

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA: INGENIERÍA DE SISTEMAS

**Tesis previa a la obtención del título de: INGENIERA E INGENIERO DE
SISTEMAS**

TEMA:

**ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO
HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA
DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE
IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE
INFORMACIÓN**

AUTORES:

**JIMENA ELIZABETH MORALES TIMBELA
SANTIAGO PAÚL MOYA PORTILLA**

DIRECTOR:

FRANKLIN EDMUNDO HURTADO LARREA

Quito, mayo de 2015

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO DEL
TRABAJO DE TITULACIÓN**

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, mayo de 2015

Jimena Elizabeth Morales Timbela
C.C. 1720403177

Santiago Paúl Moya Portilla
C.C. 1714542048

DEDICATORIA

En la vida es indispensable contar con el apoyo de las personas que nos quieren porque junto a ellas podemos salir adelante y finalmente encontrar nuestros objetivos.

Este trabajo está enteramente dedicado a mi madre; Blanca Portilla que gracias a su esfuerzo diario, lucha constante y enseñanzas ha inculcado en mi persona el deseo de luchar para salir adelante y conseguir los más grandes objetivos.

A mi familia, mis hijos, que son el impulso para día a día avanzar y alcanzar los objetivos propuestos y el deseo de superación, para fomentar la mejor herencia de la vida que es la educación.

Santiago Paúl Moya Portilla

Todo el trabajo va dedicado a mi familia que ha sido el pilar fundamental para mi vida, en especial para mis padres e hijas que con su apoyo incondicional y sus pequeñas risas hicieron que siga en la lucha para culminar una fase de mi vida.

Jimena Elizabeth Morales Timbela

AGRADECIMIENTO

A la Universidad Politécnica Salesiana que ha sido una parte importante en nuestras vidas y que gracias a su vocación de enseñanza han ayudado a muchas personas como nosotros a ser buenos profesionales conjuntamente con nuestro Tutor que ha dado lo mejor de sí para guiarnos en la consecución de este proyecto y nos ayudó no solo en lo profesional si no en las experiencia de la vida diaria.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1.....	2
MARCO TEÓRICO	2
1.1. Ingeniería de software.....	2
1.1.1. Metodología	3
1.1.1.1 Proceso Unificado de desarrollo de Software (RUP)	4
1.1.1.2 Lenguaje Unificado de Modelado (UML)	7
1.1.1.3 Notación para el Modelado de Procesos de Negocios (BPMN).....	8
1.1.1.4 Arquitectura de software.....	10
1.1.2. Sistemas informáticos	13
1.1.2.1 Programación de Escritorio.....	14
1.1.2.2 Programación web.	15
1.1.2.3 Java Script y Kendo de Telerik.....	17
1.1.2.4 Administración y Gestión de Base de Datos.....	19
1.2. Procesamiento de imagen y visión artificial	22
1.2.1 Procesamiento digital de Imágenes.....	22
1.2.2 Manejo de una Imagen.....	24
1.2.3 Muestreo y cuantificación.....	24
1.2.4 Binarización de la imagen.....	25
1.2.5 Reconocimiento de rostros.....	28
1.3. Manejo de horarios de una empresa.....	33
1.3.1 Reglamento ingreso y salida de personal.....	34
CAPÍTULO 2.....	36
ANÁLISIS Y DISEÑO DEL SISTEMA	36
2.1. Situación Actual de la Empresa	36
2.2. Propuesta de la solución implementando un sistema informático	37
2.2.1 Levantamiento de información del Negocio (Fase de Inicio de RUP).	38
2.2.1.1 Documento Visión	39
2.2.1.2 Modelo de Negocios / Diagrama de procesos.....	40
2.2.1.3 Diagrama para modelo de negocios CityVenta.	42

2.2.1.4	Diagrama de modelo de objetos.....	47
2.3.	Análisis (Primera etapa fase de elaboración).....	49
2.3.1	Especificación de Requisitos del Software.....	50
2.3.1.1	Sistema de escritorio.....	51
2.3.1.2	Aplicación web.....	54
2.4.	Diseño (Segunda etapa fase elaboración).....	63
2.4.1.	Base de datos.....	64
2.4.2.	Sistema de Escritorio.....	67
2.4.2.1	Diagramas de actividades.....	67
2.4.2.2	Diagrama de clases.....	67
2.4.2.3	Diagrama de Secuencia.....	68
2.4.2.4	Diagrama de navegación.....	69
2.4.2.5	Diagrama de Interfaces.....	70
2.4.3.	Aplicación web.....	71
2.4.3.1	Diagrama de actividades.....	71
2.4.3.2	Diagrama de clases.....	71
2.4.3.3	Diagrama de secuencia.....	73
2.4.3.4	Diagrama de navegación.....	75
2.4.3.5	Diagrama de Interfaces.....	75
CAPÍTULO 3.....	77	
CONSTRUCCIÓN Y PRUEBAS DEL SISTEMA.....	77	
3.1	Fase de construcción y pruebas del sistema.....	77
3.2	Normas de Programación.....	77
3.2.1	Selección de las Normas de Programación.....	77
3.2.2	Estándares para paquetes.....	78
3.2.3	Estándares para creación de clases.....	78
3.2.4	Estándares para creación de métodos.....	79
3.2.5	Estándares para creación de variables.....	79
3.2.6	Estándares para instancia de clases.....	79
3.2.7	Estándares para ordenamiento y estructura.....	80
3.2.8	Estándares para comentar código.....	80

3.3	Construcción (Tercera fase de RUP)	80
3.3.1	Construcción aplicación web.	80
3.3.2	Construcción aplicación de escritorio.	85
3.3.3	Diccionario de datos	91
3.3.4	Construcción Web Service.....	95
3.4	Pruebas (Cuarta fase de RUP: Transición)	97
3.4.1	Pruebas de la aplicación Web.	97
3.4.1.1	Cronograma de pruebas para aplicación web	98
3.4.1.2	Condiciones para realizar pruebas	99
3.4.2	Pruebas del Sistema Windows.....	99
3.4.2.1	Condiciones para realizar pruebas de reconocimiento.....	100
3.4.2.2	Cronograma de pruebas para aplicación de escritorio	101
3.4.3	Pruebas de la base de datos.	106
	CONCLUSIONES.....	104
	RECOMENDACIONES.....	108
	LISTA DE REFERENCIAS	109
	ANEXOS.....	115

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Gráfico de las Fases de RUP (Mezquital, 2001).....	4
<i>Figura 2.</i> Distribución de la arquitectura UML.....	7
<i>Figura 3.</i> Elementos básicos de BPM para la elaboración de BPD'S (Freund, 2013)	9
<i>Figura 4.</i> Representación del ciclo de vida de MVC (librosweb.es, 2009).....	12
<i>Figura 5.</i> Ejemplo de Grilla de Kendo para mostrar información.....	18
<i>Figura 6.</i> Gestión de base de datos	19
<i>Figura 7.</i> Ejemplo de Diseño de base de datos (SA, 1998).....	21
<i>Figura 8.</i> Explicación del proceso para captura y procesamiento de imágenes	23
<i>Figura 9.</i> Ejemplo de Binarización de una imagen	25
<i>Figura 10.</i> Tabla de imágenes binarizadas respecto a un umbral.....	26
<i>Figura 11.</i> Tabla de imágenes Binarizadas usando cambios en iluminación y fondo	26
<i>Figura 12.</i> Prueba #1 de conversión HSI.....	27
<i>Figura 13.</i> Prueba #2 de conversión HSI.....	28
<i>Figura 14.</i> Manejo de Cámara Web con librería Emgu CV	30
<i>Figura 15.</i> Reconocimiento de rostro usando Emgu CV	31
<i>Figura 16.</i> Captura de imagen usando Emgu CV	31
<i>Figura 17.</i> Binarización de imagen usando Emgu CV	31
<i>Figura 18.</i> Proceso para entrenamiento de imágenes usado por Emgu CV (Nacional, 1995)	32
<i>Figura 19.</i> Esquema de red backpropagation (Nacional, 1995)	33
<i>Figura 20.</i> Diagrama para Modelo de Negocio registro de empleados	40
<i>Figura 21.</i> Subproceso de modelo de negocio para Registro de entrada.....	41
<i>Figura 22.</i> Diagrama Subproceso de modelo de negocio para registro de salida del empleado	41
<i>Figura 23.</i> Componentes del modelo de negocio CityVenta.....	42
<i>Figura 24.</i> Diagrama de dominio para contratación	43
<i>Figura 25.</i> Diagrama de dominio para asignar jornada laboral	44
<i>Figura 26.</i> Dominio para control de asistencia de jornada laboral.	44
<i>Figura 27.</i> Diagrama de dominio para pago de sueldos por consulta de asistencias.....	45
<i>Figura 28.</i> Diagrama de casos de uso para modelo de negocios.....	46

<i>Figura 29.</i> Diagrama de modelo de objetos para contratación.	47
<i>Figura 30.</i> Diagrama de modelo de objetos para asignación de horarios.	48
<i>Figura 31.</i> Diagrama de modelo de objetos para registrar jornada laboral.	48
<i>Figura 32.</i> Diagrama de modelo de objetos para consulta de asistencia de jornada laboral.....	49
<i>Figura 33.</i> Diagrama de caso de uso Autenticar usuario escritorio.....	53
<i>Figura 34.</i> Diagrama de caso de uso para registrar usuario web.....	61
<i>Figura 35.</i> Modelo conceptual de base de datos.....	65
<i>Figura 36.</i> Modelo físico de la base de datos.....	66
<i>Figura 37.</i> Diagrama de actividad para registrar empleados en sistema de escritorio	67
<i>Figura 38.</i> Diagrama de clases para aplicación de escritorio.....	68
<i>Figura 39.</i> Diagrama de secuencia para registrar empleados en sistema de escritorio....	69
<i>Figura 40.</i> Diagrama de navegación principal para aplicación de escritorio.....	70
<i>Figura 41.</i> Diagrama de Interface para registrar usuario.....	70
<i>Figura 42.</i> Diagrama de actividades para registrar usuario aplicación web.....	71
<i>Figura 43.</i> Primera parte de diagrama de clases para aplicación web.....	72
<i>Figura 44.</i> Segunda parte diagrama de clases para aplicación web.....	73
<i>Figura 45.</i> Diagrama de secuencia para registro de usuario.....	74
<i>Figura 46.</i> Diagrama de navegación para aplicación web.....	75
<i>Figura 47.</i> Diagrama de interface para registro de usuario web.....	76
<i>Figura 48.</i> Ejemplo de campos de la tabla agencia.....	78
<i>Figura 49.</i> Prueba de reconocimiento de cámara web.....	86
<i>Figura 50.</i> Prueba de enfoque y reconocimiento de rostro.....	87
<i>Figura 51.</i> Conversión a escala de grises.....	88
<i>Figura 52.</i> Imágenes para reconocimiento facial en la red neuronal.....	89
<i>Figura 53.</i> Reconocimiento de rostro.....	90

ÍNDICE DE TABLAS

Tabla 1. Módulo de reconocimiento de rostros.....	51
Tabla 2. Módulo de registro de horarios	52
Tabla 3. Descripción caso para usuario de aplicación escritorio uso.....	54
Tabla 4. Módulo de Administración de Usuario	56
Tabla 5. ERS Módulo de parametrización	58
Tabla 6. ERS Módulo de consultas	59
Tabla 7. Detalle de caso de uso para registro de usuario web.....	62
Tabla 8. Diccionario de Datos.....	91
Tabla 9. Cronograma para elaborar pruebas aplicación web	98
Tabla 10. Cronograma de pruebas funcionamiento aplicación de escritorio	101
Tabla 11. Pruebas de base de datos	106

ÍNDICE DE ANEXOS

Anexo 1. Elementos de Bpmn.....	115
Anexo 2. Ejemplos básicos de programación del marco teórico	122
Anexo 3. Descargar e instalar kendo de Telerik	125
Anexo 4. Entrevista para tesis	126
Anexo 5. Encuesta a empleados.....	140
Anexo 6. Cronograma para ejecución del proyecto	142
Anexo 7. Modelo de negocios.....	143
Anexo 8. Casos de uso aplicación de escritorio.....	148
Anexo 9. Casos de uso aplicación web	153
Anexo 10. Diagrama de actividades y secuencias aplicación de escritorio	163
Anexo 11. Diagrama de navegación e interface para aplicación de escritorio	168
Anexo 12. Diagrama de actividades y secuencia aplicación web.....	173
Anexo 13. Diagramas de navegación e interface aplicación web.....	183
Anexo 14. Tabla clases y métodos aplicación web.....	199
Anexo 15. Tabla de clases y métodos para desarrollo de la aplicación de escritorio ...	208
Anexo 16. Prueba de caja negra y caja blanca aplicación web.....	206
Anexo 17. Pruebas para aplicación de escritorio	215
Anexo 18. Pruebas finales de software	224

RESUMEN

El presente documento muestra información sobre el análisis, diseño y construcción de un sistema informático híbrido, que procesa los horarios de ingreso y salida del personal de la empresa “CITYVENTA”, mediante identificación de rostros y administración remota de información; el sistema cumple su objetivo mediante una aplicación de escritorio y una aplicación web.

La aplicación de escritorio está compuesta de: detección del dispositivo de visión artificial, reconocimiento de rostro, registro de entrada – salida y registro de nuevos empleados.

La aplicación web contiene los módulos:

- Módulos de administración de usuario para ingresos, creación de perfiles, asignación de permiso y accesos de los usuarios al sistema, es decir, control sobre la navegación y uso del sistema.
- Módulo de parametrización para ubicaciones, horarios, jornadas laborales y registro de agencias, multas; generando una forma adecuada de administrar los datos del sistema.
- Módulo de administración del sistema encargado del control y asignación de horarios al personal.
- Módulo de consultas para obtener información de atrasos, horarios registrados de entrada y salida.

Todo el proceso fue organizado mediante las iteraciones de RUP, desde el levantamiento de información hasta la construcción del producto pasando por una transición propia de la metodología, aplicando arquitectura MVC (modelo, vista, controlador) y herramientas de desarrollo tales como: MVC4 C#, SLQ Server, LINQ.

El documento contiene una investigación teórica como base del proyecto, un capítulo en el que se muestra el análisis y diseño del sistema; finalmente documentación de construcción y pruebas realizadas.

ABSTRACT

The project for analyze, design and build a hybrid computer system for managing entry and exit times of company personnel " CITYVENTA " by face detection, and remote management of information, contain a windows system and a web application.

The windows system contain: artificial vision detection device, face detection, in - out personnel checked and new employees register.

Web application has these modules:

User management modules contain input, roles creating, permission assigning and user access system. This module has control in navigation and using system.

Programming module for location, hours, working hours and registration of agencies, generate an order for administration data of system.

System management module, performed control and assignment personnel schedules.

Consult module will get arrears information, in - out times registered.

All process be organized by RUP iterations with gathering information to construction final product, using MVC architecture for web application (model, view, and controller)

Document contains theoretical investigation, a chapter of analysis and design of product; finally a chapter has documentation to construction and testing.

INTRODUCCIÓN

El presente trabajo se realizó recalcando los aspectos técnicos, científicos necesarios para explicar el desarrollo y funcionamiento de un sistema que afirme los conocimientos adquiridos en la carrera de Sistemas. Es necesario abordar los fundamentos técnico – científicos para solucionar problemas prácticos que son parte de la vida cotidiana de una sociedad que está experimentando cambios con el uso de sistemas informáticos que faciliten sus actividades, exponiendo beneficios como ahorro de tiempo y recursos.

El sistema esta guiado a mejorar la calidad del tratamiento de información en cuanto al control de asistencia del personal de la empresa CITYVENTA. Para esto se usará un proceso de visión artificial (cámaras web); estos datos serán procesados y enviados a una base de datos remota y validados, esto con la transparencia del caso para el usuario quien automáticamente al acercarse a la cámara generará el registro de ingreso y salida; mientras que el administrador del sistema podrá revisar los datos en tiempo real de cada una de las agencias. Para esto usará dos aplicaciones: escritorio y Web.

Usando la aplicación de escritorio el empleador registrará mediante el sistema en la etapa de contratación del empleado los siguientes datos: rostro, cédula, nombre – apellido, perfil de usuario. Datos que serán guardados en el servidor y que inmediatamente podrán ser usados en cualquier agencia, es decir, el empleado al dirigirse a las agencias será identificado automáticamente.

La aplicación Web tratará la información tomada en la aplicación de escritorio, con esta entregará reportes de atrasos, hora de ingreso y salida, multas por atrasos y suspensiones de ser el caso; entregando esta información para ser utilizada por la empresa.

Finalmente con la recopilación de estos datos se podrá: “CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN”; cumpliendo así con el objetivo planteado, generando un modelo para quienes requieran información relevante acerca de estos temas y puedan obtener una ayuda bibliográfica importante que resuelva inquietudes en el desarrollo de la investigación.

CAPÍTULO 1

MARCO TEÓRICO

En este capítulo se presenta información relevante, conceptos y consideraciones para entregar un panorama lógico y consistente para comprender el contenido y desarrollo del proyecto desde su fundamento teórico, composición tecnológica y la aplicación del mismo en un ambiente real de producción, así lo ratifican las fuentes bibliográficas que señalan que el Marco teórico es “la base en la cual se puede exhibir las herramientas que van a ser utilizadas en el desarrollo de un proyecto, sea este investigativo o de desarrollo” (Monterrey, 2007). Por tal motivo la importancia de entregar una base clara y consistente de definiciones que orientará tanto al autor como a los lectores del documento creando un preámbulo indispensable para la comprensión de toda la estructura del trabajo de fin de carrera; cada vez más se han ido acumulando en texto las experiencias de los autores que de su investigación y compromiso generan rutas de solución a inconvenientes, con estos aportes se ha logrado minimizar tiempos y esfuerzos, además de conseguir excelentes resultados al momento de crear soluciones prácticas a problemas cotidianos. La información será recopilada de varios autores referidos en la bibliografía del documento y que son considerados por los desarrolladores como los más relevantes por su aporte en el enfoque de los argumentos planteados (S.L., 2013).

