegurar de proporcionar integridad, coherencia y exactitud en los datos. Si el diseño de una base de datos es incorrecto, será difícil obtener determinados datos y además se corre el riesgo de producir resultados erróneos que serán presentados al usuario y/o cliente.

La base de datos se construye en base a los requerimientos obtenidos a partir de las historias de usuarios con el fin de obtener la visión que tienen los usuarios acerca de la manipulación de los datos a través del software administrativo.

MODELO CONCEPTUAL, LÓGICO Y FÍSICO

MODELO CONCEPTUAL

Para construir el diseño de una base de datos lo primero es la elaboración del modelo conceptual, el cual consiste en abstraer la información del mundo real que se tiene y los procesos que se desean automatizar para disminuir el trabajo de las personas facilitando el manejo de los datos.

En este paso es importante tener una clara visión de los requisitos que se desean cubrir con el diseño conceptual de la base de datos, dado que indica el tipo de información que se va a procesar y expone las necesidades de los futuros usuarios, ya que son ellos quienes tendrán interacción con la base de datos a través del software administrativo.

El modelo a utilizar será el modelo entidad-relación. A continuación, se presenta el modelo propuesto: (Ver Figura 4).

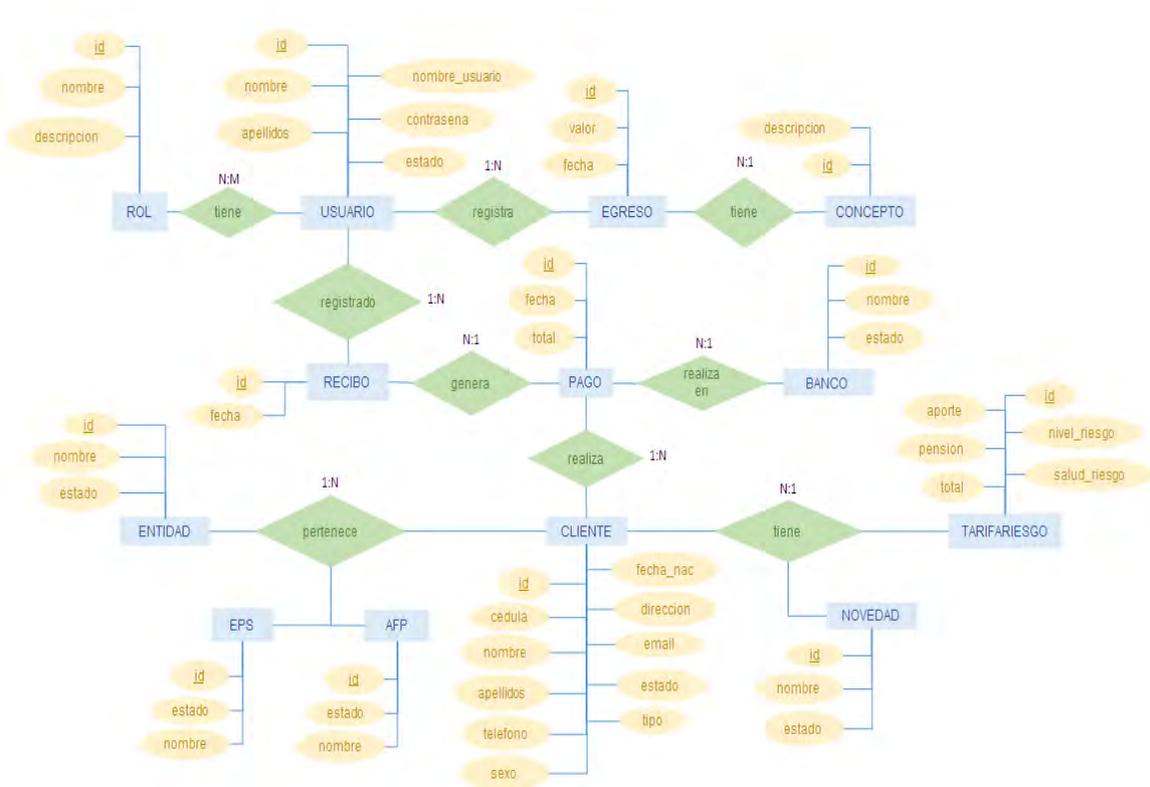


Figura 4. Modelo conceptual

MODELO LÓGICO

Una vez establecido el modelo conceptual del problema o situación, el modelo lógico permite que estos se puedan representar usando de manera eficiente recursos para estructurar los datos y modelar posibles restricciones.

El objetivo es convertir el modelo conceptual de datos en un modelo lógico que se ajuste al gestor de la base de datos que va a ser utilizado (DBMS). Para la solución propuesta se tomará el Modelo Relacional cuyo esquema relacional es

trabajado por muchos DBMS comerciales. Algunos de ellos son: Oracle, Informix, SQLServer y DB2.

VENTAJAS DEL MODELO RELACIONAL

El modelo relacional presenta muchas ventajas en cuanto a almacenamiento, rendimiento y manipulación de datos, mencionados a continuación:

- Evita la duplicidad de datos, mediante la normalización, tanto a nivel de diseño como a nivel de tablas.
- Establece relaciones entre diferentes tablas mediante llaves foráneas.
- Existe independencia de datos, tanto a nivel físico ya que su ubicación puede encontrarse en diferentes computadoras, como lógico dado que los datos se encuentran agrupados en tablas.
- Permite una recuperación rápida y fácil de los datos almacenados mediante el uso de las llaves primarias y foráneas.
- Brindar esquemas sencillos de implementar.
- Los datos pueden visualizarse mediante diferentes consultas.

MODELO PROPUESTO

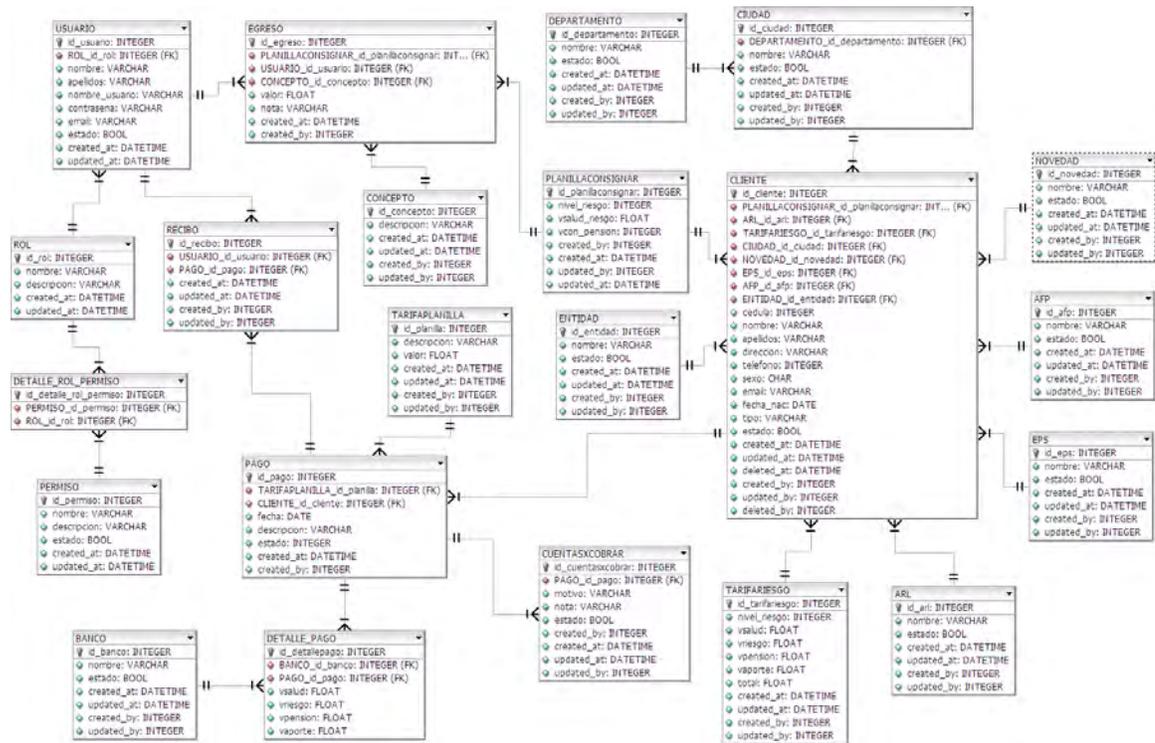


Figura 5. Modelo relacional

ENTIDADES

CLIENTE. Representa todos los clientes que se encuentran activos y no activos.

AFP. Representa el fondo de pensión al que se encuentran registrados el o los clientes.

ARL. Representa el nivel y valor de los riesgos profesionales a los que está expuesto un cliente en el desarrollo de sus actividades diarias.

NOVEDAD. Representa las novedades que se presentan cuando se registra o actualiza un cliente. Puede ser: ingreso, reintegro, retiro. Aunque se consideró que puede aparecer en un futuro una nueva novedad.

CIUDAD. Representa la ciudad de origen de los clientes registrados.

DEPARTAMENTO. Representa el conjunto de departamentos a los que pertenecen las ciudades registradas.

EPS. Representa la empresa de salud a la que se encuentra vinculado el cliente.

ENTIDAD. Representa la entidad en la cual se realiza el pago a los trabajadores.

PAGO. Representa los pagos registrado por clientes. Aquí se ha establecido que los pagos se realizan en efectivo y sin abonos.

DETALLE_PAGO. Representa los valores de los pagos de los clientes. Además aquí se almacena el banco en el que se realizará el pago.

TARIFARIESGO. Representa las tarifas que manejan actualmente en la empresa. Estas varían en función del tiempo y/o determinación del usuario administrador.

RECIBO. Representa los recibos de pagos que se entrega a los clientes. Estos son generados automáticamente cada que se registra un nuevo pago.

USUARIO. Representa las personas que interactuaran con el sistema. Para cada usuario se asignará un nombre de usuario y una contraseña, datos necesarios para acceder al sistema.

ROL. Representa los roles que se manejan dentro del sistema. Cada rol tendrá asignado permisos diferentes.

PERMISO. Representa el nivel de acceso dentro del sistema que se les brindará a los usuarios.

DETALLE_ROL_PERMISO. Representa la entidad intermedia creada para romper la relación muchos a muchos generada entre las entidades rol y permiso.

EGRESO. Representa los egresos o gastos que se generan dentro de las actividades diarios de la empresa.

CONCEPTO. Representa la descripción de los egresos registrados.

CUENTASXCOBRAR. Representa aquellos pagos que se deben cobrar a los clientes.

BANCO. Representa la entidad bancaria en la cual se realiza el pago.

TARIFAPLANILLA. Representa las tarifas que deben pagar aquellos clientes que pertenecen a la categoría de INDEPENDIENTE.

PLANILLACONSIGNAR. Representa los valores que se consignan en el banco por concepto de las afiliaciones de los clientes.

MODELO FÍSICO

El modelo físico es la implementación física de la base de datos empleando el DBMS. Una vez terminado el diseño de tablas, establecido las relaciones y los niveles apropiados de integridad de los datos, entonces se puede decir que la base de datos está completa.

Cuando ya se ha finalizado el desarrollo de la base de datos, entonces se procede a implementar la aplicación que permitirá fácilmente a los usuarios interactuar con los datos almacenados, asegurando que la aplicación proporcionará información oportuna y precisa.

DICCIONARIO DE DATOS

El objetivo de crear un diccionario de datos de las tablas empleadas en la base de datos es brindar una visión detallada de cada uno de los tipos de datos que se manejarán y conformarán la información que se va almacenar. A continuación se presenta una descripción detallada de cada una de las tablas y los campos que la conforman: (Ver Tabla 2)

Tabla 2. Cliente

Campo	Tipo de dato	Tipo de	Descripción
-------	--------------	---------	-------------

		llave	
id_cliente	Int	PK	Llave primaria que identifica de manera única cada cliente registrado
ciudad_id_ciudad	Int	FK	Llave foránea que hace referencia a la tabla CIUDAD, identificando de manera única la ciudad a la que pertenece el cliente
novedad_id_novedad	Int	FK	Llave foránea que referencia a la tabla NOVEDAD, identificando de manera única la novedad o estado de vinculación del cliente
eps_id_eps	Int	FK	Llave foránea que referencia la tabla EPS, identificando de manera única la Eps a la que se encuentra afiliado el cliente
afp_id_afp	Int	FK	Llave foránea que referencia a la tabla AFP, que identifica de manera única el fondo de pensión al que se encuentra vinculado el cliente
entidad_id_entidad		FK	Llave foránea que referencia a la tabla ENTIDAD, que identifica de manera única la entidad en la cual se encuentra vinculado el cliente
tarifariesgo_id_tarifariesgo	Int	FK	Llave foránea que referencia a la tabla TARIFARIESGO, que identifica de manera única la el nivel de riesgo que tiene

			asociado el cliente
Cedula	Int		Número de identificación del cliente
Nombre	Varchar(100)		Nombre(s) del cliente
apellidos	Varchar(100)		Apellidos del cliente
direccion	Varchar(100)		Dirección del cliente
telefono	Varchar(50)		Número telefónico del cliente
sexo	Char(1)		Género al que pertenece el cliente. Este atributo tiene un dominio definido: M (masculino), F(femenino)
email	Varchar(50)		Correo electrónico del cliente (si tiene). Necesario para enviar recordatorio de sus pagos
fecha_nac	Date		Fecha de nacimiento del cliente. Importante para determinar la edad del cliente y enviar mensajes de felicitaciones en sus cumpleaños.
tipo	Varchar(100)		Representa el tipo de cliente que ingresa en el sistema. Este atributo tiene un dominio definido(DEPENDIENTE, INDEPENDIENTE)
estado	Boolean		Indica si el cliente se encuentra activo o inactivo
Created_at	DateTime		Representa la fecha y hora en la cual se registró al cliente en

			el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información del cliente
Deleted_at	DateTime		Representa la fecha y hora en la cual se realizó la eliminación de un cliente
Created_by	Int		Representa el id del usuario que realizó el registro del nuevo cliente
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro de un cliente
Deleted_by	Int		Representa el id del usuario que realizo el proceso de eliminación de un registro

Tabla 3. Afp

Campo	Tipo de dato	Tipo de llave	Descripción
id_afp	Int	PK	Llave primaria que identifica de manera única los fondos de pensión registrados
nombre	Varchar(100)		nombre del fondo de pensión
estado	Boolean		Indica el estado del fondo de pensión, es decir si se encuentra activo o inactivo

Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 4. Eps

Campo	Tipo de dato	Tipo de llave	Descripción
id_eps	Int	PK	Llave primaria que identifica de manera única las Pesos registradas
nombre	Varchar(100)		nombre de la Eps
estado	Boolean		Indica el estado de la Eps, es decir si se encuentra activo o inactivo
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema

Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 5. Novedad

Campo	Tipo de dato	Tipo de llave	Descripción
id_novedad	Int	PK	Llave primaria que identifica de manera única las novedades registradas
nombre	Varchar(100)		Nombre o descripción de la novedad
estado	Boolean		Indica el estado de la novedad
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última

			modificación al registro
--	--	--	--------------------------

Tabla 6. Entidad

Campo	Tipo de dato	Tipo de llave	Descripción
id_entidad	Int	PK	Llave primaria que identifica de manera única las entidades registradas
nombre	Varchar(100)		Nombre o descripción de la entidad
estado	Boolean		Indica el estado de la entidad
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 7. Ciudad

--	--	--	--

Campo	Tipo de dato	Tipo de llave	Descripción
id_ciudad	Int	PK	Llave primaria que identifica de manera única las ciudades registradas
departamento_id_departamento	Int	FK	Llave foránea que referencia a la tabla DEPARTAMENTO, que identifica de manera única el departamento al que pertenecen las ciudades registradas
nombre	Varchar(100)		Nombre de la ciudad
estado	Boolean		Indica el estado de la ciudad
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 8. Departamento

Campo	Tipo de dato	Tipo de llave	Descripción
id_departamento	Int	PK	Llave primaria que identifica de manera única los departamentos registrados
nombre	Varchar(100)		Nombre del departamento
estado	Boolean		Indica el estado del departamento
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 9. Pago

Campo	Tipo de dato	Tipo de llave	Descripción
id_pago	Int	PK	Llave primaria que identifica de manera única los pagos

			registrados por los clientes
cliente_id_cliente	Int	FK	Llave foránea que referencia la tabla CLIENTE, que identifica de manera única el cliente que realizo este pago
banco_id_banco	Int	FK	Llave foránea que referencia la tabla BSNCO, que representa la entidad bancaria en la cual se realiza el pago de los clientes
total	Float		Representa el valor total que debe pagar el cliente
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registró en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 10. TarifaRiesgo

Campo	Tipo de dato	Tipo de	Descripción
-------	--------------	---------	-------------

		llave	
id_tarifariesgo	Int	PK	Llave primaria que identifica de manera única las tarifas de riesgos existentes
nivel_riesgo	Int		Representa en valor numérico los niveles de riesgo existentes
vsalud_riesgo	Float		Representa el valor que debe pagar el cliente si se encuentra afiliado a una Eps
vpension	Float		Representa el valor que debe pagar el cliente si se encuentra cotizando pensión
vaporte	Float		Representa el valor del aporte que realizan todos los clientes por concepto de los pagos que realiza la entidad
total	Float		Representa el valor total que debe pagar el cliente de acuerdo al nivel de riesgo que tenga asociado
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registró en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 11. Recibo

Campo	Tipo de dato	Tipo de llave	Descripción
id_recibo	Int	PK	Llave primaria que identifica de manera única los recibos emitidos por cada uno de los pagos registrados por el cliente
usuario_id_usuario	Int	FK	Llave foránea que referencia la tabla USUARIO, que identifica de manera única el usuario que hizo el registro y emitió la factura
pago_id_pago	Int	FK	Llave foránea que referencia la tabla PAGO, que identifica de manera única el pago realizado por el cliente
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 12. Usuario

