INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

RESIDENCIA PROFESIONAL AGOSTO-DICIEMBRE 2012

Proyecto: Sistema Reloj Checador Biometrico. (SISRECBIO)

PRESENTA:
Yoc Trujillo Ana Karen

Licenciatura en Informática

Asesor Interno	M.C. Roberto Cruz Gordillo.	
Revisor	Dra. Maria Candelaria Gutiérrez Gómez	

Tuxtla Gutiérrez, Chiapas; Enero de 2013

INDICE

CAPITULO I ---------- 2
JUSTIFICACION ---------- 2
CAPITULO II ---------- 3
OBJETIVO GENERAL ---------- 3
OBJETIVO ESPECIFICO ---------- 3
CAPITULO III CARACTERIZACION DEL AREA EN QUE PARTICIPO ---------- 4
UNIVERSIDAD AUTONOMA DE CHIAPAS (UNACH) ---------- 4
MISION ---------- 4
VISION ---------- 4
DESCRIPCION ---------- 4
FACULTAD DE CONTADURIA Y ADMINISTRACION C-I ---------- 6
MISION ---------- 6
VISION ---------- 6
DESCRIPCION ---------- 6
ORGANIGRAMA GENERAL DE LA EMPRESA ---------- 7
ORGANIGRAMA DE LA FCA C-I ---------- 7
UBICACIÓN FISICA DE LA UNACH ---------- 8
CROQUIS DE LA FACULTAD ---------- 8
INFRAESTRUCTURA DEL EQUIPO DE CÓMPUTO ---------- 9
Hardware ---------- 9
Software ---------- 9
CAPITULO IV PROBLEMAS A RESOLVER PRIORIZANDOLOS ---- 10
CAPITULO V ALCANCES Y LIMITACIONES ---------- 11
ALCANCES ---------- 11
LIMITACIONES ---------- 11
CAPITULO VI ---------- 12
FUNDAMENTO TEORICO ---------- 12
MARCO TEORICO CONCEPTUAL ---------- 12
TECNOLOGIAS PARA EL DESARROLLO DE SISTEMA ---------- 18
MARCO TEORICO ESPECÍFICO ---------- 21
CAPITULO VII PROCEDIMIENTO Y DESCRIPCION DE LAS ACTIVIDADES REALIZADAS ---------- 38
RESULTADOS, PLANOS, GRAFICAS, PROTOTIPOS Y PROGRAMAS. ---------- 42
DIAGRAMA ELKA ---------- 48
DIAGRAMAS DE CASO DE USO ---------- 50
DIAGRAMAS DE SECUENCIA ---------- 60
CONCLUSIONES Y RECOMENDACIONES ---------- 67
REFERENCIAS BIBLIOGRÁFICAS ---------- 68
ANEXOS ---------- 70

INTRODUCCION

En la actualidad los sistemas de información ofrecen una gran ventaja competitiva en las empresas tanto públicas como privadas que requieren estar a la vanguardia con la tecnología. Estos sistemas han permitido la eficiencia oportuna en la elaboración de las actividades de una empresa obteniendo de esta manera la reducción de los tiempos de respuesta al realizar una transacción y se da un mejor servicio a los clientes de una empresa.

La información presentada en este documento está basada en la creación de un sistema de información para la Facultad de Contaduría y Administración C-I de la Universidad Autónoma de Chiapas. El proceso que se llevo a cabo para obtener el sistema se describe en este documento empezando por el análisis y diseño de la arquitectura del software.

CAPITULO I

JUSTIFICACIÓN.

La Facultad de Contaduría y Administración C I de La Universidad Autónoma de Chiapas no cuenta con ningún sistema que controle las asistencias e incidencias del personal esto ocasiona una serie de inconvenientes como son: mal manejo de información e irregularidades en el reporte general de asistencias.

Por esta problemática se ve en la necesidad de implementar un sistema con la única finalidad de darle solución a todos esos inconvenientes descritos anteriormente.

La realización del sistema traerá los siguientes beneficios:

- Seguridad en el control de la información
- Se podrá aplicar las sanciones correspondientes al personal cuando se realice algún movimiento erróneo.
- La unidad de informática en el área de sistemas contara con la documentación técnica del sistema fortaleciendo el trabajo que realizan.

CAPITULO II

II.I OBJETIVO GENERAL

Analizar, diseñar y desarrollar un sistema de información para el registro y control de asistencias e incidencias del personal que labora en la Facultad de Contaduría y Administración C I de la Universidad Autónoma de Chiapas.

II.II OBJETIVOS ESPECÍFICOS

- Reunión de requisitos
- Determinar las funciones ,comportamientos y rendimientos requeridos
- Diseñar la estructura de datos
- Diseñar la arquitectura del software
- Generación de código
- Realización de pruebas necesarias

CAPITULO III

CARACTERIZACIÓN DEL ÁREA EN QUE PARTICIPÓ

UNIVERSIDAD AUTONOMA DE CHIAPAS

Mtro. Jaime Valls Esponda

Rector

MISIÓN

La universidad autónoma de Chiapas es una institución de educación superior, publica y autónoma que genera, recrea y extiende el conocimiento; forma profesionales, capaces, críticos, propositivos y creativos, con espíritu ético y humanista, conciencia histórica y social, para incidir con responsabilidad en el desarrollo de Chiapas y de México, con respeto a la identidad cultural de los pueblos, al a biodiversidad.

VISION

La Universidad Autónoma de Chiapas es en el año 2018, una institución reconocida socialmente por la calidad de sus egresados, por su actividad científica y tecnológica, y por la transparencia y credibilidad de su gestión. Con programas educativos acreditados y procesos certificados; innovadora y articulada en redes de cooperación, centrada en lo local e inspirada en el pensamiento universal, y estrechamente vinculada al desarrollo de la sociedad chiapaneca.

DESCRIPCION

La Universidad Autónoma de Chiapas es una institución educativa de giro público, con la finalidad de formar profesionistas capaces de lograr sus metas y objetivos, la UNACH ha consolidado una serie de carreras, tanto ingeniería como licenciaturas, distribuidas a lo largo del estado de Chiapas en diversas facultades. Actualmente es una de las universidades con mayor prestigio y presencia en el Estado, así como en el sureste de México, sustentando esta información, con una matrícula de 18 mil estudiantes, distribuidos en cada uno de sus centros educativos.

Así también, la Universidad, cuenta con una planta docente de mil 672 maestros, del más alto nivel académico.

La Universidad está distribuida a lo largo de la geografía chiapaneca en 17 centros educativos, siendo facultades, escuelas e instituciones:

FACULTAD DE CONTADURIA Y ADMINISTRACION I

FACULTAD DE INGENIERIA, CAMPUS I

FACULTAD DE ARQUITECTURA, CAMPUS I

FACULTAD DE MEDICINA HUMANA, CAMPUS II

FACULTAD DE CONTADURIA PÚBLICA, CAMPUS IV

FACULTAD DE CIENCIAS DE LA ADMINISTRACION, CAMPUS IV

FACULTAD DE CIENCIAS QUIMICAS, CAMPUS IV

FACULTAD DE CIENCIAS AGRICOLAS, CAMPUS IV

FACULTAD DE CIENCIAS AGRONOMICAS, CAMPUS V

FACULTAD DE HUMANIDADES, CAMPUS VI

ESCUELA DE CONTADURIA Y ADMINISTRACION, CAMPUS VII

ESCUELA DE CIENCIAS ADMINISTRATIVAS, CAMPUS VIII

ESCUELA DE CIENCIAS ADMINISTRATIVAS ISTMO-COSTA, CAMPUS IX

ESCUELA DE LENGUAS, CAMPUS TUXTLA

ESCUELA DE LENGUAS, CAMPUS SAN CRISTOBAL DE LAS CASAS

ESCUELA DE LENGUAS, CAPUS TAPACHULA

INSTITUTO DE ESTUDIOS INDIGENAS

FACULTAD DE CONTADURIA Y ADMINISTRACION C-I

Dr. Rafael Timoteo Franco Gurria

Director

MISION

La Facultad de Contaduría y Administración, Campus I, forma de manera integral profesionales en las áreas económico-administrativas y sistemas computacionales; generadores de cambio y mejoramiento continuo, capaces de analizar los problemas de su entorno y de proponer, implantar y evaluar las soluciones a los mismos, con valores éticos y morales de acuerdo a las necesidades de desarrollo científico y tecnológico que demanda la sociedad.

VISION

La Facultad de Contaduría y Administración, Campus I, se constituye como un centro educativo congruente con su entorno, con una planta docente con perfil PROMEP, con un modelo de enseñanza centrado en el alumno, programas acreditados, cuerpos académicos consolidados, y con egresados generadores de cambio y mejoramiento continuo, capaces de analizar los problemas y proponer soluciones, respondiendo a las expectativas de la sociedad

DESCRIPCION

La Facultad de Contaduría y Administración C-I es una de las más representativas de la Universidad, ya que cuenta con algunas de las carreras más solicitadas, dentro

de su oferta académica se encuentran las carreras de Contaduría, Administración, Gestión turística y Sistemas computacionales; dada a la alta demanda, la facultad cuenta con un total de alumnos de 3808, así como 166 docentes, que otorgan sus servicios.

ORGANIGRAMA DE LA FCA C-I

UNIVERSIDAD AUTONOMA DE CHIAPAS
UNACH
Reunión de Planeación, Seguimiento y Evaluación de la Facultad
UNACH
ESTRUCTURA DE LA FACULTAD
DIRECCIÓN
H. CONSEJO TÉCNICO
COORDINACIÓN DE ACREDITACIÓN Y GESTIÓN DE LA CALIDAD
COORDINACIÓN DE PLANEACIÓN , SEGUIMIENTO Y EVALUACIÓN
COORDINACIÓN DE EDUCACION CONTINUA Y VIRTUAL
COORDINACION DE TUTORIA
COORDINACIÓN DE DESARROLLO CURRICULAR
COORDINACION DE EXTENSIÓN Y VINCULACIÓN
SECRETARIA ADMINISTRATIVA
SECRETARIA ACADÉMICA
COORDINACION DE INVESTIGACION Y POSGRADO

UBICACIÓN FÍSICA DE LA EMPRESA:

La Facultad de Contaduría y Administración de la Universidad Autónoma de Chiapas está ubicada en el Boulevard Belisario Domínguez, kilometro 1081, Sin número, Terán Tuxtla Gutiérrez, Chiapas, México, C.P. 29050.

Teléfono: 52(961)617800

Contacto: www.unach.com.mx

Fig. 1 Ubicación de la Facultad de Contaduría y Administración C-I

INFRAESTRUCTURA DEL EQUIPO DE CÓMPUTO.

Las herramientas con que cuenta el proyecto son las siguientes:

Hardware:

- Computadoras de escritorio
- Impresoras
- Fax
- Reguladores de voltaje no break
- Teléfonos.
- Teléfonos ip

Software:

- Wifi
- Microsoft Office
- Página WEB Oficial

CAPITULO IV

PROBLEMAS A RESOLVER

Actualmente en la FCA C-I *(“Facultad de Contaduría y Administración Campus I”*) ubicado en la UNACH (*“Universidad Autónoma de Chiapas”*) realiza el registro de asistencia y recopilación de información manualmente, como también en formatos de Office; al no contar con otro procedimiento que resulte más eficaz y cómodo para los trabajadores, ya que estos procesos se realizan de igual manera todos los días, por consecuencia, existe una pérdida de tiempo en elaborar los reportes quincenales, tomando en cuenta que también se hace mal uso de los datos.

Con el desarrollo del sistema se pretende agilizar el procesamiento de los datos, la representación de la información en tiempo y forma, reducir la utilización de papeles así como también aminorar costos, facilitar la toma de decisiones, obtenidas del resultado de los datos procesados por el sistema y algo muy importante, hacer buen uso de la información obtenida ya que el sistema dará seguridad a los datos obtenidos.

CAPITULO V

ALCANCES Y LIMITACIONES:

ALCANCES

El desarrollo del sistema, permitirá tener un control eficiente de los registros de las entradas y las salidas del personal de la Facultad de Contaduría y Administración C-I, se permitirá el control, de la asistencia a través de una terminal biométrica para registrar el acceso del personal.

LIMITACIONES

En esta primera etapa se desarrollara el módulo que es el reloj checador. Quedara pendiente el desarrollo de el segundo modulo, el cual consistirá en la generación de nóminas.

El sistema estará enfocado a la parte del reloj checador que presenta la parte del registro de entradas y salidas del personal, registro de horarios, registro de justificantes, etc. Y no al proceso de reclutamiento de personal.

CAPITULO VI

FUNDAMENTO TEÓRICO.

VI.I MARCO TEÓRICO CONCEPTUAL.

Para poder desarrollar un sistema de información es necesario conocer conceptos y técnicas que se requieran para poder crear un software, es por esto que en este apartado del documento se describe la información necesaria para poder comprender que es y los aspectos que involucran en la creación de los sistemas de información.

Se pretende desarrollar un software que pueda ser utilizado como una herramienta útil para el control de asistencias e incidencias. Cabe recalcar que es necesario tener en cuenta que en todo desarrollo de sistemas de software es de suma importancia definir una metodología. Esto permite a los desarrolladores seguir alguna especificación en cada una de las etapas del desarrollo del sistema, desde los requerimientos iníciales hasta las pruebas finales que hacen que el sistema sea coherente y además formal.

El manejo de los conceptos y teorías que permitirán ser un soporte para desarrollar los sistemas deben ser congruentes para efectuar e implementar un sistema de información, tomando en cuenta los estándares para elaborar un análisis adecuado, diseño, implementación especificada, las pruebas y mantenimiento.

Aspectos generales

Sistema:

“es un conjunto de componentes que interaccionan entre si para lograr un objetivo común.” (Los autores, 2006, ¿Qué es un sistema de información?, para 11)

Sistema de información:

“un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de la organización. (Los autores, 2006, Una definición general de sistemas de información, para 12)

Los sistemas de información anteriormente generaban cambios técnicos que afectaban a pocas personas que estaban inmersas dentro de la empresa, a diferencia de los actuales ya que estos involucran a una mayor parte de la institución.

Ingeniería de software

El término ‘*ingeniería de software* ’fue introducido por primera vez a finales de 1968 en una conferencia destinada a su discusión, la cual fue posteriormente llamada ‘*crisis del software’*. Esta crisis de software fue el resultado directo de la Introducción del hardware de la tercera generación computacional. Es una disciplina de la ingeniería que comprende todo los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.

La ingeniería de software no es una disciplina que solo deba aplicarse en proyectos de ciertas áreas, sino que también trata con áreas diversas dentro de las ciencias computacionales, tales como: construcción de compiladores, sistemas operativos, o desarrollos empresariales como es el caso de esta aplicación de software. La ingeniería de software abarca todas las fases del ciclo de vida en el desarrollo de cualquier sistema de información aplicables a áreas tales como investigación científica, medicina, logística y negocios (Introducción a la ingeniería del software, 2005, P 5).

Adoptar un enfoque sistemático y organizado, es la forma más efectiva de producir software de alta calidad; aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en web, los cuales requieren una mezcla de técnicas de software y de diseño gráfico. En un nivel técnico la ingeniería de software empieza con una serie de tareas de modelado que llevan a una especificación completa de los requisitos y a una representación del diseño general del software a construir.

Metodología estructurada.

Todo trabajo debe claramente saber que se desea hacer, por ejemplo "cuando se desea realizar una construcción de un inmueble, previamente debe existir un plano para conocer lo que se debe hacer, evitando cometer errores para así decidir qué deben ir primero y posteriormente en las etapas de la construcción".