1.1. Ingeniería de software

La ingeniería de software se encarga de desarrollar e implementar servicios y procedimientos, aplicando los fundamentos computacionales y electrónica (hardware) para la entrega de productos capaces de procesar y organizar información de manera automática incorporando tecnología en varios niveles. El profesional que aplica ingeniería de software posee conocimientos de hardware, software, estructura de datos, lenguajes de programación, manejo de tecnologías de la información (Falgueras, 2013),

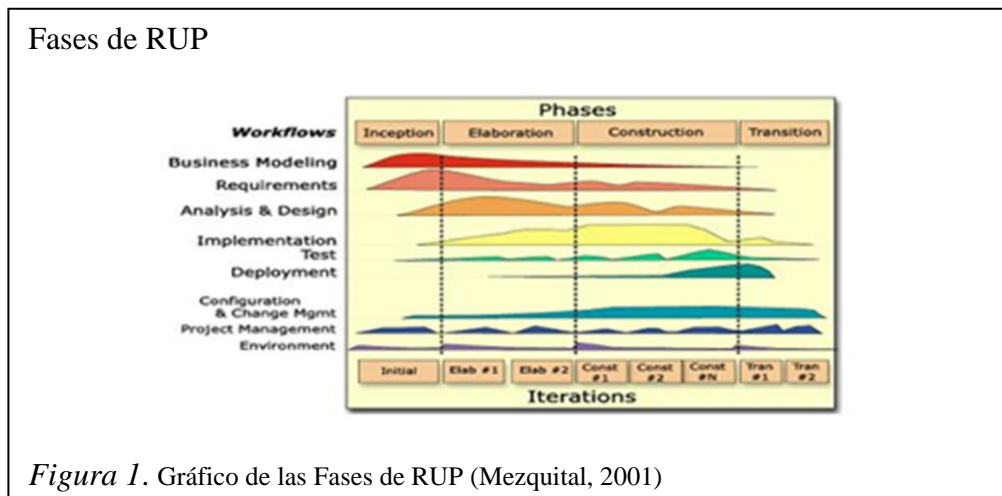
utilizando estos conocimientos puede incursionar en procesos empresariales o industriales para optimizar tiempo y recursos; desarrollando en todos sus niveles planificación, investigación, diseño y construcción de soluciones en plazos determinados y presupuestos previstos. La ingeniería de software “es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos” (Janover, 2008). El proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto que reúna los requisitos que satisfagan la necesidad del cliente, conlleva al análisis previo de la situación, diseño del proyecto, desarrollo del sistema computacional finalmente la etapa de pruebas e implementación del software, cumpliendo así con el ciclo de vida de software que los autores lo refieren como: concepción, elaboración, construcción y transición. (Calero, 2010). Directamente ligado a la ingeniería del software se encuentra la arquitectura, que determina y esquematiza la estructura del proyecto, realizando y relacionando su organización con diagramas (conocido como metodología), que contienen los distintos componentes de la solución, convirtiéndose en la columna vertebral del producto informático (Fernando, 2005). Después de organizar la arquitectura se debe decidir sobre el software en donde desarrollara el trabajo, las herramientas de almacenamiento de datos y lenguajes de programación que se utilizará para el proceso; todo esto dependerá del presupuesto que sea asignado al proyecto.

1.1.1. Metodología

La metodología es un conjunto de procedimientos que determinan una investigación de tipo científico o de construcción de un producto o servicio. Es parte esencial de toda investigación, permite sistematizar procedimientos y técnicas bajo un fundamento teórico que ha sido probado, consigue analizar en su totalidad la realidad planteada y buscar una solución al problema. (Brandon gray internet services, 2012)

1.1.1.1 Proceso Unificado de desarrollo de Software (RUP)

El Proceso Unificado de desarrollo de Software, es una metodología que se encarga de enfocar y organizar un proyecto de software para la obtención de un sistemas informático de calidad (Kruchten, 2004). El proceso de RUP es elaborado en fases, las mismas que realizan varias iteraciones a los procesos, depende del resultado obtenido y la evolución en cada iteración de las partes que lo conforman. RUP se divide en cuatro fases: inicio, elaboración, construcción y transición.



Fase de inicio

En esta fase se define el modelo del Negocio y el Alcance del Proyecto, se identifican los actores y casos de uso; estos últimos son trazados o diseñados. Los objetivos de esta fase son: El lugar en que se realizará el proyecto y sus límites; se ubica los casos de uso básicos y críticos del sistema; se selecciona una arquitectura para el diseño; se estima los costos, recursos y tiempo que se dedicará al proyecto, se verifica los riesgos y posibles problemas (Debrauwer, 2013). Los resultados de esta fase son: Documento de visión, modelo inicial de casos de uso, un glosario, caso del Negocio, lista de riesgos y contingencia frente a estos riesgos, plan del proyecto y el modelo de Negocio.

Fase de elaboración

Se analiza la raíz del problema, estableciendo las bases de la arquitectura, desarrollando el plan proyecto y se enfoca en eliminar los riesgos. Maneja la construcción del prototipo de la arquitectura la misma que irá evolucionando (Fernando, 2005). Los objetivos de esta fase son: definir, validar y dar bases a la arquitectura; completar la visión del resultado de la primera fase; crear un plan para la fase de construcción; argumentar y demostrar que la arquitectura soporta la visión del proyecto (Debrauwer, 2013). Los resultados de esta fase son: ampliación del modelo de casos de uso; descripción de requisitos adicionales; descripción de la arquitectura de software, prototipo ejecutable de la arquitectura; lista de riesgos y contingencia a estos riesgos revisada; manual de usuario preliminar.

Fase de construcción.

Tiene la finalidad de alcanzar la capacidad operativa del producto, de manera incremental siguiendo sucesivas repeticiones del proceso, en esta fase todos los componentes y requisitos deben ser integrados e implementados; incluso pueden llegar a tener un giro total en su concepción. Los objetivos de esta etapa son: minimizar los costos del desarrollo, optimizando recursos; conseguir un producto de calidad; conseguir funcionalidad del producto (Debrauwer, 2013). Los resultados que se obtienen son: modelos completos de casos de Uso, análisis, diseño, despliegue e implementación; una arquitectura integrada; solución de riesgos; manual inicial del usuario; prototipo funcional; caso de negocio actualizado.

Fase de transición.

En esta fase se entrega el producto a los usuarios finales, utilizando actualización de las versiones si fuera necesario, completar la documentación, entrenar al usuario en el uso del producto, ajustes de la configuración e instalación (López E. , 2003). Además se debe probar la versión Beta para ser validada con las expectativas del usuario,

funcionamiento en paralelo si tuviera sistemas asociados o que reciben información, evolución de las bases de datos, traspaso del producto a los impulsores. Los objetivos de esta fase: conseguir que el usuario manipule por sí mismo el producto; un producto que cumpla con los requisitos de funcionalidad y satisfacción del usuario (Debrauwer, 2013). Los resultados de esta fase: un prototipo operacional, documentación legal, el caso de negocio completo, línea completa del producto incluye todos los modelos del sistema, descripción completa de la arquitectura, conseguir una versión actualizada. Si se requiere evaluar los resultados se tomará en cuenta la satisfacción del usuario y la cuantificación de los gastos.

Selección de RUP como metodología para desarrollo del proyecto

RUP es una metodología iterativa o de repetición de procesos, mediante esta característica llega a un producto final con la calidad necesaria.

Ventajas: producto final probado y aprobado por el usuario, documentado y con raíces sólidas, evita riesgos innecesarios, reduce gastos de producción.

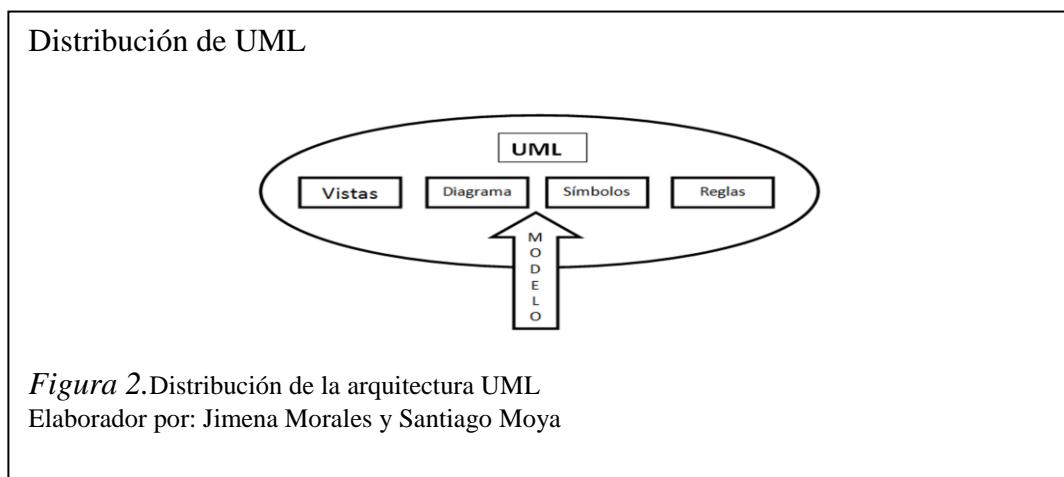
Desventaja: tiempo de ejecución muy elevado.

Autores consideran que RUP es aconsejable para proyectos extensos (3 a 10 personas participantes) sin excluir o imposibilitar su uso en proyectos de menor alcance; así lo confirman artículos publicados en la web (Betaniatech.com, 2014) mencionando que:

Gracias a la distribución de la metodología RUP, la documentación y sus procesos bien definidos los proyectos “pequeños” puedan ser desarrollados con altos estándares de calidad”. Por tal motivo se considera que “RUP es adecuado para desarrollar cualquier proyectos, la clave de la eficacia de RUP es como implementar y aprovechar sus características (Batiuk, 2012).

1.1.1.2 Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado, es un lenguaje de programación que grafica o dibuja un sistema, independientemente de los métodos de análisis y diseño. El modelado consiste en vistas, diagramas, símbolos y un conjunto de reglas que indican cómo usar los elementos, este lenguaje es orientado a objetos. UML no describe los pasos que se debe realizar para desarrollar un sistema, los documenta. (López E. , 2003).



Diagramas de UML

UML se encuentra compuesto por elementos gráficos que combinados forman un diagrama, su finalidad es presentar varias perspectivas de un sistema a las que conocemos como modelos. El lenguaje se encuentra formado de múltiples herramientas para lograr la especificación de modelado: modelado de clases, casos de uso, diagrama de Interacción, secuencia, colaboración, estados, actividades, distribución, componentes, objetos. (Debrauwer, 2013)

Diagramas de casos de uso: ayuda a describir que es lo que el sistema debe hacer, muestran operaciones que se esperan de una aplicación y como se relaciona con su entorno desde el punto de vista del usuario. El conjunto de casos de uso representa la

totalidad de operaciones que va a desarrollar el sistema. Se representa en el diagrama por elipses. (Martin, 1999)

Diagramas de clases: diagramas estáticos que tienen como finalidad describir la estructura del sistema, muestra atributos y las relaciones entre ellos, son utilizados en el proceso de análisis y diseño. Las clases definen al conjunto de objetos. (Martin, 1999)

Diagrama de objetos: se modelan en las instancias de las clases que se encuentran en el diagrama de clases. Muestra un conjunto de objetos y sus relaciones, estos diagramas son útiles para modelar estructuras (López E. , 2003).

Diagrama de actividad: representa un flujo de trabajo paso a paso, es una variación del diagrama de estados. Se compone de un inicio, actividad y transición o cambio de actividad. (Martin, 1999)

Diagrama de secuencia: muestra una interacción ordenada según la secuencia temporal de eventos y el intercambio de mensajes, manejando énfasis en el orden y el tiempo en los que se envían los mensajes de los objetos. (Martin, 1999)

1.1.1.3 Notación para el Modelado de Procesos de Negocios (BPMN)

Siglas Business Process Modeling Notation (Notación para el Modelado de Procesos de Negocios) notación grafica estandarizada sobre los procesos ejecutados en un negocio, diseñada para coordinar la secuencia de los procesos y mensajes que se obtiene durante las diferentes actividades (Freund, 2013). Define la notación y la semántica de un BPD (Diagrama de Procesos de Negocio) mediante un lenguaje común para todas las partes involucradas de forma clara, completa y eficiente. Entre las partes involucradas encontramos: Analistas de negocios son quienes definen los procesos; desarrolladores técnicos responsables de implementar los procesos; finalmente gerentes y administradores del negocio quienes monitorean y gestionan los procesos. BPMN es un lenguaje común para la comunicación entre el diseño de los procesos del negocio y su implementación, en la actualidad existen varios lenguajes para el modelado de negocios pero BPMN es el más usado para la unificación de conceptos básicos. (Hitpass, 2012)

Elementos de BPMN

BPMN se realiza mediante diagramas con un conjunto de elementos ubicados en cuatro categorías:

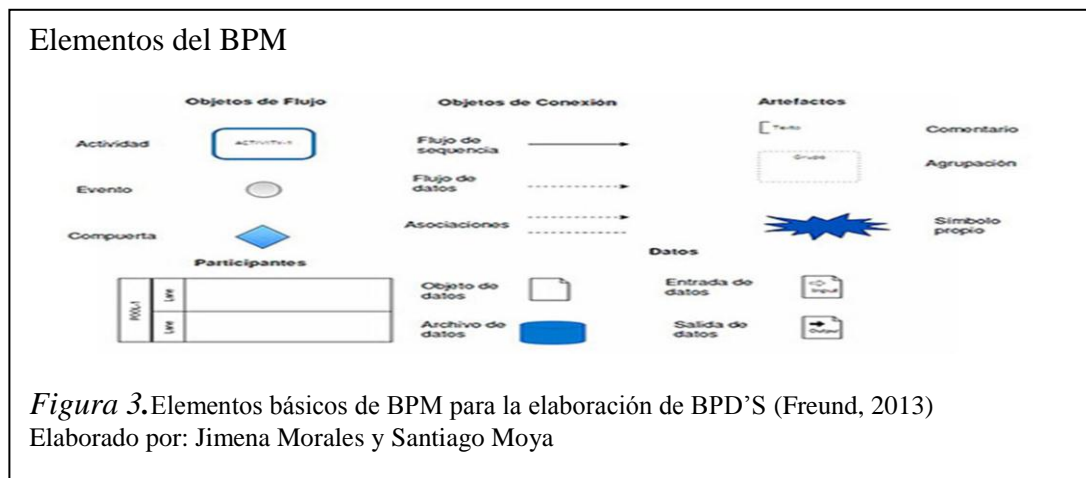
Objetos de flujo: encontramos los eventos, actividades, compuertas (Gateways).

Objetos de conexión: en estos se encuentran el flujo de secuencia, flujo de mensaje, asociación y asociación de datos.

Canales o Swimlanes: en estos se encuentra el pool y lane.

Artefactos: en estos se encuentran los objetos de datos, el Grupo y la anotación.

En el Anexo 1 del documento encontraremos a detalle los elementos de BPMN.



Usos de los BPD'S (Diagrama para procesos de negocios).

Los BPD'S son diagramas diseñados para representar gráficamente la secuencia de todas las actividades que ocurren durante un proceso, basado en la técnica de "organigramas", en donde incluye la información que se considera necesaria para el análisis (konradlorenz, 2010). Los usos principales de los BPD'S son:

Comunicación de procesos de negocios: BPD es un esquema diseñado para ser usado por los analistas de procesos, quienes diseñan, controlan y gestionan los procesos para esto se utilizan un conjunto de elementos gráficos, que se encuentran agrupados en

categorías. Igualmente sus componentes mapean las dimensiones ¿Qué?, ¿Cómo?, ¿Cuándo?, ¿Dónde? y ¿Por Qué? (Guadalajara, 1989).

Automatizar aplicaciones usando un lenguaje independiente: el objetivo de los BPD'S es proveer una notación que sea fácilmente entendida por todos los usuarios y crear un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación tecnológica, esto hace que cualquiera que sea el lenguaje para la ejecución de los procesos de negocio puedan ser visualizados con una notación común que pueda ser comprendida por los analistas (arsys internet, 2003).

Automatizar procesos con uso de plataforma: con el crecimiento de la automatización de procesos se han creado plataformas que específicamente generan esquemas BPD'S, respetando de manera integral las normas de diseño de los procesos. Software tales como Bizagi, o software libre como Work Floor representan plataformas potentes para realizar procesos de automatización de software.

1.1.1.4 Arquitectura de software

La Arquitectura de Software es la organización fundamental de un sistema representada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución (1471-2000, 2009). La Arquitectura de Software es una vista del sistema que incluye los componentes principales, la conducta y las formas en que interactúan además se coordinan para alcanzar la misión del sistema. La vista arquitectónica "es una vista genérica del proyecto pero que aporta el más alto nivel de comprensión y la supresión de la mayor parte de las abstracciones" (Inc, 2000).