Campo	Tipo de dato	Tipo de llave	Descripción
id_usuario	Int	PK	Llave primaria que identifica de manera única los usuarios registrados
rol_id_rol	Int	FK	Llave foránea que referencia la tabla ROL, que identifica de manera única el rol que tiene asociado usuario dentro del sistema
cedula	Int		Número de identificación del usuario
nombre	Varchar(100)		Nombre(s) del usuario
apellidos	Varchar(100)		Apellidos del usuario
nombre_usuario	Varchar(50)		Representa el nombre de usuario que tendrá asociado el usuario para acceder al sistema
contrasena	Varchar(50)		Representa la contraseña o password que tendrá asociado el usuario para acceder al sistema. Esta junto con el nombre de usuario representan las credenciales del usuario para ingresar al sistema
estado	Boolean		Indica si el usuario se encuentra activo o inactiva

Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información

Tabla 13. Rol

Campo	Tipo de dato	Tipo de llave	Descripción
id_rol	Int	PK	Llave primaria que identifica de manera única los roles registrados
nombre	Varchar(100)		Nombre del rol
descripcion	Varchar(200)		Descripción o características del rol registrado
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información

Tabla 14. Detalle_Rol_Permission

Campo	Tipo de dato	Tipo de llave	Descripción
id_detalle_rol_permiso	Int	PK	Llave primaria que identifica de manera única los permisos para

			cada rol
permiso_id_permiso	Int	FK	Llave foránea que referencia la tabla PERMISO, que identifica de manera única el o los permisos que tendrá asociado este rol
rol_id_rol	Int	FK	Llave foránea que referencia la tabla ROL, que identifica de manera única el rol asociado a cada permiso
fecha_asignacion	Date		Representa la fecha en la que se realizó la asignación de permisos a cada rol

Tabla 15. Permiso

Campo	Tipo de dato	Tipo de llave	Descripción
id_permiso	Int	PK	Llave primaria que identifica de manera única los permisos registrados
nombre	Varchar(100)		Nombre del permiso
descripcion	Varchar(200)		Descripción o características de cada permiso
estado	Boolean		Indica si el permiso se encuentra activo o inactivo
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro

			en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información

Tabla 16. Egreso

Campo	Tipo de dato	Tipo de llave	Descripción
id_egreso	Int	PK	Llave primaria que identifica de manera única las entidades registradas
usuario_id_usuario	Int	FK	Llave foránea que referencia la tabla USUARIO, que identifica de manera única el usuario que hizo el registro del egreso o salida de dinero que se presentó en la empresa
concepto_id_concepto	Int	FK	Llave foránea que referencia la tabla CONCEPTO, que identifica de manera única el concepto o motivo por el que se registró el egreso
Valor	Float		Representa el valor del egreso
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registró

			en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información

Tabla 16. CONCEPTO

Campo	Tipo de dato	Tipo de llave	Descripción
id_concepto	Int	PK	Llave primaria que identifica de manera única los conceptos registrados
descripcion	Varchar(200)		Descripción o características del concepto registrado
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 17. Cuentasxcobrar

--	--	--	--

Campo	Tipo de dato	Tipo de llave	Descripción
id_cuentasx cobrar	Int	PK	Llave primaria que identifica de manera única las cuentas por cobrar registradas
pago_id_pago	Int	FK	Llave foránea que referencia la tabla PAGO, que identifica de manera única el pago al que pertenece la cuenta por cobrar
motivo	Varchar(100)		Representa la razón por la que se está generando la cuenta por cobrar
nota	Varchar(300)		Representa una cadena de texto que el usuario puede utilizar para describir cuando y como se realizara el pago de la cuenta por cobrar
estado	Boolean		Representa el estado en la que se encuentra la cuenta por cobrar. Siendo 1 (uno) cuando la cuenta se encuentra saldada, y 0 (cero) en caso contrario
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al

			registro
--	--	--	----------

Tabla 18. Banco

Campo	Tipo de dato	Tipo de llave	Descripción
id_banco	Int	PK	Llave primaria que identifica de manera única los bancos registrados
nombre	Varchar(50)		Representa el nombre de la entidad bancaria
estado	Boolean		Representa un estado en el cual 1 (uno) indica que se encuentra activo, y 0 (cero) en caso contrario
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 19. TarifaPlanilla

--	--	--	--

Campo	Tipo de dato	Tipo de llave	Descripción
id_tarifaplanilla	Int	PK	Llave primaria que identifica de manera única las tarifas registradas
descripcion	Varchar(100)		Descripción o características de la tarifa registrada
Valor	Float		Representa el valor de la tarifa
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 20. PlanillaConsignar

Campo	Tipo de dato	Tipo de llave	Descripción
id_planillaconsignar	Int	PK	Llave primaria que identifica de manera única las planillas a consignar registradas
Nivel_riesgo	Int		Representa el nivel de riesgo que puede tener asociado un cliente

Vsalud_riesgo	Float		Representa el valor que se consigna por concepto de salud y riesgo
Vcon_pension	Float		Representa el valor que se consigna por concepto de pensión
Created_at	DateTime		Representa la fecha y hora en la cual se realizó el nuevo registro en el sistema
Updated_at	DateTime		Representa la fecha y hora en la cual se actualizo la información
Created_by	Int		Representa el id del usuario que realizó el nuevo registro
Updated_by	Int		Representa el id del usuario que realizó la última modificación al registro

Tabla 21. Detalle_Pago

Campo	Tipo de dato	Tipo de llave	Descripción
Id_detalle_pago	Int	PK	Llave primaria que identifica de manera única el detalle de cada pago
Id_banco	Int	FK	Llave foránea que referencia la tabla BANCO, que identifica de manera única el banco en el cual se realizará el pago

Id_pago	Int	FK	Llave foránea que referencia la tabla PAGO, que identifica de manera única el pago al que pertenece.
Valor_salud	Float		Representa el valor de salud pagado por el cliente
Valor_riesgo	Float		Representa el valor de riesgo pagado por el cliente
Valor_pension	Float		Representa el valor de pensión pagado por el cliente
Valor_aporte	Float		Representa el valor del aporte pagado por el cliente

HERRAMIENTAS SELECCIONADAS

Para realizar el desarrollo del sistema y la implementación de la base de datos, es necesario contar con diversas herramientas tanto hardware como software.

HARDWARE

Las características de la computadora en la que se desarrolla la aplicación son las siguientes:

- Sistema operativo: Windows 8.1
- Procesador: Intel Core i3
- Memoria RAM: 4 GB
- Disco Duro: 500 GB

SOFTWARE

Para el desarrollo del sistema se seleccionaron los siguientes tipos de software: para el SGBD se eligió Microsoft SQL Server 2008 Express, para el diseño relacional se utilizó DBDesigner y para el desarrollo del software se implementó Microsoft Visual Studio community 2013.

La selección del software se realizó en base a la distribución libre de los productos, así como de la documentación en línea existente y sobre todo en la funcionalidad, buscando cubrir en su totalidad con los requerimientos del cliente.

MICROSOFT SQL SERVER 2008 EXPRESS

Microsoft SQL Server Express es un sistema de administración de datos gratuito, eficaz y confiable que ofrece un almacén de datos completo y confiable para sitios web ligeros y aplicaciones de escritorio desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (línea de comandos o interfaz gráfica de Management Studio) es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crea tablas y definir relaciones entre ellas (DDL).



Figura 6. SQLServer 2008

TRANSACT SQL

SQL es un lenguaje de consultas para bases de datos relacionales, pero carece de la potencia de los lenguajes de programación. No permite el uso de variables, estructuras de control de flujo, bucles y otros elementos característicos de la programación.

Sin embargo, SQL es la mejor herramienta para trabajar con bases de datos. Cuando se requiere construir una aplicación que soporte el manejo de una base de datos relacional, se hace necesario utilizar un sistema que soporte las consultas SQL's y la versatilidad de los lenguajes de programación tradicionales. Nace entonces TRANSACT SQL, lenguaje de programación que proporciona Microsoft SQL Server para extender el SQL estándar.

TRANSACT SQL va más allá de un lenguaje SQL ya que aunque no permite crear interfaces de usuario o crear aplicaciones ejecutables, incluye características propias de cualquier lenguaje de programación, características que nos permiten definir la lógica necesaria para el tratamiento de la información, como ejemplo:

- Tipos de datos
- Estructuras de control de flujo
- Excepciones
- Funciones predefinidas

Estas características permitirán crear bloques de código más complejos; no programas sino procedimientos o funciones que podrán ser llamados en cualquier momento. SQLServer permite definir dos tipos de bloques de código, procedimientos almacenados y triggers.

PROCEDIMIENTOS ALMACENADOS (STORE PROCEDURE)

Un procedimiento almacenado o Store Procedure por sus siglas en inglés, está formado por un conjunto de instrucciones Transact SQL que definen un determinado proceso, puede aceptar parámetros de entrada y devolver un valor o un conjunto de resultados. Este procedimiento se almacena en el servidor y puede ser llamado en cualquier instante.

Los procedimientos almacenados son instrucciones o lotes SQL que están pre compilados. La primera vez que se ejecuta un procedimiento, el procesador de SQL Server lo analiza y lo almacena en una tabla del sistema. Luego el procedimiento es ejecutado cada que vez que es llamado.

Los procedimientos almacenados se ejecutan casi de manera instantánea, por lo que el uso de estos mejora notablemente la potencia y eficacia del SQL. Estos se crean mediante el comando CREATE PROCEDURE y se ejecutan con EXECUTE. Para ejecutarlo también se puede utilizar el nombre del procedimiento sólo.

TRIGGERS

Un Trigger es una clase especial de procedimiento almacenado que se ejecuta automáticamente cuando se produce un evento en el servidor de base de datos.

SQL Server permite crear varios tipos de triggers para una instrucción específica. Se clasifican en:

- Trigger DML: Se ejecutan cuando un usuario intenta modificar los datos mediante un evento de lenguaje de manipulación de datos (DML). Los eventos DML son instrucciones INSERT, UPDATE o DELETE en una tabla o vista.
- Trigger DDL: Se ejecutan en respuesta a una variedad de eventos de lenguaje de definición de datos (DDL). Estos eventos corresponden principalmente a instrucciones CREATE, ALTER, DROP de Transact SQL.

- Trigger LOGON: Se activan en respuesta al evento Logon que se genera cuando se establece una sesión de usuario.

DML (DATA MANIPULATION LANGUAGE)

Lenguaje de manipulación de datos. Es un lenguaje proporcionado por los sistemas gestores de base de datos que brinda a los usuarios la posibilidad de realizar tareas de consulta y modificación de los datos contenidos en la base de datos.

Dentro de los elementos que se encuentran en el lenguaje de manipulación de datos, se pueden destacar los siguientes: Select, Insert, Delete, Update, Order by, Group by, From y Where.

DDL (DATA DEFINITION LANGUAGE)

Lenguaje de definición de datos. Es un lenguaje proporcionado por los sistemas gestores de base de datos que permite a los usuarios tareas de definición de las estructuras que almacenaran los datos así como los procedimientos o consultas que permitirán consultarlos o modificarlos.

SQL utiliza una colección de verbos cuyo objetivo es modificar el esquema de la base de datos, permitiendo añadir, modificar o eliminar tablas u otros objetos. Estas declaraciones se pueden combinar con otras sentencias SQL, por lo que el DDL no es realmente un lenguaje independiente. La sentencia más común es CREATE TABLE.

El lenguaje de programación SQL (el más difundido entre los gestores de bases de datos) admite las siguientes sentencias de definición: CREATE, DROP, ALTER, cada uno de los cuales se pueden aplicar a tablas, vistas, procedimientos, funciones y triggers.

DBDesigner

DBDesigner es un sistema totalmente visual de diseño de base de datos, que combina muchas características profesionales con un diseño simple, claro y fácil de usar, con el fin de ofrecer un método efectivo para gestionar las bases de datos.

DBDesigner fue desarrollado por FabForce y permite desarrollar una base de datos teniendo en cuenta el diseño y las funcionalidades independientemente del gestor de base de datos que se utilizará.

DBDesigner es un proyecto Open Source disponible para Microsoft y Linux. Es capaz de trabajar con MySQL, Oracle, MSSQL y cualquier ODBC, por lo que se puede utilizar con la mayoría de bases de datos existentes.

Dentro de las funcionalidades encontradas en DBDesigner se pueden destacar las siguientes:

- Fácil de usar, al conocer en detalle el modelo que se quiere representar.
- Permite construir una base de datos bajo una interfaz intuitiva.
- Proporciona buena documentación acerca de su uso.
- Permite conectar directamente a una base de datos.
- Permite construir una base de datos a partir del modelo.
- Puede guardar el diseño en formato XML para mantener toda la información.



Figura 7. DBDesigner

MICROSOFT VISUAL STUDIO COMMUNITY 2013

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación, como C++, C#, J#, ASP.Net y Visual Basic .Net, aunque actualmente se han desarrollado extensiones necesarias para otros lenguajes como Python, Ruby, entre otros.

Visual Studio permite a los desarrolladores crear aplicaciones y sitios web, así como varios servicios web en cualquier entorno que soporte la plataforma .Net

Estas características han hecho que se elija Visual Studio como la herramienta de desarrollo para este trabajo.



Figura 8. Visual Studio Community

CRIPTOGRAFIA

La criptografía es el arte de cifrar y descifrar información mediante técnicas especiales y se emplea para permitir el intercambio de información de manera segura. Para encriptar la información se emplean complejas fórmulas que son las funciones hash.

FUNCIONES HASH

Las funciones hash son algoritmos que permiten crear a partir de una entrada (sea texto, archivo, password) una salida alfanumérica de longitud fija, que representa un resumen de toda la información que se ha enviado. Es decir, a partir de los datos de entrada se crea una cadena solo puede volver a ser creada a partir de esos mismos datos. Las funciones hash buscan asegurar que la información que se transmite no se ha modificado.

FUNCION HASH MD5 (Message-Digest Algorithm 5)

MD5 o Algoritmo de resumen del mensaje 5, es un algoritmo de reducción criptográfico de 128 bits. Toma como entrada un mensaje de tamaño desconocido y produce una salida de 128 bits respecto a la entrada.

MD5 es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (Massachusetts Institute of Technology). Desarrollado 1991 con el fin de reemplazar el algoritmo MD4 el cual Hans Dobbertin demostró que era inseguro.

En 1996 Dobbertin anuncio una colisión de Hash en la función de compresión del MD5. Esto hizo que los criptógrafos comenzaran a recomendar el reemplazo de la codificación MD5 a otras como SHA1.

El algoritmo MD5 ha sido ampliamente utilizado por la industria del software e inicialmente se pensó que era criptográficamente seguro. No obstante, algunas investigaciones han encontrado vulnerabilidades que hacen que se cuestione este algoritmo. En el año 2004, Xiaoyun Wang, Dengguo Feng, Xuejia Lai y Hongbo Yu anunciaron que habían descubierto colisiones de hash para MD5. Aunque el ataque realizado por este grupo de investigadores era de tipo analítico se demostró que el tamaño del hash (128 bits) es demasiado pequeño y que era vulnerable frente ataques de este tipo.

APLICACIONES DE MD5

- El MD5 es utilizado en software para proporcionar seguridad a los archivos y a la información.
- En sistemas UNIX y GNU/Linux se utiliza MD5 para la autenticación de usuarios.
- MD5 es utilizado ampliamente en internet para comprobación de ficheros.

- MD5 puede ser utilizado para comprobar la integridad de los correos electrónicos que se envían o reciben a través de claves públicas o privadas.

VENTAJAS DE MD5

La ventaja principal de este algoritmo es que es casi imposible reconstruir la cadena original a partir del resultado y encontrar dos cadenas de texto que generen el mismo resultado. Esto permite utilizar el algoritmo para enviar contraseñas a través de un medio inseguro como es internet.

DESVENTAJA DE MD5

En la actualidad se considera que el algoritmo MD5 que cifra la información a partir de 128 bits, no brinda total seguridad. Se recomienda utilizar el algoritmo SHA1, dado que cuenta con 160 bits.

CODIFICACION MD5

La codificación MD5 de 128 bits es representada como una cadena de 32 dígitos hexadecimal. La siguiente cadena será tratada con MD5 y se observará su respectiva salida:

MD5 ("prueba de md5") = 9b3b07046a058ac013f5e8f4d31ab8ce

A continuación, se observa que un simple cambio en la cadena cambia la codificación:

MD5 ("prueba md5") = 138665e789c1d66f848efd534bd955ec

FUNCION HASH SHA-1 (Secure Hash Algorithm)

SHA-1 o Algoritmo seguro de hash es un algoritmo de funciones hash criptográficas, el cual toma una entrada de cualquier longitud y produce una salida de 160 bits.

VENTAJAS DE SHA1

- Es un poco más lento que MD5, pero es computacionalmente más complejo y su salida es de mayor longitud, por lo que es considerado más seguro.
- Es considerado seguro para todo tipo de aplicaciones, además cuenta con versiones como SHA-256, SHA-384 y SHA-512 que generan hash de 256, 384, y 512 bits respectivamente; estas versiones reemplazarán en el futuro a SHA1 mientras continúa su desarrollo.
- SHA1 es un excelente método criptográfico que convierte una cadena de texto en otra de 40 caracteres sin importar la longitud de la cadena original, cifrándola de tal manera que se hace casi imposible obtenerla ya que SHA1 no posee un método de reversa para obtener la cadena original a partir de la cadena cifrada.

CARACTERISTICAS

- Genera una cadena de 160 bits, su funcionamiento es igual que el algoritmo MD5, pero con un buffer de 160 bits
- Es más seguro que el algoritmo MD5 dado que utiliza un mayor número de bits.

APLICACIONES DE SHA1

- Es utilizados en aplicaciones de correo electrónico, transferencias electrónicas de fondos, almacenamiento de datos y en general en aplicaciones que requieren integridad en los datos y autenticación.
- Es implementado por organizaciones privadas y comerciales

CODIFICACION SHA-1

A continuación, se muestra un ejemplo de la cadena obtenida después de encriptar en SHA1.