El análisis y diseño de la construcción de modelos para representar las funciones que realizará el sistema desde su concepción física hasta la deducción lógica de los procesos y resultados. La técnica estructurada se basa en el concepto top-Down de dividir el sistema en funciones; En un primer nivel se representan las entradas y salidas del sistema, para bajar a niveles inferiores, donde se describe en qué consiste cada proceso. Permite que las personas observen los elementos lógicos (lo que hará el sistema) separado de los componentes físicos (computadoras, terminales, sistemas de almacenamiento, etc.) después de esto se puede desarrollar un diseño físico eficiente.

El modelo está determinado por tres partes bien diferenciadas y esenciales como símbolos gráficos, diagrama de flujo de datos y diccionario de datos. (Metodología Del Análisis **Estructurado** de Sistemas, 2005 - P 59)

Descripción gráfica

Esta prepara un bosquejo que señala sus características, identifica la función e indica como esta interactúa con otros elementos, describe la manera como

desarrollar un sistema grande con procesos tediosos y propensos a errores. Ya que es más fácil omitir algún detalle o dar una explicación que quizá los demás lo entiendan.

Diagrama de flujo de datos (DFD)

Muestra las fuentes y destino de los datos, identifica y da nombre a los procesos que se llevan a cabo, identifica y da nombre a los grupos de datos que relacionan una función con otra y señalan los almacenes de datos a los que tienen acceso,

Recibe este nombre para la descripción completa de un sistema que está formado por un conjunto de diagramas de flujo de datos para desarrollar una descripción del sistema por el método de análisis estructurado se sigue un proceso descendente top-Down. Ese modelo se detalla en diagramas de bajo nivel que muestra características adicionales del sistema cada proceso puede desglosarse en diagramas de flujo de datos cada vez más detallados esta secuencia se repite hasta que se obtienen suficientes detalles que permiten al analista comprender en su totalidad al sistema.

MÉTODO DE DESARROLLO DE SISTEMAS DE INFORMACIÓN.

Desarrollo del Sistema:

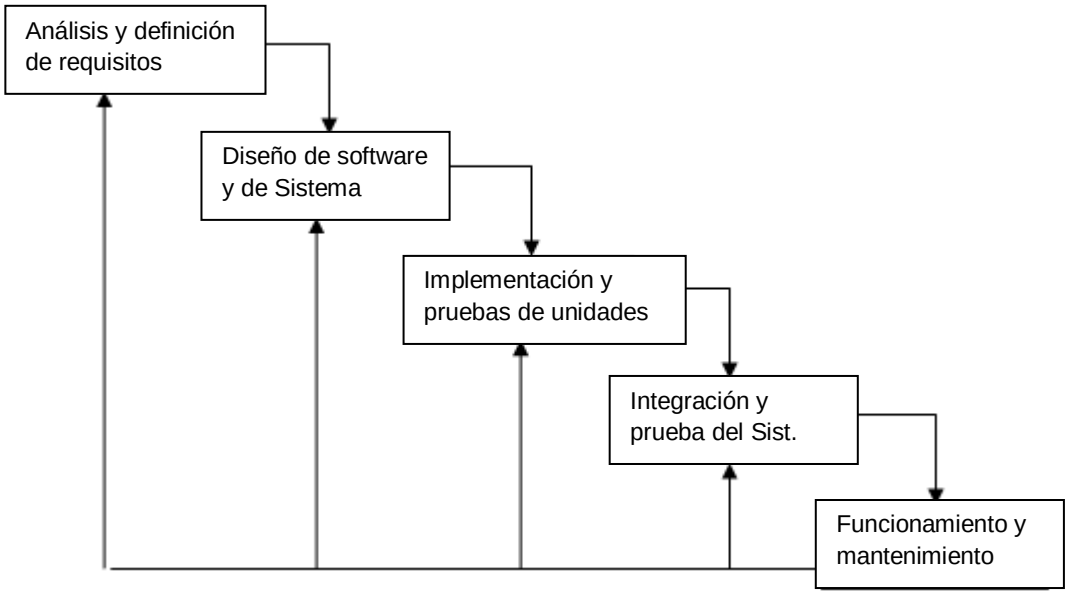


Figura.2: *desarrollo en cascada (modelo cascada) enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software.*

Es el primer modelo de proceso de desarrollo de software que se publicó se derivó de procesos de ingeniería de sistemas más generales (Royce, 1970). Se le nombra cascada debido a la cascada de una fase a otra, dicho modelo se conoce como modelo en cascada o como ciclo de vida del software **Figura 2**.

Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

Análisis y definición de requerimientos

Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema.

Diseño del sistema

El proceso de diseño del sistema divide los requerimientos en sistemas hardware o software. Establece una arquitectura completa del sistema. El diseño del software identifica y describe las abstracciones fundamentales del sistema software y sus relaciones.

1. **Implementación y prueba de unidades.**
   Durante esta etapa, el diseño del software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.

2. **Integración y prueba del sistema.**
   Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema software se entrega al cliente.

3. **Funcionamiento y mantenimiento.**
   Por lo general (aunque no necesariamente), ésta es la fase más larga del ciclo de vida. El sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del Sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos. La segunda fase no debe empezar hasta que la

   Fase previa haya finalizado. En la práctica, estas etapas se superponen y proporcionan información a las otras. Durante el diseño se identifican los problemas con los requerimientos; durante el diseño del código se encuentran problemas, y así sucesivamente. Debido a los costos de producción y aprobación de documentos, las iteraciones son costosas e implican rehacer el trabajo. Por lo tanto, después de un número reducido de

Iteraciones, es normal congelar partes del desarrollo, como la especificación, y continuar con las siguientes etapas de desarrollo.

Ventajas:

La documentación se produce en cada fase y que éste cuadra con otros modelos del proceso de ingeniería. El modelo en cascada sólo se debe utilizar cuando los requerimientos se comprendan bien y sea improbable que cambien radicalmente durante el desarrollo del sistema. Sin embargo, el modelo refleja el tipo de modelo de proceso usado en otros proyectos de la ingeniería. Por consiguiente, los procesos del software que se basan en este enfoque se siguen utilizando para el desarrollo de software, particularmente cuando éste es parte de proyectos grandes de ingeniería de sistemas.

TECNOLOGIAS PARA EL DESARROLLO DEL SISTEMA

Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++,Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. Aunque actualmente

Visual Basic 6.0

Visual Basic es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper paraMicrosoft. Este lenguaje de programación es un dialecto de BASIC, con importantes agregados. Su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando unambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y, en cierta medida, también la programación misma.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008.

En 2001 Microsoft propuso abandonar el desarrollo basado en la API Win32 y pasar a un framework o marco común de librerías, independiente de la versión del sistema operativo, .NET Framework, a través deVisual Basic .NET (y otros lenguajes como C Sharp(C#) de fácil transición de código entre ellos); fue el sucesor de Visual Basic 6.

Aunque Visual Basic es de propósito general, también provee facilidades para el desarrollo de aplicaciones de bases de datos usando Data Access Objects,Remote Data Objects, o ActiveX Data Objects.

Visual Basic contiene un entorno de desarrollo integrado o IDE que integra editor de textos para edición del código fuente, un depurador, un compilador (y enlazador) y un editor de interfaces gráficas o GUI. Se han desarrollado las extensiones necesarias para muchos otros.

Access

Posiblemente, la aplicación más compleja de la suite Office, sea Access, una base de datos visual. Como todas las modernas bases de datos que trabajan en el entorno Windows, puede manejarse ejecutando unos cuantos clic de mouse sobre la pantalla. Access contiene herramientas de diseño y programación reservadas a los usuarios con mayor experiencia, aunque incluye bases de datos listas para ser usadas; están preparadas para tareas muy comunes, que cualquiera puede realizar en un momento determinado –ordenar libros, archivar documentación, etc.-.

Biometría

El concepto Biometría proviene de las palabras bio (vida) y metria (medida),

Por lo tanto con ello se refiere que todo equipo biométrico mide e identifica alguna característica propia de la persona.

La biometría es una tecnología de seguridad basada en el reconocimiento de una característica de seguridad y en el reconocimiento de una característica física e intransferible de las personas, como por ejemplo la huella digital.

Los sistemas biométricos incluyen un dispositivo de capacitación y un software biométrico que interpreta la muestra física y la transforma en una secuencia numérica. En el caso del reconocimiento de la huella digital, se ha de tener en cuenta que ningún caso se extrae la imagen de la huella, sino una secuencia de números que la representan. Sus aplicaciones abarcan un gran número de sectores: desde el acceso seguro a computadores, redes, protección de ficheros electrónicos, hasta el control de horario y control de acceso físico a una sala de acceso restringido.

Por esta razón la definen como una rama de las matemáticas estadísticas que se ocupa del análisis de datos biológicos y que comprende temas como población, medidas físicas, tratamientos de enfermedades y otros por el estilo.

Todos los seres humanos tenemos características morfológicas únicas que nos diferencian. La forma de la cara, la geometría de parte de nuestro cuerpo como las manos, nuestro ojos y tal vez la más conocida, la huella digital, son algunos rasgos que nos diferencian del resto de los seres humanos.
La medición biométrica se ha venido estudiando desde tiempo atrás y es considerada en la actualidad como el método ideal de identificación humana.
La identificación por medio de huellas digitales constituye una de las formas mas representativas de la utilización de la biometría, una huella digital esta formada por una serie de surcos. Las terminaciones o bifurcaciones de los mismos son llamados “puntos de minucia”. Cada uno de estos puntos tiene una característica y una posición única, que puede ser medida. Comparado esta distribución es posible obtener la identidad de una persona que intenta acceder a un sistema en general.

MARCO TEÓRICO ESPECÍFICO

Tipos de sistemas de información

Kennet E. Kendall y Julie E. Kendall (2005), definen los siguientes tipos de sistemas de informacion en su libro analisis y diseños de sistemas:

Sistemas de nivel operativo: son sistemas de información computarizada creados para procesar grandes cantidades de datos relacionadas con transacciones rutinarias de negocios, como las nominas y los inventarios.

Sistemas de nivel de conocimientos: sistemas de información en los que se apoyan los trabajadoras del conocimiento y de la información en una institución.

Sistemas de nivel gerencial: son sistemas de información computarizada cuyo propósito es contribuir a la correcta interacción entre kis usuarios y las computadoras. Dan apoyo a un espectro de tareas organizacionales como el análisis y la toma de decisiones.

Sistema de nivel estratégico: son sistemas de información computarizados que ayudan a organizar actividades relacionadas con el entorno externo mediante herramientas graficas y de comunicaciones. Amplían y apoyan las capacidades de los ejecutivos al darles la posibilidad de comprender sus entornos. (P 2,3,4).

Procesos de Desarrollo de software.

"El proceso de desarrollo de software se define como: la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software." (Roger S. Presman, 2002. P 14).

Para la creación de los sistemas de información los desarrolladores de software deben de guiarse por uno de los modelos que contempla el proceso de desarrollo de software porque ayudan a construir un sistema de información de forma ordenada dividiendo las actividades en una serie de etapas.

Los modelos pueden ser considerados como métodos porque indican como hacer mas eficiente ek desarrollo de sistema de información. Para ello se suelen estructurar en fases la vida de dichos sistemas con el fin de facilitar su planificación, desarrollo y mantenimiento.

Los modelos de desarrollo de sistema deben definir: objetivos, fases, tareas, productos, y responsables, necesarios para la correcta realización del proceso y su seguimiento.

Los principales objetivos de un modelo de desarrollo son:

- Asegurar la uniformidad y calidad tanto del desarrollo como del sistema en si.

- Satisfacer las necesidades de los usuarios del sistema.
- Conseguir un mayor nivel de rendimiento y eficiencia del personal asignado al desarrollo.
- Ajustarse a los plazos y costes previstos en la planificación.
- Generar de forma adecuada la documentación asociada a los sistemas.
- Facilitar el mantenimiento posterior de los sistemas.

El proceso de desarrollo de software contempla los siguientes modelos: Modelo lineal secuencial (ciclo de vida clásico o modelo de cascada), modelo incremental, modelo DRA, modelo de construcción de prototipos, modelo en espiral, modelo de desarrollo concurrente , desarrollo basado en componentes, modelo de métodos formales y Proceso Unificado de Rational (RUP).

Un modelo de desarrollo de software debe de ser seleccionado de acuerdo a la complejidad del proyecto a las necesidades que se requieran solucionar, esto es porque cada modelo cumple una serie de características que solo aptan para proyectos específicos por lo cual debe tener cuidado al seleccionar algún modelo para evitar problemas futuros durante el desarrollo del software.

Por lo tanto en este documento se describirán los modelos considerados aptos para el desarrollo de Sistema Reloj Checador Biométrico, los cuales son:

- Modelo de ciclo de vida clásico
- Modelo de construcción de prototipos
- Modelo en espiral
- Proceso Unificado de Rational

Modelo de Ciclo de Vida Clásico

En el modelo de ciclo de vida clásico o cascada, un sistema avanza a través de una secuencia ordenada de pasos, empezando por el análisis del sistema hasta el mantenimiento del mismo.

En este modelo el avance de cada una de las etapas del desarrollo conduce a volver a rediseñar el proyecto, esto genera aumento en el costo del proyecto y aumenta el tiempo de desarrollo.

Este modelo es uno de los mas antiguos por lo tanto es de los mas utilizados en la actualidad por los desarrolladores de software por su fácil comprensión.

Este modelo presenta los siguientes inconvenientes:

- Los proyectos reales raras veces sigue un modelo secuencial.
- Es difícil que el cliente exponga explícitamente todos los requisitos.
- El cliente debe tener paciencia debido a que el proyecto no esta disponible hasta que se tiene un avance considerable de este.

A pesar de estos inconvenientes que presenta el modelo se considera útil implementar este modelo cuando los requerimientos son fijos y no tienden a cambiar en poco tiempo y en donde el trabajo se realiza hasta su conclusión de una manera lineal.

El modelo de ciclo de vida clásico se conforma de las siguientes etapas: Análisis, Diseño, Generación de código, Pruebas y Mantenimiento, la forma en cómo las etapas de comunican se ilustra en la figura 1. A continuación se describe cada una de estas etapas.

Análisis: esta fase es la parte inicial del desarrollo del software y está conformado por las siguientes actividades: Reunión de los requisitos, comprender el dominio de la información del software y determinar las funciones, comportamientos, rendimiento e interconexión requeridos por el software.

Diseño: corresponde a la fase en donde se diseña la estructura funcional del software. El diseño esta conformado por las siguientes actividades: Diseñar la estructura de datos, diseñar la arquitectura del software, diseñar la interfaz del software, diseñar los algoritmos (forma en como se comportará internamente las funciones del sistema antes de pasar a la codificación).

Generación de código: El diseño debe de ser traducido a la forma en que la computadora entiende como realizar las funciones.

Pruebas: En esta fase se realizan las pruebas necesarias sobre la funcionalidad del software verificando que este cumpla con todos los requisitos que se contemplaron y no provoque errores.

Mantenimiento: esta fase entra en operación cuando se producen cambios debido a errores presentados, porque el software sebe ser adaptado a los cambios de su entorno externo o porque el cliente requiere mejoras funcionales o de rendimiento del software.