Modelo Vista Controlador (MVC)

Siglas de Model View Controller (Modelo Vista Controlador) aparece en 1979 impulsada por Trygve Reenskaug¹, es una arquitectura utilizada para el diseño de software, su característica principal es diferenciar los componentes de un software en

¹ De nacionalidad Noruega nació en 1930, informático y Profesor de la Universidad de Oslo, trabajo en Smalltalk para Xerox

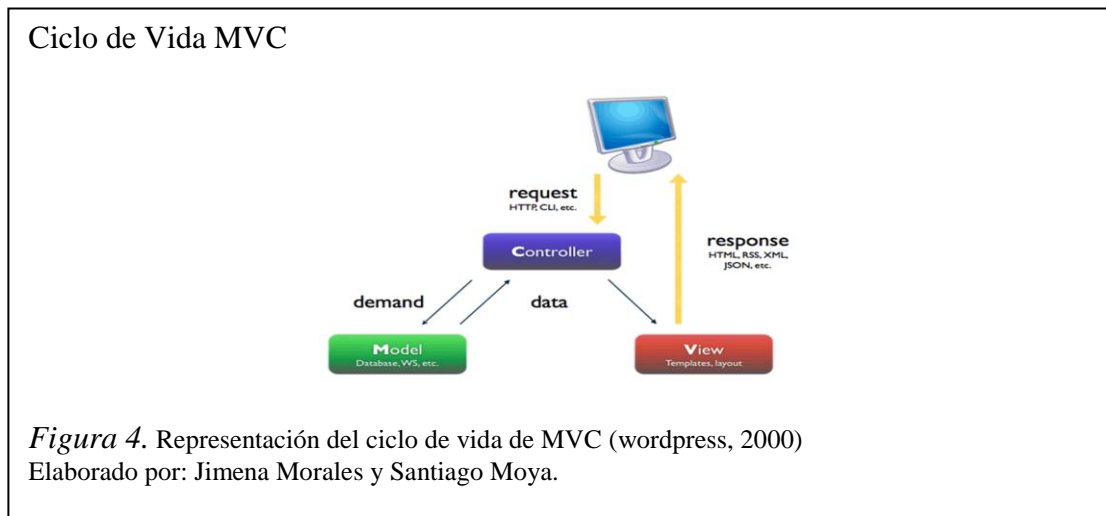
tres sectores (capas) interactuantes, interrelacionadas e intercomunicadas estas capas son: El interfaz de usuario (lo que el usuario tendrá como imagen o presentación visual del software); la lógica de control (programación utilizada por el desarrollador en modo paquetes esencialmente) y la lógica del Negocio o Modelo (o conocido como nicho de mercado para donde fue creado).

Características de MVC

Maneja la lógica del negocio para quien fue diseñado el software bajo un modelo, en el modelo se encuentra la información que estará almacenada en un gestor de datos y las reglas del negocio con las que interactúa el usuario. Muestra al/los usuario la información que requiere mediante una interfaz (vista) o página HTML. Para el desarrollador las vistas son un conjunto de clases que muestran al usuario la información contenida en el modelo. Mediante un controlador toma e interpreta la acción del usuario, actuando sobre el interfaz y la forma del negocio para bajo la interacción realizar cambios de estado en los datos, así como en su visualización, se encarga de dirigir o controlar la aplicación dependiendo de los datos ingresados por el usuario, partiendo de estos datos el controlador se encarga de la apertura o cierre de las vistas (microsoft.com, 2003). Como en todo lo relacionado a los sistemas computacionales y la programación debemos recordar que todo este procedimiento debe ser escalable y adaptable, un programa desarrollado con MVC pueda ser compactado con alguno otro creado por el mismo u otro programador (Aumaille, 2002). De acuerdo a las múltiples páginas revisadas para este trabajo la arquitectura MVC es muy utilizada con punto Net en ASP.Net MVC, por su estructura define las áreas de trabajo en las que el programador realiza el desarrollo (Dollon, 2012).

Ciclo de vida de MVC

- a) **Solicitud del navegador:** el navegador solicita una URL², ingreso a una página cualquiera. (CityVenta.com/home/index.)
- b) **Ruteo del MVC:** la URL indicada es interpretada por un método que depende de la plataforma de programación que se esté usando.
- c) **Métodos del controlador y acción:** siguiendo la secuencia del MVC creamos un controlador que contiene diferentes métodos de acción que han sido previamente programados. Invocar estos métodos resultantes retornan al usuario con informes puede ser texto o imágenes para esto se utiliza vistas.
- d) **Llamada a las vistas:** realizando una llamada a las vistas accederá a una página de la misma programación que se encuentra preparada por el desarrollador que llegará nuevamente al navegador y será presentado al usuario.



En conclusión MVC maneja un funcionamiento particular en la que participan sus componentes de forma ordenada para generar una respuesta adecuada a la petición del cliente (Aumaille, 2002). Como se describe en estos pasos: El cliente envía una petición HTTP que tiene como destino un módulo de una clase; el módulo de la clase requerida

²Uniform resource locator o localizador de recurso uniforme o como conocemos la dirección global de documentos WWW, depende a los protocolos a usar.

recupera los datos enviados en la petición HTTP y delega el proceso de los datos a componentes en este caso particular Linq; en función del proceso realizado por los componentes se puede acceder a la fuente de datos; el método envía el conjunto de procesos de la petición hacia la página Web. La interface de usuario recupera los datos almacenados por el método generando la respuesta HTTP; la respuesta HTTP es visualizada por el cliente.

Ventajas del MVC para la construcción del proyecto.

La ventajas de utilizar MVC es la posibilidad de escalabilidad, es decir, aumentar la capacidad de producción del software ya que se puede trabajar más de un programador o desarrollador en el proyecto siguiendo parámetros similares predefinidos, además que las pruebas o TEST de cada componente programado se la puede realizar sin que afecte en el trabajo del otro; gracias a esto se pueden generar con facilidad entregables según la necesidad del cliente (Berzal, 2003). El MVC es una arquitectura que organiza y distribuye de forma adecuada la construcción o desarrollo de una página web dedicada, identificando capas que hacen que el tiempo de respuesta sea corto, esto en situaciones de medición es un parámetro que hace que podamos identificar y calificar a una página como de alto rendimiento (Berzal, 2003). Muchas veces al usuario es más fácil identificarlo con la página si está guiado por rutas en las que las imágenes para la exploración son las predominantes, pero para quienes manejan programación esto puede causar que el tiempo de respuesta de la página sea elevado, usando MVC esto no es notorio porque identifica la necesidad del usuario canalizando sus solicitudes.

1.1.2. Sistemas informáticos

Programar es una forma de tratar y manejar información puntual requerida, es decir, tiene la particularidad de coordinar, condicionar, involucrar e incluso calcular esta información consiguiendo los resultados deseados (Sala, 2003). La programación es la forma adecuada, segura, ágil de organizar y tratar información. En las primeras incursiones de los sistemas informáticos y los procesadores era muy común encontrar

programas de escritorio pero manteniendo dificultad para el manejo y difusión de la información, por tal motivo se generaron aplicaciones web que facilitaban la portabilidad y administración de los datos. (Dollon, 2012)

1.1.2.1 Programación de escritorio

Conocido como “app” derivado de su significado en inglés application, es un programa informático diseñado para usuarios que realicen actividades o tareas específicas y se aloja en un ordenador. Existen varios ejemplos de este tipo de aplicaciones, empresariales, contables gráficas o reproductoras de medios. Las aplicaciones de escritorio están desarrolladas bajo un sistema operativo y hardware concretos, tienen un mayor acceso e interacción con el hardware, destacándose su trabajo en aplicaciones industriales de monitoreo de procesos, acceso a datos y video Juegos (Mackenzie, 2003). Su período de gestación es reducido y llegan al usuario en menor tiempo, se concentran en el usuario individual. No depende de la conexión a internet, pueden funcionar en una red local o puede trabajar de manera online únicamente usando a la Web como puente de comunicación (Ramirez, 2007).

Visual C# o C Sharp

Parte del paquete de Visual estudio .NET, es un lenguaje de programación orientado a objetos (entidades del mundo real), fue creado para realizar aplicaciones en un entorno virtual y puede desenvolverse en múltiples plataformas de hardware y software. Su sintaxis es similar a Java pero tiene mejoras derivadas de otros lenguajes (Rodriguez, 2007). Como cualquier lenguaje de programación dentro de la estructura podemos declarar: variables, métodos, clases, objetos que puedan ser compilados y entreguen el resultado deseado.

Partes de C#

Se debe identificar claramente la estructura de C#, a continuación se menciona las partes que conforman a la plataforma de desarrollo (Rodriguez, 2007). En el ANEXO 2 del documento se puede visualizar un ejemplo de la estructura del programa.

Variables: es un espacio de memoria que ha sido separado y que contiene información: los tipos de variables son: int, bool, short, float, double, decimal, char.

Clases: son construcciones usadas como patrones para crear objetos, una clase puede tener atributos y métodos que la hacen interactuar consigo misma o con otras clases. En C# encontraremos las herencias, es decir, podemos mantener una clase base que se puede implementar en varios interfaces.

Métodos: Es una parte del código que puede ser invocado por el programa o por otro método, debe ser llamado por su identificador por una secuencia de parámetros que son devueltos después de ser procesados.

Objetos: es el resultado de la instancia de una clase, es capaz de recibir mensajes, procesar datos y enviar mensajes a otros objetos.

1.1.2.2 Programación web.

Con el paso del tiempo y la necesidad del manejo remoto de información empresas informáticas estructuraron plataformas en la Web que soportan gran cantidad de datos además del desarrollo de seguridades certifican al usuario confidencialidad de la información (Dollon, 2012). El uso y transmisión de la información ha evolucionado gracias a Internet (Red mundial de transmisión de información conformada por redes interconectadas entre sí) (Berzal, 2003). Los autores coinciden que a finales de los 70 e inicios de los 80 con el crecimiento de las redes se debió crear los protocolos; enunciando TCP/IP o protocolo de control de transmisión de la información junto al protocolo de internet; este último muy importante para localizar un ordenador dentro de una red por asignación de un número que la identifica. En el inicio de la programación Web se debía manejar una gran cantidad de pseudocódigo (HTML) que claramente a pesar de ser ordenado no generaba un ambiente amigable para los programadores. En la

actualidad existen muchos software desarrolladores que hacen más amigable la creación de páginas que interactúan con el usuario.

Programación Razor

Razor es parte del paquete de Visual Basic punto Net, permite programación con código simple y fluido apegado a la arquitectura MVC, proporciona una sintaxis amigable para escribir código de una página web, el código se ejecuta en el servidor antes de que sea enviado al navegador proporcionando una alternativa para los formularios, haciendo las páginas más ligeras. El carácter que lo diferencia es el @, que se incluye en la interface HTML. (Chadwick, 2011). Razor no es un lenguaje de programación, es un motor de vistas que se rige al MVC en donde separamos la gestión y presentación mediante un modelo el que contiene la lógica de programación. (microsoft, 2012). En el Anexo 2 del documento podemos observar un ejemplo de la estructura de razor.

Las ventajas que presenta razor son: Es compacto, expresivo y fluido reduciendo la cantidad de código para la elaboración de las vistas. Fácil de implementar para el programador por su funcionalidad, además que trabaja en cualquier editor de texto.

Web Service

Desarrollado para intercambio de datos entre software programados en plataformas diferentes por medio de internet su objetivo transmitir información de manera ordenada y segura. Un ejemplo constante son los ejecutables de consulta que se ponen a disposición del cliente para buscar de información concreta. “El web service es una forma estandarizada para integrar aplicaciones mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de internet” (wordpress.com, 2000). Su funcionalidad compartir la lógica del negocio, datos y procesos sin una interface gráfica definida, es decir, no interactúan directamente con el usuario final, quien consume el servicio sin darse cuenta de la acción que está realizando.

1.1.2.3 Java Script y Kendo de Telerik

Java Script en su inicio llamado como ECMAScript por la Asociación Europea de Fabricantes de computadoras, una marca registrada por Sun Microsystem, Inc. Es un lenguaje ligero e descifrado, orientado a objetos aunque también es un lenguaje de procedimientos que genera un resultado alojándose en páginas web, pero que también se han ido incorporando en entornos que no manejan navegador. La sintaxis básica es similar al Java y al C++, pero maneja características independientes (Innova, 2001). Es usado para crear pequeños programas con funciones específicas y que son insertados en programas web u otros programas más complejos, particularmente son creadores de efectos que interactúan con el usuario. El código de Java Script se lo encuentra dentro de las etiquetas <body> e inician con la simbología <script>

Las características más importantes recopiladas en los textos bibliográficos y como ejemplo (Gutierrez, 2009) menciona los siguientes datos: Permite la creación de páginas web dinámicas, un ejemplo de esta función son los efectos de texto, animaciones, etc. Todo a respuesta de movimientos de mouse, teclado y en los programas para aperturas y formas de cargar páginas web. No necesita tener instalado ningún Framework para su compilación. Las variables son declaradas con la palabra reservada var y son interpretadas por todos los compiladores en los que se aloja. Al ser basado en acciones tiene menos restricciones de funcionamiento.

Kendo de Telerik

Kendo es una estructura de software para diseño de páginas Web basado en JQuery HTML5, usado para construir aplicaciones Web modernas, cuenta con gran cantidad de widgets (aplicaciones pequeñas) de interfaz de usuario para la visualización de datos (telerik.com, 2001). Kendo integra su funcionalidad con java Script y Ajax (técnica para cargar datos o partes de HTML sin refrescar la ventana del navegador, permitiendo comunicación con el servidor utilizando el objeto XMLHttpRequest) para ejecutar las pequeñas aplicaciones propuestas y mostrar datos al usuario. Su utilidad y versatilidad

hace que sea compatible con la mayoría de navegadores web. Siendo un paquete de librerías desarrolladas íntegramente en java Script y basadas en jQuery (simplifica la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones, etc.). La variedad de componentes para interfaces de usuario de Kendo UI incluye menús dinámicos, gráficas, paneles, rejillas de datos, árboles, ventanas, sistemas de upload de archivos, etc. En el Anexo 3 del documento se encuentra información sobre la instalación de las librerías de Kendo.

Grilla de Kendo

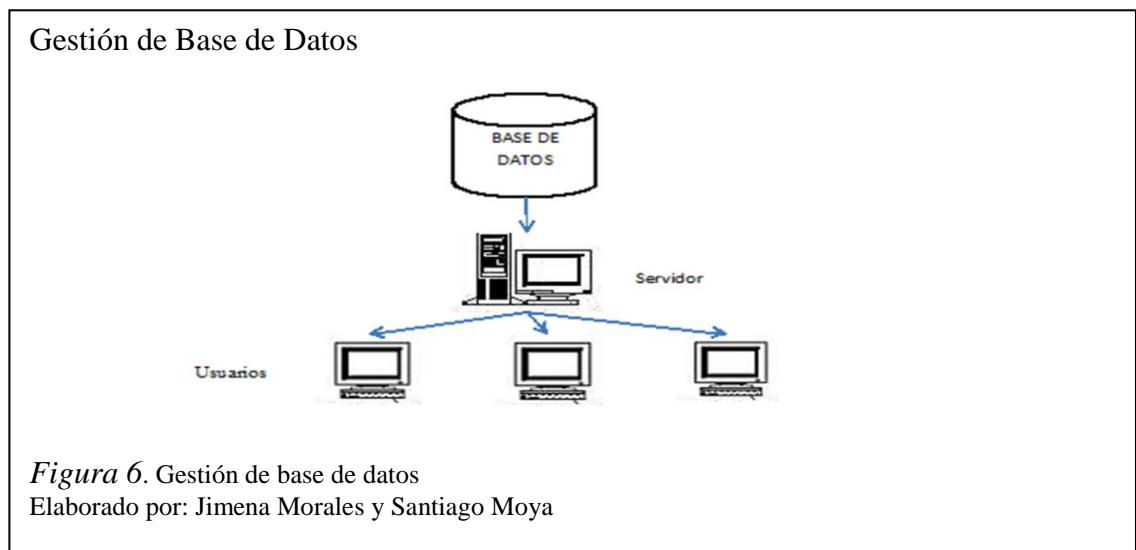
La Grilla es una de las funcionalidades más comunes que se está seleccionando en la actualidad para mostrar datos de forma ordenada además con la posibilidad de filtrarlos, ordenarlos y manejar paginación (división de registros mostrados en pequeñas partes). El usuario puede visualizar la información requerida además de obtener funcionalidades a las que se encuentra acostumbrado convirtiéndose en una herramienta muy importante y útil para el despliegue de información (Paz, 2010). La grilla en MVC la podemos denominar como la esencia de la vista que está siendo controlada por un modelo de datos que han sido guardados en un contenedor. Dentro de la grilla se puede enviar, consultar y actualizar los datos, representado una forma eficaz de manejar la información. (julitogtu.com, 2013).



Las ventajas que representa el uso de Kendo frente al diseño de páginas Web y su relación con Ajax para administración de Java Script (Adams, 2013) son: Ahorro de tiempo por uso de plantillas dedicadas; aplicación dedicada hacia el cliente por su gestión para mostrar información; son controles adaptables fáciles de incorporar en el cuerpo del programa, por ejemplo la grilla mantiene las condiciones de un ambiente Excel para trabajo en filas y columnas pero adjuntando otras propiedades que facilitan la interacción con el usuario.

1.1.2.4 Administración y gestión de base de datos

Una base o banco de datos nos permite guardar grandes cantidades de información de manera organizada, es decir, es una serie de registros creados y relacionados entre sí, los cuales son recolectados y finalmente puedan ser utilizados fácilmente manipulando un programa informático (Yeta, 2012). Los sistemas de gestión de base de datos son software dedicados a entregar una interfaz entre la base de datos, los usuarios y las aplicaciones que manejan la información. (López O. P., 2000).