SHA1 ("prueba de sha1") = c39fc2b4d7f771e4fed1d288b2c24d34b875715b

Un pequeño cambio en el mensaje resulta un hash totalmente diferente, como se puede observar a continuación:

SHA1 ("prueba de sha") = 32fad87cd9d875d23328b2ade0f13559c2e46b34

COMPARATIVA ENTRE MD5 Y SHA-1

MD5	SHA1
Genera una salida de 128 bits de longitud	Genera una salida de 160 bits de longitud
La dificultad de generar un mensaje que tenga un resumen dado es de 2 ¹²⁸ operaciones	La dificultad de generar un mensaje que tenga un resumen dado es 2 ¹⁶⁰ operaciones

La diferencia de bits a favor de SHA1 lo convierte en un algoritmo más seguro y resistente a ataque de fuerza bruta que el algoritmo MD5

Tabla 22. Comparación entre Algoritmo Md5 y Sha1

ARQUITECTURA DE TRES NIVELES

La arquitectura de software incluye los elementos estáticos y dinámicos del sistema que se desea construir. De igual manera la arquitectura de tres niveles incluye estos aspectos, y brinda además mejores opciones y mayor organización para el desarrollo de software.

La arquitectura de tres niveles define como organizar el modelo en capas. La primera capa se denomina capa de presentación y consiste normalmente en la interfaz con la que interactúa el usuario. La capa intermedia, consiste en la aplicación o lógica de negocio, y la tercera capa, llamada capa de datos, contiene los datos necesarios para la aplicación. Este patrón es importante ya que mejora la comprensión y organización en el desarrollo de sistemas complejos. Además ayuda a identificar partes del código que pueden ser reutilizables.

Esta separación entre la lógica de la aplicación y la interfaz de usuario brinda gran flexibilidad al desarrollo de software. Dado que podrían construirse múltiples interfaces de usuario sin afectar la lógica de la aplicación.

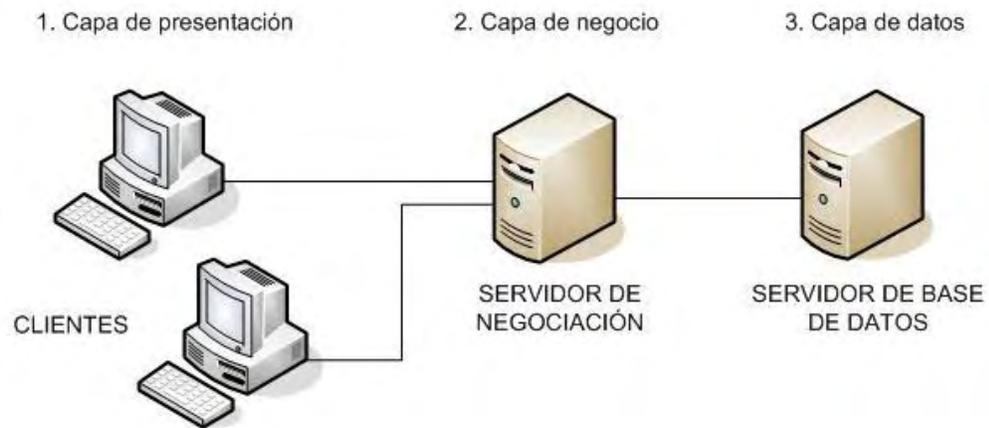


Figura 9. Arquitectura de tres niveles

CAPA DE PRESENTACION

Es la encargada de la interacción entre el sistema y los usuarios. En el mundo de la informática es conocida como la interfaz gráfica y debe poseer ciertas características como ser amigable a los usuarios, ser entendible y fácil de usar. Esta capa se comunica únicamente con la capa intermedia o lógica de negocio.

CAPA DE LOGICA DE NEGOCIO

Aquí se almacenan las funciones y procedimiento que se ejecutan. Se capturan las peticiones del usuario, se procesa la información y se envían las respuestas al usuario. Se denomina lógica de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir las solicitudes del usuario y presentar los resultados y con la capa de datos para solicitar a la base de datos almacenar o recuperar datos de él.

CAPA DE DATOS

Esta capa es la encargada de almacenar y devolver los datos a la capa de lógica negocio, aunque para que pueda realizar este proceso se hace necesario que tenga almacenados funciones y procedimientos que permitan el fácil acceso a los datos.

3. EJECUCION DEL PROYECTO

PLANIFICACION

La primera fase en todo proyecto de desarrollo es la planificación; en esta etapa se analiza el desarrollo de cada Sprint. Sin embargo, antes de iniciar con las iteraciones, se hace necesario realizar una definición general de las iteraciones, esto es, realizar una planificación inicial, que permitirá identificar el propósito de cada iteración y definir en forma general lo que se realizará en cada Sprint. Este proceso inicial se denomina Sprint 0.

Antes de definir la cantidad y los objetivos de cada uno de los Sprint's a desarrollar durante la ejecución de este proyecto, se hace necesario definir los requisitos funcionales y no funcionales que debe cumplir el sistema a implementar.

SPRINT 0

REQUISITOS FUNCIONALES Y NO FUNCIONALES

Teniendo claro el problema y orientando el presente proyecto hacia una solución óptima, se han identificado los siguientes requisitos funcionales y no funcionales que deberá cumplir el sistema.

REQUISITOS FUNCIONALES

- El sistema debe permitir registrar la información de los usuarios.
- Los usuarios no deben ser eliminados del sistema para mantener la consistencia en la información histórica
- El sistema debe permitir asignar permisos a los usuarios registrados
- El sistema debe permitir asignar un rol a los usuarios registrados
- El sistema debe permitir que los usuarios registren egresos que se presenten dentro de las actividades diarias
- El sistema debe permitir registrar la información de los clientes.
- Los clientes no deben ser eliminados del sistema para mantener la consistencia en la información histórica
- El sistema debe permitir registrar la información correspondiente a los pagos realizados por los clientes
- El sistema debe permitir emitir un comprobante de los pagos realizados por los clientes
- El sistema debe permitir registrar los valores de las tarifas de riesgo que están rigiendo en la actualidad
- El sistema debe permitir generar un reporte que muestre el último pago realizado por cada cliente
- El sistema debe permitir generar reportes de pagos realizados por clientes (semanales, mensuales)

REQUISITOS NO FUNCIONALES

Facilidad de uso:

- El sistema debe ser de fácil uso para los usuarios y permitir una fácil adaptación en la entidad.

Ingreso de información:

- El sistema no debe permitir el cierre de una operación o transacción hasta que el proceso haya finalizado satisfactoriamente.
- El ingreso de información en el sistema debe diseñarse con procedimientos que permitan el ingreso de datos de manera adecuada en la base de datos.
- El sistema debe presentar mensajes de error que permitan al usuario identificar el tipo de error y comunicarse con el administrador del sistema.

Desempeño:

- El sistema debe garantizar la confiabilidad y seguridad de la información almacenada en la base de datos.
- El sistema debe estar en capacidad de obtener en un corto tiempo la información contenida en la base de datos.

Disponibilidad:

- El sistema debe garantizar el acceso a la información para aquellos usuarios autenticados.
- El sistema debe estar disponible para su uso en todo momento.

Escalabilidad:

- El sistema debe ser construido sobre la base de una metodología ágil, Scrum. De manera que nuevas funcionalidades y requerimientos pueden ser incorporados afectando el código de la menor manera posible.
- El sistema debe estar en la capacidad de permitir el desarrollo de nuevas funcionalidades.

Seguridad:

- El acceso al sistema debe estar restringido por el uso de nombres de usuarios y claves asignadas a cada uno de los usuarios. Solo podrán ingresar las personas registradas. Estos usuarios serán clasificados

además por roles con acceso a las opciones de trabajo definidas para cada rol.

- El sistema debe estar en la capacidad de rechazar el acceso de usuarios no autorizados.
- El sistema debe contar con mecanismos que permitan llevar un control de registro de actividad con la identificación del usuario que realizó dicho proceso.

Confidencialidad:

- La información contenida en la base de datos debe estar protegida de acceso no autorizado.

Validación de información:

- El sistema debe validar la información contenida en los formularios de ingreso de datos. En este proceso se deben tener en cuenta aspectos como: campos obligatorios, longitud de caracteres permitidos y manejo de tipo de datos.

CONFORMACIÓN DEL EQUIPO DE TRABAJO

En todo proyecto es importante definir el equipo de trabajo y además conocer el equipo humano con el que se cuenta para el desarrollo éxito del mismo. El equipo de trabajo para llevar a cabo el desarrollo del software administrativo en la entidad PREVENIR estará conformado según lo descrito en la siguiente tabla: (Ver Tabla 23)

Rol	Persona	Área de trabajo
-----	---------	-----------------

PRODUCT OWNER	Mireya Obando	Administradora
SCRUM MASTER	Jesús Francisco Solis	Análisis y Desarrollo
TEAM	Jesús Francisco Solis	Análisis y Desarrollo
	Mireya Obando	Administradora
	Patricia	Secretaria

Tabla 23. Equipo de trabajo

BACKLOG DEL PRODUCTO

El Backlog del producto contiene toda la funcionalidad que el producto final debería tener. Para el desarrollo del sistema administrativo se ha definido el Backlog del producto identificando las funcionalidades, especificando además la prioridad que tendrá cada una y realizando una estimación del tiempo requerido para su implementación.

El Backlog del producto para el presente proyecto se encuentra definido en la siguiente tabla: (Ver Tabla 24)

Módulo	Importancia	Tiempo estimado (en semanas)	Descripción
Módulo Clientes	1000	4	Funcionalidad para gestionar la información de los clientes
Módulo usuarios	700	2	Funcionalidad para gestionar la información de los usuarios que

			interactuaran con el sistema
Módulo Grupos	600	2	Funcionalidad que permite crear grupos de trabajo
Módulo permisos	600	3	Funcionalidad que permite crear permisos para acceder a ciertos lugares del sistema
Módulo tarifas	1000	3	Funcionalidad que permite gestionar la información de las tarifas de pagos
Módulo egresos	500	2	Funcionalidad que permite registrar los gastos que se generan en la entidad en el desarrollo de sus actividades diarias
Módulo pagos	1000	6	Funcionalidad que permite registrar los pagos de los clientes por concepto de salud, riesgo y pensión.
Módulo reportes	800	4	Funcionalidad que permite observar reportes que facilitan la toma de decisiones de la entidad

Tabla 24. Backlog del Producto

La escala de importancia ha sido definida por el Product Owner y el Scrum Master de acuerdo a la siguiente figura:



Figura 10. Escala de importancia

SPRINT 1

El primer Sprint tiene como objetivo implementar los módulos con mayor importancia definidas en el Backlog del Producto. Estos son: Clientes, Tarifas y Pagos.

PLANIFICACION

Para la planificación del Sprint 1, se llevó a cabo una reunión con el Product Owner. En esta reunión se realizó un análisis de los procesos y las funcionalidades que serán implementadas para cada uno de los módulos.

De esta reunión se logró determinar la historia de usuario 1 para el Sprint 1, detallada en la siguiente tabla:

Historia de usuario	Importancia Product Owner ¹	Importancia técnica ²	Descripción

¹ Determina la importancia estimada por el propietario del producto y está cuantificada de acuerdo a la escala establecida en la figura 9.

² Determina la importancia técnica estimada por el Scrum Master

Gestionar fondos de pensión	800	1000	Registrar, modificar y consultar la información correspondiente a los fondos de pensión en la cual están afiliados los clientes
Gestionar ciudad	500	200	Registrar, modificar y consultar la información correspondiente a las ciudades donde se encuentra viviendo el cliente
Gestionar novedad	700	500	Registrar, modificar y consultar la información correspondiente a los motivos por el cual el cliente ingresa en la entidad. Estos pueden ser: Ingreso, Reingreso o Retiro
Gestionar departamento	500	200	Registrar, modificar y consultar la información correspondiente a los departamentos al que pertenecen las ciudades
Gestionar Eps	800	1000	Registrar, modificar y consultar la información correspondiente a las empresas prestadoras de salud en la cual se encuentran vinculados los cliente
Gestionar entidad	800	1000	Registrar, modificar y

			consultar la información correspondiente a las entidades en la cual se realizan los pagos de los trabajadores
Gestionar Arl	800	1000	Registrar, modificar y consultar la información correspondiente a las empresas que prestan el servicio de riesgos profesionales en la que se pueden afiliar los clientes
Gestionar el cliente	1000	1000	Registrar, modificar, consultar y eliminar clientes. Cuando se realiza un nuevo registro de cliente se debe especificar la Eps, Arl, Afp y entidad en la que se encuentra afiliado el cliente.
Gestión de tarifas de riesgo	1000	1000	Registrar, modificar y consultar la información correspondiente a las tarifas que se cobran de acuerdo al nivel de riesgo del cliente. Además la tarifa aumenta si el cliente se encuentra vinculado a una Eps o a un fondo de pensión.
Gestión de tarifas de planilla	800	1000	Registrar, modificar y consultar la información correspondiente a las tarifas que tienen las

			planillas que diligencian aquellos clientes tipo INDEPENDIENTE. Este valor varía en función del tiempo.
Gestión de tarifas de planillas a consignar	800	1000	Registrar , modificar y consultar la información correspondiente a los valores que se consignan y envían a la sede principal localizada en la ciudad de Cali
Registrar pagos de clientes dependientes e independientes	1000	1000	Registrar la información correspondiente a los pagos que realizan los clientes dependientes por concepto de salud, riesgo y pensión

Tabla 25. Historia de usuario 1

SPRINT 2

El segundo Sprint tiene como objetivo implementar los módulos de permisos, grupos, usuarios, egresos y reportes.

PLANIFICACION

Para la planificación del Sprint 2, se llevaron a cabo dos reuniones con el Product Owner. En estas reuniones se determinaron los procesos y funcionalidades que serán implementadas para cada uno de los módulos mencionados.

De estas reuniones se lograron determinar la historia de usuario 2 para el Sprint 2, detallada en la siguiente tabla: (Ver Tabla 26)

Historia de usuario	Importancia Product Owner ³	Importancia técnica ⁴	Descripción
Gestionar de permisos	300	1000	Registrar, modificar y consultar la información correspondiente a los permisos que se pueden asignar a los grupos del sistema
Gestión de grupos	500	1000	Crear, Asignar o modificar permisos a los grupos
Gestión de conceptos	200	500	Registrar, modificar y consultar la información correspondiente a los conceptos o razones por las que se pueden generar egresos en la entidad
Gestión de egresos	800	500	Registrar y consultar la información correspondiente a los egresos que se generen en la empresa
Generación de reportes	1000	1000	Consultar información de aquellos reportes definidos entre el Scrum Master y el Product Owner

Tabla 26. Historia de usuario 2

³ Determina la importancia estimada por el propietario del producto y está cuantificada de acuerdo a la escala establecida en la figura 9.

⁴ Determina la importancia técnica estimada por el Scrum Master

CONCLUSIONES

Con base en el planteamiento del objetivo general y los objetivos específicos, se ha desarrollado una aplicación que cumple con la finalidad de sistematizar los procesos de pagos y gestión de la información en la FUNDACIÓN PREVENIR. Se buscó que el sistema desarrollado sea amigable con los usuarios para mejorar el proceso de adaptación y familiarización con la nueva herramienta software.

El sistema administrativo no solo ha agilizado los procesos y gestión de la información en la entidad, sino que ha mostrado una manera más rápida, fácil y óptima de realizar las cosas.

Haber implementado la metodología de desarrollo ágil Scrum dio como resultado una experiencia gratificante y un aprendizaje constante de todas las etapas de desarrollo del mismo. El hecho de haber terminado el desarrollo del sistema obteniendo como resultado un producto funcional y satisfacción por parte del cliente (FUNDACIÓN PREVENIR) muestra que los objetivos trazados se cumplieron.

Uno de los objetivos planteados fue apropiar el conocimiento de la metodología ágil Scrum con el fin de conocer el método y poder aplicarlo en el desarrollo del sistema. Luego de su estudio se llegó a la conclusión de que la simplicidad en la implementación de la metodología evita perder de vista los objetivos. Es sencilla, flexible y está enfocada en las necesidades verdaderas del cliente. Es ideal para sistemas complejos y con requisitos variantes.

Como conclusión se quiere exponer que el hecho de ir realizando entregas constantes al cliente, no solo aumenta la confianza del mismo al saber que su producto se encuentra en proceso de desarrollo sino que además aumenta la confianza del desarrollador dado que ir concretando etapas en el desarrollo es una motivación para seguir adelante.

RECOMENDACIONES

- Implementar nuevas funcionalidades dentro de la aplicación con la finalidad de brindar nuevas características.
- Desarrollar e integrar un módulo para cuentas por pagar, de esta manera se podría llevar un mejor control de los servicios que generan un egreso.
- Implementar un módulo que periódicamente envíen emails para aquellos clientes que se encuentra cerca su fecha de pago. Cabe destacar que para realizar esta acción el o los equipos de cómputo deberán tener conexión a internet.

BIBLIOGRAFIA

- [1] <http://www.md5.cz/>
- [2] <http://www.sha1.cz/>
- [3] WinPYME. Software de gestión empresarial.
- [4] <http://www.winpyme.com/>
- [5] Roger S. Pressman. Ingeniería del Software Un Enfoque Práctico: McGraw-Hill, 2002.
- [6] SIIGO. Sistema Integrado de Información Gerencial Operativo.
- [7] http://noqualityinside.com/nqi/nqifiles/XP_Agil.pdf
- [8] <http://www.agilealliance.org/>
- [9] http://www.scrummanager.net/files/sm_proyecto.pdf
- [10] Diseño de sistemas administrativos capítulo 5, pág. 173-209

ANEXO

ANEXO 1. MANUAL DE USUARIO

MANUAL DE USUARIO

AUTENTICACIÓN DE USUARIO

Mediante este formulario podrá acceder a las diferentes opciones que presenta la aplicación.

1. Ingresar en el formulario de autenticación de usuarios (figura 1) el nombre de usuario y contraseña que le fueron asignados por el administrador del sistema.



The screenshot shows a web form titled "AUTENTICACIÓN DE USUARIO". On the left side, there is a 3D illustration of a white figure holding a red padlock. The main form area is titled "ACCESO AL SISTEMA" and contains two input fields: "NOMBRE DE USUARIO" with the text "francisco" and "CONTRASEÑA" with asterisks. Below the fields are two buttons: "SALIR" with a door icon and "INGRESAR" with a green checkmark icon.