Retroalimentación

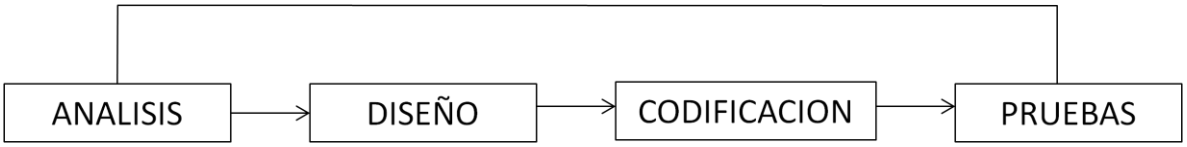


Figura 1. Diagrama del Ciclo de Vida Clásico

Construcción de Prototipos

Este modelo se basa en construir un software de forma rápida que describa la relación que tiene el usuario con el software.

Es necesario centrar la construcción del software en las funciones mas importantes para el usuario porque son los puntos críticos que el software controlara.

Este modelo es recomendado utilizarse cuando los requerimientos cambian constantemente, cuando no se tiene suficientemente especificados los requerimiento.

El desarrollo de un software por medio de este modelo comienza en la comunicación con el cliente en donde especifican los objetivos globales para el desarrollo del software, identifica los requisitos y las áreas en donde es necesaria mas definición. Entonces se lleva a cabo un plan de desarrollo rápido en donde verifican lo que es sistema debe realizarse después se procede a la construcción del sistema mediante un diseño rápido. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el usuario final. El diseño rápido conduce a la construcción de un prototipo. Después, el software que se desarrollara. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer. El proceso que se menciono en este párrafo sobre el recorrido que se realiza un sistema durante su desarrollo mediante el modelo de prototipo se ilustra en la figura 2.

Figura 2. Modelo de construcción de prototipos.

Ventajas

- No modifica el flujo del ciclo de vida.
- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios.
- Reduce costos y aumenta la probabilidad de éxito.

- Exige disponer de las herramientas adecuadas.
- No presenta calidad ni robustez.

Desventajas

A los usuarios les gusta el sistema real y a los desarrolladores les gusta construir algo de inmediato. Sin embargo, la construcción de prototipos se torna problemática por las siguientes razones:

- El cliente ve funcionando lo que para el es la primera versión del prototipo y puede decepcionarse al indicarle que el sistema aun no ha sido construido.
- El desarrollador puede caer en la tentación de aumentar el prototipo para construir el sistema final sin tener en cuenta las obligaciones de calidad y de mantenimiento que tiene con el cliente.
- Desconocimiento del tiempo que se tardara en crear un producto aceptable.
- Los desarrolladores de software pueden adoptar practicas de prueba y error sin un análisis y diseño formales previos.

Modelo en espiral

Es un modelo de proceso se software evolutivo. Fue desarrollado por BOHEM en el año 1985 en donde toma en cuenta fuertemente el análisis de riesgo a la hora de desarrollar software. Este modelo une características tanto del modelo de ciclo de vida clásico o cascada y la de prototipos incluyendo una nueva actividad la de análisis de riesgos, en este punto los demás modelos tienen su debilidad debido a que no contemplan, los riesgos permiten determinar si el sistema puede continuar desarrollándose en cada etapa del modelo o no.

Al desarrollar un software bajo este modelo se construye a través de una serie de versiones incrementales. Durante las primeras iteraciones o vueltas de la espiral podría ser un modelo en papel o un prototipo, durante las últimas iteraciones se produce una versión más completa del software.

Cuando se comienza a desarrollar un software por este modelo se gira a través de la espiral en dirección a las agujas del reloj comenzando por el centro. Primero se establecen las alternativas de desarrollo, se evalúan los riesgos y se realiza la primera vuelta a la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelve a evaluar las distintas nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado y no necesite seguir mejorándose con otro nuevo ciclo. El proceso que se describió en este párrafo se puede observar en la figura 3.

Es uno de los modelos mas recomendables en el desarrollo y creación de software porque consta de pocas etapas o fases, las cuales se van realizando de manera continua y cíclica.

Características.

- Incluye la etapa de análisis de riesgos, no incluida anteriormente.
- Es ideal para crear productos con diferentes versiones mejoradas como se hace con el software moderno de microcomputadoras.
- La ingeniería puede desarrollarse a través del ciclo de vida clásico o el de construcción de prototipos.
- Este es el enfoque mas realista actualmente.
- El modelo en espiral se divide en un numero de actividades estructurales, también llamadas regiones de tareas. Generalmente, existen entre tres y seis regiones de tareas.

Etapas

Planificación: las tarea requeridas para definir recursos, el tiempo y otras informaciones relacionadas con el proyecto. Son todos los requerimientos.

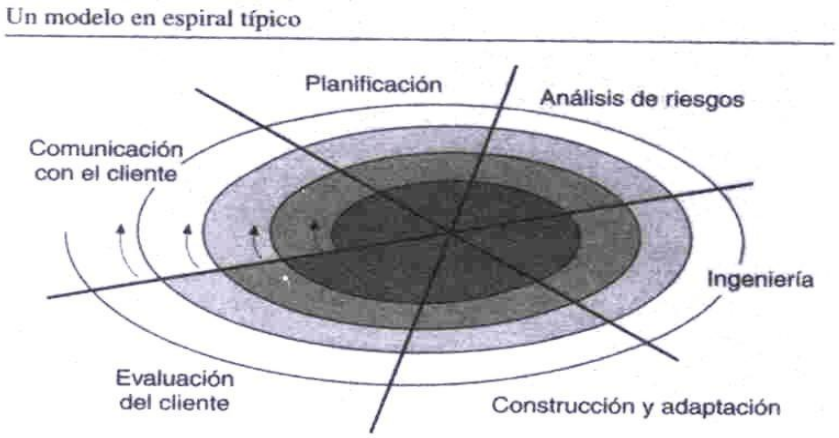
Análisis de riesgos: las tareas requeridas para evaluar riesgos técnicos y otras informaciones relacionadas con el proyecto.

Ingeniería: las tareas requeridas para construir una o mas representaciones de la aplicación.

Construcción y adaptación: las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario.

Evaluación del cliente: las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementación durante la etapa de instalación.

Figura 3. Modelo espiral



Proceso unificado de Rational (RUP)

El proceso unificado de Rational es un proceso de desarrollo de software y trabaja en conjunto con el Lenguaje Unificado de Modelo (UML) es considerado un proceso porque “define quien esta haciendo que, cuando hacer y como alcanzar cierto objetivo”, constituye el proceso estándar mas utilizado para el análisis, implementación y documentación de sistemas orientados a objetos.

El proceso unificado de Rational puede ser adaptado a cualquier tipo de proyecto y empresa (grandes y pequeñas).

Este proceso se caracteriza por ser iterativa, basada en la técnica de caso de uso y flexible.

El proceso de desarrollo esta dividido en fases a lo largo del tiempo cada una de las cuales tienen objetivos específicos y conjunto de pasos definidos que deben alcanzarse. La duración de cada fase depende del equipo y del producto a generar. En la figura 4 se ilustra cada una de las etapas que contempla el modelo.

A continuación se describe las cuatro fases involucradas en este proceso:

Inicio: en esta fase se suele especificar el ámbito del proyecto, se identifican los requisitos de negocio del software presentados a través de un conjunto de casos de uso que describen las características y funciones que son deseables para cada usuario que interactuara con el sistema, se identifican los riesgos y las posibles arquitecturas (es un esquema tentativo de los subsistemas mas importantes y de las funciones y características que lo forman) y se concluyen con un plan de ejecución.

Elaboración: en esta fase de eligen los casos de uso necesarios para cubrir la arquitectura básica de sistema y se desarrollan. La arquitectura del sistema es la pieza fundamental cobre la que se añade el resto del desarrollo.

El plan realizado en la etapa de inicio se revisa de manera cuidadosa al termino de esta fase para asegurar que el ámbito, los riesgos y los datos entregados aun son razonables. Si el resultado de la evaluación no es satisfactorio se debe proceder a la corrección del plan antes de pasar a la siguiente fase para así poder entregar un sistema que cumpla con las características pedidas por el cliente.

Construcción: esta fase se concentra en la elaboración de un producto totalmente operativo y eficiente es decir se realiza el diseño de las interfaces (pantallas) del sistema, se da inicio a la programación de esas interfaces para hacer funcional al software y se realizan las pruebas necesarias para detectar posibles errores y verificar el rendimiento del sistema.

Implementación: esta fase contempla la instalación y liberación del sistema entre las actividades realizadas se incluye las siguientes.

- Se hace entrega del software al cliente
- Se da la capacitación a los usuarios sobre el uso del software

- Se brinda soporte técnico a los usuarios por parte de los desarrolladores de software cuando se requiera
- Se entrega el manual de usuario al cliente.

Tras cada iteración de la fase de transición, se pone a disposición de los usuarios una nueva versión de la aplicación, utilizándose la información recibida de estos para mejorar el sistema en la siguiente etapa.

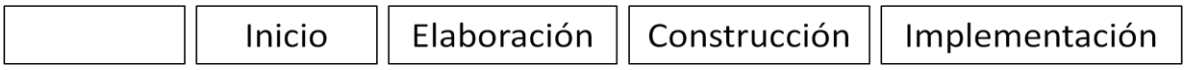


Figura 4. Diagrama del Proceso de Rational

Lenguaje Modelado Unificado (UML).

UML es una herramienta que mejora la calidad del análisis y díselo de los sistemas además que se puede tener una mejor comunicación cliente-desarrollador de software porque permite que los desarrolladores diseñen la estructura de un software mostrando las ideas que el cliente ha planteado de manera que las dos partes puedan comprender la estructura funcional del software mas fácilmente visualizando asi todos los procesos que el software realizara.

Conceptos Orientados a Objetos.

Es importante que usted comprenda los siguientes conceptos orientados a objetos antes de empezar a describir cada uno de los diagramas que UML contempla.

Objetos

Los objetos son personas , lugares o cosas que son relevantes para el sistema bajo análisis. Los objetos podrían ser clientes, artículos, pedidos, etc. Los objetos también podrían ser pantallas GUI o áreas de texto en la pantalla.

Clases

Define el conjunto de atributos y comportamientos compartidos por cada objeto de la clase. Cada clase debe tener un nombre que lo distinga de todas las demás. Los nombres de las clases normalmente son sustantivos o frases cortas y empiezan con una letra mayúscula.

En UML una clase se representa como un rectángulo. El rectángulo contiene otras dos características importantes: una lista de atributos y una serie de métodos.

Un atributo describe alguna propiedad de todos los objetos de la clase. Un método es una acción que se puede solicita a cualquier objeto de la clase. Los métodos son los procesos que una clase sabe como realizar. Los métodos se llama operaciones.

Herencia

Las clases pueden tener hijos; es decir, una clase se puede crear a apartir de otra clase. En UML la clase original o madre se conoce como base. La clase hija se denomina clase derivada. Esta se puede crear de tal manera que herede todos los atributos y comportamientos de la clase base. La herencia reduce el trabajo de programación usando fácilmente objetos comunes.

Abstracción

“La abstracción de refiere a quitar las propiedades y acciones de un objeto para dejar solo aquellas que sean necesarias” (Schmuller Joseph, 2004,P 22)

Polimorfismo

Es cuando una operación tiene el mismo nombre en diferentes clases pero se diferencian del tipo de dato que represente a la operación o de la cantidad de parámetros que se envíen a la operación.

Encapsulamiento

Se refiere a que pueden existir algunas funcionalidades de algunos objetos que no son visibles para otros objetos esto conlleva a un ocultamiento de las funciones.

Envío de mensajes:

Se refiere al envío de mensaje que se puede dar entre un objeto a otro para realizar alguna operación es decir un objeto enviara un mensaje a otro para que realice una operación al recibir el mensaje el objeto receptor procesara y ejecutara la operación.

Asociaciones

Se refiere a que los objetos se pueden relacionar entre si. En los objetos puede existir una relación es una sola dirección y asociarse en mas de una forma. Las clases se pueden asociar con mas de una clase distinta.

Agregación

Representa la conformación del conjunto de varias clases para poder funcionar.

Modelo de Caso de Uso

Se puede definir a un caso de uso como la descripción de la secuencia de eventos que un usuario (actor) realiza en un sistema. Un caso de uso muestra la funcionalidad de un sistema desde la perspectiva de un usuario fuera del sistema no se visualiza la forma en que el sistema realiza las funciones

Simbología

Un diagrama de caso de uso está conformado por actores, caso de uso y líneas de interconexión. Estos símbolos se ilustran en la figura 5 donde muestra la representación del diagrama de caso de uso.

Un actor es aquel que realiza alguna operación en el sistema este puede ser humano, otro sistema o un dispositivo (teclado, modem o conexión web, etc.).

Un caso de uso es representado mediante una elipse se encarga de mostrar una función que el sistema realiza.

Las líneas de interconexión representa la comunicación que existe entre un actor y un caso de uso o de un actor a otro.

“Un caso de uso describe siempre tres cosas: un actor que inica el evento, el evento que activa un caso de uso y el caso de uso que desempeña las acciones activada por el evento. Un caso de uso se nombra con un verbo y un sustantivo.” (E. Kendall Kennet y E. Kendall Julie, 2005, P 666).

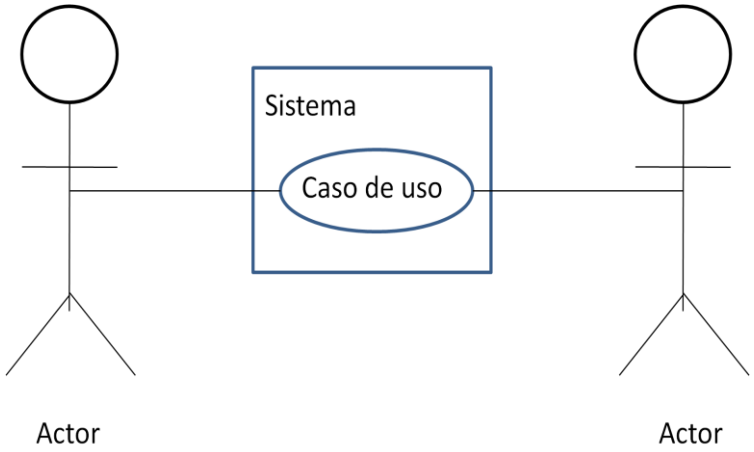


Figura 5. Ejemplo de un diagrama de caso de uso

Relaciones

Existen básicamente cuatro tipos de relaciones en los diagramas de caso de uso de las cuales son: comunica, inclusión, extensión y generalización. Estas relaciones a continuación se describen.

- Comunica: se usa para conectar un actor con un caso de uso
- Inclusión: describe la situación en que un caso contiene un comportamiento que es común para mas de un caso de uso.
- Extensión: describe la situación en la que un caso de uso posee el comportamiento que permite al nuevo caso de uso manejar una variación o excepción del caso de uso básico.

- Generalización: se da cuando un caso de uso o actor tiene o posee las mismas características o acciones que el caso de base o padre además que el caso de uso hijo puede agregar sus propias acciones.

Escenarios

Un escenario representa la descripción de la serie de actividades o procesos posibles que intervienen en un caso como por ejemplo que pasa si un articulo comprado ya se encuentra agotado.

Cada escenario de un caso de uso puede ser representado como se muestra en la figura 6, este formato no es estándar, encontraras con diversos autores formatos diferentes, todos son validos para seleccionar un formato es necesario centrarse en las necesidades que se tienen para elegir el adecuado.

A continuación describiré cada una de las opciones que muestra el formato de la figura 6.