Una base de datos dirige la información de manera lógica, posee un orden que debe ser respetado para acceder a los datos mediante consultas; cada base de datos contiene una o

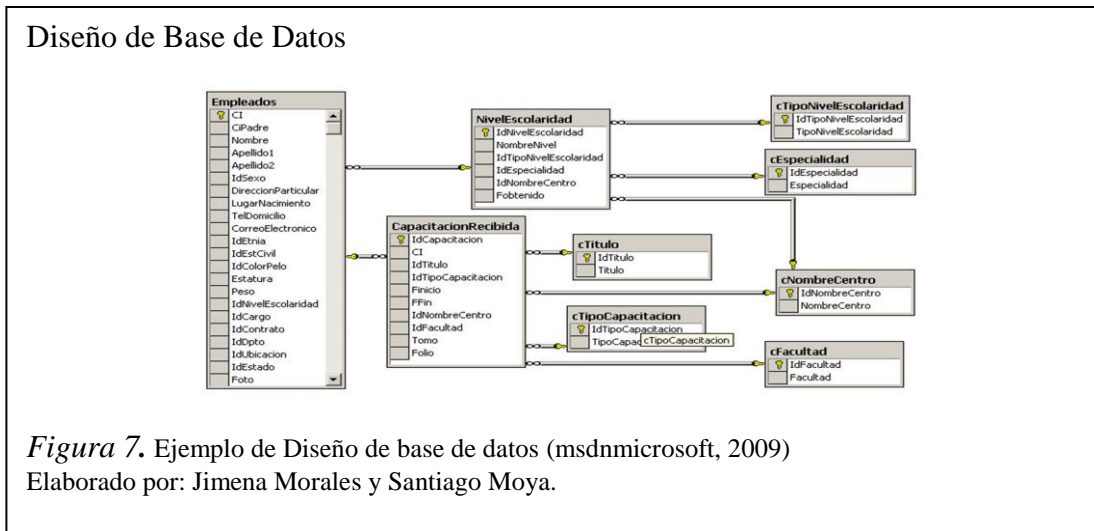
más tablas que anidan campos en donde se almacena información, el tipo de información que contendrán las tablas se define en el diseño del contenedor y se habilita en las propiedades de cada uno de los campos, estas propiedades deben ser rigurosamente respetadas al momento de crear, modificar o acceder a los datos (Abelló, 2006).

Linq

Lenguaje integrado de consultas, nos permite manipular la información en el desarrollo de aplicaciones, orientado a objetos basado en funciones agrupadas que manipulan datos para obtener una respuesta (Freeman, 2010), utiliza funciones propias, que unifican las operaciones más comunes en todos los entornos, con esto se relaciona el lenguaje con los datos; Linq es un componente de Microsoft.Net que añade la capacidad de realizar consultas a través de lenguajes .Net usando la sintaxis similares a SQL, facilita la implementación de varias arquitecturas existentes para acceso a datos. linq utiliza API's (Interfaces de programación para aplicaciones) para conectarse con los diferentes gestores de base de datos (Paul, 2009). Linq de SQL, es el nombre de la interface que permite interactuar con Microsoft SQL server, para entablar una conexión entre el software de desarrollo y el contenedor de datos se debe llamar a la referencia System.Data.Linq.dll. Esta función nos permite modelar base de datos relacionales con clases de .Net que se crean y ejecutan automáticamente, finalmente desde el código en los diferentes métodos se puede actualizar, añadir y borrar datos. (slideshare.net, 2006).

SQL Server 2012.

SQL Server es un gestor de base de datos desarrollado por Microsoft, utilizando modelos relacionales; su función principal es almacenar y consultar datos que son dependientes de otras aplicaciones. Los usuarios del gestor son todas aquellas personas que organizan datos para luego reutilizarlos. (Gabillaud, 2013).



Es necesario normalizar las bases de datos para impedir dependencias de datos, esto hará que la actualización de la base sea fácil y eficiente. Aunque si se requiere alguna consulta es posible se necesite la combinación de varias tablas; a medida que el número de tablas aumenta el tiempo de ejecución será mayor, por este motivo es importante regular y reducir las tablas al mínimo posible para que no se dificulte el proceso de actualización. (microsoft, 2003).

Las características principales de SQL server son seguridades (protección, control de accesos), integración de datos; varios clientes por diferentes medios, recuperación de datos, lenguaje propio (aulaClic, 2013). SQL Server 2012, tiene como particularidad de plataforma Icloud, servicio en la nube, previene perdida de tiempos, reporte, visualización, diseño, depuración y despliegue; proporcionando herramientas gráficas para diseño.

Las ventajas de SQL son las transacciones que es el medio para interactuar con el usuario, siendo un conjunto de funciones que se ejecutan en bloques, si fallara alguna de las operaciones de este conjunto todas se revierten y no afectan al conjunto de datos. Estas transacciones pueden ser activadas desde cualquier aplicación que tenga comunicación con estos datos. (microsoft, 2012)

1.2. Procesamiento de imagen y visión artificial

El ser humano como parte de sus habilidades innatas maneja el reconocimiento e interpretación de su entorno usando la visión como herramienta para conseguir este objetivo; la tecnología con su evolución ha generado sensores que se asemejan al ojo humano y que interpretan un entorno dedicado (Elizondo, 2005). Los sensores proporcionan a un procesador información que va a ser interpretado dependiendo del enfoque para el cual fue creado el sistema informático, el sensor entrega información relevante de una imagen las veces que sean necesarias, para que un intérprete (software) pueda reconocerla utilizando la segmentación. En el caso de reconocimiento de imágenes al segmentarse una porción importante de una imagen es necesario revisar las propiedades geométricas, tamaño y forma que son la base para reconocer si la imagen representa o no el objeto que se desea reconocer. (Bustio, 1994).

1.2.1 Procesamiento digital de Imágenes

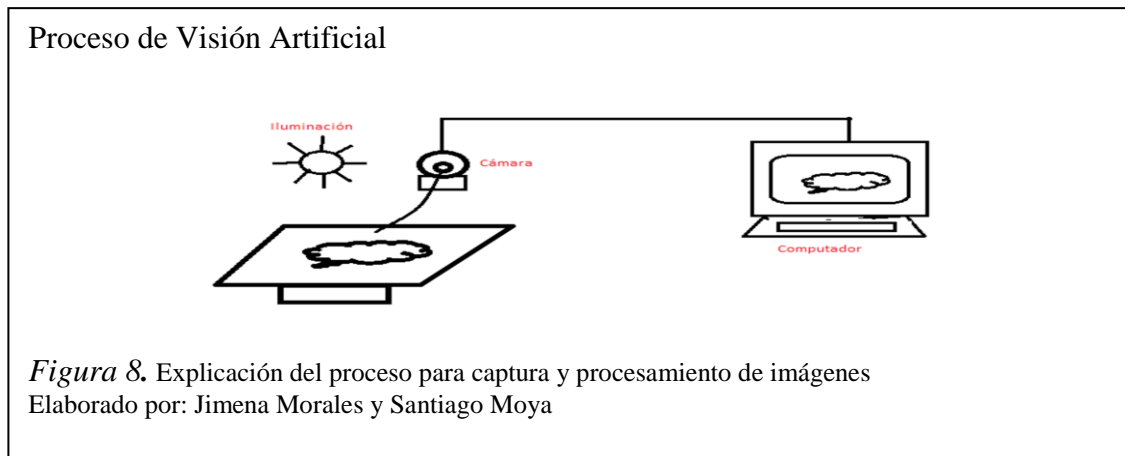
El procesamiento se basa en la forma, tamaño, localización, color, iluminación textura y composición de una imagen, las mismas son una forma de reconocimiento para los humanos quienes identifican objetos basados en las propiedades mencionadas. (Chávez, 2010). En el caso de un sistema computacional de visión artificial para identificación cimienta su trabajo en dos componentes principales: Hardware y Software como se lo muestra en la figura 8 del documento.

Hardware: compuesto de sistemas interactuantes que obtienen la imagen externa por medio de dispositivos como son las cámaras y en el computador que cuenta con una tarjeta de video utilizada como transductor de la imagen y está profundamente ligada a la iluminación del objeto a ser capturado.

Software: o programa para manipulación de la imagen capturada por el hardware; la función del software es digitalizar, procesar y controlar la imagen.

Las fuentes bibliográficas sugieren que el éxito para un sistema de visión Artificial depende de: la óptica, sensores para captura de la imagen y principalmente la

iluminación. La iluminación es la parte principal del reconocimiento y procesamiento de imágenes de esta depende el que se pueda analizar un objeto de manera eficiente, la fuente de luz a la que está expuesto el objeto, la superficie en donde se encuentra ubicado y finalmente la distancia y ángulo de ubicación del receptor de la imagen (cámara) son factores preponderantes en el reconocimiento de rostro. Las fuentes de luz pueden ser incandescentes, luminiscente, descarga de arco, para los sistemas de visión artificial los autores coinciden que la luz infrarroja y ultravioleta son las indicadas por el procesamiento. De la iluminación se generan rayos de luz que afectan de varias formas a él o los objetos de una imagen; la información generada por el sistema de adquisición de visión artificial es representada por valores conocidos como píxeles (Elizondo, 2005).



Para adquirir la imagen es importante poseer los siguientes dispositivos:
una cámara la cual produzca una señal eléctrica que se proporciona al nivel de energía receptada; un digitalizador o transductor que convierte la señal eléctrica proporcionada por la cámara en señal digital; un Hardware digitalizador que acepta la señal analógica y la convierte en digital.

1.2.2 Manejo de una imagen

Una imagen es cualquier representación visual de un objeto, esta imagen generalmente de tres dimensiones se convierte en una imagen plana para finalmente ser digitalizada (Zanuy, 2000). La imagen digital es una matriz de dos dimensiones de números (reales - complejos) representados por bits. La imagen digitalizada se compone de píxeles que son áreas a las cuales asignamos números de identificación los cuales almacenan la información de esta imagen. Dentro de este proceso se debe reconocer la resolución espacial que es el área representada por el pixel y la frecuencia espacial que es la rapidez con la que cambia el valor de la imagen. Importante destacar el manejo del RGB o combinación de los colores básicos para la interpretación de la imagen y la transformación que debe sufrir a escala de grises para ser tratada en menor tiempo. Se conocen como colores primarios al rojo, verde, azul y secundarios la combinación de los primarios representados en el amarillo, cyan y magenta. En el proceso digital de una imagen se utiliza 8 bits para representar el valor de grises siendo el valor mínimo 0 (negro) y el máximo 255 (blanco) (Molina, 2006).

1.2.3 Muestreo y cuantificación.

Para procesar una imagen mediante un ordenador debemos adquirir y digitalizar la imagen, es decir, se la debe converger a una matriz de números. Cuando se digitaliza la imagen se muestrea la misma convirtiéndola en una matriz discreta en donde se cuantifica cada pixel encontrado en la matriz consiguiendo como resultado un número determinado de bits, hay que enfatizar que estos procesos producen pérdida de información (Artigas, 2002). Existen diferentes tipos de formatos para realizar muestreo pero los más conocidos son los datos vectoriales que se refieren a una forma de representar líneas, polígonos, curvas u objetos que sean lineales en las que se puede especificar numéricamente puntos claves y mapas de bits que están formados por conjunto de valores numéricos que especifican los colores de los píxeles.

1.2.4 Binarización de la imagen.

La digitalización o binarización de una imagen es el proceso que permite convertir una señal analógica en una digital. Para realizar el proceso, en primer lugar se crea un mapa de la imagen haciendo una división del espacio a modo de cuadrícula de puntos, llamados píxeles. Cada píxel lleva asociada la información referente al tono y la luminosidad, representada por un código binario. Los datos binarios son almacenados en una secuencia comprimida que el ordenador reproduce como analógica para visualizar la imagen (Ilce, 2008). La Binarización de una imagen consiste en un proceso de reducción de información de la imagen, en la que sólo persisten dos valores: verdadero y falso. En una imagen digital los valores, verdadero y falso, pueden representarse por 0 y 1 o frecuentemente, por los colores negro (valor de gris 0) y blanco (valor de gris 255); la binarización se emplea para separar las regiones u objetos de interés en una imagen del resto. Las imágenes binarias se aplican en operaciones booleanas o lógicas para identificar individualmente objetos de utilidad o crear máscaras sobre regiones. (dimages, 2006). A continuación en la figura 9 un ejemplo de binarización de la imagen en rangos 3,59 y 11 para RGB.



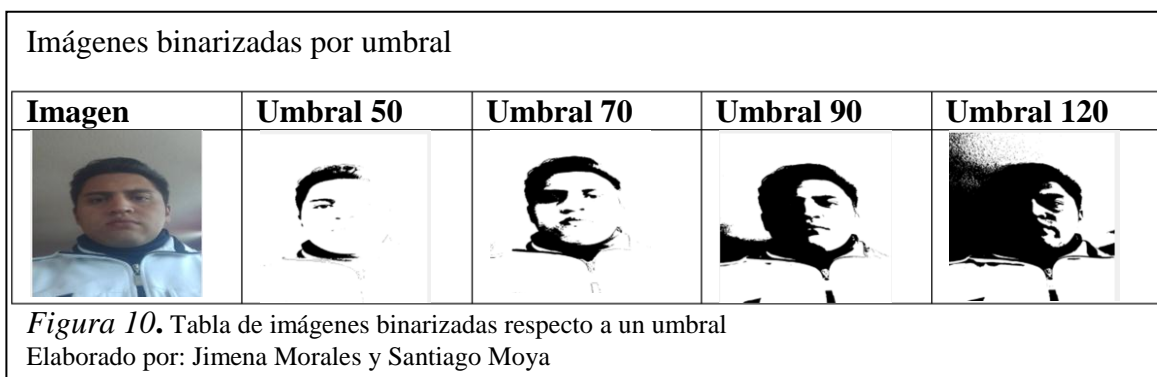
Para emplear la funcionalidad se debe aplicar un valor conocido como umbral que es el valor de color que deseamos diferenciar y representa el límite que determinará si el valor es blanco o negro.

Código para realizar la binarización usando umbral.

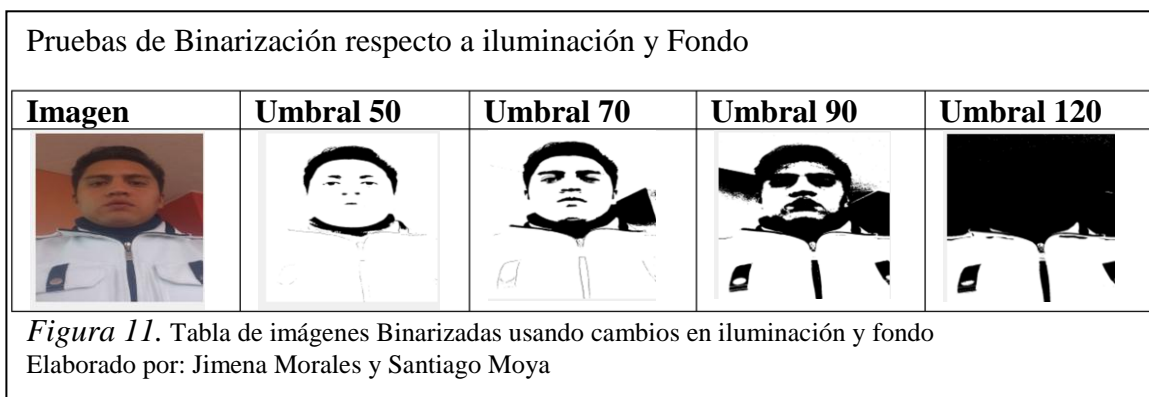
```

// Color del pixel
Color col = source.GetPixel(i, e);
// Escala de grises
byte gray = (byte)(col.R * 0.3f + col.G * 0.59f + col.B * 0.11f);
// Blanco o negro
byte value = 0;
if (gray > umb) { value = 255; }
// Asignar nuevo color
Color newColor = System.Drawing.Color.FromArgb(value, value, value);
imagen_salida.SetPixel(i, e, newColor);

```



Hay que recordar que la binarización de una imagen depende de la iluminación y fondo que posea, a continuación se muestra la misma binarización usando menor iluminación y colores opacos en el fondo. Los resultados son evidentes.



Hay otras técnicas efectivas para realizar binarización que son óptimas para aplicarlas en el reconocimiento facial, una de estas técnicas es la conversión a HSI.

Modelo HSI

El modelo HSI (Hue, Saturation, intensity) o tono, saturación e intensidad es utilizado para procesamiento de imágenes basado en las propiedades del sistema de visión humano, este modelo representa el color de la forma que el ser humano lo percibe. El modelo HSI es usado preferentemente para operaciones de procesamiento básico de imágenes a color, como convolución, ecualización del histograma, etc.; las operaciones que pueden realizarse sobre la componente I (intensidad), que depende a partes iguales, de R, G y B (Sevilla, 2007). En la conversión de RGB a HSI, la componente I depende de los datos de R, G y B: $I = 1/3 * (R+G+B)$

Se aplica la siguiente forma para la programación:

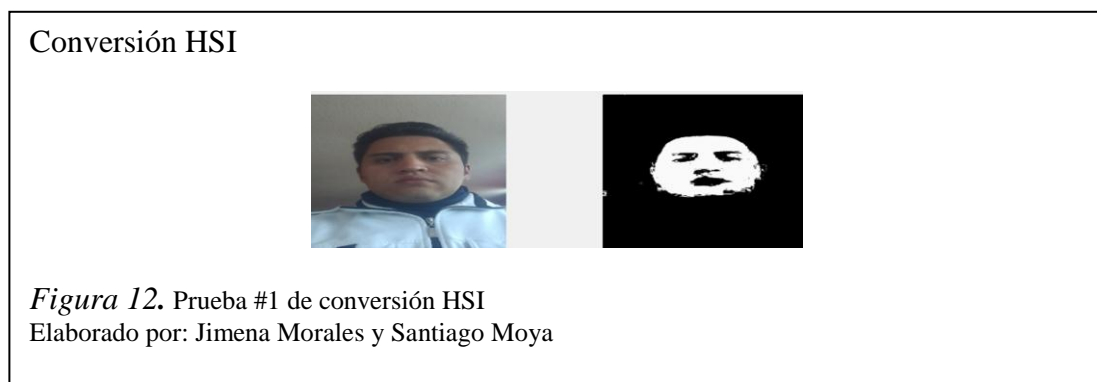
```
//Fórmula para manejo conversión RGB a HSI
```

```
colorI = (1 / 3) * ((double)col.R + (double)col.G + (double)col.B);
```

```
colorS = (1 - ((3 * m)/((double)col.R + (double)col.G + (double)col.B)));
```

```
colorH = (180 * Math.Acos((((R1 - G1) + (R1 - B1)) / 2) / Math.Sqrt(((Math.Pow((R1 - G1), 2)) + ((R1 - B1) * (G1 - B1)))))))/3.1415926535;
```

Se realiza pruebas utilizando un fondo equitativo e iluminación baja y obtenemos:



Se realiza una prueba contrastando el fondo y con mayor iluminación obtenemos:

Conversión HSI

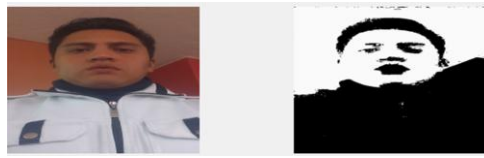


Figura 13. Prueba #2 de conversión HSI
Elaborado por: Jimena Morales y Santiago Moya

Se observa que dependemos del fondo e iluminación para realizar la conversión de una imagen, confirmando lo tratado en los apartados anteriores. En el caso de reconocimiento se debe considerar las propiedades mencionadas para obtener buenos resultados en el reconocimiento.