Figura 11. Autenticación de usuario

2. Si no desea ingresar presionar el botón SALIR de lo contrario presionar el botón INGRESAR para realizar la validación correspondiente e ingresar al sistema.
3. Si la información ingresada es correcta entonces se mostrará la aplicación.



Figura 12. Aplicación principal

GESTIONAR CIUDADES

REGISTRAR NUEVA CIUDAD

A través de esta opción podrá registrar una nueva ciudad.

1. Presionar el botón REGISTRAR
2. Ir a la opción CIUDAD
3. En el siguiente menú seleccionar la opción NUEVA CIUDAD. Como se muestra en la siguiente imagen.



Figura 13. Acceder a Nueva Ciudad

4. Aparecerá un formulario en el cual deberá ingresar el nombre de la ciudad que se desea agregar y seleccionar el departamento al que pertenece.

Figura 14. Agregar nueva ciudad

5. Presionar el botón GUARDAR en la sesión de acciones.
6. Luego aparecerá una ventana confirmando que se ha realizado la inserción de la nueva ciudad de manera satisfactoria, como muestra la siguiente imagen.

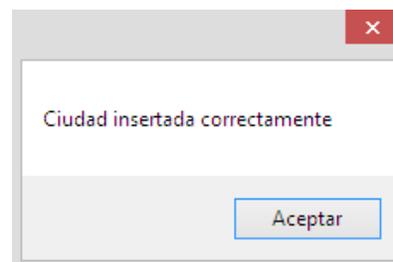


Figura 15. Confirmación de ingreso de ciudad

ACTUALIZAR CIUDAD

A través de esta opción podrá actualizar la información correspondiente a una ciudad.

1. Presionar el botón REGISTRAR
2. Ir a la opción CIUDAD
3. En el siguiente menú seleccionar la opción LISTAR CIUDADES. Como se muestra en la siguiente imagen.



Figura 16. Acceder a listado de ciudades

4. El siguiente formulario muestra todas las ciudades registradas.

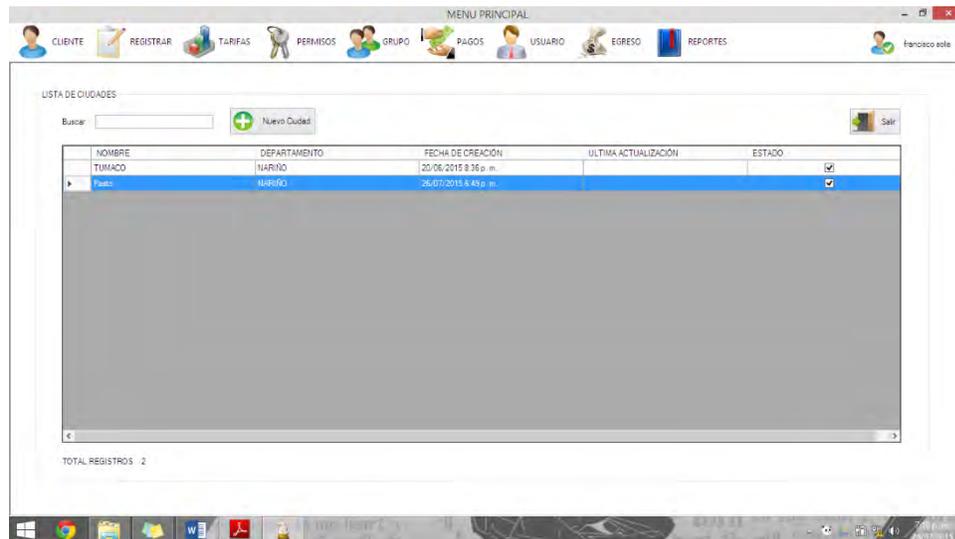


Figura 17. Lista de ciudades

5. Dar doble clic sobre el registro que se desea actualizar.
6. En el formulario que se muestra se deberá ingresar la nueva información.

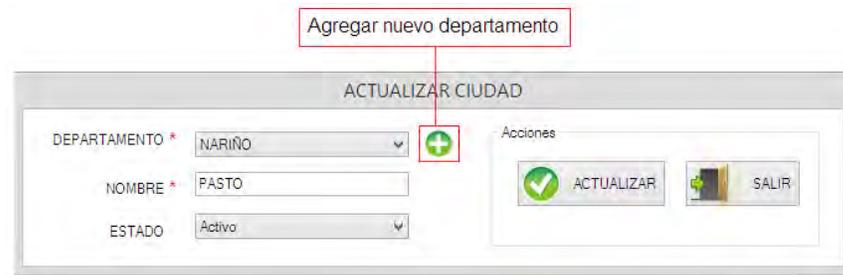


Figura 18. Actualizar ciudad

7. Presionar el boton ACTUALIZAR en la sesión de acciones.
8. Aparecerá una ventana confirmando que se ha realizado la actualización de la ciudad de manera satisfactoria.

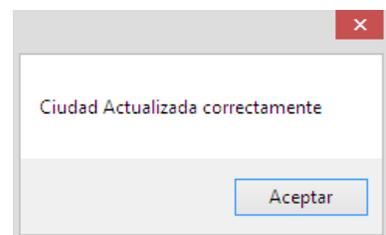


Figura 19. Confirmación de actualización de ciudad

GESTIONAR DEPARTAMENTOS

REGISTRAR UN NUEVO DEPARTAMENTO

A través de esta opción podrá registrar un nuevo departamento.

1. Presionar el botón REGISTRAR
2. Ir a la opción DEPARTAMENTO
3. En el siguiente menú seleccionar la opción NUEVO DEPARTAMENTO. Como se muestra en la siguiente imagen.



Figura 20. Acceder a nuevo departamento

4. Aparecerá un formulario en el cual deberá ingresar el nombre del departamento que se desea agregar.

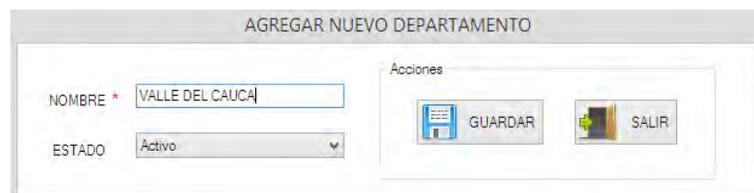
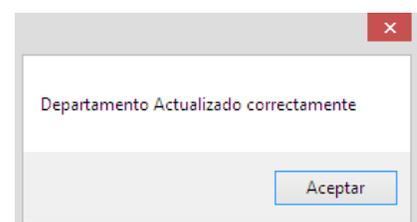
A screenshot of a form titled 'AGREGAR NUEVO DEPARTAMENTO'. It contains two input fields: 'NOMBRE' with the value 'VALLE DEL CAUCA' and 'ESTADO' with a dropdown menu set to 'Activo'. To the right, there is an 'Acciones' section with two buttons: 'GUARDAR' and 'SALIR'.

Figura 21. Agregar nuevo departamento

5. Presionar el botón GUARDAR en la sesión de acciones.



6. Luego aparecerá una ventana confirmando que se ha realizado la inserción de la nueva ciudad de manera satisfactoria, como muestra la siguiente imagen.

ACTUALIZAR DEPARTAMENTO

A través de esta opción podrá actualizar la información correspondiente a los departamentos.

Figura 22. Confirmación de ingreso de departamento

1. Presionar el botón REGISTRAR
2. Ir a la opción DEPARTAMENTO
3. En el siguiente menú seleccionar la opción LISTAR DEPARTAMENTOS. Como se muestra en la siguiente imagen.



Figura 23. Acceder a listado de departamentos

4. El siguiente formulario muestra todos los departamentos registrados.

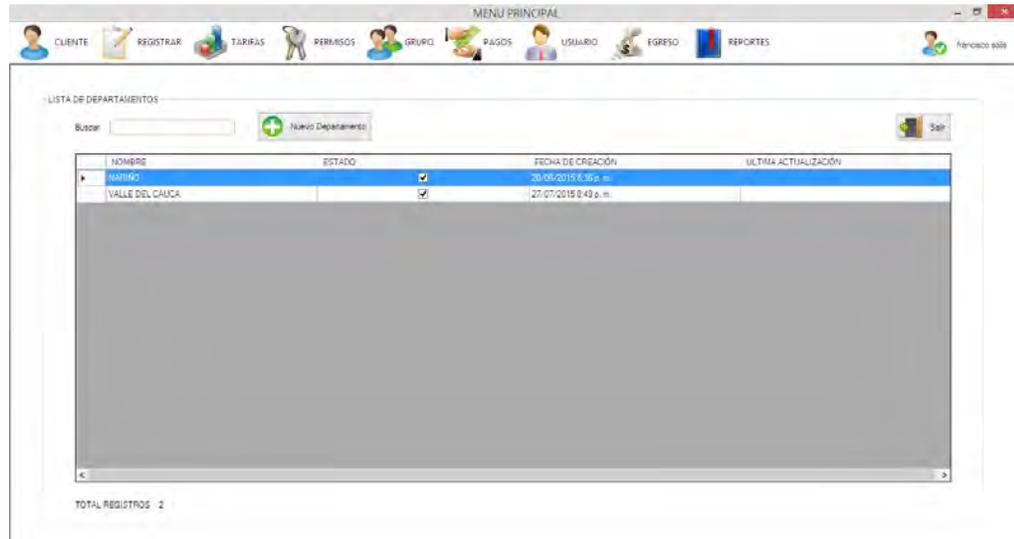


Figura 24. Lista de departamentos

5. Dar doble clic sobre el registro que se desea actualizar.
6. En el formulario que se muestra se deberá ingresar la nueva información.

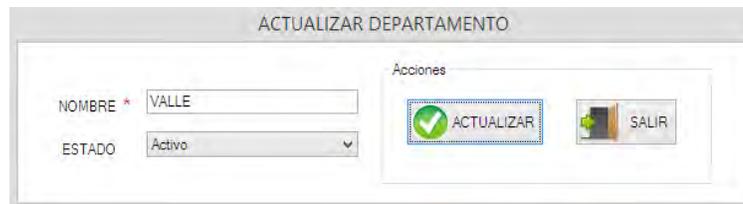


Figura 25. Actualizar departamento

7. Presionar el boton ACTUALIZAR en la sesión de acciones.
8. Aparecerá una ventana confirmando que se ha realizado la actualización de la ciudad de manera satisfactoria.

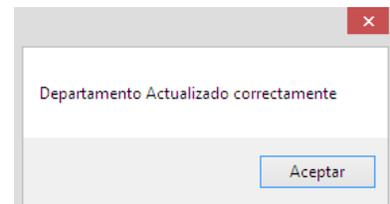


Figura 26. Confirmación de actualización de departamento

GESTIONAR CLIENTES

REGISTRAR UN NUEVO CLIENTE

A través de esta opción podrá registrar un nuevo cliente.

1. Presionar el botón CLIENTE
2. Ir a la opción NUEVO CLIENTE



Figura 27. Acceder a nuevo cliente

3. Aparecerá un formulario en el cual deberá ingresar la información correspondiente al nuevo cliente.

The screenshot shows a web form titled "AGREGAR CLIENTE" with a sub-section "INSERTAR CLIENTE". The form contains the following fields and callouts:

- NOVEDAD:** Dropdown menu with "INGRESO" selected. Callout: "Agregar nueva novedad".
- ENTIDAD:** Dropdown menu with "PREVENIR" selected. Callout: "Agregar nueva entidad".
- CIUDAD:** Dropdown menu with "TUMACO" selected. Callout: "Agregar nueva ciudad".
- EPS:** Dropdown menu with "COOMEVA" selected. Callout: "Agregar una Eps".
- AFP:** Dropdown menu with "No Aplica" selected. Callout: "Agregar nueva Afp".
- ARL:** Dropdown menu with "POSITIVA" selected. Callout: "Agregar nueva Arl".

Other fields include: CEDULA (1087), NOMBRE(S) (francisco), APELLIDOS (solis), FECHA NACIMIENTO (domingo, 09 de agosto de 2015), DIRECCION, TELEFONO, SEXO (M), EMAIL, and TIPO (DEPENDIENTE). At the bottom, there are three buttons: "ACEPTAR" (with a green checkmark), "LIMPIAR" (with a broom icon), and "CANCELAR" (with a red X icon).

Figura 28. Agregar nuevo cliente

4. Presionar el botón GUARDAR en la sesión de acciones.
5. Luego aparecerá una ventana confirmando que se ha realizado la inserción del nuevo cliente de manera satisfactoria, como muestra la siguiente imagen.

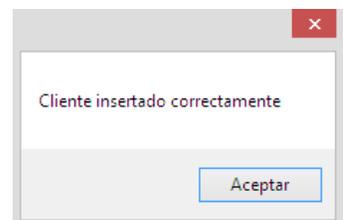


Figura 29. Confirmación de ingreso de cliente

ACTUALIZAR CLIENTE

A través de esta opción podrá actualizar la información correspondiente a los clientes.

1. Presionar el botón CLIENTE
2. Ir a la opción LISTA DE CLIENTES

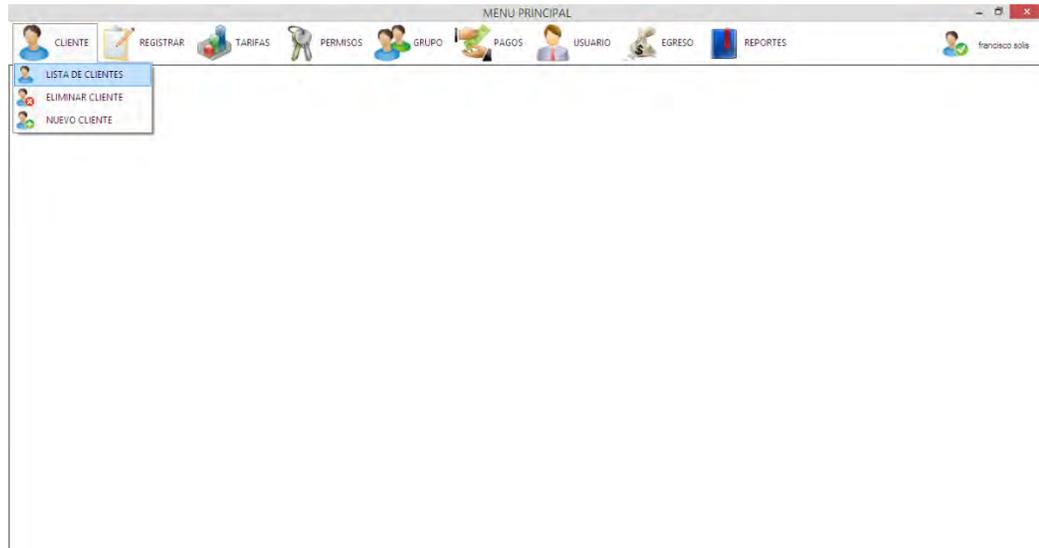


Figura 30. Acceder a listado de clientes

3. El siguiente formulario muestra todos los clientes registrados.

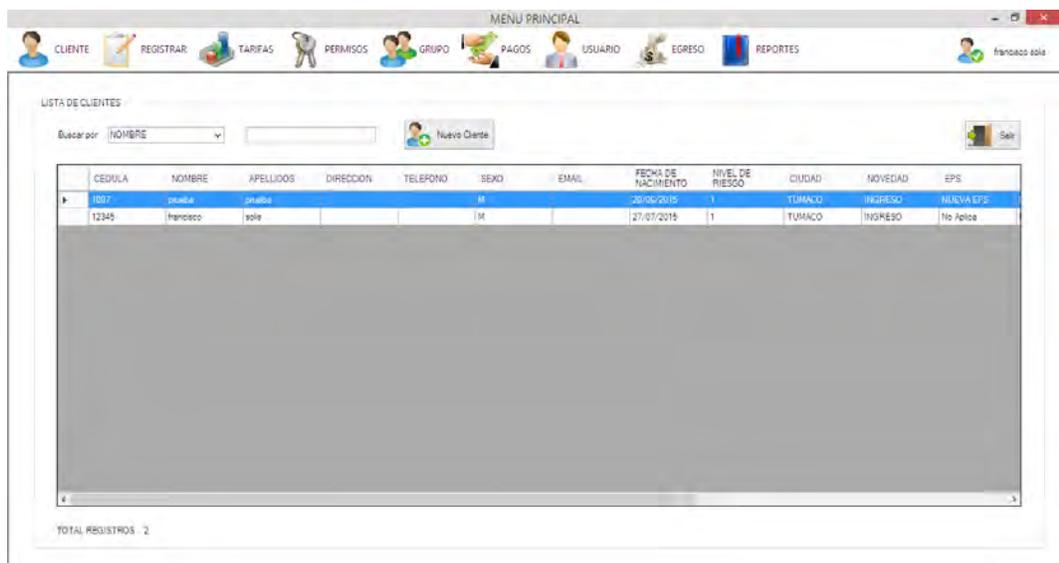


Figura 31. Lista de clientes

4. Dar doble clic sobre el registro que se desea actualizar.
5. En el formulario que se muestra se deberá ingresar la nueva información.

ACTUALIZAR CLIENTE

ACTUALIZAR CLIENTE

CEDULA * 12345 NOVEDAD INGRESO

NOMBRE(S) * jesus francisco ENTIDAD No Aplica

APELLIDOS * sola rengfo FECHA NACIMIENTO lunes, 27 de julio de 2015

DIRECCION NIVEL DE RIESGO 1

TELEFONO CIUDAD TUMACO

SEXO * M EPS No Aplica

EMAIL AFP PROTECCIÓN

TIPO * DEPENDIENTE ARL No Aplica

ACCIONES

ACTUALIZAR CANCELAR

Figura 32. Actualizar cliente

6. Presionar el boton ACTUALIZAR en la sesión de acciones.
7. Aparecerá una ventana confirmando que se ha realizado la actualización del cliente de manera satisfactoria.