- Nombre de caso de uso (ACCESO AL SISTEMA): En este apartado de debe se escribir en nombre del caso de uso.
- Descripción breve: Se describe que es lo que hace el caso de uso de manera general no debe de ser tan especifico.
- Flujo de eventos: describir de manera breve cada uno de los eventos que el usuario visualizara en primera instancia al inicializar el caso de uso.
- Flujo básico: describir cada uno de los pasos que contempla el caso de uso que serán realizados con éxito.
- Flujos alternativos: Describir las condiciones que deben ser cumplidas de algunas acciones o todas del flujo básico para poder completar la operación o acción, las operaciones que caigan en un flujo alternativo se tienen que numerar de acuerdo al numero que tenga el flujo básico.
- Requerimientos especiales: cualquier requerimiento extra del sistema, asociado al caso de uso especificado.
- Precondición: describir la condición que se debe de cumplir antes de iniciar el caso de uso

- Post_Condition: Describir la condición en la que debe quedar el caso de uso una ves que los pasos han sido realizados con éxito.
- Puntos de extensión: Puntos donde se extiende el caso de uso mediante una relación de <<extend>>.

ACCESO AL SISTEMA
1. DESCRIPCION BREVE: En este caso de uso es iniciado por el administrador quien introduce su usuario y contraseña para asi poder utilizar el sistema
2.Flujo de eventos: Se le presenta al administrador la pantalla inicial del sistema solicitándole su nombre de usuario y contraseña y la opción aceptar y cancelar
2.1 Flujo básico: 1. El usuario selecciona un usuario de la lista que parece en la pantalla, posteriormente se activa la casilla contraseña para que la persona que utilizara el sistema pueda introducir la contraseña. 2. El usuario selecciona la opción “aceptar” 3. Si el nombre de usuario existe y la contraseña son validas la aplicación permite el acceso. 4. Al entrar al sistema se identifica el privilegio de usuario y se le presenta un menú con las acciones que deba realizar
2.2 flujos alternativos 2.2.1 Nombre de usuario y/o contraseña incorrecta Si el nombre de usuario y contraseña es incorrecto el sistema no se le da acceso y manda un mensaje de contraseña no valida y regresa a solicitar los datos requeridos, el sistema permite realizar tres intentos en el acceso si a la tercera vez no se logra acceder sale del sistema.

3. Requerimientos Especiales: Que el usuario este dado de alta.
4. Pre_Condiciones: Que el usuario este registrado previamente en el sistema
5. Post_Condicion: Ninguno
6. Puntos de Extension: Ninguna.

Figura 6. Ejemplo del escenario de un caso de uso denominado Acceso al Sistema

CAPITULO VII

PROCEDIMIENTOS Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.

En esta parte del documento detallare como se aplicaron los métodos, técnicas y tecnología para el desarrollo del proyecto.

Análisis de requisitos

La primera actividad dentro de esta fase fue elaborar un formato de entrevista en donde se abordaron preguntas enfocadas y dirigidas a recolectar la información suficiente sobre la problemática que encerraba la situación actual del proceso de registro de asistencias en la facultad. Se aplico la entrevista al administrador de la facultad ya que el es el que esta encargado de ese proceso. La implementación de la entrevista hizo entender mejor el panorama de la situación y además se descubrieron nuevas necesidades. Con la información obtenida mediante la entrevista se da paso a planear la mejor solución para resolver la problemática.

Teniendo aclarada la situación y determinada la mejor solución para este, se inicio a la realización del estudio de factibilidad para determinar la viabilidad del proyecto, los estudios se hicieron en el ámbito técnico y operático obteniendo como resultado que el proyecto era viable en los dos ámbitos para poder proceder a la creación de este.

Contando con toda la información se determinan los principales requerimientos del sistema obteniendo de esto los principales casos de uso y actores que estarán involucrados con el uso del sistema.

Diseño y arquitectura

En el diseño del sistema como primera actividad se detallaron aun mas los casos de uso eligiendo los que tenían mayor relevancia en el funcionamiento del sistema. Contando con toda esta información se procedió a elaborar los diagramas de caso

de uso en donde se visualiza la iteración que tienen los actores con cada caso de uso.

Teniendo los casos de uso establecidos se procedió a elaborar las especificaciones de cada caso de uso en donde se describieron el flujo de actividades que intervienen al utilizar un caso de uso.

Una vez ya desarrollados los casos de uso, se comenzó a elaborar la base de datos en el manejador de base de datos Access, en las cuales se estructuraron detalladamente cada tabla, campo, catalogo y cada detalle de la misma. Se realizo el diseño de la base de datos, utilizando el modelo Entidad-Relación.

Nombre del campo	Tipo de datos	Descripción
catjus_justificacion	Texto	
catjus_descripcion	Texto	
status	Texto	
observacion	Texto	
fecha	Fecha/Hora	

Tabla 1. Catalogo de justificantes

Nombre del campo	Tipo de datos	Descripción
catol_tipo	Texto	
catol_tiempo_minuto	Número	
status	Texto	
observacion	Texto	
fecha	Fecha/Hora	
movtouser	Texto	

Tabla 2. Catalogo de tolerancia

Nombre del campo	Tipo de datos	Descripción
NumChequeo	Número	
Fecha	Fecha/Hora	
Hora	Fecha/Hora	
Dia	Texto	
TipoChequeo	Texto	Em=Entrada matutino, Sm=Salida matutino, Ev=Entrada vesp, Sv=salida ves
status	Texto	
observacion	Texto	
movtouser	Texto	
Rfc	Texto	
status_incidencia	Texto	V=validado
Tipo_incidencia	Texto	F=Falta, R=Retardo
Fecha_texto	Texto	

Tabla 3. Contiene los registros de asistencias

Nombre del campo	Tipo de datos	Descripción
num_plaza	Texto	
nombre	Texto	
apellido_pat	Texto	
apellido_mat	Texto	
categoria	Texto	
rfc	Texto	
num	Autonumérico	
entrada_mat	Fecha/Hora	
salida_mat	Fecha/Hora	
entrada_vesp	Fecha/Hora	
salida_vesp	Fecha/Hora	
hrs_lab	Fecha/Hora	
sueldo_men	Número	
horario_act	Texto	
sabado_ent	Texto	
tipo_personal	Texto	
area	Texto	
status_imp	Texto	
NumChequeo	Número	
Status	Texto	A=Activo,B=Baja,BT=Baja Temporal,I=Inactivo
Observacion	Memo	

Tabla 4. Contiene los datos de los empleados de la facultad

NumChequeo	Número	
Fecha	Fecha/Hora	
Hora_ENT	Fecha/Hora	
Hora_SAL	Fecha/Hora	
Dia	Texto	
TipoChequeo	Texto	Em=Entrada matutino, Sm=Salida matutino, Ev=Entrada vesp, Sv=salida ves
status	Texto	
observacion	Texto	
movtouser	Texto	
Rfc	Texto	
status_incidencia	Texto	V=validado
Tipo_incidencia	Texto	F=Falta, R=Retardo
Fecha_texto	Texto	
turno	Texto	
tipo_personal	Texto	
Fecha_Reporte	Fecha/Hora	

Tabla 5. Esta tabla guarda los registros para generar el reporte de asistencia diaria

Nombre del campo	Tipo de datos	Descripción
NumChequeo	Número	
Tipo_incidencia	Texto	
Fecha_incidencia	Fecha/Hora	
Dia	Texto	
Fecha_texto	Texto	
Status_incidencia	Texto	J=Justificada
Observaciones	Texto	
turno	Texto	
Origen_incidencia	Texto	

Tabla 6. En esta tabla se registran las incidencias de cada quincena

Nombre del campo	Tipo de datos	Descripción
NumChequeo	Número	
Tipo_Justificacion	Texto	
Fecha_JustificacionIni	Fecha/Hora	
Fecha_JustificacionFin	Fecha/Hora	
Dia	Texto	
Fecha_texto	Texto	
Status_Justificacion	Texto	J=Justificada
Observaciones	Texto	
turno	Texto	
Oficio_ref	Texto	

Tabla 7. En esta tabla se registran todas las justificaciones quincenales

Una vez establecida la arquitectura del sistema se realizo el diseño de las interfaces (pantalla) del software tomando en cuenta los colores de la institución.

Programación

En esta etapa se dio inicio a la codificación de las clases del sistema haciendo funcional las operaciones que el software controla.

Prueba

Se llevaron a cabo la prueba de caja negra, que son las pruebas que se realizan en los sistemas informáticos con el propósito de causar algún tipo de error en el sistema, y de esta manera corregirlos. Estas pruebas tratan especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario.

Documentación

Se realizo el manual de usuario correspondiente, especificando detalladamente el uso del software para que el usuario pueda manejar con seguridad el sistema y sobre todo que haga buen uso de el.

CAPITULO VII

RESULTADOS, PLANOS, GRAFICAS, PROTOTIPOS Y PROGRAMAS

RESULTADOS

Fig. 1 Pantalla principal del sistema

Fig. 2 Pantalla de chequeo

Fig. 3 Pantalla de acceso del Administrador

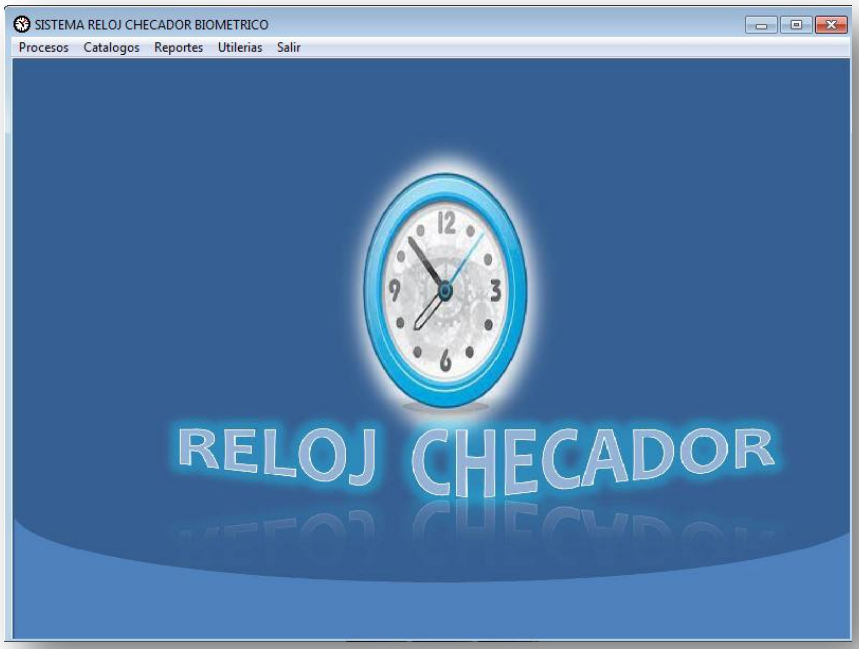


Fig. 4 Pantalla principal del Administrador

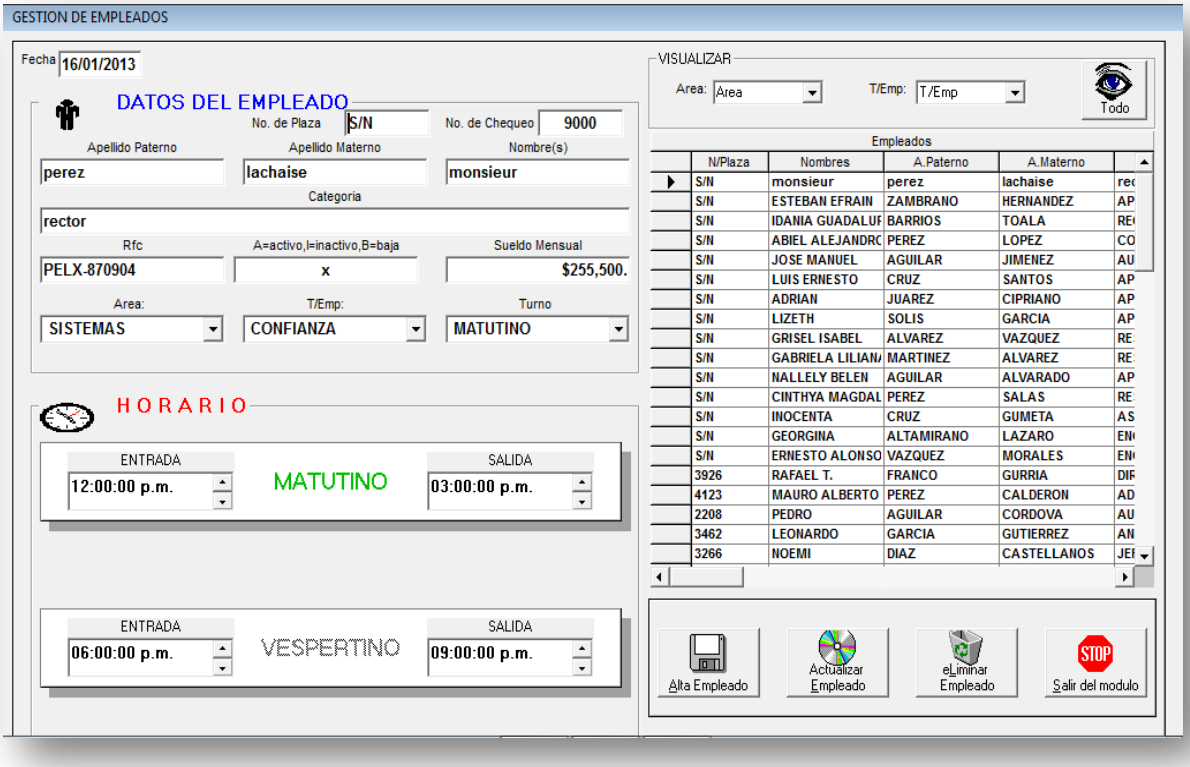


Fig. 5 Pantalla de asignación de horarios, actualizaciones, alta al sistema, bajas y consulta de datos.