1.2.5 Reconocimiento de rostros.

El paso del tiempo ha evolucionado las técnicas de reconocimiento facial entre las principales tenemos: algoritmos, redes neuronales, histogramas, etc. De la revisión e indagación bibliográfica, se proponen un sin número de algoritmos dedicados para el reconocimiento facial, los más representativos que generan registros sobre los resultados obtenidos son: algoritmo de Viola-Jones (s.a.c., 2012) que se basa en el cálculo de la diferencia de la suma de los píxeles de dos o más zonas rectangulares adyacentes; el algoritmo de Boosting que se usa como clasificador de los sectores de la imagen, el algoritmo EBGM (Domains, 2008) (Elastic bunch graph matching o Correspondencia entre agrupaciones de grafos elásticos) que construye rasgos locales extraídos usando procesos matemáticos (transformada de wavelet de Gabor) y una figura global de la cara representada como un grafo (Americas-Puebla, 1990).

Las redes neuronales son sistemas ideados como parte de las estructuras neurobiológicas (cerebros) encontradas en la naturaleza y tienen la característica de ser sistemas desordenados capaces de guardar información. (Sonora, 1994). Se puede describir a una red neuronal como modelos computacionales capaces de adaptarse, aprender, generalizar, agrupar y organizar datos; se basa en procesamiento de información ingresada en sus capas que son tratadas por modelos matemáticos, la red está formada

por múltiples capas, donde las neuronas de cada capa están conectadas con todas las neuronas de la siguiente capa, la conexión está ponderada mediante un peso que determina la activación o no de la neurona. (Bolívar, 1999). Las redes neuronales son usadas dentro de un sistema de reconocimiento facial para clasificar las características del individuo, estas características son extraídas aplicando algoritmos que consideran los niveles de grises como la entrada; una limitación del modelo es la sensibilidad a las variaciones de iluminación y orientación (Artigas, 2002). El histograma de una imagen es la función de distribución de probabilidad o de la frecuencia en que se repiten los píxeles de una imagen en función de sus niveles de intensidad, es decir, es una técnica estadística basada en el tratamiento de imágenes, es necesario convertir la imagen en escala de grises para un procesamiento rápido y efectivo, los pasos a seguir son: segmentar la imagen utilizando la técnica de análisis de tonalidad; transformar la imagen a escala de grises; obtener, ecualizar y normalizar el histograma; calcular los índices de ponderación, aplicar método de tratamiento de imágenes para medir la similitud de los histogramas (Ejemplo método de Hamming) y finalmente obtener los resultados de la comparación de las imágenes. (Nacional I. P., 1995). EMGUCV, librería desarrollada para el tratamiento y reconocimiento de imágenes utiliza las técnicas anteriormente descritas para su desarrollo incorporando la base teórica para implementar los algoritmos, librerías y métodos que se utilizan en el desarrollo del trabajo.

Emgu CV

Emgu CV es la adaptación de la librería OpenCV para trabajar en C# de punto net en el manejo de visión artificial. OpenCV es una biblioteca de código abierto desarrollado por Intel; gratuito para uso comercial e investigación bajo la licencia BSD (Berkeley Software Distribution), es decir, muy cercana al dominio público permitiendo el uso del código fuente en software propietario. Bajo las condiciones que la licencia otorga para su uso. (Andalucía, 2010). Su primera versión apareció en enero de 1999, con el propósito de generar un entorno de desarrollo fácil de usar y altamente eficiente por su potencia y variedad de algoritmos para detección de características 2D y 3D, estimación

de movimiento, segmentación y reconocimiento, identificación de objetos, seguimiento de movimiento, reconocimiento facial y de gestos entre los más relevantes (Bradski, 2008). OpenCV originalmente fue escrita en lenguaje C pero por su utilidad y gran acogida a ido adaptándose y mejorando su interface a otros lenguajes, en el 2008 migra a C++, finalmente pasando a java y C#. En C# la librería se la renombra como EMGU CV, escrita enteramente en lenguaje C#. Su beneficio es compilación en Mono (una tarea a la vez), por lo tanto puede ser ejecutada en cualquier plataforma que soporte Mono, incluyendo Linux, Mac OS X, iOS y Android. (Shi, 2013). Gracias al código abierto se la puede adaptar al desarrollo de una solución sin problemas. En el documento se considera las funcionalidades de la librería en captura de imagen, reconocimiento de rostro y red neuronal.

Librerías de Emgu CV

Las librerías están compactadas en .dll que son llamadas desde el software a desarrollarse, estas dll usadas para la funcionalidad de captura de imagen, manejo de cámara Web, reconocimiento de rostro y red neuronal son: EMGU.CV, EMGU.CV.UI, EMGU.UTIL. De las librerías mencionada se puede llamar a métodos y componentes (Shi, 2013). En el trabajo observaremos:

Administración de cámara web

Manejo de Cámara Web

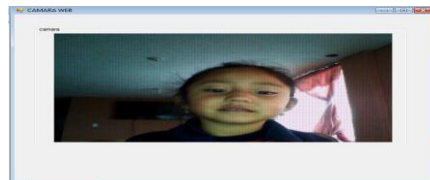


Figura 14. Manejo de Cámara Web con librería Emgu CV
Elaborado por: Jimena Morales y Santiago Moya

Reconocimiento facial

Reconocimiento de rostros

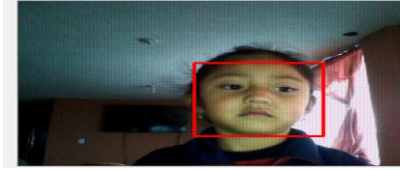


Figura 15. Reconocimiento de rostro usando Emgu CV
Elaborado por: Jimena Morales y Santiago Moya

Captura de Imagen.

Adquisición de imagen

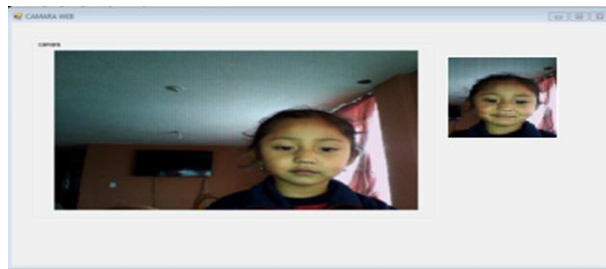


Figura 16. Captura de imagen usando EmguCV
Elaborado por: Jimena Morales y Santiago Moya

Binarización de imagen.

Conversión a escala de grises

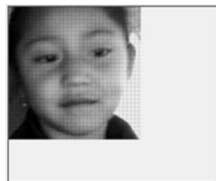
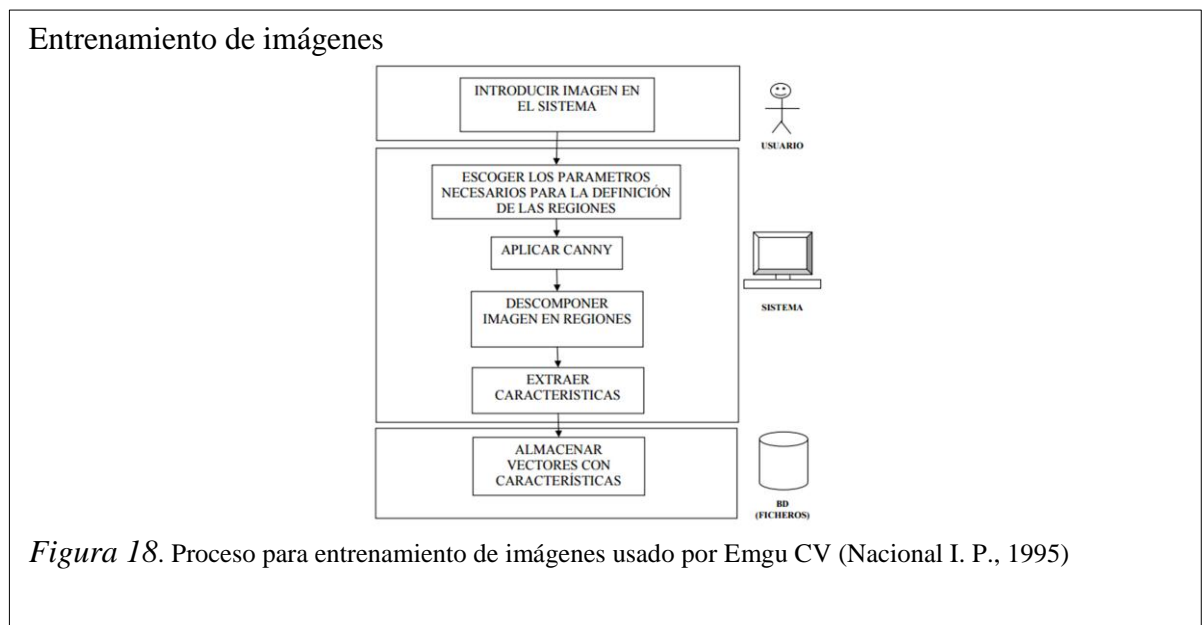


Figura 17. Binarización de imagen usando Emgu CV
Elaborado por: Jimena Morales y Santiago Moya

Redes Backpropagation

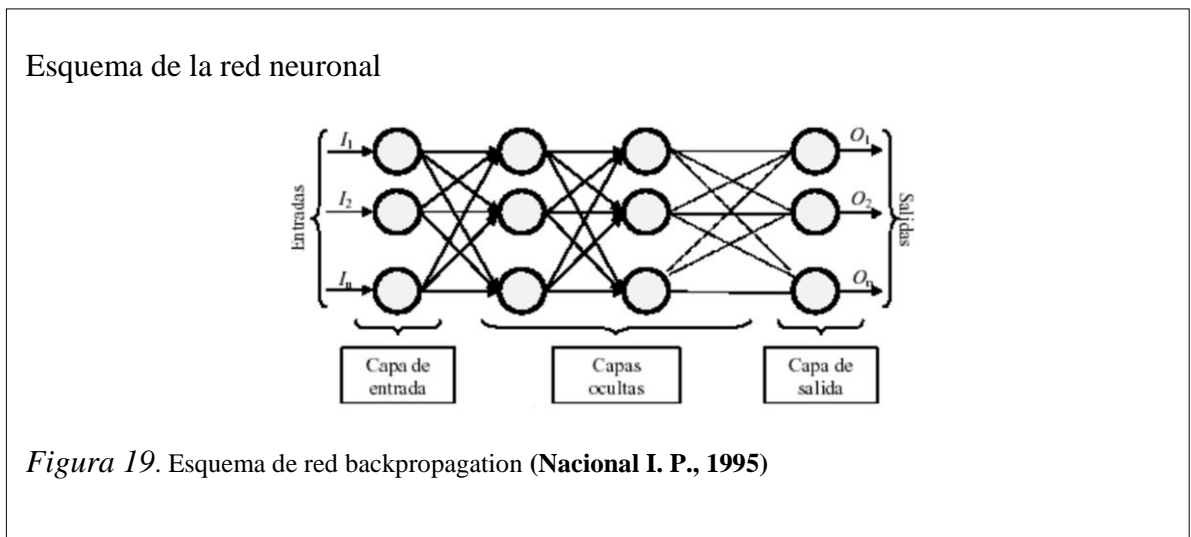
Emgu CV utiliza la red de propagación de error hacia atrás para realizar entrenamiento de imágenes que son previamente procesadas a escala de grises, guardando los vectores con las características de las imágenes, finalmente el sistema clasifica los vectores en varias clases para ser usados en lo posterior en el reconocimiento de objetos (Catalunya, 2000). En la figura 18 se muestra la secuencia que usa Emgu CV para el entrenamiento de imágenes, en el software se invoca al método “MCvTermCriteria” asociado a las funciones del paquete del algoritmo (open source).



Las redes backpropagation aplican un valor de entrada para estimular a la primera capa de neuronas el valor se propaga por todas las capas hasta generar una salida, el resultado del proceso es comparado con la salida deseada y se calcula un valor de error, los errores se propagan hacia atrás para ser corregidos ajustando los valores de entrada conocidos como pesos, las redes backpropagation tienen la capacidad de auto adaptar los pesos de las neuronas, simplificando el proceso, los pasos a cumplir para entrenamiento son:

- Inicializar los pesos de la red con valores pequeños aleatorios.

- Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.
- Calcular la salida actual de la red. Para ello se presentan las entradas a la red y se calcula la salida de cada capa hasta llegar a la capa de salida final.
- Calcular los términos de error para todas las neuronas de la red.
- Actualización de los pesos: para ello utilizamos un algoritmo recursivo, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada, ajustando los pesos.
- El proceso se repite hasta que el término de error sea aceptable.



1.3. Manejo de horarios de una empresa.

Cuando el talento humano se administra adecuadamente y se combina con otros recursos, se cimentan las bases mismas de la producción económica de una sociedad. La estandarización aplicada en cada programa hace que los nuevos usuarios adquieran un rápido dominio de todo el sistema y se sientan en un ambiente de trabajo confortable.

Los pilares fundamentales de una gestión moderna y eficiente de las organizaciones, son la medición y control de las variables relevantes definidas. El principal objetivo de implantar un sistema para controlar la asistencia es garantizar cumplimiento de horarios

de los trabajadores. Sin embargo, la implantación de estos métodos también permite el establecimiento de horarios flexibles. Son varios los sistemas para controlar el tiempo de trabajo. Algunas empresas por medio de un parte de firmas, dando un margen de diez minutos antes de retirar el parte, otras por medio de un reloj con tarjetas personalizadas para firmar; este último no es muy efectivo porque siempre hay algún compañero que ficha por otros, causando una gestión poco efectiva (trabajo, 2005). Actualmente, son muchas las empresas que a través del ordenador, controlan las entradas y salidas de sus empleados de forma segura; utilizando datos únicos por cada empleado. De esta manera, el departamento de talento humano consigue llevar control personalizado y automático de todos sus trabajadores. Los controles de jornadas laborales permiten a la empresa obtener estadísticas y analizar comportamientos colectivos determinados, tendencias, productividad, clima laboral, etc., además de adoptar las medidas oportunas. (mcgraw-hill, 1999).

1.3.1 Reglamento ingreso y salida de personal

El departamento de Talento Humano según los estudios generales realizados por expertos maneja consideraciones universales o reglas para los colaboradores. A continuación se mencionan las más relevantes:

Todos los colaboradores tienen la obligación de concurrir puntualmente a sus labores, de acuerdo al horario establecido y de registrar su asistencia al ingreso y salida en los sistemas de control. El colaborador que no marque o registre su ingreso y salida, es considerado como inasistencia. Vencida la hora oficial de ingreso, el colaborador tendrá una tolerancia definida por cada empresa; se define como inasistencia: no concurrir al centro de trabajo, ingresar después del tiempo establecido en el reglamento de la empresa, no registrar su ingreso al lugar de trabajo (trabajo, 2005). El personal que incurra en retrasos reiterados se hará merecedor a las sanciones que correspondan, según la gravedad de la falta. Toda ausencia al trabajo debe ser justificada por el colaborador dentro del tercer día de producida. Dicho plazo se contará por días laborables. Las

inasistencias ocurridas por motivos de enfermedad se acreditarán con la constancia médica expedida por un certificado médico de acuerdo a ley, el cual se debe presentar en la oficina de Talento Humano, o Área de Administración, según corresponda. El colaborador deberá de permanecer en su puesto dentro del horario de trabajo. El desplazamiento fuera de su puesto de trabajo se hará con conocimiento del jefe inmediato a quien compete bajo responsabilidad, el control de permanencia del personal a su cargo. (Prisma, 2000)

CAPÍTULO 2

ANÁLISIS Y DISEÑO DEL SISTEMA

Para realizar el análisis y diseño del proyecto se ha empleado RUP como metodología; usando el aporte de las iteraciones en todos los niveles del proceso se puede obtener un software de calidad. En este capítulo se mostrarán los datos finales de las dos primeras fases de RUP: Inicio y Elaboración, presentando los diagramas e información necesaria para el desarrollador convirtiéndose en base fundamental de las fases de Construcción y Transición que se mostrarán en el capítulo 3 del documento.

2.1. Situación actual de la empresa

La empresa CITYVENTA, fundada el año de 1997, con su oficina principal ubicada en la ciudad de Quito, sus locales comerciales con el nombre Vanguardia se encuentran en los principales centros comerciales de la capital ecuatoriana con proyección de mercado a nivel nacional; desde su fundación se ha dedicado a la comercialización de prendas de vestir para toda la familia.

Su misión: Vestir a las familias ecuatorianas con prendas de calidad, ofreciendo una atención personalizada a todos sus clientes.

Su visión: Ser un proveedor confiable de prendas de vestir para toda la familia que incorpore nuevas líneas y tendencias que satisfagan a todos sus clientes en el mercado ecuatoriano.

Todos sus procesos administrativos son efectuados de forma manual, por ejemplo el registro de sus empleados en el momento de la contratación se efectúa en libros escritos donde se recopila la información más relevante del nuevo empleado. En cuanto a los registros de asistencia el ente administrativo encargado de los procesos organizacionales internos manejan las siguientes políticas de conducción basados en la plantilla universal de gestión de talento humano:

Todos los colaboradores tienen la obligación de concurrir puntualmente a sus labores, rigiéndose a los horarios de entrada y salida asignados.