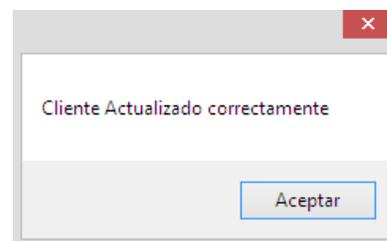


Figura 33. Confirmación de actualización de cliente

ELIMINAR CLIENTE

A través de esta opción podrá eliminar uno o varios clientes

1. Presionar el botón CLIENTE

2. Ir a la opción ELIMINAR CLIENTE



Figura 34. Acceder al formulario eliminar cliente

3. El siguiente formulario muestra los clientes registrados.

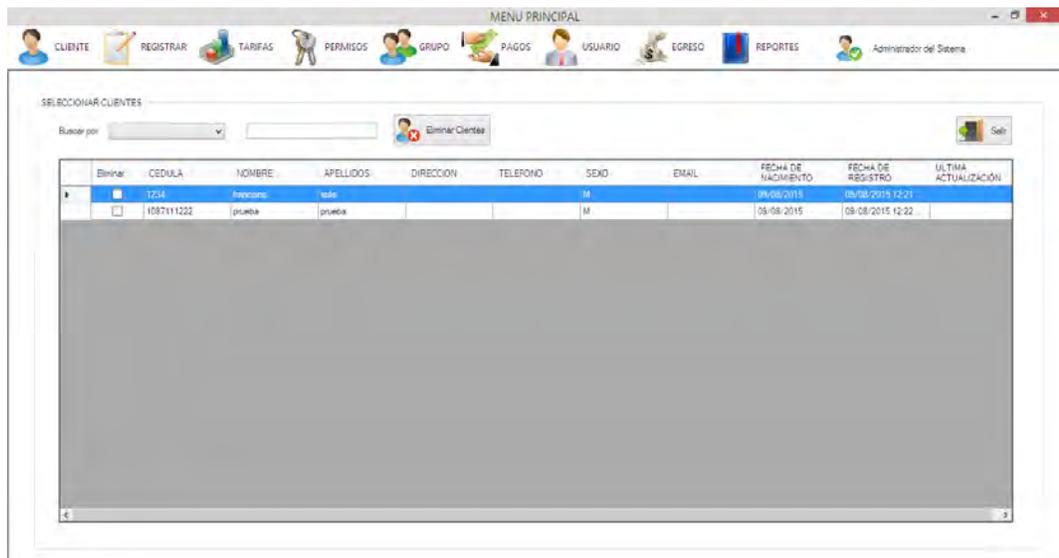


Figura 35. Listado de cliente para eliminar

4. Seleccionar en la casilla de verificación aquellos clientes que desea eliminar

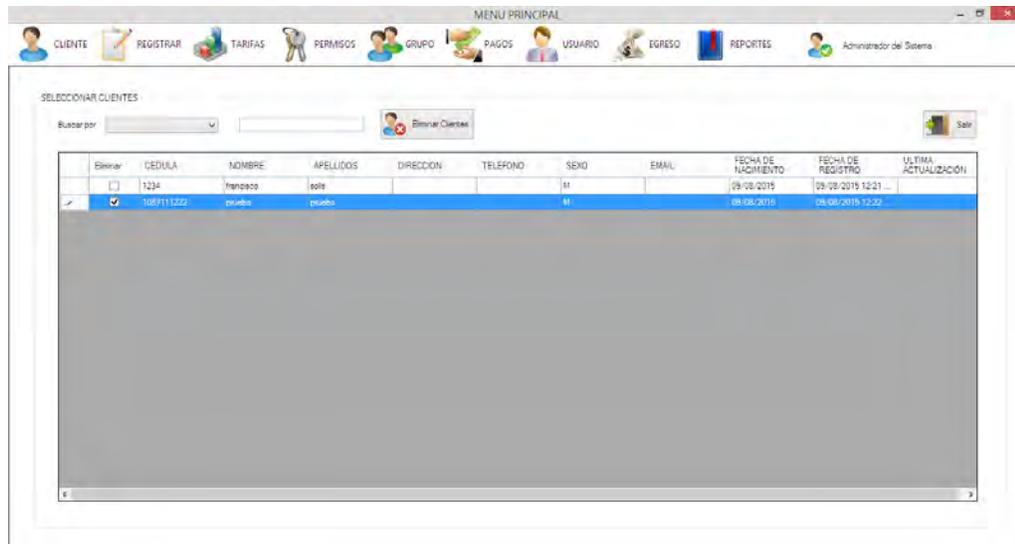


Figura 36. Seleccionar clientes

5. Presionar el botón Eliminar Clientes. Se mostrará una ventana solicitando confirmación, como muestra la siguiente imagen.

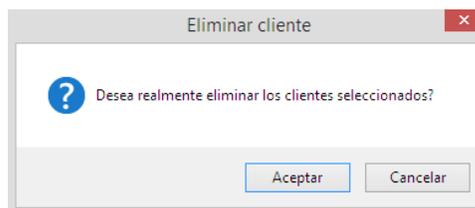
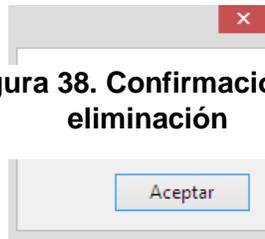


Figura 37. Confirmar eliminación de clientes

6. Aparecerá una ventana confirmando que se ha realizado la eliminación de los clientes de manera satisfactoria.

Figura 38. Confirmación de eliminación



GESTIONAR FONDOS

REGISTRAR UN NUEVO FONDO DE PENSION

A través de esta opción podrá registrar un nuevo fondo de pensión.

1. Presionar el botón REGISTRAR
2. Ir a la opción AFP
3. En el siguiente menú seleccionar la opción NUEVA AFP. Como se muestra en la siguiente imagen.



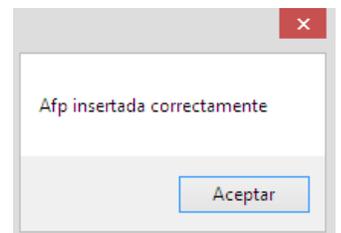
Figura 39. Acceder a formulario nueva Afp

4. Aparecerá un formulario en el cual deberá ingresar el nombre del fondo de pensión que se desea agregar.



Figura 40. Agregar nueva Afp

5. Presionar el botón GUARDAR en la sesión de acciones.
6. Luego aparecerá una ventana confirmando que se ha realizado la inserción del nuevo fondo de pensión de manera satisfactoria, como muestra la siguiente imagen.



**Figura 41.
Confirmación de
ingreso de Afp**

ACTUALIZAR FONDO DE PENSION

A través de esta opción podrá actualizar la información correspondiente a los fondos de pensión registrados.

1. Presionar el botón REGISTRAR
2. Ir a la opción AFP
3. En el siguiente menú seleccionar la opción LISTAR AFP. Como se muestra en la siguiente imagen.



Figura 42. Acceder a listado de Afp

4. El siguiente formulario muestra todos los fondos de pensión registrados.

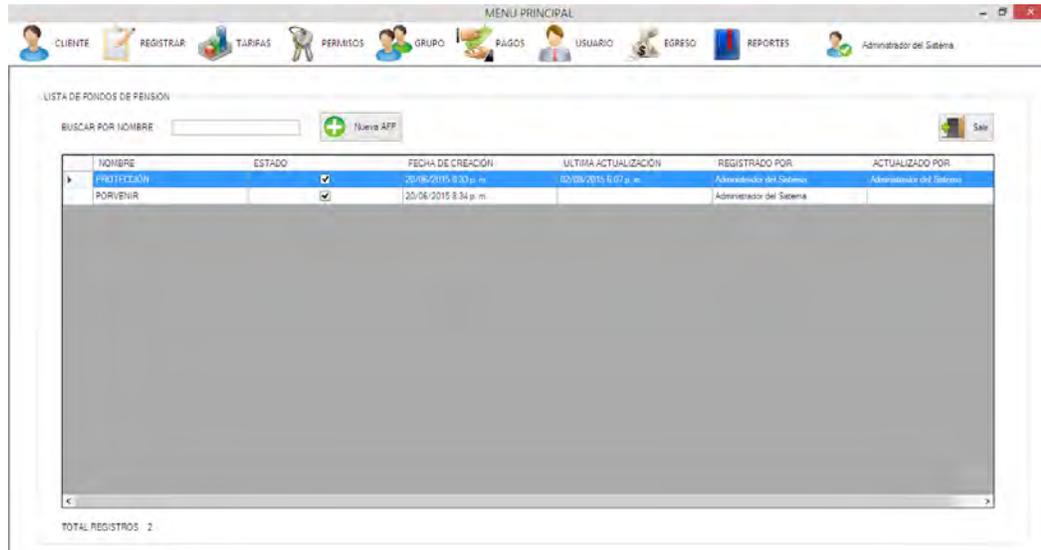


Figura 43. Lista de Afp

5. Dar doble clic sobre el registro que se desea actualizar.
6. En el formulario que se muestra se deberá ingresar la nueva información.

Figura 44. Actualizar Afp

7. Presionar el boton ACTUALIZAR en la sesión de acciones.
8. Aparecerá una ventana confirmando que se ha realizado la actualización del fondo de pensión de manera satisfactoria.

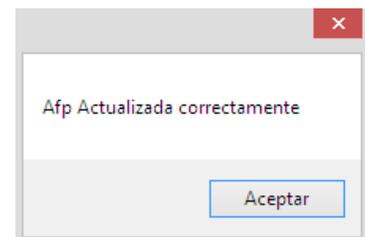


Figura 45. Confirmación de actualización de Afp

GESTIONAR EPS

REGISTRAR UNA NUEVA EPS

A través de esta opción podrá registrar una nueva EPS

1. Presionar el botón REGISTRAR
2. Ir a la opción EPS
3. En el siguiente menú seleccionar la opción NUEVA EPS. Como se muestra en la siguiente imagen.



Figura 46. Acceder al formulario nueva Eps

4. Aparecerá un formulario en el cual deberá ingresar el nombre de la EPS que se desea agregar.

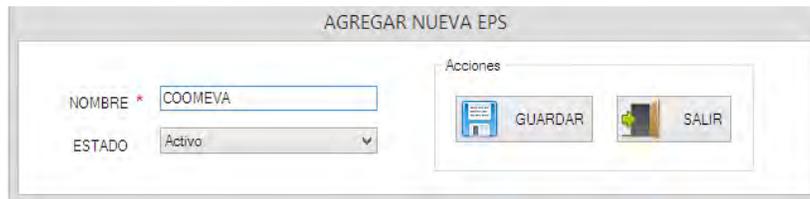


Figura 47. Agregar nueva Eps

5. Presionar el botón GUARDAR en la sesión de acciones.
6. Luego aparecerá una ventana confirmando que se ha realizado la inserción de la nueva EPS de manera satisfactoria, como muestra la siguiente imagen.

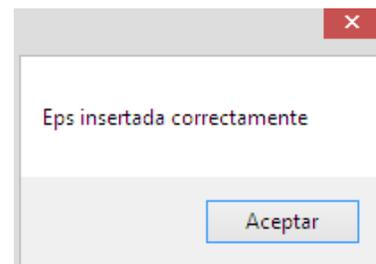


Figura 48. Confirmación de ingreso de Eps

ACTUALIZAR EPS

A través de esta opción podrá actualizar la información correspondiente a las EPS registradas.

1. Presionar el botón REGISTRAR
2. Ir a la opción EPS
3. En el siguiente menú seleccionar la opción LISTAR EPS. Como se muestra en la siguiente imagen.



Figura 49. Acceder a listado de Eps

4. El siguiente formulario muestra todas las EPS registradas.

The screenshot shows the 'LISTA DE EPS' form. It includes a search bar 'BUSCAR POR NOMBRE', a '+ Nueva Eps' button, and a 'Salir' button. The table below lists the registered EPS:

NOMBRE	ESTADO	FECHA DE CREACIÓN	ULTIMA ACTUALIZACIÓN	REGISTRADO POR	ACTUALIZADO POR
NUEVA EPS	<input checked="" type="checkbox"/>	20/06/2015 8:34 p. m.		Administrador del Sistema	
COOMEVA	<input checked="" type="checkbox"/>	20/06/2015 8:34 p. m.		Administrador del Sistema	
SALUDCOOP	<input checked="" type="checkbox"/>	20/06/2015 8:38 p. m.	02/08/2015 7:15 a. m.	Administrador del Sistema	Administrador del Sistema

TOTAL REGISTROS: 3

Figura 50. Lista de Eps

5. Dar doble clic sobre el registro que se desea actualizar.
6. En el formulario que se muestra se deberá ingresar la nueva información.



Figura 51. Actualizar Eps

7. Presionar el boton ACTUALIZAR en la sesión de acciones.
8. Aparecerá una ventana confirmando que se ha realizado la actualización de la EPS de manera satisfactoria.

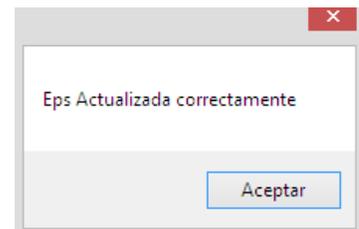


Figura 52. Confirmación de actualización de Eps

GESTIONAR TARIFAS DE RIESGO

REGISTRAR UNA NUEVA TARIFA DE RIESGO

A través de esta opción podrá registrar las tarifas de riesgos profesionales.

1. Presionar el botón TARIFAS
2. Ir a la opción TARIFAS DE RIESGO
3. El siguiente formulario muestra un listado de las tarifas de riesgo registradas en el sistema. Presionar el botón Nueva Tarifa

MENU PRINCIPAL

CLIENTE REGISTRAR TARIFAS PERMISOS GRUPO PAGOS USUARIO EGRESO REPORTES Administrador del Sistema

LISTA DE TARIFAS

Nueva Tarifa Salir

NIVEL DE RIESGO	VALOR SALUD	VALOR RIESGO	VALOR PENSION	VALOR APORTE	TOTAL	FECHA DE CREACIÓN	ULTIMA ACTUALIZACIÓN	REGISTRADO POR	ACTUALIZADO POR
1	81000	6700	103000	19000	207700	20/06/2015 8:43 p. m.	02/08/2015 7:40 p. m.	Administrador del Sist...	Administrador del Sist...
2	73800	8200	103000	18000	211000	22/06/2015 9:50 p. m.		Administrador del Sist...	
3	73000	21000	103000	19000	216000	22/06/2015 9:51 p. m.		Administrador del Sist...	
4	66000	30000	103000	19000	218000	22/06/2015 9:56 p. m.		Administrador del Sist...	

TOTAL REGISTROS: 4

Figura 53. Lista de tarifas de riesgo

- En el siguiente formulario deberá ingresar la información correspondiente a la nueva tarifa de riesgo.

AGREGAR TARIFA

INSERTAR TARIFA

NIVEL DE RIESGO * VALOR PENSION *

VALOR SALUD * VALOR APORTE *

VALOR RIESGO * TOTAL

Acciones

GUARDAR LIMPIAR SALIR

Figura 54. Agregar nueva tarifa de riesgo

Presionar el botón GUARDAR en la sesión de acciones.

- Luego aparecerá una ventana confirmando que se ha realizado la inserción de la nueva tarifa de riesgo de manera satisfactoria, como muestra la siguiente imagen. ACTUALIZAR TARIFA DE

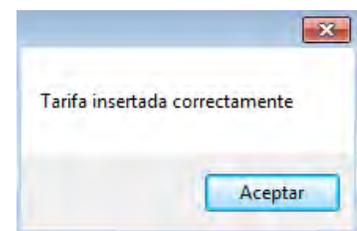


Figura 55. Confirmación de ingreso de tarifa de riesgo

RIESGO

A través de esta opción podrá actualizar la información correspondiente a las tarifas de riesgo registradas en el sistema.

1. Presionar el botón TARIFAS
2. Ir a la opción TARIFAS DE RIESGO
3. El siguiente formulario muestra un listado de las tarifas de riesgo registradas en el sistema. Presionar doble clic sobre el registro que se desea actualizar.
4. En el formulario que se muestra deberá ingresar la nueva información.



ACTUALIZAR TARIFAS DE RIESGOS

ACTUALIZAR TARIFA

NIVEL DE RIESGO *	2	VALOR PENSION *	103000
VALOR SALUD *	79800	VALOR APORTE *	19000
VALOR RIESGO *	9200	TOTAL	211000

Acciones

ACTUALIZAR LIMPIAR SALIR

Figura 56. Actualizar tarifa de riesgo

5. Presionar el botón ACTUALIZAR en la sesión de acciones.
6. Aparecerá una ventana confirmando que se ha realizado la actualización de manera satisfactoria.

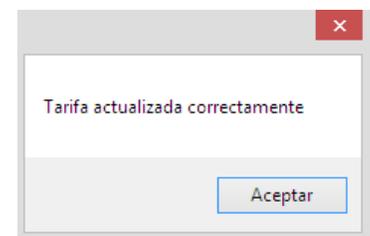


Figura 57. Confirmación de actualización de tarifa de riesgo

GESTIONAR TARIFAS DE PLANILLAS

REGISTRAR UNA NUEVA TARIFA DE PLANILLA

A través de esta opción podrá registrar las tarifas de las planillas.

1. Presionar el botón TARIFAS
2. Ir a la opción TARIFAS DE PLANILLA

3. El siguiente formulario muestra un listado de las tarifas de las planillas registradas en el sistema. Presionar el botón Nueva Tarifa de Planilla

DESCRIPCION	VALOR	FECHA DE CREACION	ULTIMA ACTUALIZACION	REGISTRADO POR	ACTUALIZADO POR
Descargar planilla de ingresos	4000	21/06/2015 12:12 p. m.	21/06/2015 12:16 p. m.	Administrador del Sistema	Administrador del Sistema
Descargar planilla y realizar pago en el...	12000	21/06/2015 5:24 a. m.		Administrador del Sistema	
Planilla de internet + intereses	15000	20/07/2015 1:05 a. m.		Administrador del Sistema	

Figura 58. Lista de tarifas de planilla

4. En el siguiente formulario deberá ingresar la información correspondiente a la nueva tarifa de planilla.

AGREGAR TARIFA DE PLANILLA

INSERTAR TARIFA DE PLANILLA

DESCRIPCION * Planilla mas interes

VALOR * 12000

Acciones

GUARDAR LIMPIAR SALIR

Figura 59. Registrar nueva tarifa de planilla

5. Presionar el botón GUARDAR en la sesión de acciones.
6. Luego aparecerá una ventana confirmando que se ha realizado la inserción de la nueva

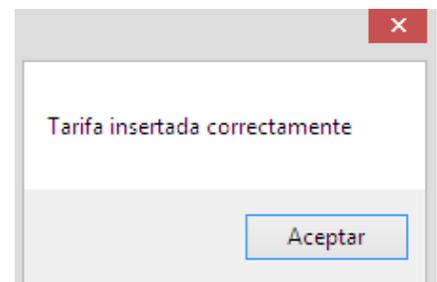


Figura 60. Confirmación de ingreso de tarifa de planilla

tarifa de planilla de manera satisfactoria, como muestra la siguiente imagen.