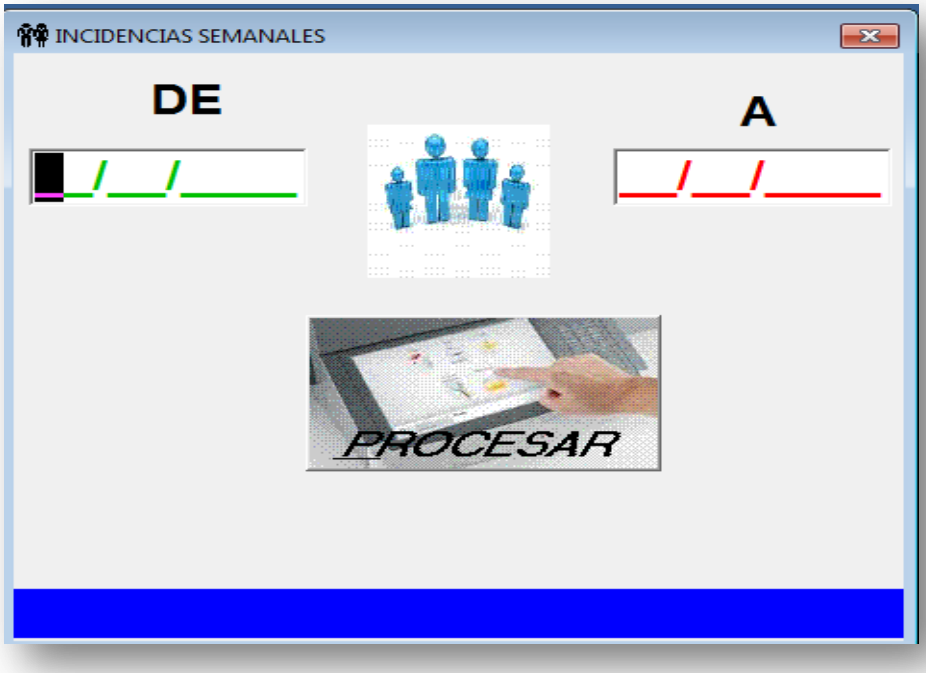


Fig. 6 Pantalla de generación de incidencias

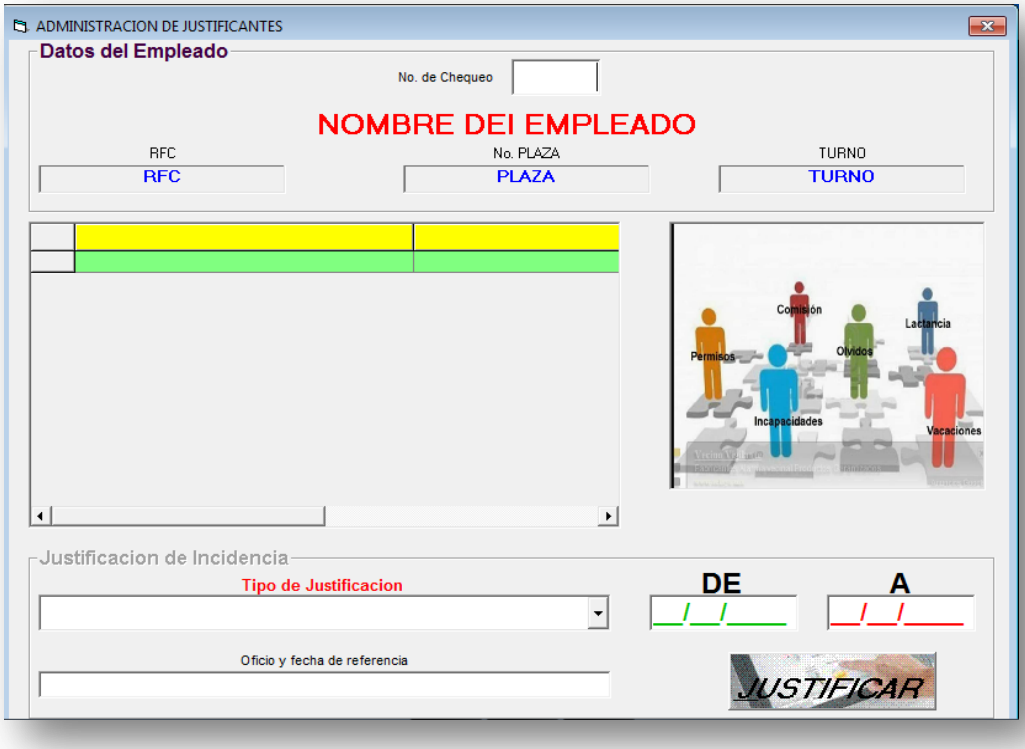


Fig. 7 Pantalla de asignación de justificantes

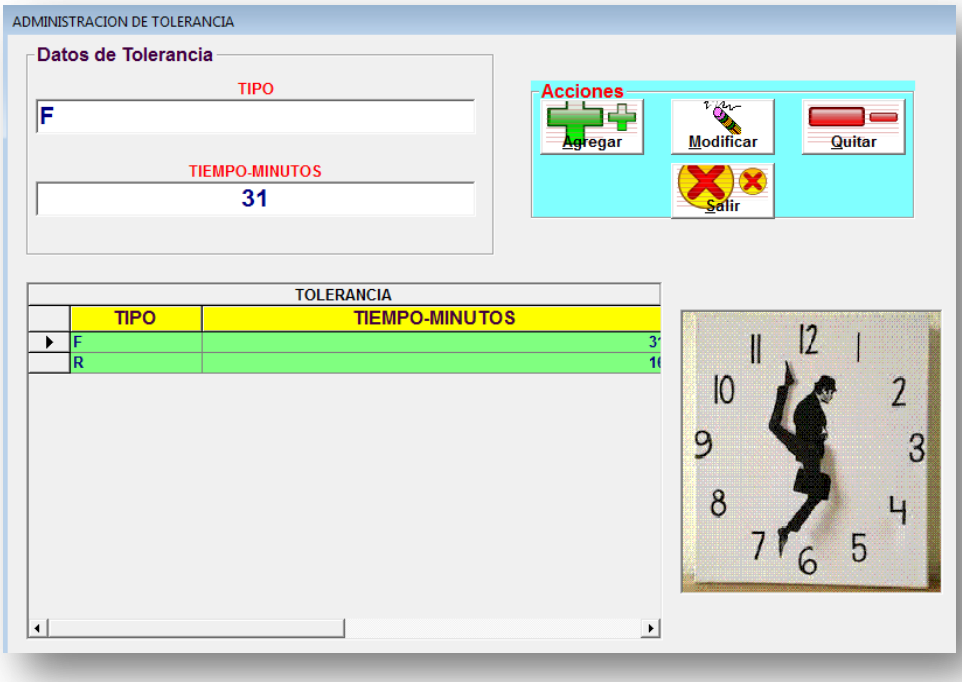


Fig. 8 Catalogo de tolerancia

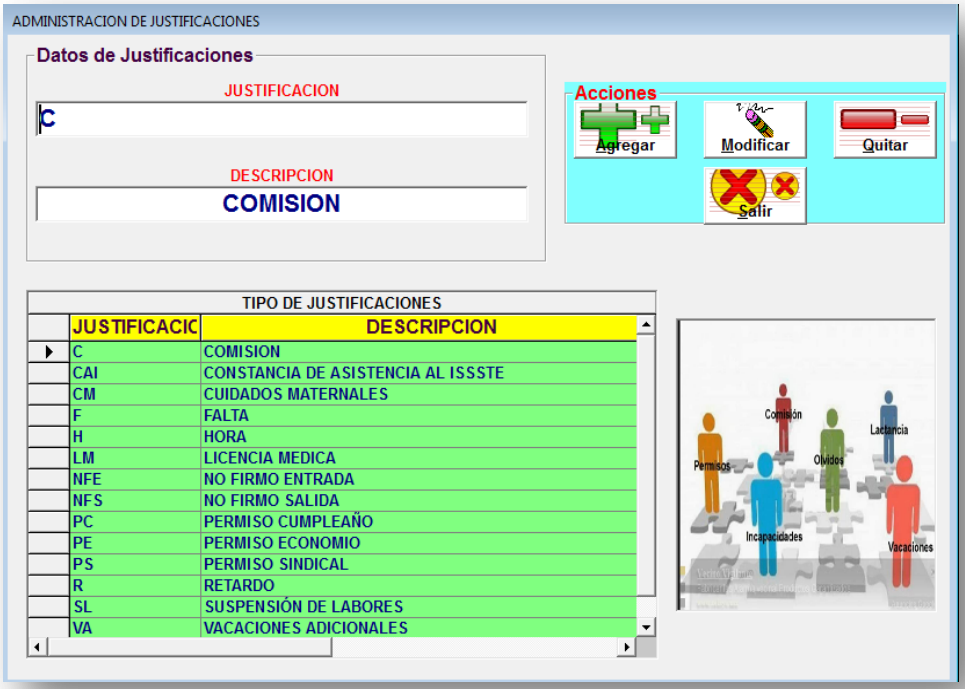


Fig. 9 Catalogo de justificantes

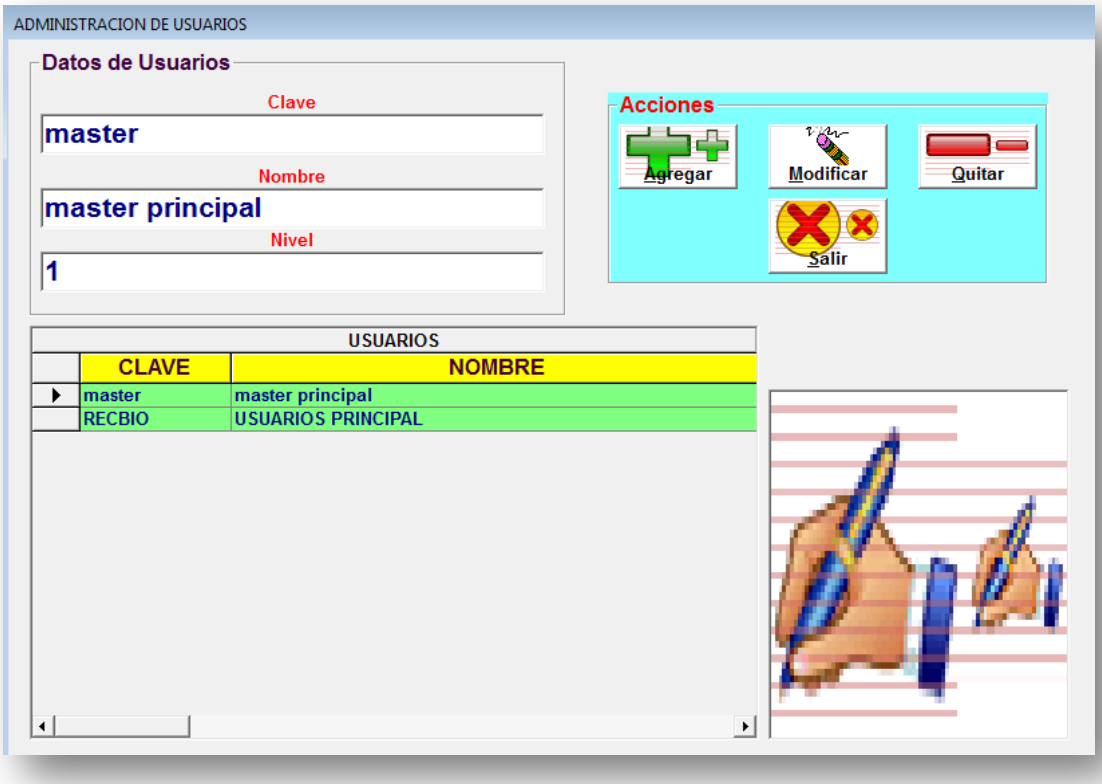


Fig. 10 Catalogo de usuarios

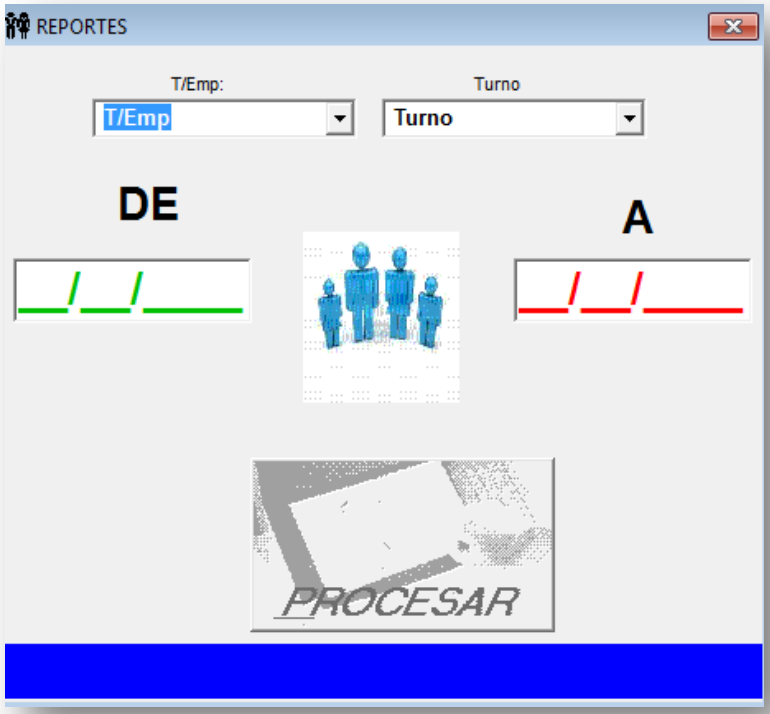
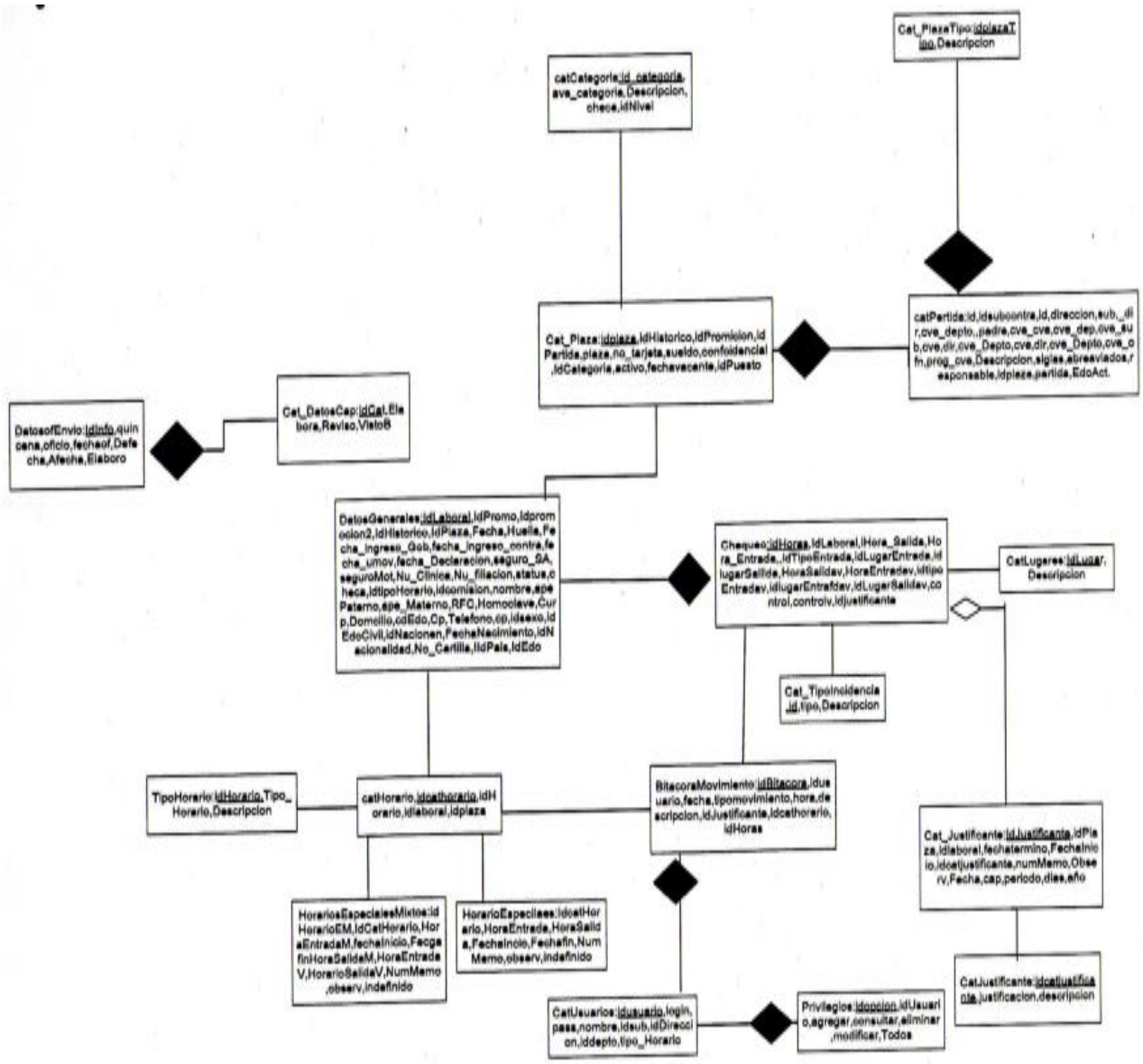