Los colaboradores llenan su ingreso de entrada y salida en un cuaderno, en el que consta su nombre, hora de ingreso - salida y su firma de responsabilidad, este control es retirado por personal administrativo quien ingresa la información en un documento de office. El colaborador que no marque o registre su horario es considerado como inasistencia, en el documento no se registran atrasos, únicamente si son encontrados en flagrancia al realizar inspecciones sorpresa o cuando se comunica vía telefónica el Jefe de personal se procede con lo estipulado en las políticas internas. Si algún empleado es sorprendido en varias ocasiones incurriendo en atrasos es sancionado, generalmente con multa económica o llamados de atención según la política vigente de la empresa. Al ser un proceso netamente manual se incurre en varios errores, voluntarios e involuntarios que disminuyen así la confiabilidad de la información y por ende su administración al momento del procesamiento. Los colaboradores están sujetos a este registro, el no llevar un control de asistencias perjudica sus intereses del mismo modo que los intereses de la organización principalmente lo económico. Importante mencionar que a pesar de poseer ordenadores distribuidos en cada uno de los locales no son aprovechados de la manera adecuada.

2.2. Propuesta de la solución implementando un sistema informático

El representante legal de CityVenta Johan Schmollgruber, con el compromiso de mejora continua, además de evitar pérdidas de talento humano y económico generado por la falta de control del personal por diversos factores, ha decidido escuchar la propuesta de automatización del proceso mediante registros automáticos usando reconocimiento facial y administración de la información usando un sitio Web.

Los autores del proyecto han examinado los procesos para generar una solución informática adecuada; al seleccionar la metodología se deben especificar requisitos del software que se visualizan al indagar la raíz del problema; así se definen los procesos que se pueden automatizar, los usuarios o actores que intervienen y finalmente las

actividades a realizar para generar un software que solvete las necesidades de los usuarios.

2.2.1 Levantamiento de información del Negocio (Fase de Inicio de RUP).

La función de la etapa inicial que propone RUP es recopilar información relevante del negocio abordando sus principales actores, que son los expertos para el manejo de procesos actuales en la empresa, en este punto se utilizó una entrevista que se puede observar en el Anexo 4 del documento, las preguntas se enfocan al conocimiento de cómo se realizan las actividades. El representante legal de la organización, permitió el acceso a los libros de registro de jornada laboral, al tomar una muestra del mes de junio de 2012 se observó un registro de 72 horas injustificadas por atrasos de todos sus empleados, es decir, si se genera un cálculo elemental se puede afirmar que un empleado no laboro durante 9 días, esto bajo un control poco eficaz con bajo porcentaje de credibilidad, el resumen se lo puede observar en el mismo Anexo 4. Seguido a esta actividad se efectuó una visita de campo para validar cada uno de los procesos que se realizan para el control de la jornada laboral en los locales: Vanguardia de C.C. El bosque y Vanguardia de C.C.I. como en las oficinas de CityVenta, recopilando información importante para la fase de elaboración del software. Para finalizar con el proceso se generó una encuesta a las empleadas de la empresa con el objetivo de conocer su realidad, el enfoque sobre la forma de llevar el control de asistencia y la aceptación al cambio del modelo actual usando tecnología. La encuesta se la puede observar en el Anexo 5 del documento. Resumiendo los resultados con una muestra de 25 personas se obtiene: un 92% coincide que las faltas perjudican su economía personal, un 64% estaría dispuesto a cambiar la forma de llevar el control de la jornada laboral usando tecnología y un 92% piensa que los programas informáticos son útiles y simplifican tareas.

Bajo estos lineamientos se desarrolló los primeros artefactos (documentos), que RUP propone para levantamiento y organización de información. En el Anexo 6 se puede observar el cronograma de entrega de los artefactos y del producto terminado. A continuación se destaca los aspectos más relevantes del artefacto principal de esta fase “Documento Visión” y los artefactos entregados en la fase de inicio.

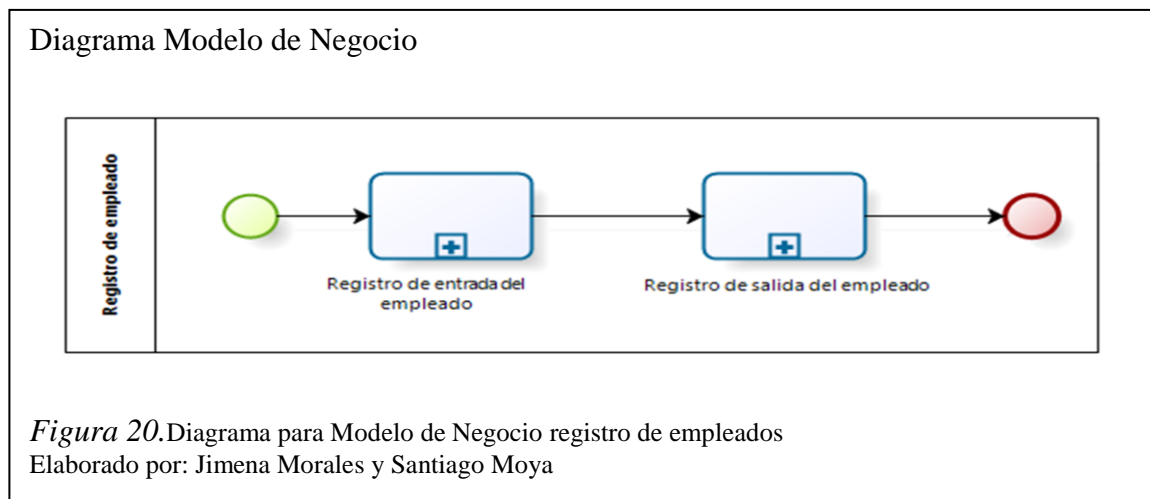
2.2.1.1 Documento Visión

El documento Visión recopila la información relevante sobre la gestión actual y procesos para la administración de asistencia en los diferentes lugares en donde se encuentra laborando personal de la empresa CityVenta, los datos obtenidos son pilares fundamentales en la construcción del software. El sistema permitirá a los encargados de la empresa controlar las actividades de su personal y evitar gastos innecesarios de recursos. La oportunidad del negocio se plantea con un sistema que permitirá administrar la jornada laboral de sus empleados y controlar bajo este medio la asistencia del personal, utilizando tecnología que facilita la gestión y acceso a datos con un interface gráfico sencillo, los datos a los cuales accedan se tomarán en tiempo real, es decir, permanentemente se actualiza la información generando control de asistencia en los locales comerciales, también permite a los empleados acceder a consultas sencillas en las que conocerán su asignación de jornada laboral y agencia. El objetivo del documento es abarcar la información obtenida del negocio, es decir, definir el problema, reconocer los actores, describir el producto y plantear la solución además proponer los primeros riesgos y un glosario de términos que aclaren la perspectiva de la información; se concluye el documento Visión en la tercera iteración propuesta en la versión 3.0. A partir de estos artefactos se conseguirá el documento especificación de requisitos de software propuesto por la norma IEEE.830.

Se debe señalar que ningún proceso de CityVenta se encuentra asentado bajo diagramas de comprensión que muestren el modelo del negocio, por este motivo para el proyecto se ha recopilado información necesaria y relevante, específicamente del proceso de control de asistencia de la jornada laboral además de la contratación del personal que es la base para la automatización, la información se la muestran en notación BPMN. A continuación los diagramas del proceso.

2.2.1.2 Modelo de negocios / Diagrama de procesos

Los diagramas de modelo de negocio ayudan a entender el funcionamiento de los procesos de una empresa. Tienen la misión de esquematizar o dibujar las actividades registradas en el proceso del negocio. Su nomenclatura se derivada de su traducción a Ingles BPD (Bussines Process Diagram). En la figura 20, 21 y 22 se diagrama el modelo de negocio para registro de empleados en los locales comerciales usando BPD'S con notación estándar, en el Anexo 7 del documento se puede observar el levantamiento total del modelo de negocios.



El modelo de negocio para registro de empleados en los locales tiene dos subprocesos que definen la entrada y salida del personal:

- **Subproceso de registro de entrada del empleado**

El subproceso se refiere a la forma que los empleados registran su ingreso en los locales comerciales.

Subproceso modelo de negocio

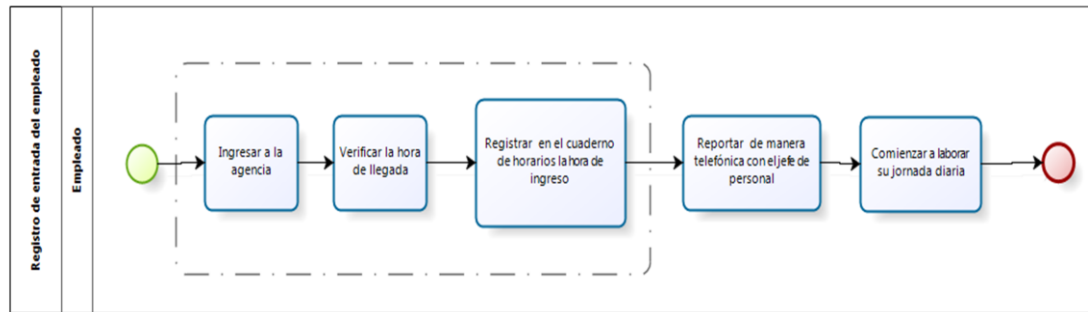


Figura 21. Subproceso de modelo de negocio para Registro de entrada
Elaborado por: Jimena Morales y Santiago Moya

- **Subproceso de registro de salida del empleado**

El subproceso se refiere a la forma que los empleados registran su salida en los locales comerciales.

Subproceso modelo de negocio

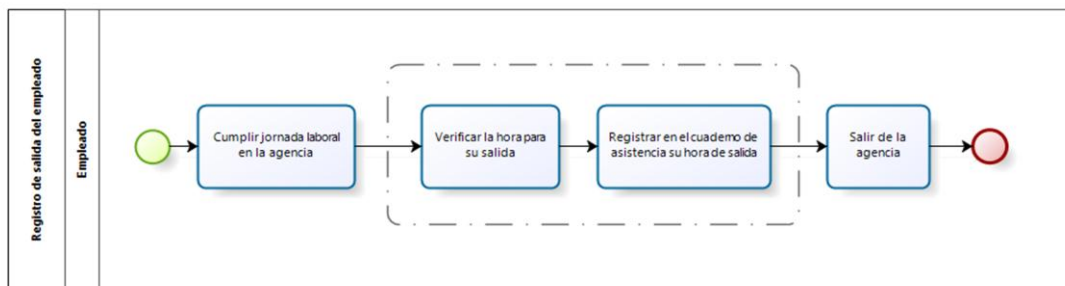
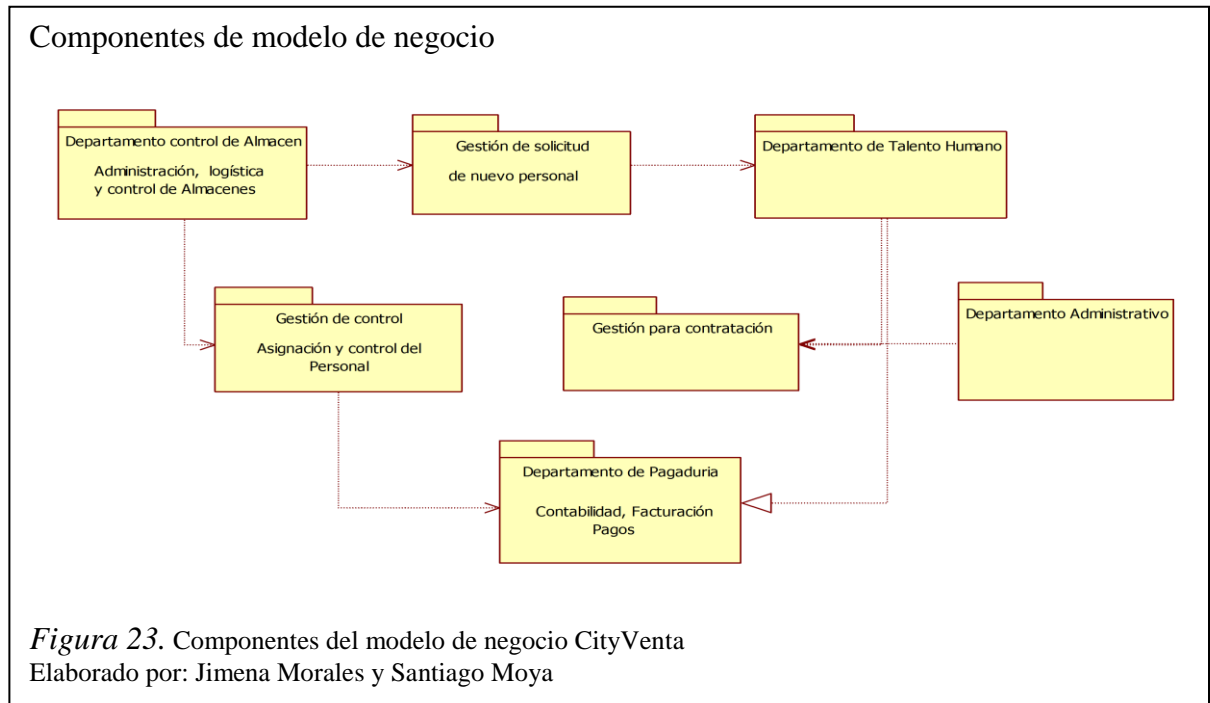


Figura 22. Diagrama Subproceso de modelo de negocio para registro de salida del empleado
Elaborado por: Jimena Morales y Santiago Moya

2.2.1.3 Diagrama para modelo de negocios CityVenta.

Después del levantamiento de información que ha representado en los diagramas del modelo de negocio, siguiendo las condiciones solicitadas por la metodología RUP se propone el diagrama de dominio, diagrama de objetos y diagrama de caso de uso para el modelo de Negocios, de esta manera se consigue el objetivo de la fase de inicio de RUP y se deja sentado el precedente para el inicio de la fase de elaboración; a continuación se propone cada una de los diagramas aplicados a la parte del negocio que se desea sistematizar.

- **Diagrama de componentes.-** Es un esquema que muestra las interacciones y relaciones de los componentes de un modelo que tienen dependencia. En la figura 23 se muestra la relación entre los entes que componen el proceso de negocio de CityVenta.



- **Diagrama de dominio.-** Captura y expresa el proceso de un área que se está analizando como paso previo al diseño de software, es usado para comprender el sector de negocio al cual el sistema va a servir. Además contribuirá a identificar personas, eventos, transacciones y objetos que están involucrados en el sistema.

Diagrama de dominio para contratación.

En la figura 24 se muestra la relación entre los usuarios que son dependientes de cada uno de los departamentos y la consolidación del proceso para contratación siguiendo el esquema actual de la empresa.

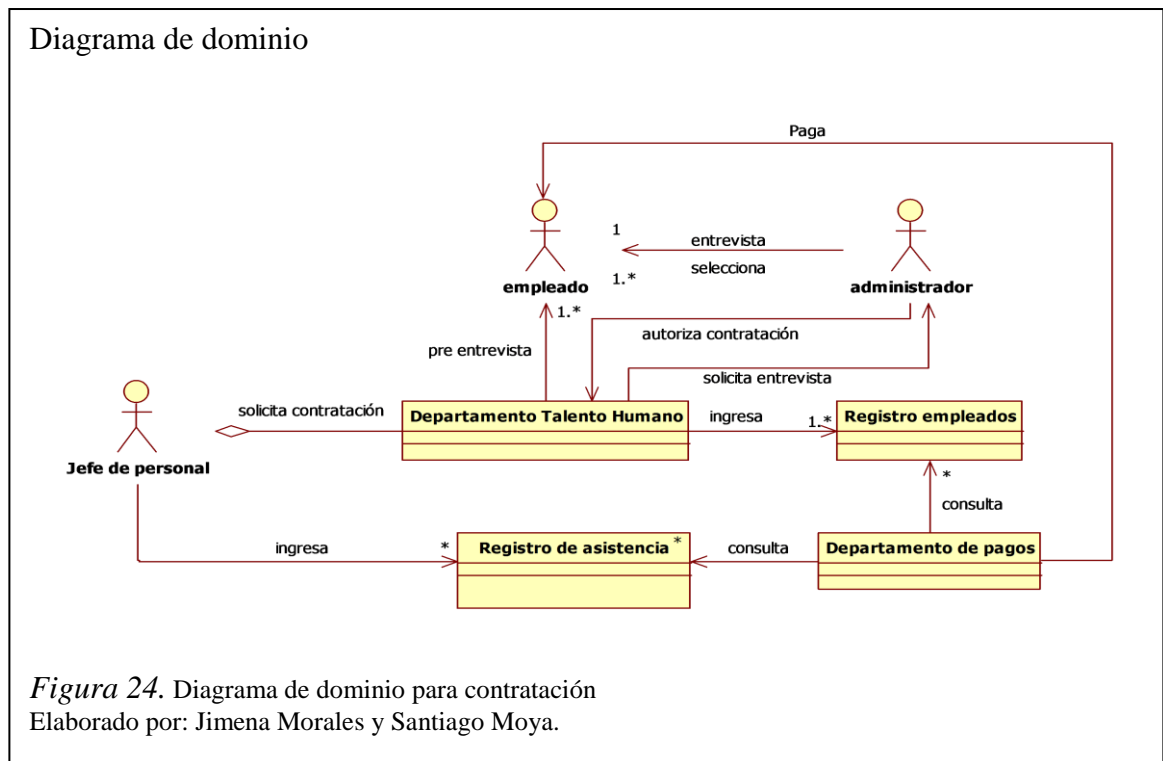


Diagrama de dominio para asignar jornada laboral.