ACTUALIZAR TARIFA DE PLANILLA

A través de esta opción podrá actualizar la información correspondiente a las tarifas de planillas registradas en el sistema.

1. Presionar el botón TARIFAS
2. Ir a la opción TARIFAS DE PLANILLA
3. El siguiente formulario muestra un listado de las tarifas de planillas registradas en el sistema. Presionar doble clic sobre el registro que se desea actualizar.
4. En el formulario que se muestra deberá ingresar la nueva información.



Figura 61. Actualizar tarifa de planilla

5. Presionar el botón ACTUALIZAR en la sesión de acciones.
6. Aparecerá una ventana confirmando que se ha realizado la actualización de manera satisfactoria.

GESTIONAR PERMISOS

A través de esta opción podrá listar, modificar o crear un nuevo nivel de acceso en el sistema.

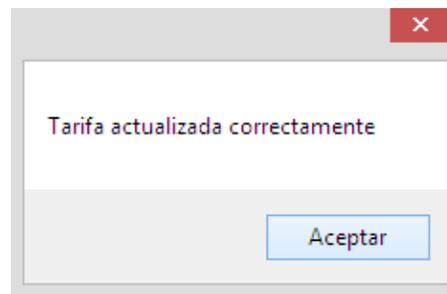


Figura 62. Confirmación de actualización de tarifa de planilla

LISTAR PERMISOS

1. Presionar el botón PERMISOS.
2. Ir a la opción LISTAR PERMISOS
3. El siguiente formulario muestra los permisos y la descripción registrados en el sistema

NOMBRE	DESCRIPCION	FECHA DE CREACION	ULTIMA ACTUALIZACION
Clientes	Acceso a todas las opciones de clientes	05/07/2015 10:09 a. m.	05/07/2015 10:17 a. m.
Registrar informacion	Acceso a registro de informacion correspondiente a ciudad	05/07/2015 10:10 a. m.	05/07/2015 10:19 a. m.
Tarifas	Acceso al menu de tarifas para registrar y actualizar	05/07/2015 10:15 a. m.	
Pagos	Acceso a las opciones de pagos	05/07/2015 10:20 a. m.	
Usuarios	Acceso a la gestion de usuarios. Agregar y actualizar	05/07/2015 10:20 a. m.	
Egresos	Acceso a las opciones de egreso	05/07/2015 10:21 a. m.	
Reportes	Acceso a los reportes	05/07/2015 10:21 a. m.	

TOTAL REGISTROS 7

Figura 63. Lista de permisos

ACTUALIZAR PERMISOS

1. Ubicarse en el formulario de LISTAR PERMISOS
2. Dar doble clic sobre el registro que se desea actualizar su información
3. En el siguiente formulario registrar la nueva información.

Figura 64. Actualizar permiso

4. Presionar el boton ACTUALIZAR en la sesión de acciones.
5. Aparecerá una ventana confirmando que se ha realizado la actualización de manera satisfactoria.

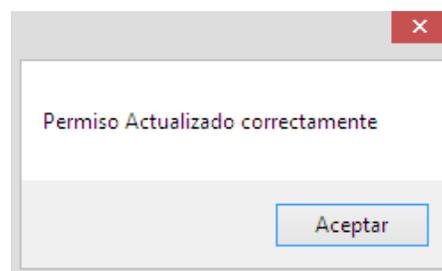


Figura 65. Confirmación de actualización de permiso

REGISTRAR UN NUEVO PERMISO

1. Ubicarse en el formulario de LISTAR PERMISOS
2. Presionar el botón Nuevo Permiso
3. En el siguiente formulario registrar la información correspondiente al nuevo permiso.

Figura 66. Registrar nuevo permiso

4. Presionar el boton **ACEPTAR** en la sesión de acciones.
5. Aparecerá una ventana confirmando que se ha realizado la inserción de manera satisfactoria.

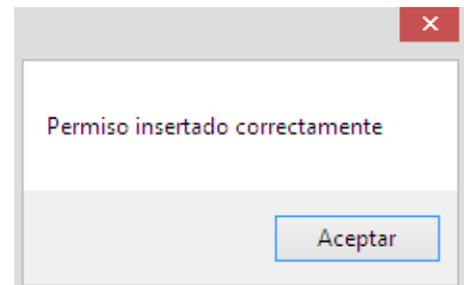


Figura 67. Confirmación de inserción de permiso

GESTIONAR GRUPOS

A través de esta opción podrá listar, modificar o crear un nuevo grupo en el sistema.

LISTAR GRUPOS

4. Presionar el botón **GRUPOS**.
5. Ir a la opción **LISTAR GRUPOS**
6. El siguiente formulario muestra los grupos registrados en el sistema.

CLIENTE REGISTRAR TARIFAS PERMISOS GRUPO PAGOS USUARIO EGRESO REPORTE Administrador del Sistema

MENÚ PRINCIPAL

LISTA DE GRUPOS

Buscar

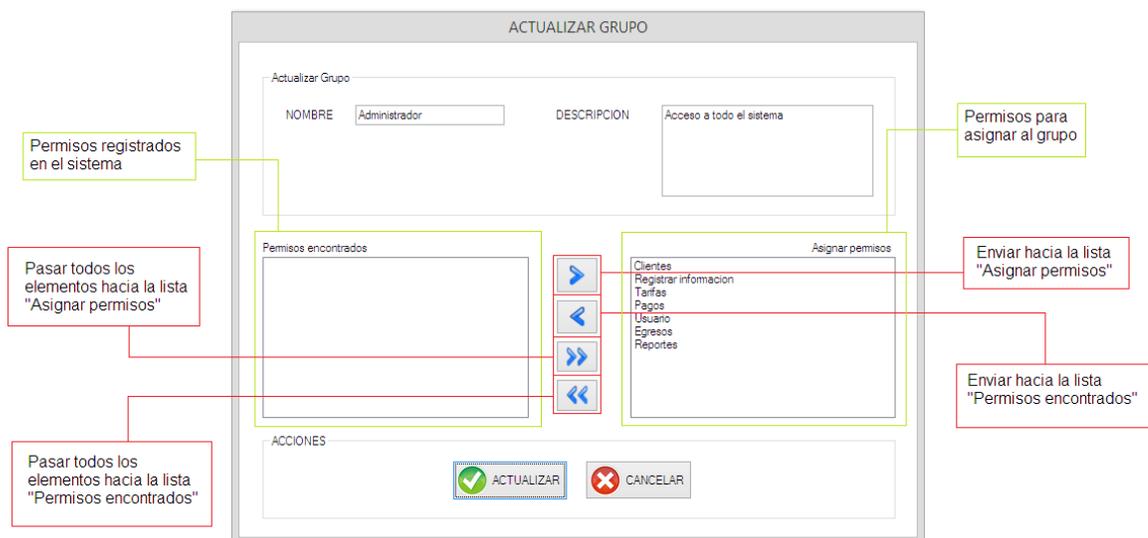
NOMBRE	DESCRIPCIÓN	FECHA DE CREACIÓN	ÚLTIMA ACTUALIZACIÓN
Administrador	Acceso a todo el sistema	20/06/2015 8:20 a. m.	02/08/2015 1:58 p. m.
Gerente	Acceso a biblioteca	05/07/2015 10:20 a. m.	05/07/2015 9:00 p. m.

TOTAL REGISTROS: 2

Figura 68. Lista de grupos

ACTUALIZAR GRUPOS

1. Ubicarse en el formulario de LISTAR GRUPOS
2. Dar doble clic sobre el registro que se desea actualizar su información
3. En el siguiente formulario podrá observar los permisos que tiene asignado ese grupo. Además podrá agregar o quitar permisos.



4. Presionar el boton ACTUALIZAR en la sesión de acciones.
5. Aparecerá una ventana confirmando que se ha realizado la actualización de manera satisfactoria.

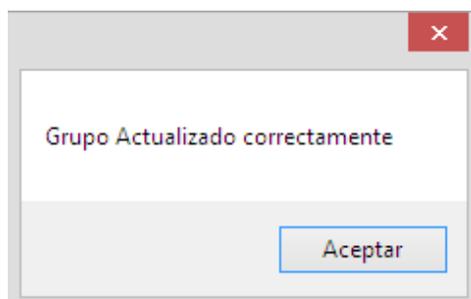


Figura 69. Confirmación de actualización de grupo

REGISTRAR UN NUEVO GRUPO

1. Ubicarse en el formulario de LISTAR GRUPOS
2. Presionar el botón Nuevo Grupo

3. En el siguiente formulario registrar la información correspondiente al nuevo grupo. Deberá además agregar los permisos que considere necesario para el nuevo grupo.

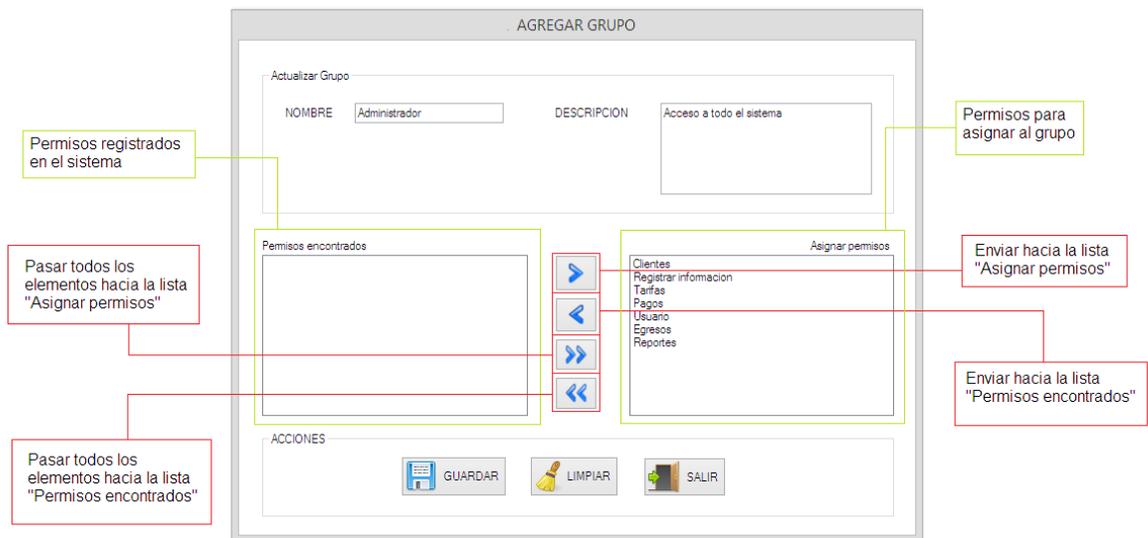


Figura 70. Registrar nuevo grupo

4. Presionar el boton GUARDAR en la sesión de acciones.
5. Aparecerá una ventana confirmando que se ha realizado la inserción de manera satisfactoria.

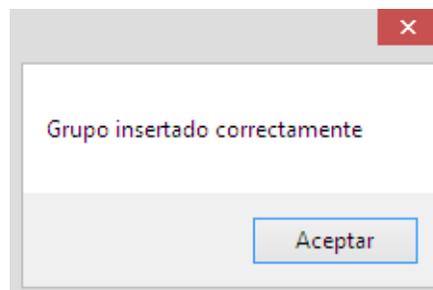


Figura 71. Confirmación de registro de grupo

A través de este módulo podrá registrar los pagos de los clientes tipo dependiente e independiente.

REGISTRAR PAGO CLIENTE DEPENDIENTE

1. Ir a la opción PAGOS
2. Presionar el botón REGISTRAR NUEVO PAGO
3. En el submenú ir a la opción CLIENTE DEPENDIENTE



Figura 72. Ir a formulario registrar pago cliente dependiente

4. El siguiente formulario muestra la información que deberá registrar para realizar el pago del cliente de manera satisfactoria.

The screenshot shows a web form titled "REGISTRAR NUEVO PAGO". It is divided into several sections:

- Seleccionar Cliente:** Contains two input fields labeled "CEDULA" and "NOMBRE". To the right is a "Nuevo Cliente" button with a green plus icon.
- Otras Opciones:** Contains two checkboxes: "15 Dias" and "Cuentas x cobrar".
- Payment Values:** Four input fields labeled "VALOR SALUD", "VALOR RIESGO", "VALOR PENSION", and "VALOR APORTE".
- Totals:** Two input fields labeled "TOTAL RECIBIDO" and "TOTAL A DEVOLVER".
- Acciones:** Two buttons: "GUARDAR PAGO" (with a save icon) and "CANCELAR" (with a red X icon).
- Total:** A section with the label "Total:" and a large empty input field.

Figura 73. Registrar nuevo pago. Cliente dependiente

5. En la opción seleccionar cliente, podrá buscar aquellos clientes dependientes que se encuentren registrados. El sistema permitirá buscar por cedula o nombre de cliente.

REGISTRAR NUEVO PAGO

Seleccionar Cliente

CEDULA
1
109723 (pedro perez)
1234 (francisco solis)
18989 (Maria Quiñones)

NOMBRE Nuevo Cliente 

Otras Opciones
 15 Dias
 Cuentas x cobrar

VALOR SALUD VALOR RIESGO
VALOR PENSION VALOR APORTE

TOTAL RECIBIDO **TOTAL A DEVOLVER**

Acciones

 GUARDAR PAGO  CANCELAR

Total

TOTAL:

Buscar cliente por cédula

Figura 74. Buscar cliente dependiente por cedula

REGISTRAR NUEVO PAGO

Buscar cliente por nombre

Seleccionar Cliente

CEDULA

NOMBRE
francisco solis (1234)

Nuevo Cliente

Otras Opciones

15 Dias

Cuentas x cobrar

VALOR SALUD

VALOR RIESGO

VALOR PENSION

VALOR APORTE

TOTAL RECIBIDO

TOTAL A DEVOLVER

Acciones

GUARDAR PAGO CANCELAR

Total

TOTAL:

Figura 75. Buscar cliente dependiente por nombre

- Una vez se ha seleccionado el cliente, el sistema se encargará de traer la información correspondiente a VALOR SALUD, VALOR PENSION, VALOR RIESGO Y VALOR APORTE. Esta información la obtendrá basado en el nivel de riesgo que tiene asociado el cliente y en los valores de las tarifas registrados en el módulo de TARIFAS -> TARIFAS DE RIESGO. Y en caso que el cliente no tenga asociado alguna entidad, entonces el valor correspondiente será cero.

The screenshot shows a web form titled "REGISTRAR NUEVO PAGO". It contains several sections:

- Selección de Cliente:** Fields for "CEDULA" (containing "1234 (francisco solis)") and "NOMBRE". A "Nuevo Cliente" button with a plus sign is also present.
- Otras Opciones:** Two checkboxes: "15 Dias" and "Cuentas x cobrar".
- Valores:** Four input fields: "VALOR SALUD" (81000), "VALOR RIESGO" (6700), "VALOR PENSION" (0), and "VALOR APOORTE" (19000).
- Totales:** Two input fields: "TOTAL RECIBIDO" and "TOTAL A DEVOLVER".
- Acciones:** "GUARDAR PAGO" and "CANCELAR" buttons.
- Total:** A display area showing "TOTAL: 106700".

Callouts provide the following information:

- "Si no existe el cliente, a través de esta opción podrá registrarlo" (pointing to the "Nuevo Cliente" button).
- "Si esta opción esta activada, entonces el cliente estará vinculado por solo 15 días" (pointing to the "15 Dias" checkbox).
- "Si esta opción esta activada entonces el valor total pagar será enviado a las cuentas por cobrar" (pointing to the "Cuentas x cobrar" checkbox).
- "Ingresar el dinero recibido por el cliente" (pointing to the "TOTAL RECIBIDO" field).
- "El sistema calculará el total a devolver (total recibido - total)" (pointing to the "TOTAL A DEVOLVER" field).

Figura 76. Opciones de pago dependiente

7. El siguiente campo TOTAL RECIBIDO, representa el dinero físico que se recibe del cliente, posteriormente el sistema calculará y reportará el TOTAL A DEVOLVER.

REGISTRAR NUEVO PAGO

Seleccionar Cliente

CEDULA: 1234 francisco solis

NOMBRE: []

Nuevo Cliente +

Otras Opciones

15 Dias

Cuentas x cobrar

VALOR SALUD: 81000

VALOR RIESGO: 6700

VALOR PENSION: 0

VALOR APORTE: 19000

TOTAL RECIBIDO: 120000

TOTAL A DEVOLVER: 13300

Acciones

GUARDAR PAGO

CANCELAR

Total

TOTAL: 106700

Figura 77. Ingresar valor recibido

- Una vez se ha registrado toda la información necesaria entonces se presiona el botón GUARDAR PAGO en la sesión de acciones. El sistema emitirá un mensaje reportando que el pago se ha registrado de manera satisfactoria.

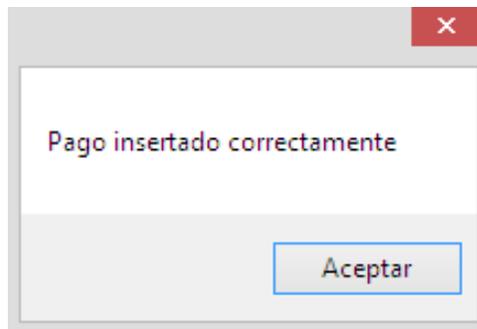


Figura 78. Confirmación de pago dependiente

- Posteriormente el sistema solicitará confirmación para imprimir el recibo de pago. Presionar Aceptar para imprimir el recibo de pago.