Fig. 11 Pantalla para la generación de reportes

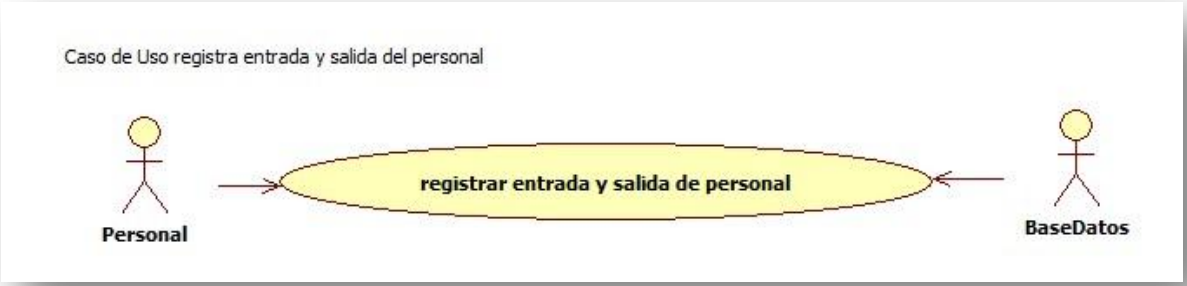
DIAGRAMA ELKA


Cat_PlazaTipo:idplazaTipo,Descripcion
catCategoria:id_categoria,ave_categoria,Descripcion,checa,idNivel
Cat_Plaza:idplaza,idHistorico,idPromocion,idPartida,plaza,no_tarjeta,sueldo,confidencial,idCategoria,activo,fechavacante,idPuesto
catPartida:id,idsubcontra,id,direccion,sub,_dir,cve_depto.,padre,cve_cve,cve_dep,cve_sub,cve,dir,cve_Depto,cve,dir,cve_Depto,cve_cfn,prog_cve,Descripcion,siglas,abreaviados,responsable,idplaza,partida,EdoAct.
Cat_DatosCap:idCat,Elabora,Reviso,VistoB
DatosofEnvio:idinfo,quincena,oficio,fechaof,Defecha,Afecha,Elaboro
DatosGenerales:idLaboral,idPromo,Idpromocion2,IdHistorico,IdPlaza,Fecha,Huella,Fecha_ingreso_Gob,fecha_ingreso_contra,fecha_umov,fecha_Declaracion,seguro_SA,seguroMot,Nu_Clinica,Nu_filiacion,status,checa,idtipoHorario,idcomision,nombre,apePaterno,ape_Materno,RFC,Homoclave,Curp,Domicilio,cdEdo,Cp,Telefono,op,idsexo,idEdoCivil,idNacionen,FechaNacimiento,idNacionalidad,No_Cartilla,IldPais,IdEdo
Chequeo:idHoras,IdLaboral,iHora_Salida,Hora_Entrada,,IdTipoEntrada,idLugarEntrada,idlugarSalida,HoraSalidav,HoraEntradav,idtipoEntradav,idlugarEntradav,idLugarSalidav,control,controlv,idjustificante
CatLugares:idLugar,Descripcion
Cat_TipoIncidencia:id,tipo,Descripcion
TipoHorario:idHorario,Tipo_Horario,Descripcion
catHorario,idcathorario,idHorario,idlaboral,idplaza
BitacoraMovimiento:idBitacora,idusuario,fecha,tipomovimiento,hora,descripcion,idJustificante,idcathorario,idHoras
Cat_Justificante:idJustificante,idPlaza,idlaboral,fechatermino,FechaInicio,idcatjustificante,numMemo,Observ,Fecha,cap,periodo,dias,año
HorariosEspecialesMixtos:idHorarioEM,IdCatHorario,HoraEntradaM,fechaInicio,FecgfinHoraSalideM,HoraEntradaV,HorarioSalideV,NumMemo,observ,indefinido
HorarioEspeciales:IdcatHorario,HoraEntrada,HoraSalida,FechaInicio,Fechafin,NumMemo,observ,indefinido
CatJustificante:idcatjustificante,justificacion,descripcion
CatUsuarios:idusuario,login,pass,nombre,idsub,idDireccion,iddepto,tipo_Horario
Privilegios:idopcion,idUsuario,agregar,consultar,eliminar,modificar,Todos

DIAGRAMAS DE CASO DE USO

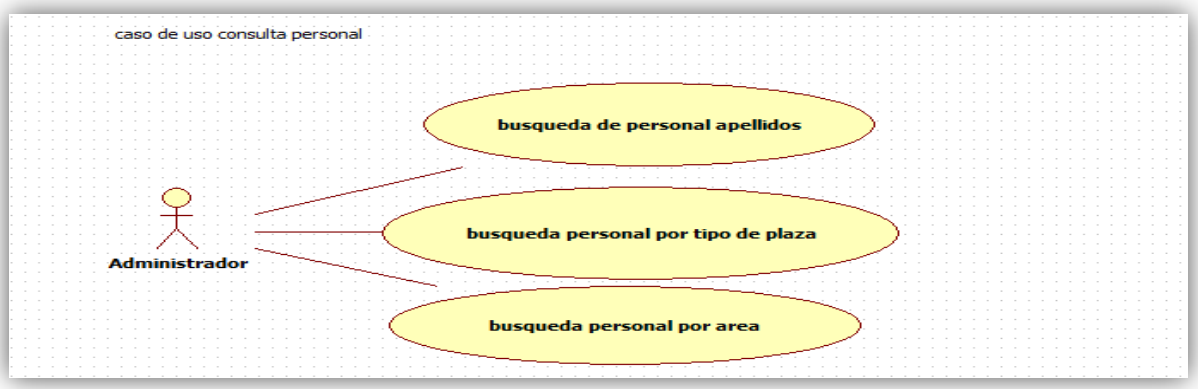
ACCESO AL SISTEMA
1. Descripción breve: Este caso de uso es iniciado por el administrador quine introduce el nombre de usuario y password para ingresar al sistema.
2. Flujo de eventos: Se le presenta al administrador la pantalla inicial del sistema solicitándole su nombre de usuario y contraseña y la opción aceptar y cancelar.
2.1 Flujo básico: 1. El usuario introduce el nombre de usuario y el password . 2. Selecciona “Entrar”. 3. Si el nombre de usuario y contraseña sonvalidas la aplicación permite el acceso al sistema. 4. Al entrar al sistema se identifica el privilegio de usuario y se le presenta un menú con las acciones que beba realizar.
2.2 Flujos alternativos: 2.2.1 Nombre de usuario y/o contraseña incorrecta

Si el nombre de usuario y contraseña es incorrecto el sistema no le da acceso y manda un mensaje de contraseña no valida y regresa a solicitar los datos requeridos, el sistema permite realizar tres intentos, si a la tercera vez no se logra acceder sale del sistema.
3. Requerimientos especiales: Que el usuario este dado de alta
4. Precondiciones: Que el usuario este registrado previamente en el sistema.
5. Postcondicion: Ninguna
6. Puntos de extensión: Ninguna



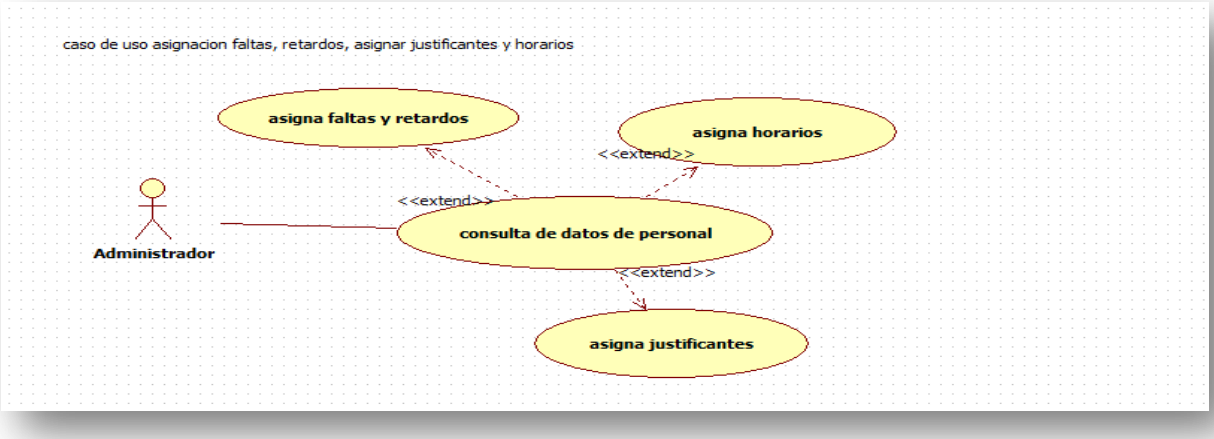
REGISTRO DE ENTRADA Y SALIDA
1. Descripción breve: Este caso de uso se inicia cuando el personal registra su hora de entrada y salida
2. Flujo de eventos: Se presenta al personal la pantalla “Checar”
2.1 Flujo básico: 1. El usuario visualiza la pantalla de registro de asistencia. 2. El usuario introduce la pantalla en la terminal para registrar su asistencia. 3. La terminal identifica la huella y envía imagen a la base de datos. 4. Base de datos verifica existencia de la huella en la tabla de empleados. 5. Se registra la hora de entrada en la base de datos. 6. El usuario visualiza los siguientes datos en la pantalla: Nombre, No. Plaza, Huella y hora de chequeo.
2.2 Flujos alternativos: 2.2.1 Personal no dado de alta Si el sistema no encuentra los registros del personal manda un mensaje “Personal no existe, verifique con el administrador”.

3	Requerimientos especiales: Ninguna
4	Precondiciones: Ninguna
5	Postcondicion: Ninguna
6	Puntos de extensión: Ninguna



CONSULTA DE PERSONAL
1. Descripción breve: Este caso de uso se inicia cuando el administrador desea consultar datos de una persona en específico.
2. Flujo de eventos: Se presenta al personal la pantalla “Asignar”
2.1 Flujo básico: 1. El administrador visualiza selecciona la opción “Asignar” 2. El administrador visualiza los datos generales de un personal.
2.2 Flujos alternativos: Ninguno
3. Requerimientos especiales: Ninguna
4. Precondiciones:

Deben existir registros almacenados en la base de datos para poder generar la lista de todo el personal.
5. Postcondicion: Ninguna
6. Puntos de extensión: Ninguna



ASIGNACION DE HORARIOS
1. Descripción breve: Este caso de uso se inicia cuando el administrador asigna el horario al personal
2. Flujo de eventos: Se presenta al personal la pantalla “Asignar”
2.1 Flujo básico:

1. El administrador selecciona la opción Asignar de la pantalla principal del sistema.
2. El administrador selecciona a un usuario de la lista que aparece en pantalla y los datos del usuario se ven reflejados a un costado de la lista.
3. Posteriormente se dirige a un apartado de la pantalla y asigna los siguiente datos: tipo de horario, hora de entrada , hora de salida, observaciones.
4. El administrador captura datos y posteriormente selecciona la opción “Asignar”.

2.2 Flujos alternativos:

2.2.1 Cancelar operación

2.2.2 Falta registro por llenar

Si el administrador deja un campo en blanco cuando seleccione la opción “Asignar”, el sistema enviara un mensaje “Faltan Datos, verifique”.

3. Requerimientos especiales:

El administrador debe capturar todos los datos solicitados.

4. Precondiciones:

Ninguna

5. Postcondicion:

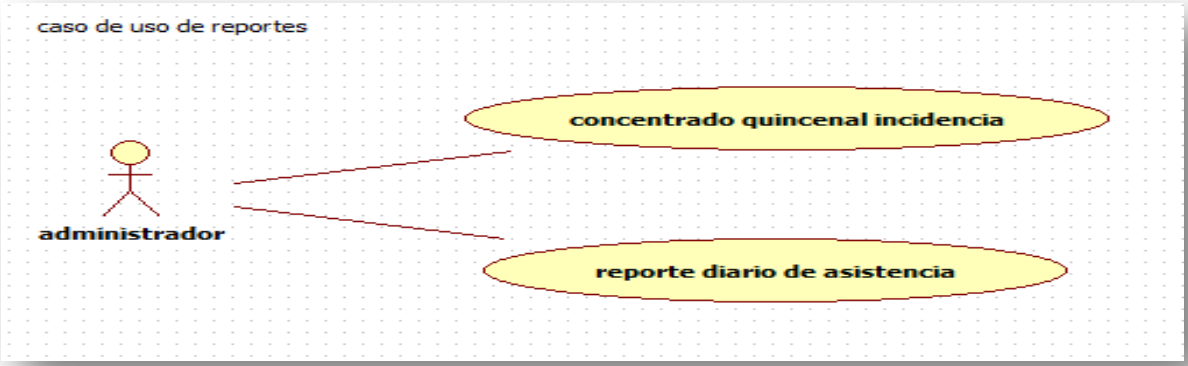
Ninguna

6. Puntos de extensión:

Ninguna

ASIGNAR JUSTIFICANTES
1. Descripción breve: Este caso de uso se inicia cuando el administrador tiene que asignar un justificante de un personal.
2. Flujo de eventos: Se presenta al personal la pantalla de asignación de justificantes.
2.1 Flujo básico: 1. El administrador selecciona la opción Asignar de la pantalla principal del sistema, enseguida la opción “Asignar justificante”. 2. El administrador visualiza la pantalla asignar justificantes con los siguientes datos: Nombre, No. Plaza, Tipo justificante, Fecha inicio, Fecha fin, Observaciones. 3. El administrador captura cada uno de los datos y selecciona la opción “Asignar”
2.2 Flujos alternativos: 2.2.1 Falta registro por llenar Si el administrador deja un campo en blanco cuando seleccione la opción “Asignar”, el sistema enviara un mensaje “Faltan Datos, verifique”. 2.2.2 Selecciona opción Cancelar Se cierra la pantalla de Asignar justificante y se regresa a la pantalla principal del sistema. Si aun no se ha seleccionado la opción agregar la información capturada se perderá.
3 Requerimientos especiales: Ninguna

4 Precondiciones: Ninguna
5 Postcondicion: Ninguna
6 Puntos de extensión: Ninguna



CONCENTRADO QUINCENAL DE INCIDENCIA
1. Descripción breve: Este caso de uso de inicia cuando el administrador desea generar un reporte del concentrado quincenal de inasistencia de un los empleados que laboran en la facultad.
2. Flujo de eventos: Se presenta al administrador, pantalla principal del sistema.
2.1Flujo básico:

1. El administrador selecciona la opción Reporte posteriormente elige la opción entradas y salidas, enseguida la opción global. 2. El administrador visualiza el reporte selecciona imprimir.
1.2 Flujos alternativos: 1.2.1 El administrador selecciona la opción “cerrar” Se cierra la pantalla del reporte y se regresa a la pantalla principal del sistema. 1.2.2 No existen datos El sistema manda el siguiente mensaje “ no existen registros en la fecha que ha especificado”, cuando no encuentre en la base de datos dentro del rango de fecha s que se ha especificado.
2. Requerimientos especiales: Ninguna
3. Precondiciones: 4. Deben existir registros en la base de datos para poder iniciar este caso de uso . 5. La impresora debe estar encendida.
6. Postcondicion: Ninguna
7. Puntos de extensión: Ninguna