En la figura 25 se muestra la relación entre actores, de paramentos y el proceso secuencial usado para asignar jornada laboral a un empleado.

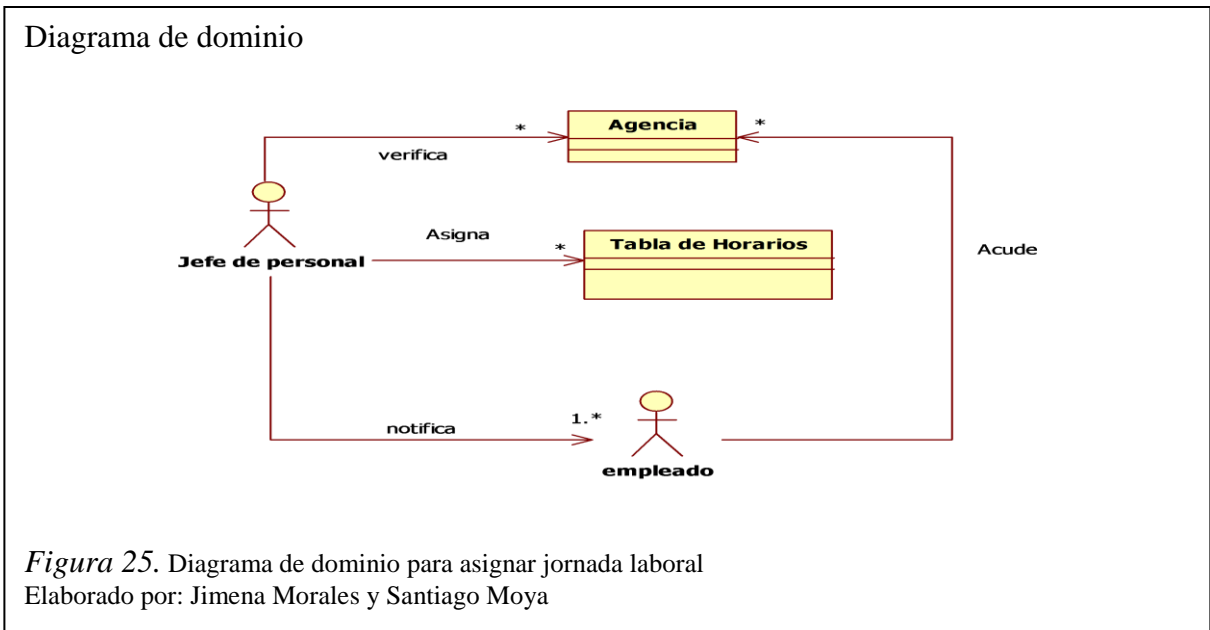


Diagrama de dominio control de asistencia de jornada laboral.

En la figura 26 se muestra la relación de los procesos para control de asistencia de la jornada laboral de los empleados de CityVenta según el proceso actual.

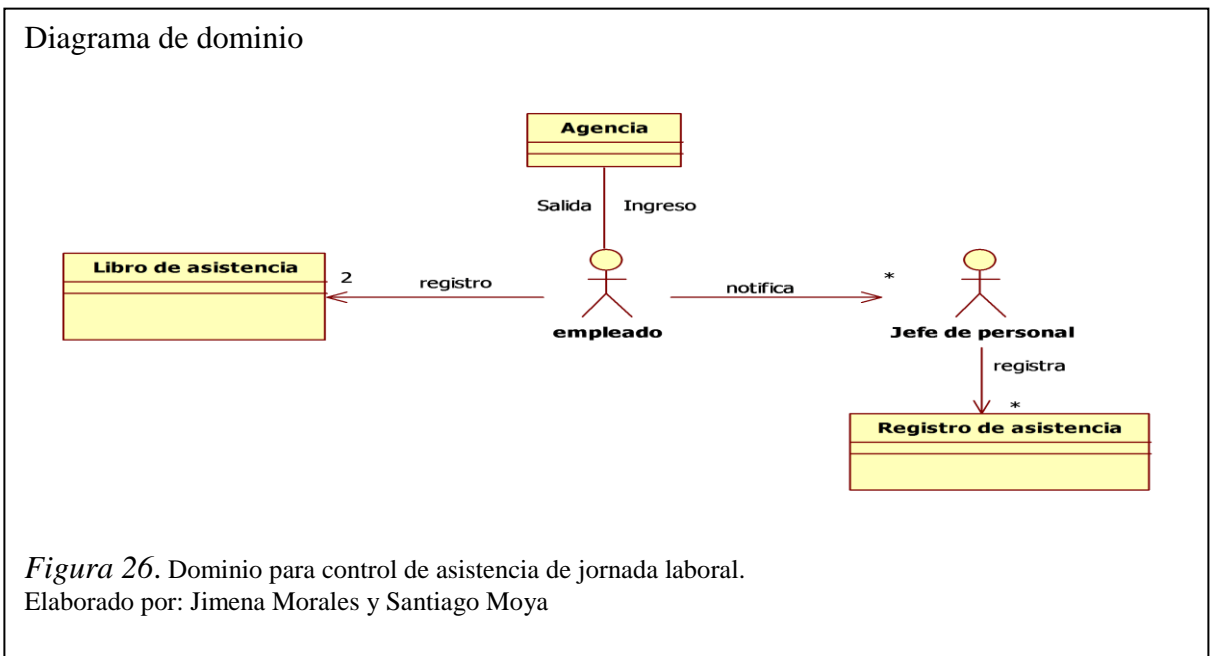
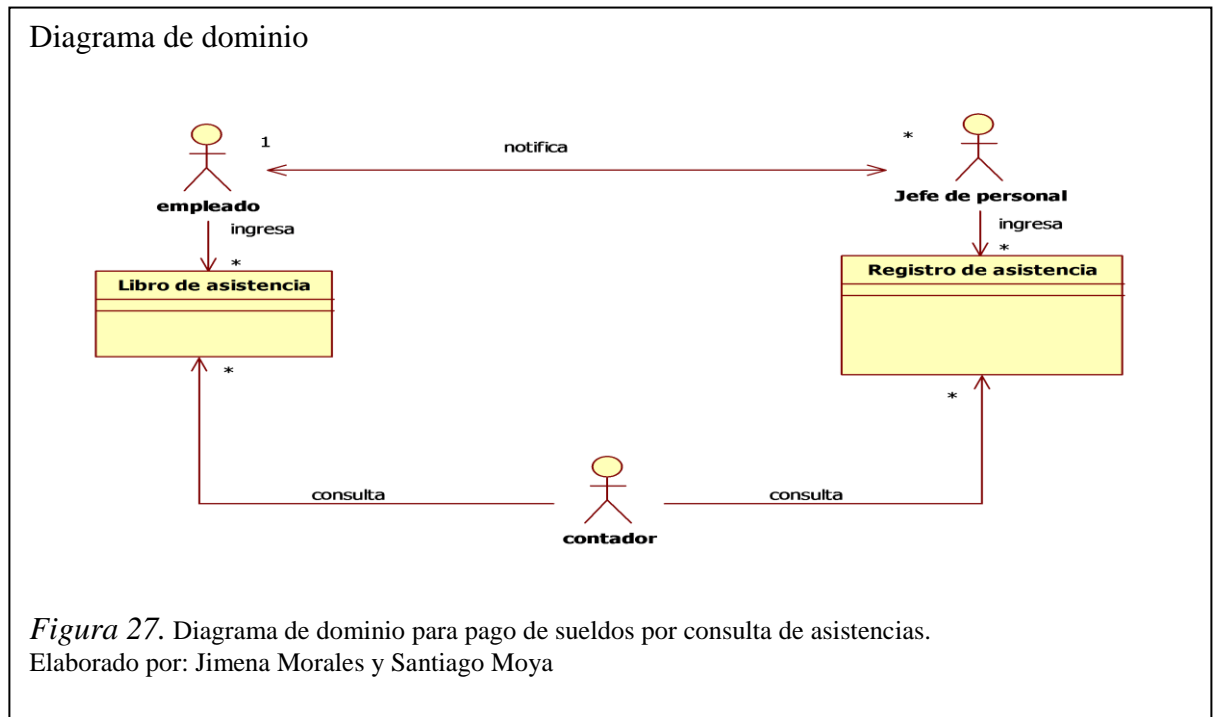


Diagrama de dominio pago de sueldos por consulta de asistencias.

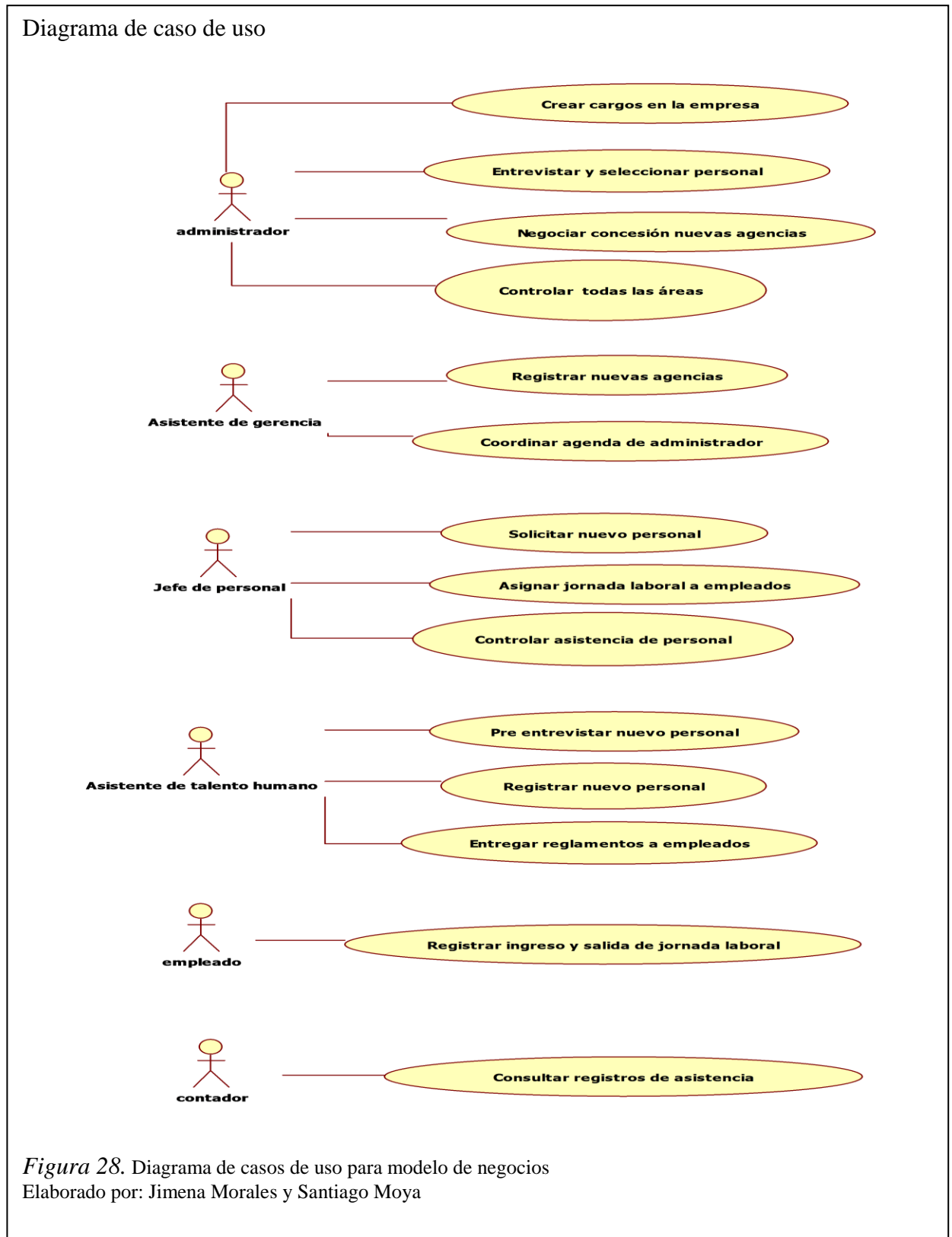
En la figura 27 se muestra los pasos a seguir y la dependencia entre departamentos y actores para realizar un pago de sueldos según la asistencia a su jornada laboral.



Después de haber visualizado el proceso que se realiza para contratación, asignación de jornada laboral, control de asistencia y pago de sueldos en los diagramas de dominio se pueden identificar claramente los actores que interactúan con el proceso y que serán sin duda quienes deban manipular el software y para quienes la aplicación será un puente de comunicación.

Diagrama de casos de uso.- Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas (Santos, 2014).

Diagrama de casos de uso para modelo de negocios

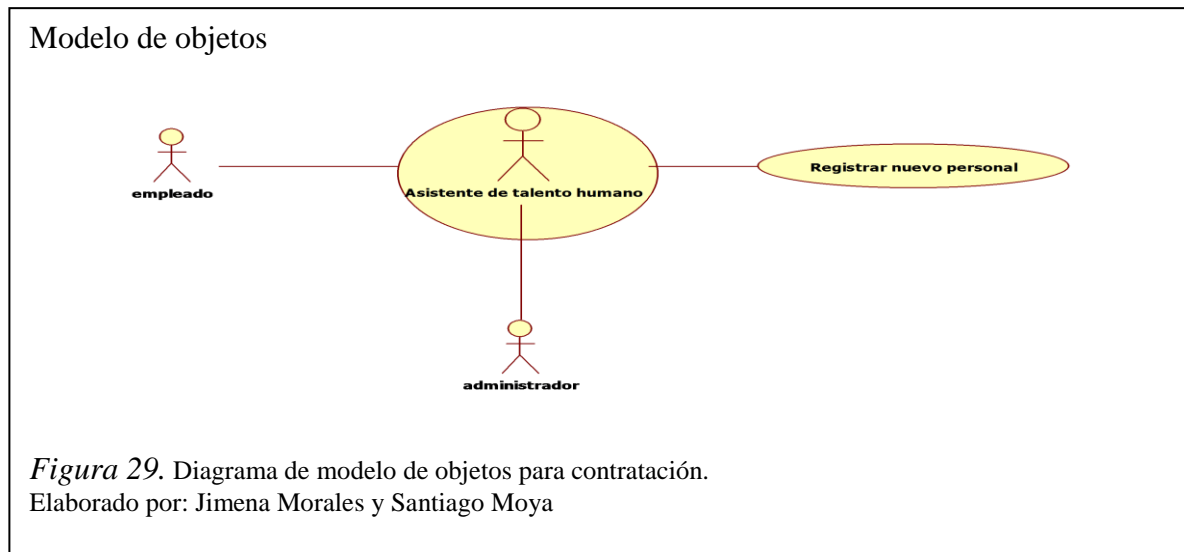


- **Diagrama de modelo.-** Es un diagrama que muestra una vista completa o parcial de los objetos de un sistema en un instante de ejecución específico de las actividades, describe como se relaciona un grupo particular de objetos entre sí y muestra su resultado además de reconocer sobre quien recae el eje central de la actividad (Falgeras, 2003).

2.2.1.4 Diagrama de modelo de objetos

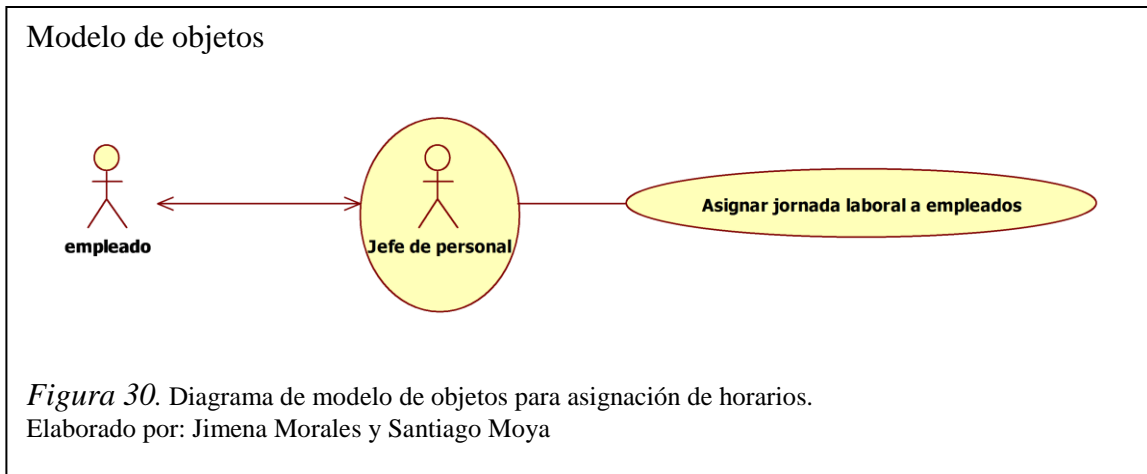
Modelo de objetos para contratación.

En la figura 29 se muestra la relación particular entre los actores del proceso contratación de personal, se observa que la Asistente de Talento Humano es quien lleva a cabo y dirige la continuidad de la contratación relaciona a los actores empleado y administrador para concluir con la contratación.