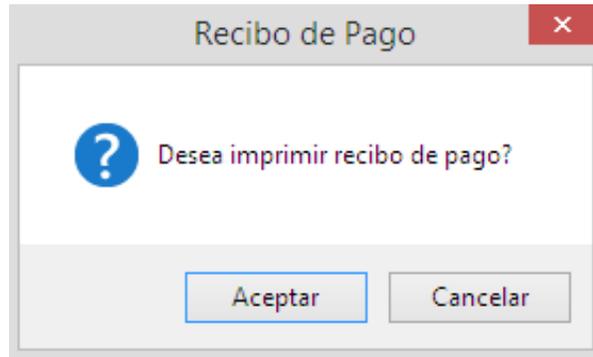


Figura 79. Confirmar impresión de recibo de pago dependiente

10.A continuación se muestra el recibo de pago generado para el cliente. El cual tiene en detalle el valor de su pago y los servicios por lo que ha pagado.

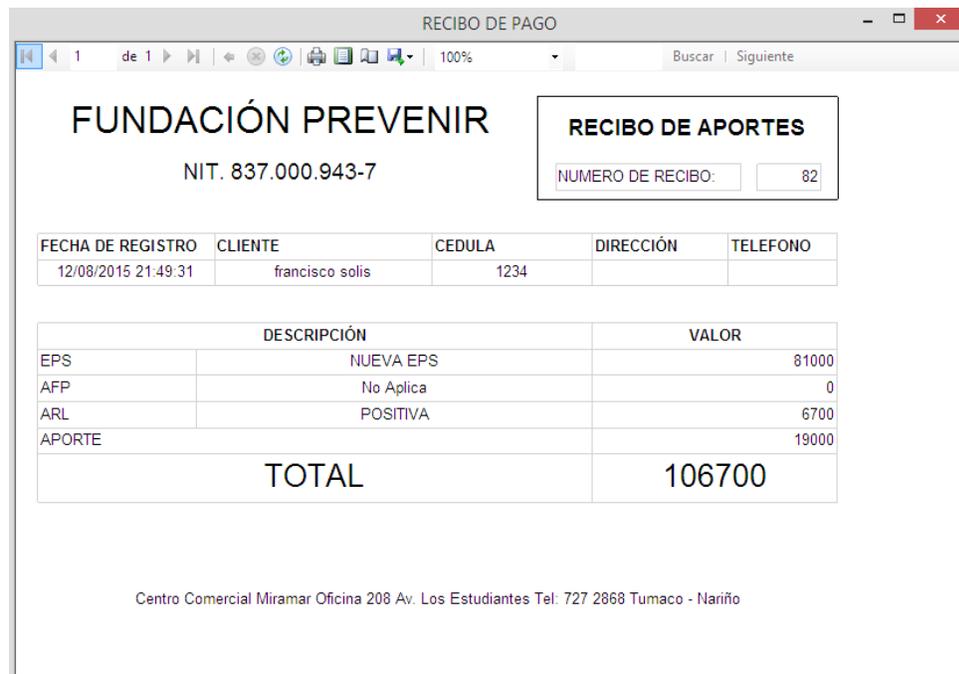


Figura 80. Recibo de pago dependiente

11. Para enviar a imprimir el recibo de pago generado, dar clic en la opción imprimir. Como muestra la siguiente imagen:



Figura 81. Recibo de pago (Dependiente). Opción imprimir

12. Se mostrará un formulario en el cual seleccionará la impresora que se encuentra configurada en el equipo y posteriormente dará clic en la opción Imprimir.

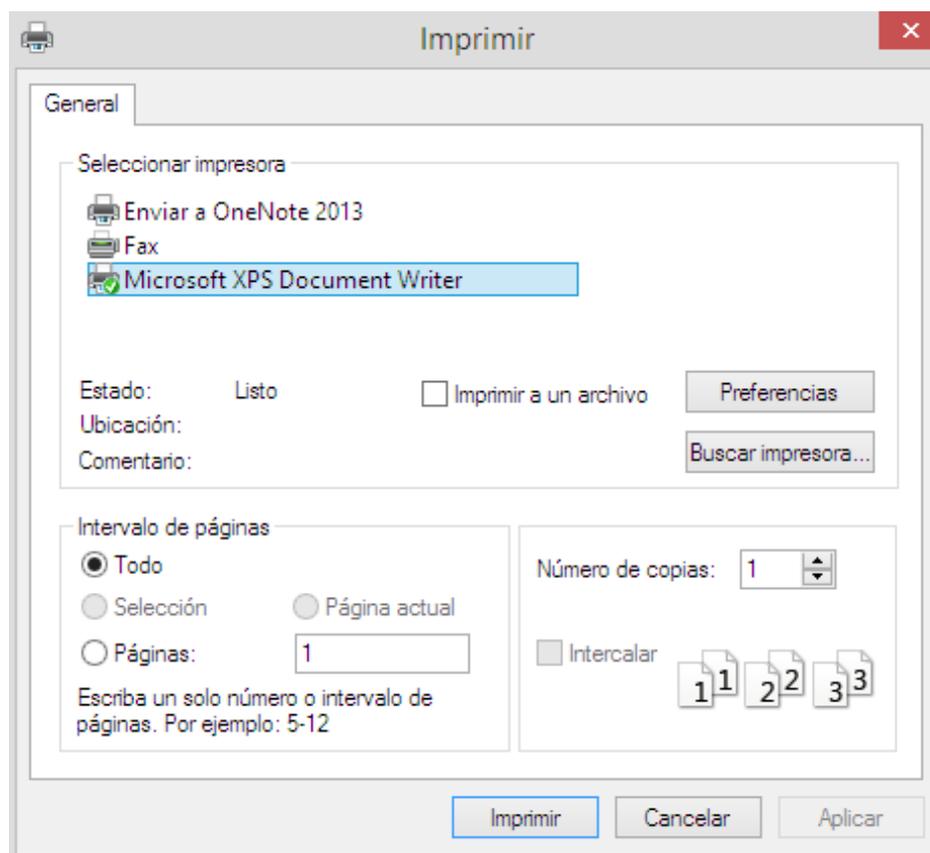


Figura 82. Formulario de impresión

REGISTRAR PAGO CLIENTE INDEPENDIENTE

1. Ir a la opción PAGOS
2. Presionar el botón REGISTRAR NUEVO PAGO
3. En el submenú ir a la opción CLIENTE INDEPENDIENTE



Figura 83. Ir a formulario registrar pago cliente dependiente

4. El siguiente formulario muestra la información que deberá registrar para realizar el pago del cliente de manera satisfactoria.

REGISTRAR PAGO

Seleccionar Cliente

<input type="text" value="CEDULA"/>	<input type="text" value="NOMBRE"/>	<input type="button" value="Nuevo Cliente +"/>
-------------------------------------	-------------------------------------	--

Seleccionar Banco

BANCO <input type="text" value="BANCOLOMBIA"/>	<input type="button" value="+"/>
--	----------------------------------

Valor

VALOR SALUD <input type="text"/>	PLANILLA <input type="text"/>
VALOR PENSION <input type="text"/>	VALOR A PLANILLA <input type="text"/>
VALOR RIESGO <input type="text"/>	

TOTAL RECIBIDO <input type="text"/>	TOTAL A DEVOLVER <input type="text"/>
-------------------------------------	---------------------------------------

<p>Acciones</p> <div style="display: flex; justify-content: space-around;"> <input type="button" value="GUARDAR PAGO"/> <input type="button" value="CANCELAR"/> </div>	<p>Total</p> <p>TOTAL:</p>
--	----------------------------

Figura 84. Registrar nuevo pago. Cliente independiente

5. En la opción seleccionar cliente, podrá buscar aquellos clientes dependientes que se encuentren registrados. El sistema permitirá buscar por cedula o nombre de cliente.

The screenshot shows the 'REGISTRAR PAGO' form. In the 'Seleccionar Cliente' section, the 'CEDULA' dropdown menu is open, displaying a search result: '109723 (pedro perez)'. A red box highlights this dropdown, and a red arrow points to it from a label 'Buscar cliente por cedula'. Other fields include 'NOMBRE', 'Banco' (set to 'BANCOLOMBIA'), and various value input fields for 'VALOR SALUD', 'VALOR PENSION', 'VALOR RIESGO', 'PLANILLA', and 'VALOR A PLANILLA'. At the bottom, there are buttons for 'GUARDAR PAGO' and 'CANCELAR', and a 'Total' section with a 'TOTAL:' label.

Figura 85. Buscar cliente independiente por cedula

The screenshot shows the 'REGISTRAR PAGO' form. In the 'Seleccionar Cliente' section, the 'NOMBRE' dropdown menu is open, displaying a search result: 'pedro perez (109723)'. A red box highlights this dropdown, and a red arrow points to it from a label 'Buscar cliente por nombre'. Other fields include 'CEDULA', 'Banco' (set to 'BANCOLOMBIA'), and various value input fields for 'VALOR SALUD', 'VALOR PENSION', 'VALOR RIESGO', 'PLANILLA', and 'VALOR A PLANILLA'. At the bottom, there are buttons for 'GUARDAR PAGO' and 'CANCELAR', and a 'Total' section with a 'TOTAL:' label.

Figura 86. Buscar cliente por nombre

- Una vez se ha seleccionado el cliente, el sistema se encargará de traer la información correspondiente a VALOR SALUD, VALOR PENSION y VALOR RIESGO. Esta información la obtendrá basado en el nivel de riesgo que tiene asociado el cliente y en los valores de las tarifas registrados en el módulo de TARIFAS -> TARIFAS DE RIESGO. Y en caso que el cliente no tenga asociado alguna entidad, entonces el valor correspondiente será cero.

The screenshot shows a web form titled "REGISTRAR PAGO". It is divided into several sections:

- Seleccionar Cliente:** Includes fields for "CEDULA" (with value "109723 (pedro perez)") and "NOMBRE". A "Nuevo Cliente +" button is highlighted with a red box and labeled "Registrar un nuevo cliente".
- Seleccionar Banco:** Includes a "BANCO" dropdown menu (with value "BANCOLOMBIA") and a "+" button. A red box highlights this section with the label "Registrar un nuevo banco".
- Valor:** Includes input fields for "VALOR SALUD" (81000), "VALOR PENSION" (103000), and "VALOR RIESGO" (6700). It also has a "PLANILLA" dropdown (value "Descargar planilla de internet") and a "VALOR A PLANILLA" field (6000).
- TOTAL RECIBIDO:** A text input field highlighted with a red box and labeled "Registrar el dinero recibido por el cliente".
- TOTAL A DEVOLVER:** A text input field highlighted with a red box and labeled "El sistema calculará el total a devolver (total recibido - total)".
- Acciones:** Includes "GUARDAR PAGO" and "CANCELAR" buttons.
- Total:** A box showing "TOTAL: 196700".

Figura 87. Opciones de pago independiente

- El siguiente campo TOTAL RECIBIDO, representa el dinero físico que se recibe del cliente, posteriormente el sistema calculará y reportará el TOTAL A DEVOLVER.

Figura 88. Ingresar valor recibido en pago independiente

8. Una vez se ha registrado toda la información necesaria entonces se presiona el botón GUARDAR PAGO en la sesión de acciones. El sistema emitirá un mensaje reportando que el pago se ha registrado de manera satisfactoria.

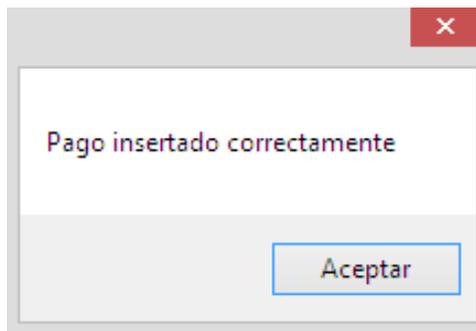


Figura 89. Confirmación de pago independiente

9. Posteriormente el sistema solicitará confirmación para imprimir el recibo de pago. Presionar Aceptar para imprimir el recibo de pago.

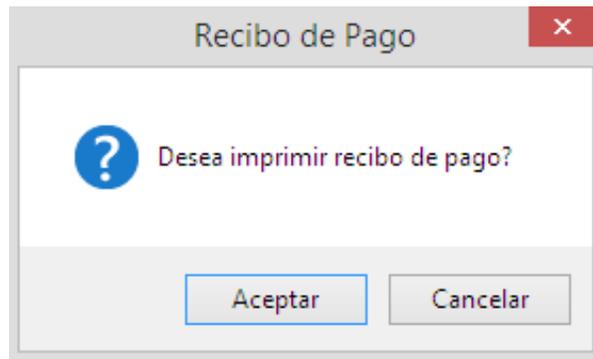


Figura 90. Confirmar impresión de recibo de pago independiente

10. A continuación se muestra el recibo de pago generado para el cliente. El cual tiene en detalle el valor de su pago y los servicios por lo que ha pagado.

FUNDACIÓN PREVENIR
NIT. 837.000.943-7

RECIBO DE APORTES

NUMERO DE RECIBO:

FECHA DE REGISTRO	CLIENTE	CEDULA	DIRECCIÓN	TELEFONO
13/08/2015 22:09:55	pedro perez	109723		

DESCRIPCIÓN	VALOR
EPS NUEVA EPS	81000
AFP PROTECCIÓN	103000
ARL POSITIVA	6700
PLANILLA	6000
TOTAL	196700

Centro Comercial Miramar Oficina 208 Av. Los Estudiantes Tel: 727 2868 Tumaco - Nariño

Figura 91. Recibo de pago independiente

11. Para enviar a imprimir el recibo de pago generado, dar clic en la opción imprimir. Como muestra la siguiente imagen:



Figura 92. Recibo de pago (Independiente). Opción imprimir

12. Se mostrará un formulario en el cual seleccionará la impresora que se encuentra configurada en el equipo y posteriormente dará clic en la opción Imprimir.

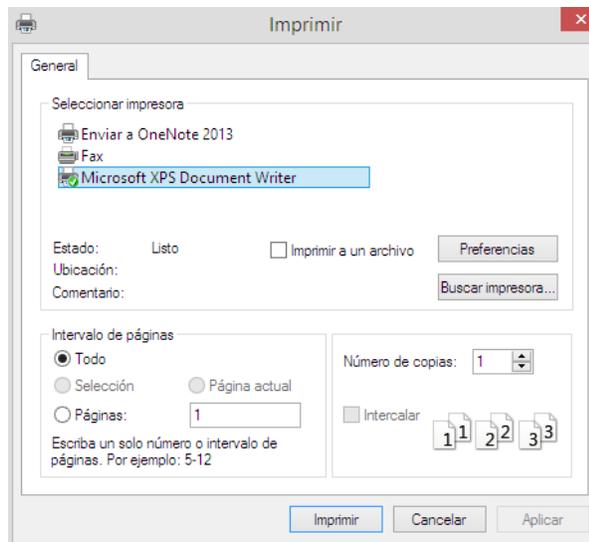


Figura 93. Formulario de impresión recibo independiente

VER PAGOS REGISTRADOS

A través de esta opción podrá observar todos los pagos registrados en el sistema. Observara además el cliente que realizo el pago y el valor total pagado.

1. Ir a opción PAGOS
2. Presionar el botón LISTAR PAGOS



Figura 94. Ir a opción Listar pagos

3. En el siguiente formulario podrá observar los pagos registrados.

ID_RECIBO	CLIENTE	SALUD	RIESGO	PENSION	APORTE	TOTAL PAGADO	FECHA DE REGISTRO	USUARIO
52	francesca solis	81000	8700	0	13000	104700	12/08/2015 9:43 p. m.	Administrador del S
53	pedro perez	81000	8700	100000	0	150700	13/08/2015 10:09 p. m.	Administrador de S

TOTAL REGISTROS: 2

Figura 95. Lista de pagos

CUADRE DE CAJA

Este módulo permite observar las entradas y salidas de dinero dentro de un rango de fechas definidas por el usuario.

1. Ir a la opción PAGOS
2. Presionar el botón CUADRE DE CAJA



Figura 96. Opción cuadro de caja

3. En el siguiente formulario deberá ingresar el rango de fechas en el cual desea obtener los ingresos y egresos registrados en el sistema. Posteriormente presionar el botón BUSCAR

CUADRE DE CAJA

DESDDE: sábado, 01 de agosto de 2015 HASTA: viernes, 14 de agosto de 2015 [BUSCAR] [Salir]

CLIENTE	VALDR PAGO
Eduardo ariz	106700
pedro perez	196700

CONCEPTO	VALDR PAGO	CLIENTE
Cuentas en liquidación	25000	

TOTAL INGRESOS: 303400 — Suma de ingresos

TOTAL EGRESOS: 25000 — Suma de egresos

SALDO EN CAJA: 278400 — Saldo en caja = suma de ingresos - suma de egresos

Figura 97. Detalle de cuadro de caja

CUENTAS POR COBRAR

A través de esta opción podrá observar en detalle aquellos pagos de clientes que se encuentran pendientes por cobrar.