DIAGRAMAS DE SECUENCIA

DIAGRAMA ACCESO AL SISTEMA

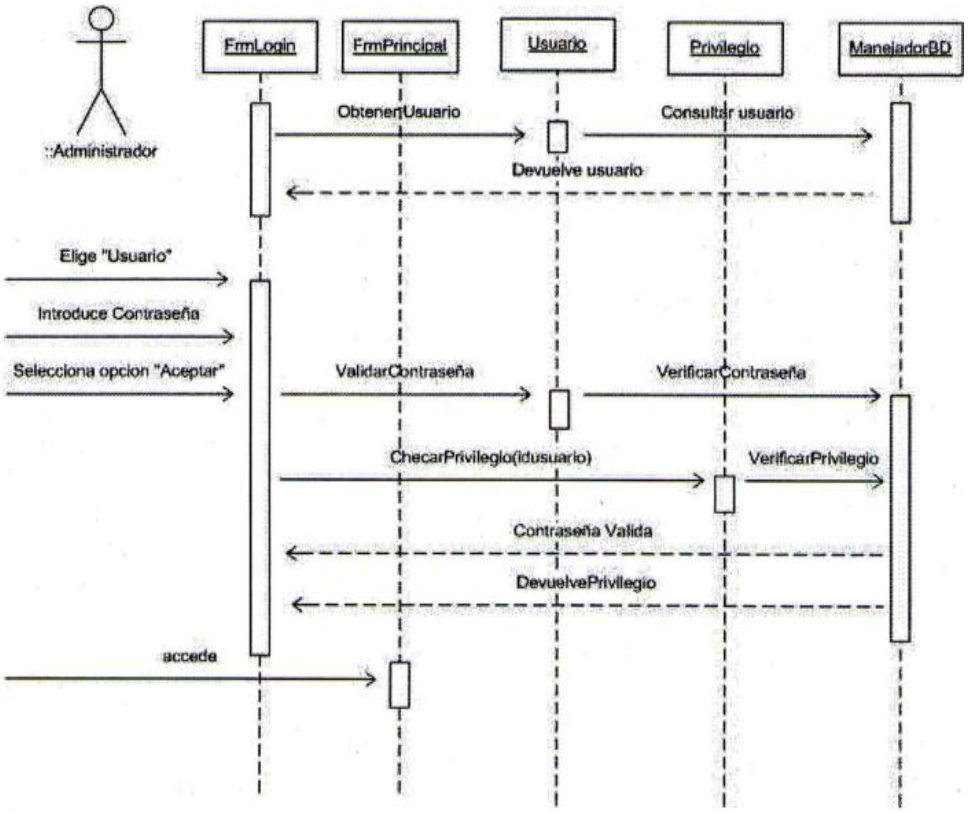


DIAGRAMA DE SECUENCIA ENTRADA Y SALIDA

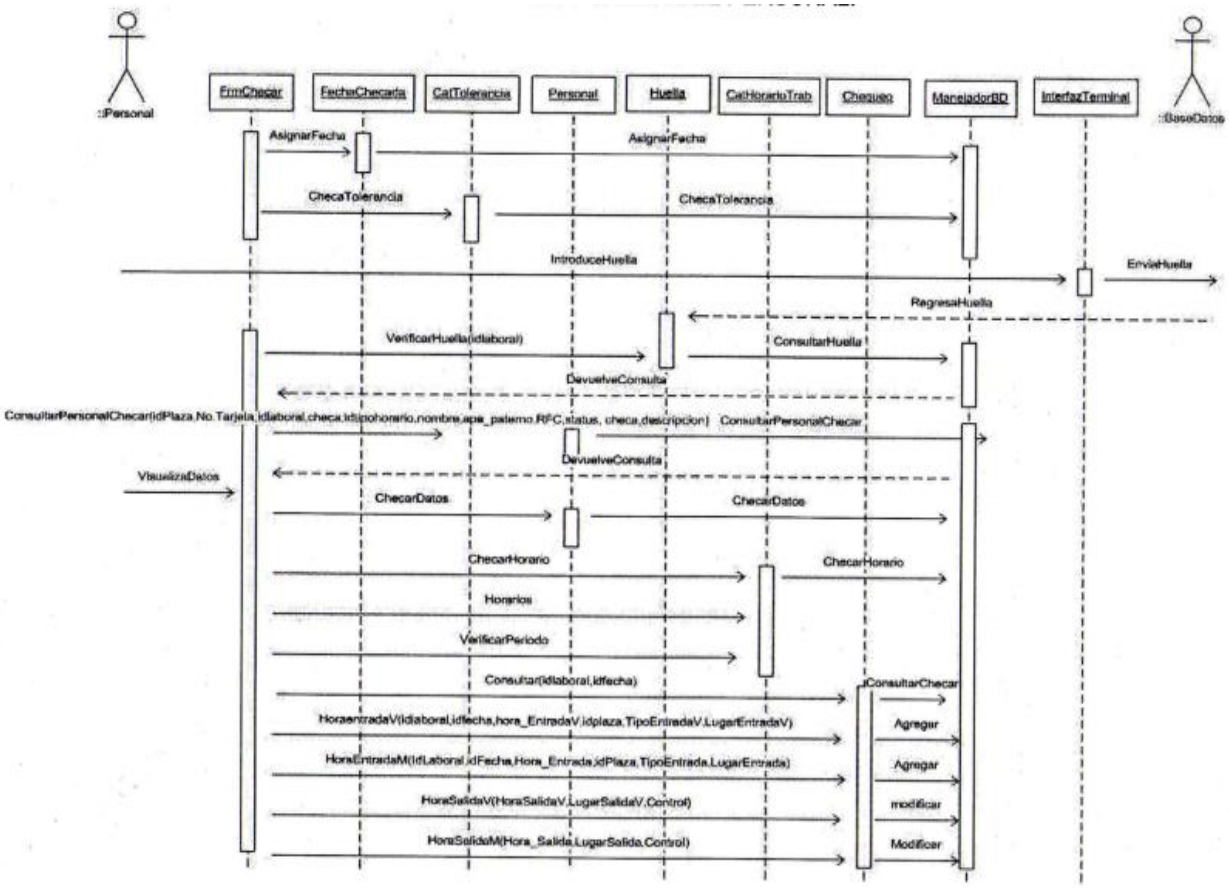


DIAGRAMA DE SECUENCIA ASIGNACION DE HORARIOS

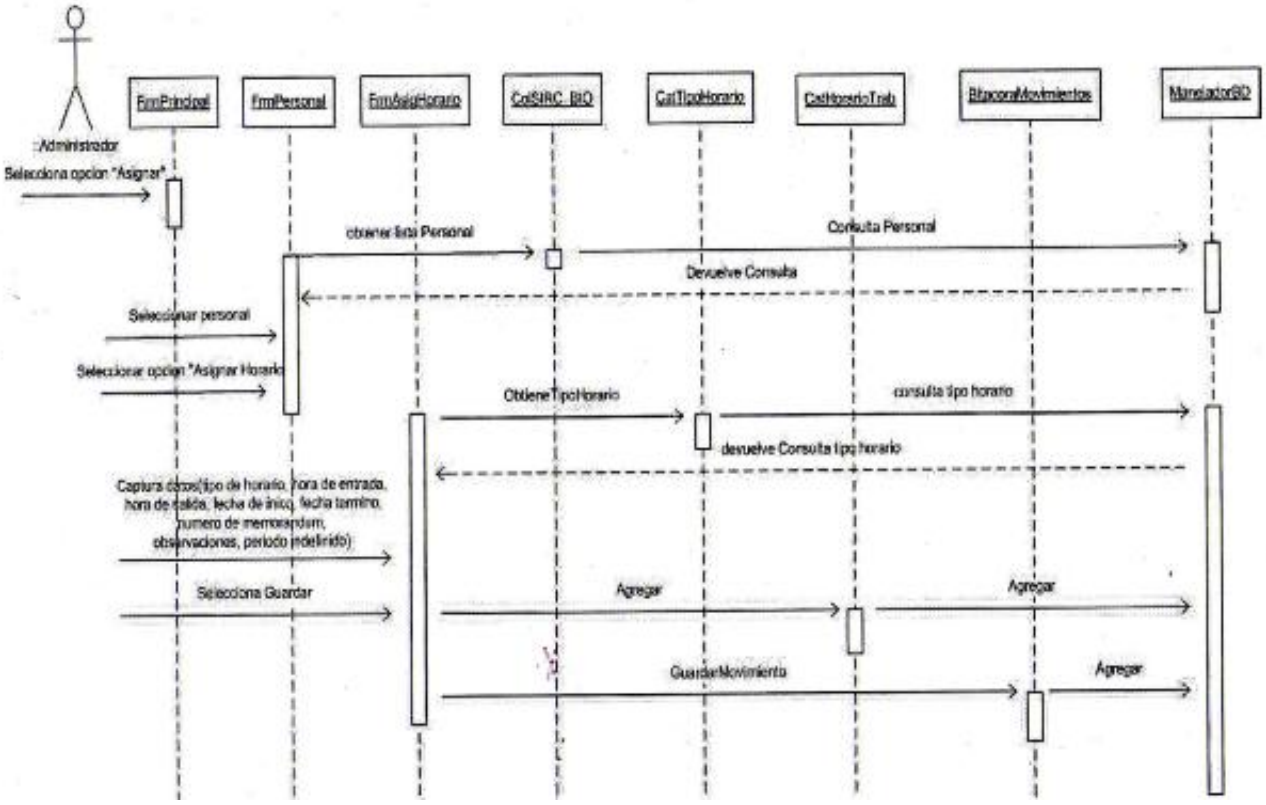


DIAGRAMA DE SECUENCIA CONSULTA DE PERSONAL

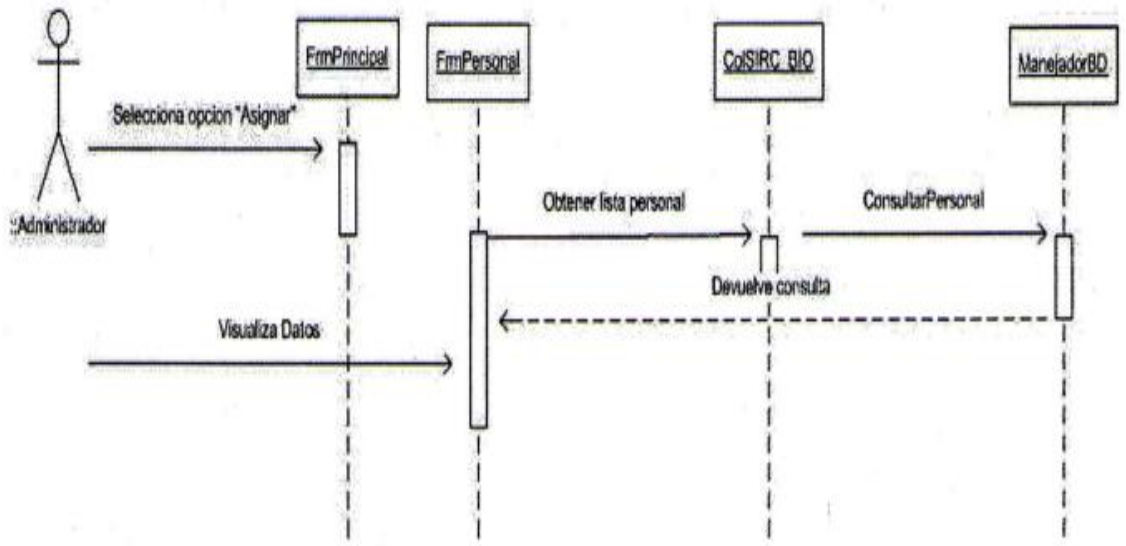


DIAGRAMA DE SECUENCIA CATALOGO DE JUSTIFICANTES

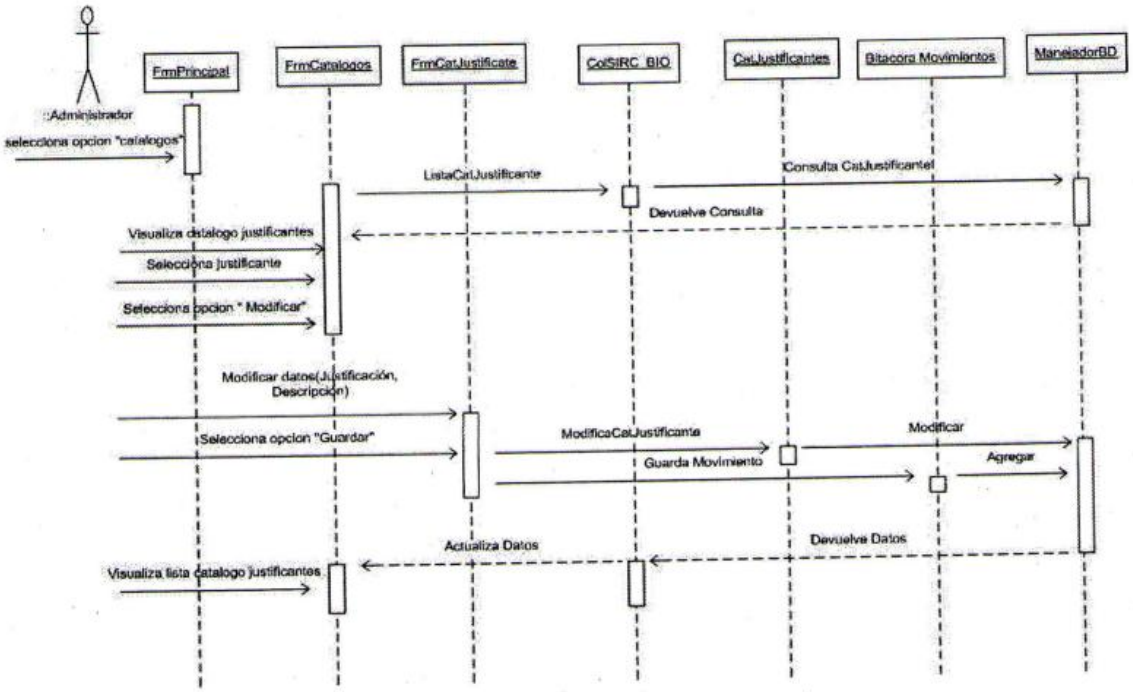
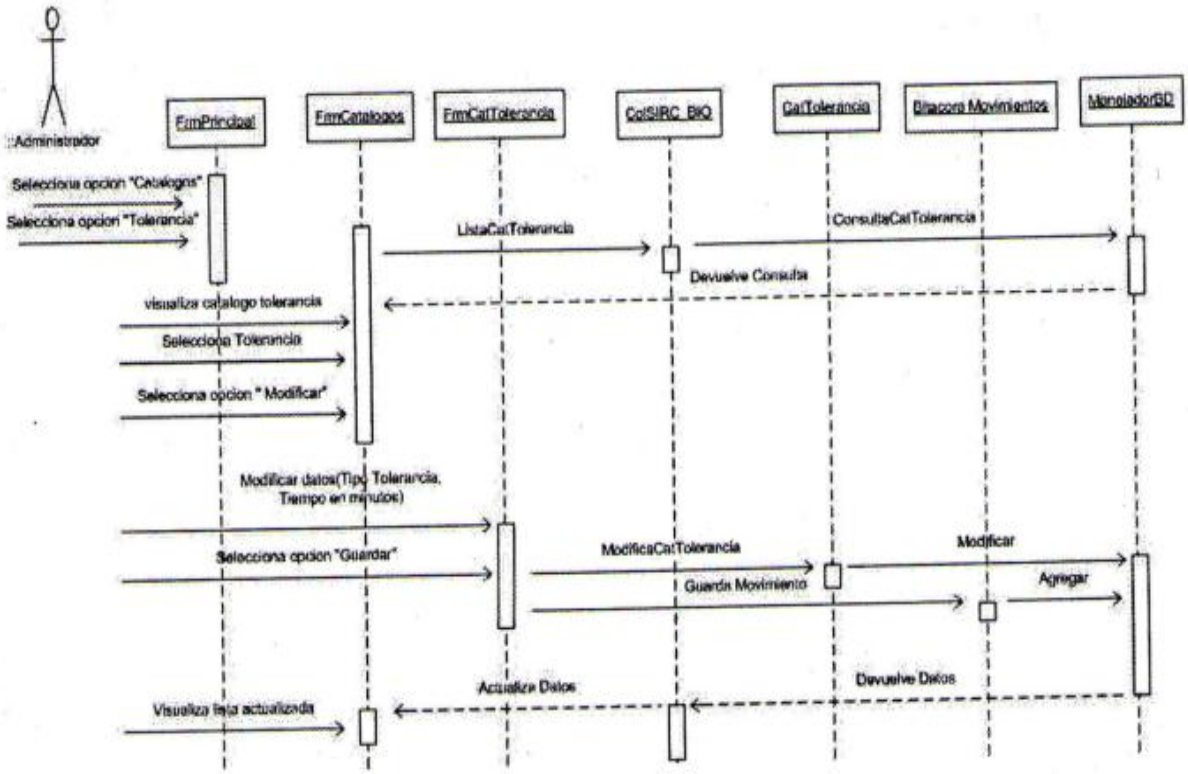
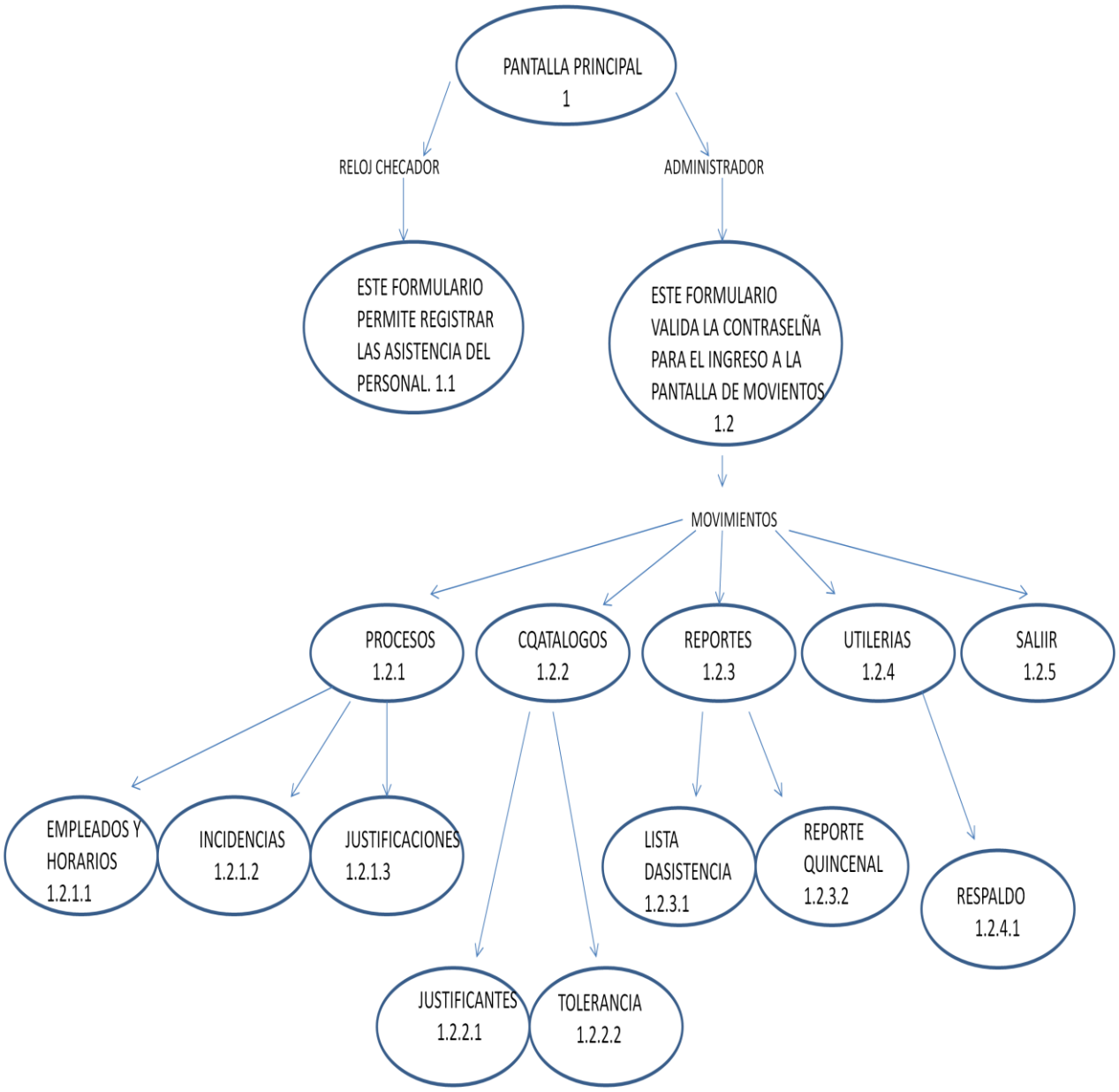


DIAGRAMA DE SECUENCIA CATALOGO DE TOLERANCIA



DISEÑO ESTRUCTURAL



CONCLUSIONES Y RECOMENDACIONES

Sin duda el apoyarse de los sistemas de información es de gran ayuda se hace imprescindible su utilización, contar con estas herramientas que ayudan a obtener información precisa y confiable para la toma de decisiones que auxilien en el rendimiento de las empresas.

La utilización de papeles en las oficinas es vital para los procesos que se utilizan pero sin tener en cuenta el daño que se le hace al medio ambiente, es importante aprovechar estas herramientas que nos ofrece la tecnología como lo es la documentación digital y así crear oficinas sin papel.

El avance de la tecnología hacen posible el mundo de herramientas disponibles para desarrollar trabajos como planos, diseños en 3D, prototipos, que facilitan la elaboración y la comprensión del mismo anudado a eso reducen en gran medidas los costos de tiempo y económicos.

Las recomendaciones dadas a la empresa para el buen funcionamiento del Sistema Reloj Checador Biométrico son las siguientes:

- Mantenimiento al software y al hardware
- Realizar respaldo de la base de datos
- Actualización de antivirus
- Al realizar una restauración o respaldo de lavase de datos es necesario que se comunique al personal sobre esta situación porque no se podrán realizar movimientos en ese momento.

REFERENCIA BIBLIOGRAFICA

C.J Date. "*Introduccion a los sistemas de base de datos".*2001. Prentice Hall.

Elmasrsi y Navathe. *"Sistemas de base de datos.Conceptos Fundamentales"*. Addison-Wesley Iberoamericana.

Cohen Karen Daniel y Asin Lares Enrique. "*Sistemas de información para los negocios. Un enfoque de toma de desiciones".*

E Kendall Kennet y E Kendall Julie. *"Analisis y Diseños de sistemas".* 2005. Pearson Education.

Larman Craig. "*Uml y Patrones. Introduccion al análisis y diseño orientado a objetos"*. Prentice Hall.

Schmuller Joseph. *"Uml en 24 hrs".* Prentice Hall.

Sommerville Ian. "*Ingenieria del software*". 2005. Pearson Addison Wesley.

S. Presman Roger. "*Ingenieria del software. Un enfoque Practico".*

PETER ROB/CARLOS CORONEL 5ª edición "*sistema de base de datos Diseño, implementación, y administración."*

Cavsi, ¿Qué es un sistema gestor de base de datos (SGBD)?, de Http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-base-de-datos-o-sgbd/

ANEXOS

CUESTIONARIO

- ¿Cuál es el problema que se tiene la empresa?
- Detalles del problema
- Importancia del problema
- ¿Cuál cree el solicitante que puede ser la solución al mismo?
- ¿En qué forma ayuda un sistema de información?
- Breve resumen de los reportes usados y funciones que se realizan
- ¿Qué otras personas tienen conocimiento del problema y que se pueden contactar?
- ¿Existe apoyo suficiente para el proyecto por parte de la administración?, ¿Y por parte de los usuarios?
- Los métodos que actualmente se usan en la empresa, ¿son aceptados por los usuarios?
- ¿El sistema propuesto causará perjuicios?
- ¿Producirá resultados pobres en alguna área?
- ¿Se perderá control en alguna área específica?
- ¿Se perderá la facilidad de acceso a la información?
- ¿La productividad de los empleados será menor después de instalado el sistema?
- ¿Existe o se puede adquirir la tecnología necesaria para realizar lo que se pide?
- ¿El equipo propuesto tiene la capacidad técnica para soportar todos los datos requeridos para usar el nuevo sistema?
- ¿El sistema propuesto ofrecerá respuestas adecuadas a las peticiones sin importar el número y ubicación de los usuarios?
- Si se desarrolla el sistema, ¿se puede crecer con facilidad?
- ¿Existen garantías técnicas de exactitud, confiabilidad, facilidad de acceso y seguridad de los datos?

- ¿Cuál es el proceso básico de la empresa?
- ¿Qué datos utiliza o produce este proceso?
- ¿Cuáles son los límites impuestos por el tiempo y la carga de trabajo?
- ¿Qué controles de desempeño utiliza?
- ¿Cuál es la finalidad de la actividad dentro de la empresa?
- ¿Qué pasos se siguen para realizarla?
- ¿Dónde se realizan estos pasos?
- ¿Quiénes los realizan?
- ¿Cuánto tiempo tardan en efectuarlos?
- ¿Con cuánta frecuencia lo hacen?
- ¿Quiénes emplean la información resultante?
- ¿Cuántos empleados laboran para la organización en el área(s) que se pretende desarrollar el sistema; o sea, cuántos tienen relación directa con el proyecto que se está investigando. ?
- ¿Cuáles son las personas claves en el sistema? ¿Por qué son importantes?
- ¿Existen obstáculos o influencias de tipo político que afectan la eficiencia del sistema?
- ¿Existen manuales de procedimientos, políticas o lineamientos de desempeño documentados oficial o no oficialmente?. Si los hay, ¿Se cumplen en forma cabal en el 100% de las ocasiones?, es decir, ¿se respetan dichos procedimientos?
- ¿Existen métodos para evadir el sistema?, ¿Por qué se presentan?
- ¿Qué tan frecuente es el ciclo con el que se desarrolla dicha actividad?
- De acuerdo al ciclo con el que se presenta la actividad, ¿Cuál es el volumen de información que aquí se procesa?
- ¿Qué pasos, sub-procesos, o funciones constituyen la actividad? (describir la actividad paso a paso)
- ¿Existe algún tipo de control desarrollado en el proceso en cuestión?

- ¿De dónde proviene la información que se utiliza en esta actividad? (*fuentes*)
- ¿Cuáles son específicamente los datos que recibe esta actividad? (*dts de flujos*)
- ¿De qué manera ingresan a este proceso? (*flujos*)
- ¿Qué tablas de referencia y diagramas u otros datos intervienen en la actividad? (*documentación involucrada*)
- ¿Qué información se genera en esta actividad? (*producto de la actividad*)
- El resultado identificado anteriormente producto de los datos que se procesan ¿Hacia qué o quién van dirigidos? -persona o entidad- (*destinos)*
- ¿Con qué finalidad la utilizan?
- ¿Cuáles datos se conservan o almacenan en este proceso? Y ¿en qué forma quedan almacenados?
- ¿Existe información que se genera pero que no es utilizada nunca por nadie? (*partes extrañas*)
- ¿Cómo se deben procesar los datos para producir la información necesaria?
- ¿Cómo debe presentarse la información.

UNIVERSIDAD AUTÓNOMA DE CHIAPAS

Facultad de Contaduría y Administración, Campus I
Gestión 2010 – 2014
"Construyendo Valores y Actitudes"

DIRECCIÓN

Tuxtla Gutiérrez, Chiapas.
Enero 16 del 2013.
Oficio No. D/012/2013

ASUNTO: CARTA DE TERMINACIÓN

ING. RODRIGO FERRER GONZÁLEZ
JEFE DEL DEPARTAMENTO DE
GESTIÓN TECNOLÓGICA Y VINCULACIÓN
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ
CIUDAD

Por este medio, informo a usted que la **C. ANA KAREN YOC TRUJILLO**, con número de control 08270938 de la Licenciatura en Informática del Instituto Tecnológico de Tuxtla Gutiérrez ha cumplido de manera satisfactoria 640 horas de Residencia Profesional en la Secretaría Administrativa de esta Facultad con el proyecto denominado Reloj Checador Biométrico, durante el periodo Agosto-Diciembre del 2012.

Se extiende la presente constancia, a petición de la parte interesada y para todos los fines a que esta tenga lugar, a los dieciséis días del mes de enero del año don mil trece.

RECIBIDO
24 ENE 2013
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

ATENTAMENTE
"POR LA CONCIENCIA DE LA NECESIDAD DE SERVIR"

DR. RAFAEL T. FRANCO GURRÍA
DIRECTOR

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN C-I
DIRECCIÓN

C.c.p. Lic. Mauro Alberto Pérez Calderón.- Administrador.- Edificio.
Alumno.
Archivo/minutario.
*AAPL/HTFG.

P. FCA-01

Boulevard Belisario Domínguez, Km 1051, Sin Número
Tuxtla, Tuxtla Gutiérrez, Chiapas, México, C.P. 29050,
Tel. 32 (961) 61 5 84 46 y 61 5 84 79

INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE CIENCIAS ECONÓMICO ADMINISTRATIVAS
SEGUIMIENTO DE PROYECTO DE RESIDENCIAS PROFESIONALES

ALUMNO: ANA KAREN YOC TRUJILLO — No. DE CONTROL: 09270093

NOMBRE DEL PROYECTO: RELOJ CHECADOR BIOMETRICO — EMPRESA: FACULTAD DE CONTADURIA Y ADMINISTRACION C.I. DE LA UNIVERSIDAD AUTONOMA DE CHIPAS

ASESOR EXTERNO: DR. RAFAEL TIMOTEO FRANCO GURREA — ASESOR INTERNO: MC. ROBERTO CRUZ GORDILLO

PERIODO DE REALIZACIÓN: AGOSTO-DICIEMBRE 2012

ACTIVIDAD		SEMANAS 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ANALISIS DE REQUISITOS	P	x	x													
	R															
ESPECIFICACION	P			x												
	R															
DISEÑO Y ARQUITECTURA	P				x	x	x	x								
	R															
PROGRAMACION	P								x	x	x	x				
	R															
PRUEBA	P												x	x		
	R															
DOCUMENTACION	P														x	x
	R															
MANTENIMIENTO	P															x
	R															
OBSERVACIONES		Septiembre 24-25							Octubre 22-23				Noviembre 21-22			
ENTREGA DE REPORTES	Docente															
	Alumno															
SNEST-AD-PO-007-05	Jefe Depto.															Rev.5