1. Ir a opción PAGOS
2. Presionar el botón CUENTAS POR COBRAR



Figura 98. Ir a formulario cuentas por cobrar

3. En el siguiente formulario el sistema mostrará aquellos clientes que tienen pagos pendientes.

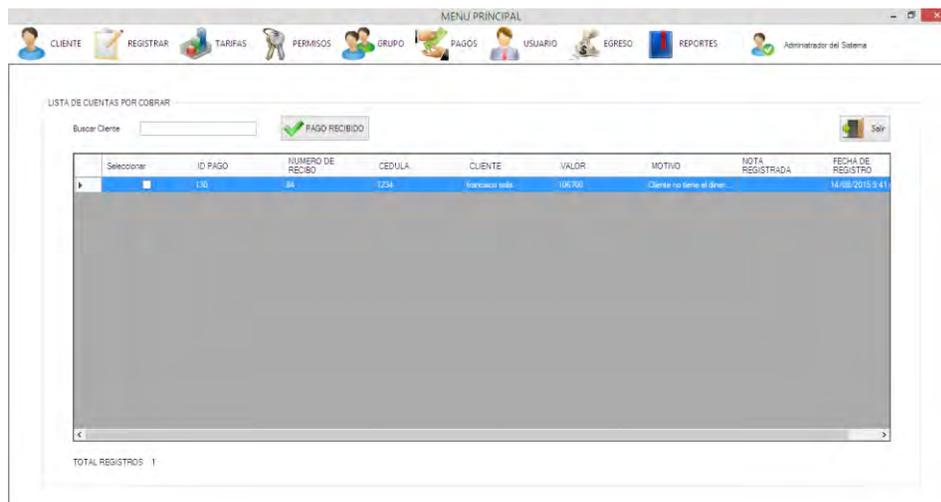
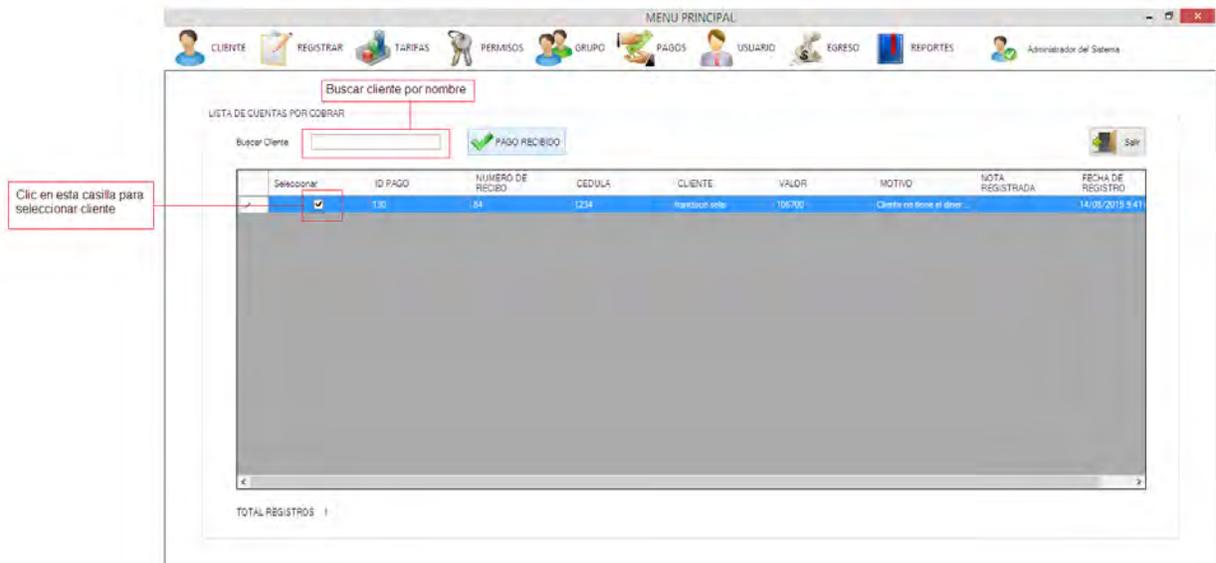


Figura 99. Lista de cuentas por cobrar

- Una vez se haya recibido el dinero por parte del cliente entonces dar clic en



la casilla del cliente, luego presionar el botón PAGO RECIBIDO

Figura 100. Opciones en formulario cuentas por cobrar

- Aparecerá un formulario solicitando confirmación del pago. Si todo esta correcto clic en Aceptar.

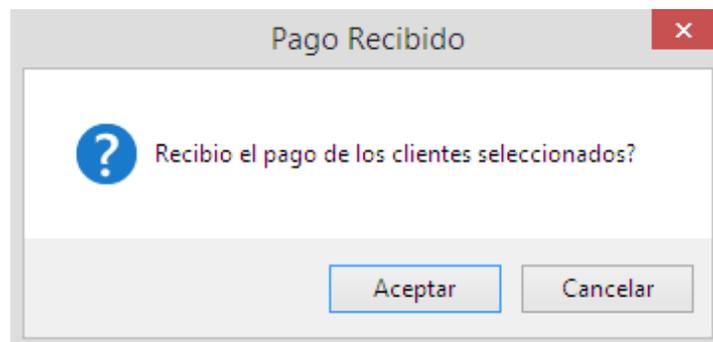
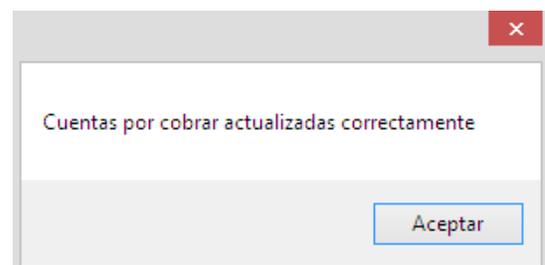


Figura 101. Confirmación de pago

- Si todo esta correcto entonces se mostrará un mensaje confirmación que se ha



actualizado las cuentas por cobrar correctamente.

PAGOS POR TRABAJADORES

Figura 102. Confirmación de actualización de cuentas por cobrar

A través de esta opción podrá ver en detalle aquellos clientes que se les realizará la consignación por los servicios en los que se encuentra vinculado. Estos clientes serán aquellos que han realizado su pago dentro de un rango de fechas seleccionadas.

1. Ir a opción PAGO
2. Seleccionar la opción PAGO POR TRABAJADORES



Figura 103. Ir a formulario Pago por trabajadores

3. Se mostrara el siguiente formulario

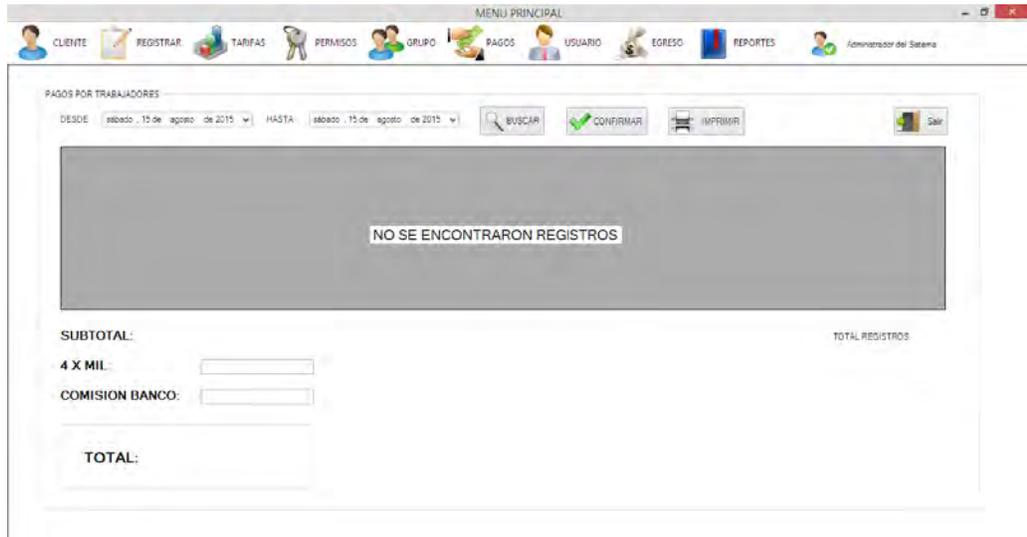


Figura 104. Pago por trabajadores

4. Deberá seleccionar un rango de fechas y clic en el botón BUSCAR. El sistema mostrará aquellos clientes que hayan realizado su pago dentro del rango de fechas seleccionado. El sistema calculara además el valor de 4 por mil basado en el subtotal (suma de los valores pagados por los clientes).

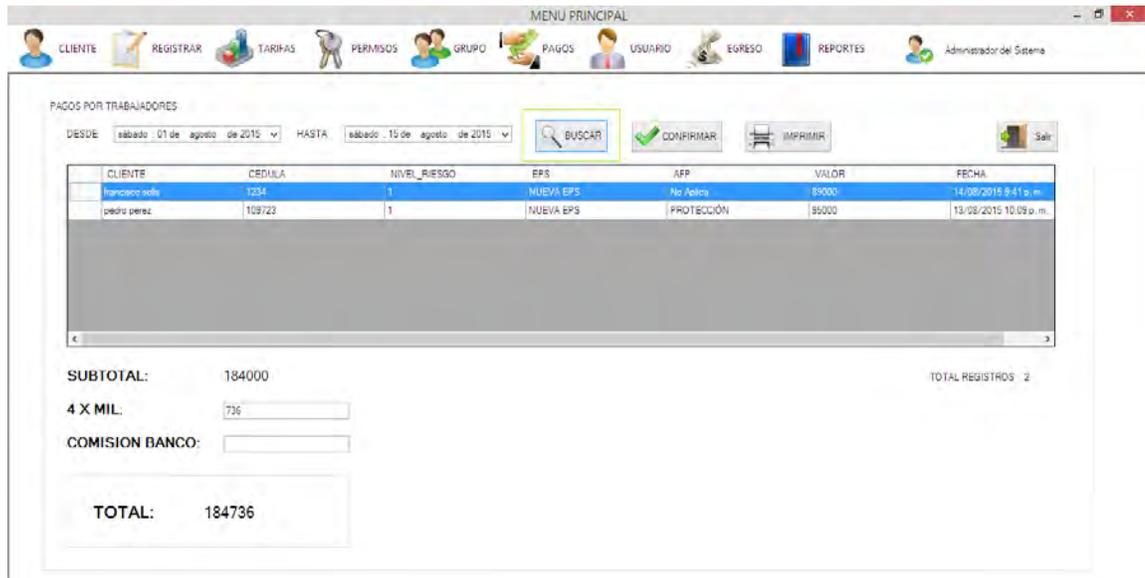


Figura 105. Clientes a realizar el pago

- El usuario deberá ingresar el valor de comisión del banco. Si toda la información es correcta entonces clic en el botón CONFIRMAR. El sistema mostrará un mensaje de confirmación indicando que el pago se registró en los egresos.

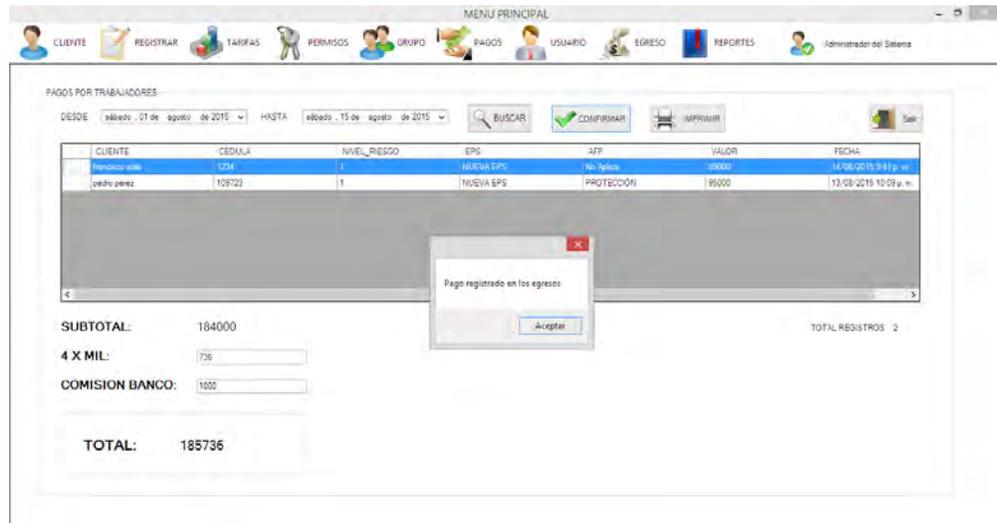


Figura 106. Pago por trabajadores registrado en egresos

- Si desea imprimir el reporte de clientes que se ha realizado el pago, entonces clic en el botón IMPRIMIR.

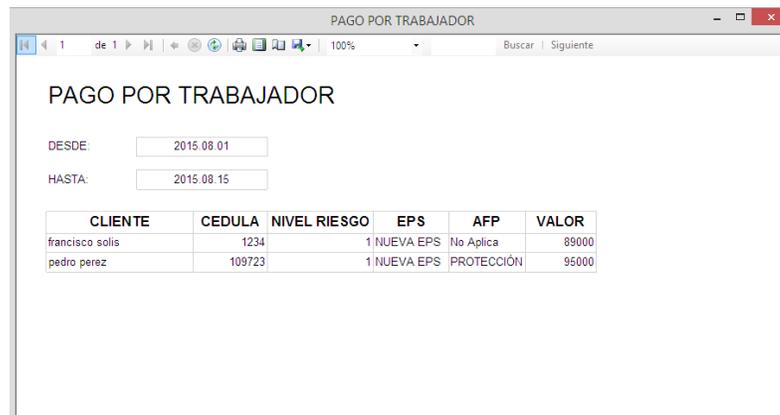


Figura 107. Reporte de pago por trabajador

REPORTE DE CLIENTES SIN PAGO

A través de este reporte podrá ver aquellos clientes que no han realizado su pago durante el último mes.

1. Ir a opción REPORTES
2. Seleccionar la opción CLIENTES SIN PAGO



Figura 108. Ir a reporte Cliente sin pago

3. Se mostrará un reporte de aquellos clientes que no han realizado su pago dentro del último mes.

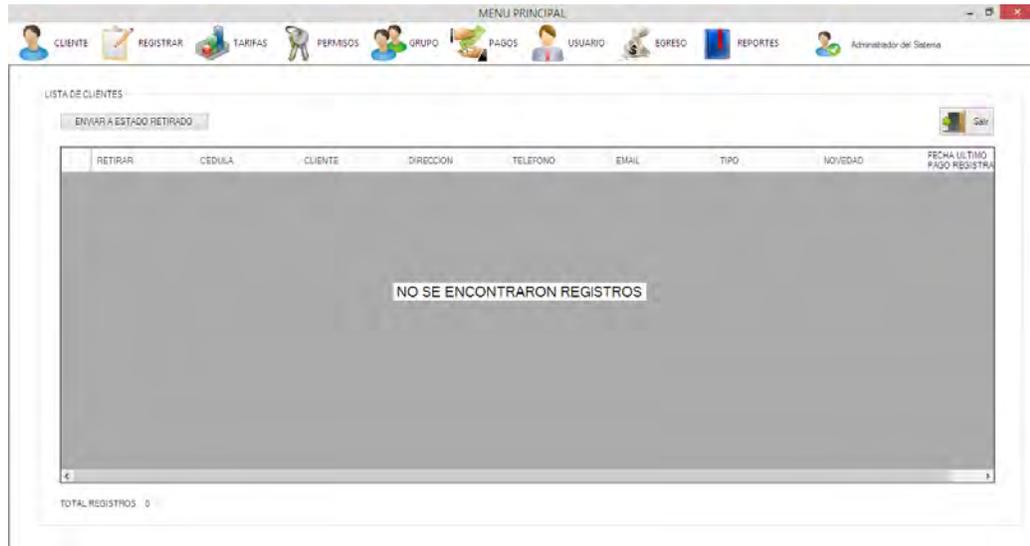


Figura 109. Reporte de clientes sin pago

4. Si el usuario lo considera necesario podrá enviar aquellos clientes sin pagar a estado de RETIRADO. Para ello, seleccionar el cliente y dar clic en el botón ENVIAR A ESTADO RETIRADO

REPORTE DE ÚLTIMO INGRESO POR CLIENTE

A través de este reporte podrá ver el último pago registrado por cada cliente.

1. Ir a opción REPORTES
2. Seleccionar la opción ULTIMO INGRESO POR CLIENTE



Figura 110. Ir a formulario ultimo ingreso

3. El siguiente formulario contiene la información correspondiente al último pago registrado por cada cliente.

CEDULA	CLIENTE	DIRECCION	TELEFONO	EMAIL	VALOR PAGO	FECHA ULTIMO PAGO REGISTRADO
10023	Francisco Solis				126700	13/05/2015 10:17 p. m.
1234	Francisco Solis				126700	14/05/2015 9:41 a. m.

TOTAL REGISTROS: 2

Figura 111. Reporte de último ingreso por cliente

REPORTE DE ÚLTIMO EGRESO POR CLIENTE

A través de esta opción podrá ver en detalle el último pago que se ha realizado para los clientes.

1. Ir a opción REPORTES
2. Seleccionar la opción ULTIMO EGRESO POR CLIENTE

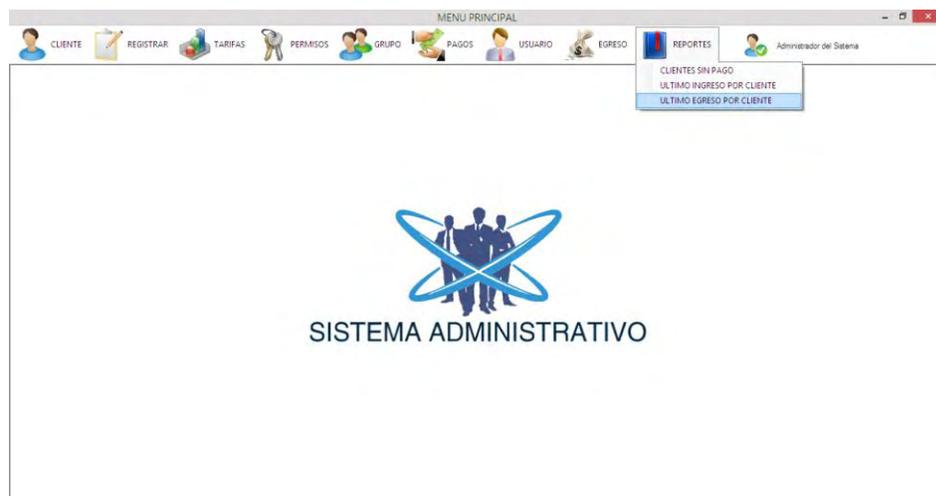


Figura 112. Ir a formulario ultimo egreso

3. El siguiente formulario muestra el último egreso que se ha generado por concepto de los pagos que se han realizado para los clientes.

MENU PRINCIPAL

CLIENTE REGISTRAR TAPIRAS PERROS GRUPO PAGOS USUARIO EGRESO REPORTES Administrador de Sistema

LISTA DE CLIENTES

Buscar Cliente

CEDEJA	CLIENTE	DIRECCION	TELEFONO	EMAIL	VALOR	FECHA ULTIMO PAGO REGISTRADO
100723	pedro perez				10000	15-08-2014 4:08 p.m.

TOTAL REGISTROS: 2

Figura 113. Reporte de último egreso por cliente