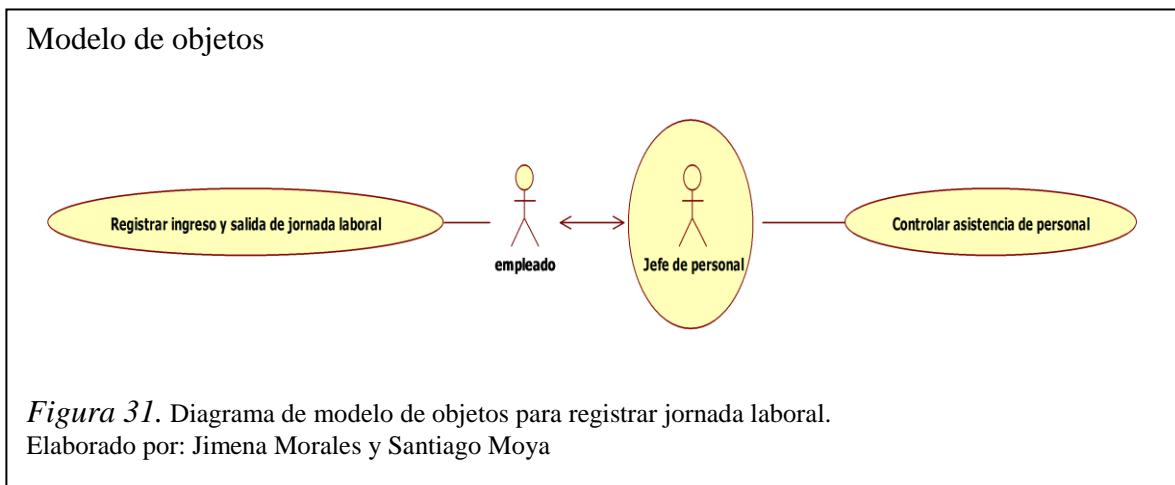
Modelo de objetos para asignación de horarios.

En la figura 30 se muestra la relación entre Jefe de Personal y Empleado quien comparten información en el proceso para la asignación de jornada laboral.



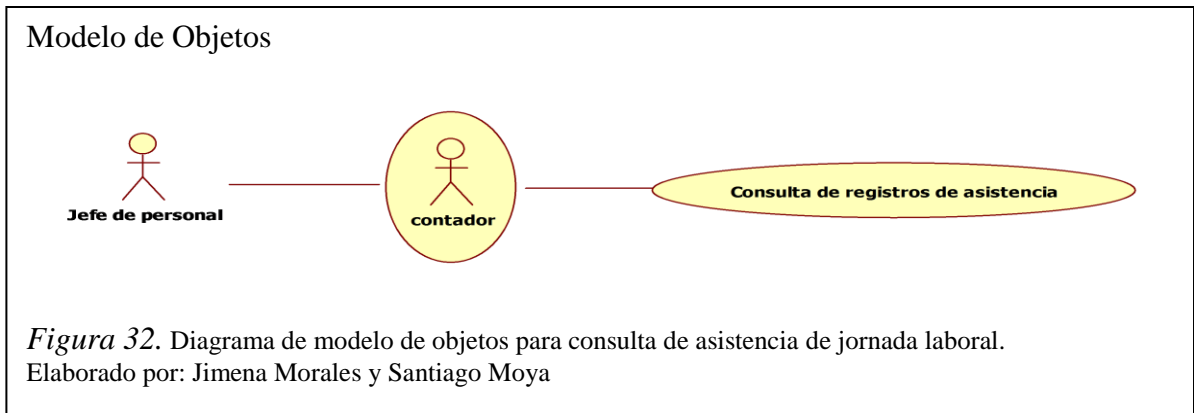
Modelo de objetos para registrar jornada laboral.

En la figura 31 se muestra una nueva relación entre Empleado y Jefe de Personal esta vez para el control de asistencia del empleado que registra su ingreso y salida de labores.



Modelo de objetos para consulta de asistencia de jornada laboral.

En la figura 32 se muestra la relación de dependencia que existe cuando el Jefe de Personal entrega información al Contador según la asistencia receptada.



Al concluir con la fase de inicio se han entregado los artefactos propuestos por RUP: documento visión, glosario de términos, la primera evaluación de posibles riesgos y los diagramas. Estos documentos se los puede revisar en el formato digital que se encuentra anexo al escrito.

2.3. Análisis (Primera etapa fase de elaboración)

Después de tratar y elaborar la primera fase propuesta por la metodología RUP y con los refinamientos necesarios se puede continuar con la fase de elaboración, en la que se debe seleccionar la arquitectura con la que se elaborará el sistema; la arquitectura elegida para la construcción del software es MVC (Modelo, Vista y Controlador), la arquitectura fundamenta su desarrollo en la orientación a objetos, es decir, delega responsabilidades, sectoriza la información en pequeños bloques independientes y finalmente enlaza estos bloques para conformar el producto final.

Pasando por las etapas de refinamiento los resultados que se han obtenido aplicando los procesos y documentación tenemos: Especificación de requisitos de software, diagramas de casos de uso del software y la descripción de los casos de uso, la información se encuentra evidenciada en el artefacto "Casos de uso del Negocio" concluyendo en la versión 4.0 a continuación se presenta la propuesta final de estos diagramas.

2.3.1 Especificación de requisitos del software

La especificación de requisitos es una descripción completa del funcionamiento y cumplimiento del software que se quiere desarrollar, para el documento será abreviado usando las iniciales ERS. La información surge de la fase de inicio en donde se ha determinado los actores y las actividades que realizan para el control de la jornada laboral de la empresa y que se encuentran reflejados en las figuras de la sección anterior (20 al 32) del documento. Al realizar el análisis de la secuencia además de las actividades se ha determinado el orden y los objetos que se deberán implementar en el software para automatizar el proceso; estos procesos se han dividido en módulos independientes pero a la vez interactuantes. Para el análisis del sistema de control de horarios entrada y salida se ha definido módulos para la aplicación web y la aplicación de escritorio usando tablas en las que se reflejan la especificación de requisitos, las tablas constan de siete campos que manejan los siguientes datos:

- Código: identificador del requisito, iniciado por las siglas de especificación de requisitos de software ERS, las iniciales del módulo a tratarse y finalmente un código numérico de cuatro números para identificación única.
- Descripción: se ingresará información de lo que debe realizar el requisito.
- Actor: la persona que va a intervenir en el proceso que se está desarrollando.
- Tipo: discriminado si este es funcional que lo marcaremos con una (F) y no funcional o de diseño (D).
- Entradas: es la información o datos que inducirán una respuesta adecuada de lo que se requiere que ejecute el sistema.
- Salidas: son los resultados o cumplimientos del requisito que se le realiza al software.
- Observaciones: texto informativo sobre especificación especial del requerimiento.

2.3.1.1 Sistema de escritorio

ERS Módulo reconocimiento de rostros

Tabla 1.
Módulo de reconocimiento de rostros

CÓDIGO	DESCRIPCIÓN	ACTOR	TIPO	ENTRADAS	SALIDAS	OBSERVACIONES
ERS-RRS0001	Registrar empleado	Administrador Asistente de talento humano	F	<ul style="list-style-type: none"> Nombre del empleado Captura de imagen de usuario 	Mensaje de información registrada	El usuario tendrá asignado un número único de identificación encriptado automático por el sistema.
ERS-RRS0002	Reconocer usuario	Sistema	F	<ul style="list-style-type: none"> Captura de imagen. 	Mensaje de usuario registrado o no registrado	El sistema enfocará al usuario y comparará si se encuentra registrado.

Nota: Especificación de requisitos para módulo reconocimiento de rostros del sistema de escritorio.
Elaborado por: Jimena Morales y Santiago Moya

ERS Módulo de registro de horarios

Tabla 2.
Módulo de registro de horarios

CÓDIGO	DESCRIPCIÓN	ACTOR	TIPO	ENTRADAS	SALIDAS	OBSERVACIONES
ERS-RH0001	Registrar asistencia	Empleado	F	<ul style="list-style-type: none"> • Captura de imagen • Captura automática de hora del sistema 	Mensaje de ingreso OK	<p>El cliente solicita: 10 minutos que no serán tomados en cuenta como atraso y el empleado solo podrá retrasarse hasta una hora, el sistema no permitirá registró después de esta hora.</p> <p>En el caso de salida deberá realizarlo en el horario asignado, Se registrará todo horario después de la hora de salida.</p>

Nota: Especificación de requisitos para módulo de registro de horarios del sistema de escritorio.
Elaborado por: Jimena Morales y Santiago Moya

Con la especificación requisitos realizada en la etapa de análisis a continuación usando los diagramas que integran los casos de uso y su interacción con el actor se propone una descripción detallada de la secuencia a seguir del software. Importante señalar que los diagramas parten de la información de los casos de uso del negocio (Figura 28) pero esta vez alineada a la concepción del software.

Diagramas de casos de uso

Para la presentación de los casos de uso se ha utilizado la herramienta StarUML y posterior una tabla que especifica a detalle la información necesaria para que se efectuó el proceso. En la figura 33 se muestra un ejemplo de casos de uso para autenticar usuario aplicación de escritorio y en la tabla 3 se describe de manera textual el procedimiento, el Anexo 8 del documento muestra el artefacto “Casos de uso para software aplicación de escritorio” en donde se puede observar la propuesta final de los casos de uso.

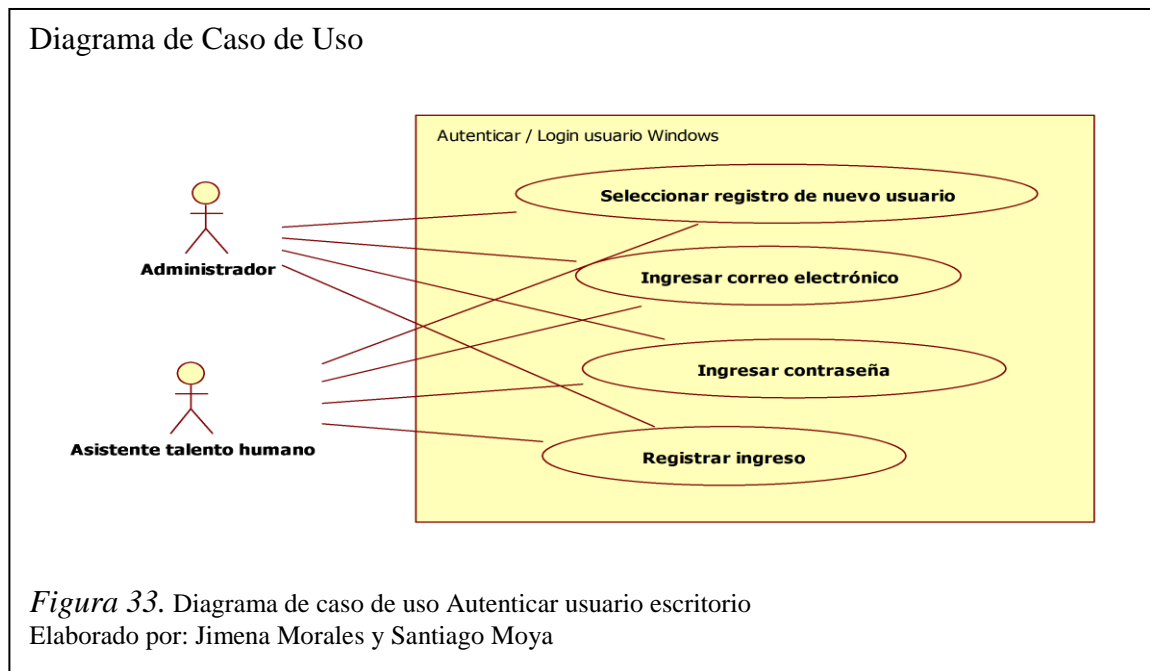


Tabla 3.
Descripción caso de uso

Nombre	Autenticar usuarios escritorio
Autor(es)	Jimena Morales / Santiago Moya
Fecha	22 de noviembre de 2013.
Descripción	Permite autenticar al usuario que tiene permisos para ingresar a la opción registro de usuario y realizar captura de imágenes.
Actores	Administrador, Asistente de talento humano
Precondiciones	El Administrador o la Asistente de talento humano deben estar registrados para acceder a la función. Deben estar creados perfiles de usuarios en la aplicación web.
Flujo normal	<ol style="list-style-type: none"> 1. La Asistente de talento humano debe seleccionar registro de nuevo usuario. 2. La Asistente de talento humano deben ingresar sus credenciales para acceder a las opciones de ingreso de datos del personal. 3. El sistema valida ingreso de información. 4. El sistema permite acceso a ventana de registro de empleado
Flujo alternativo	<ol style="list-style-type: none"> 3.1. El sistema muestra un mensaje de alerta si la información ingresada no pasa el filtro de validación.
Post condiciones	Se ingresa a la ventana de registro de empleado.

Nota: Descripción caso de uso Autenticar usuarios en sistema de escritorio
Elaborado por: Jimena Morales y Santiago Moya.

2.3.1.2 Aplicación web

Continuando en la etapa de análisis es importante ratificar e incorporar la Arquitectura de Software que es la organización fundamental de un sistema representada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución (IEEE, 2011).

De la misma manera que fue presentada la aplicación de escritorio se ha analizado la aplicación web mediante la arquitectura MVC, el Modelo, Vista y Controlador será utilizado en mayor nivel para la construcción del producto web. El requisito principal planteado en este apartado es el trato de información por los actores de manera remota

en tiempo real, dependiendo de la actividad que realizan dentro de la organización y enfocados al control del horario del personal.

Al tener claro el panorama en cuanto a las actividades de los actores es necesario incorporar los procesos más relevantes para ser automatizados en el software, esta fase de elaboración tiene como objetivo organizar la información recolectada en la fase de inicio con los cambios necesarios que son visualizados en el proceso de mejora (iteraciones).

Para esta actividad las revisiones en sitio con los actores han hecho que la planificación y estructura sean las adecuadas además de mitigar los posibles riesgos que fueron planteados al iniciar el proceso.

Al final del análisis se podrá diagramar (diseñar) el proyecto, esta etapa es sin duda una herramienta fundamental para la ejecución del sistema en donde los involucrados cubren sus expectativas, evitando cambios que alteren el enfoque del producto y sobre todo evitando gastos innecesarios (Holdings, 2009) .

ERS Módulo de Seguridad/ administración de usuarios

Tabla 4.
Módulo de Administración de Usuario

CÓDIGO	DESCRIPCIÓN	ACTOR	TIPO	ENTRADAS	SALIDAS	OBSERVACIONES
ERS-AU0001	Autenticar usuario	Administrador, Asistente de talento humano, Asistente de gerencia, Jefe de personal, Contador, Empleado.	F	<ul style="list-style-type: none"> • Correo de Usuario. • Contraseña de usuario. 	Ingreso a página Home del sistema.	Si el usuario no se encuentra registrado no podrá ingresar en el sistema.
ERS-AU0002	Registrar perfiles de usuario	Administrador o Asistente de talento humano	F	<ul style="list-style-type: none"> • Nombre del perfil. • Descripción de Perfil. • Dar permisos al usuario del perfil ingresado. 	Mensaje Ingreso OK	Se podrá asignar permisos o accesos al sistema al perfil ingresado.

ERS-AU0003	Registrar usuarios / empleados	Administrador o Asistente de talento humano	F	<ul style="list-style-type: none"> • Nombre y apellidos del usuario • Cédula de usuario • Mail de usuario. • Clave de usuario. • Edad. • Estado Civil. • Dirección. • Teléfono Fijo. • Número Celular 	Mensaje Ingreso de datos OK, generación de código de usuario	<p>La clave deberá ser creada y encriptada de forma automática por el sistema.</p> <p>El usuario por defecto tendrá estado de "activo" y si es un nuevo ingreso tendrá como estado "Nuevo" en la base de datos.</p>
------------	--------------------------------	---	---	--	--	---

Nota: Especificación de requisitos para módulo de administración de usuarios
 Elaborado por: Jimena Morales y Santiago Moya

ERS Módulo de Parametrización

Tabla 5.

ERS Módulo de parametrización

CÓDIGO	DESCRIPCIÓN	ACTOR	TIPO	ENTRADAS	SALIDAS	OBSERVACIONES
ERS-PR0001	Registro de ciudades	Administrador/ Asistente de gerencia	D	<ul style="list-style-type: none"> Nombre de la ciudad 	Mensaje de ingreso OK.	
ERS-PR0002	Registrar agencias	Administrador/ Asistente de gerencia	D	<ul style="list-style-type: none"> Nombre de agencia Ubicación de agencia. Dirección de agencia 	Mensaje ingreso de datos OK	
ERS-PR0003	Registrar horarios	Jefe de personal	F	<ul style="list-style-type: none"> Nombre del horario. Hora de entrada Hora de salida 	Mensaje ingreso de datos OK	

	Asignar jornada laboral	Jefe de personal	F	<ul style="list-style-type: none"> • Nombre de empleado. • Nombre de Horario. • Agencia. • Asignar días para el horario. 	Mensaje de Asignación OK	El horario se podrá asignar a varios empleados y en diferentes locales, el cruce de horarios está controlado por el sistema.
--	-------------------------	------------------	---	--	--------------------------	--

Nota: Especificación de requisitos para módulo de parametrización
Elaborado por: Jimena Morales y Santiago Moya

ERS Módulo de consultas

Tabla 6.
ERS Módulo de consultas

CÓDIGO	DESCRIPCIÓN	ACTOR	TIPO	ENTRADAS	SALIDAS	OBSERVACIONES
ERS-CON0001	Consultar horarios asignados	Empleado	F	<ul style="list-style-type: none"> • Ingresar nombre de la agencia. • Ingresar el nombre del empleado. • Consultar horarios 	Horarios del empleado.	Se desplegará los horarios que han sido asignados al empleado desde la fecha de consulta.

ERS-CON0002	Consultar asistencias	Jefe de personal, Asistente contable	F	<ul style="list-style-type: none"> • Seleccionar fecha de inicio • Seleccionar Fecha de Fin. • Ingresar el nombre de la agencia. • Ingresar el nombre del empleado. 	Registros de asistencia del empleado.	El registro de asistencia depende del rango de la fecha inicio y fin que selecciona el actor.
-------------	-----------------------	---	---	---	---------------------------------------	---

Nota: Especificación de requisitos para módulo de consultas
Elaborado por: Jimena Morales y Santiago Moya

Con la especificación de requisitos para el software planteada, se propone los diagramas de casos de uso, en la figura 34 se muestra un ejemplo de casos de uso para registrar usuarios en la aplicación web y en la tabla 7 la descripción textual del diagrama. En el Anexo 9 del documento se puede revisar todos los diagramas propuestos para la aplicación web en el documento “Modelos de casos de uso para software aplicación web” Versión 5.0.

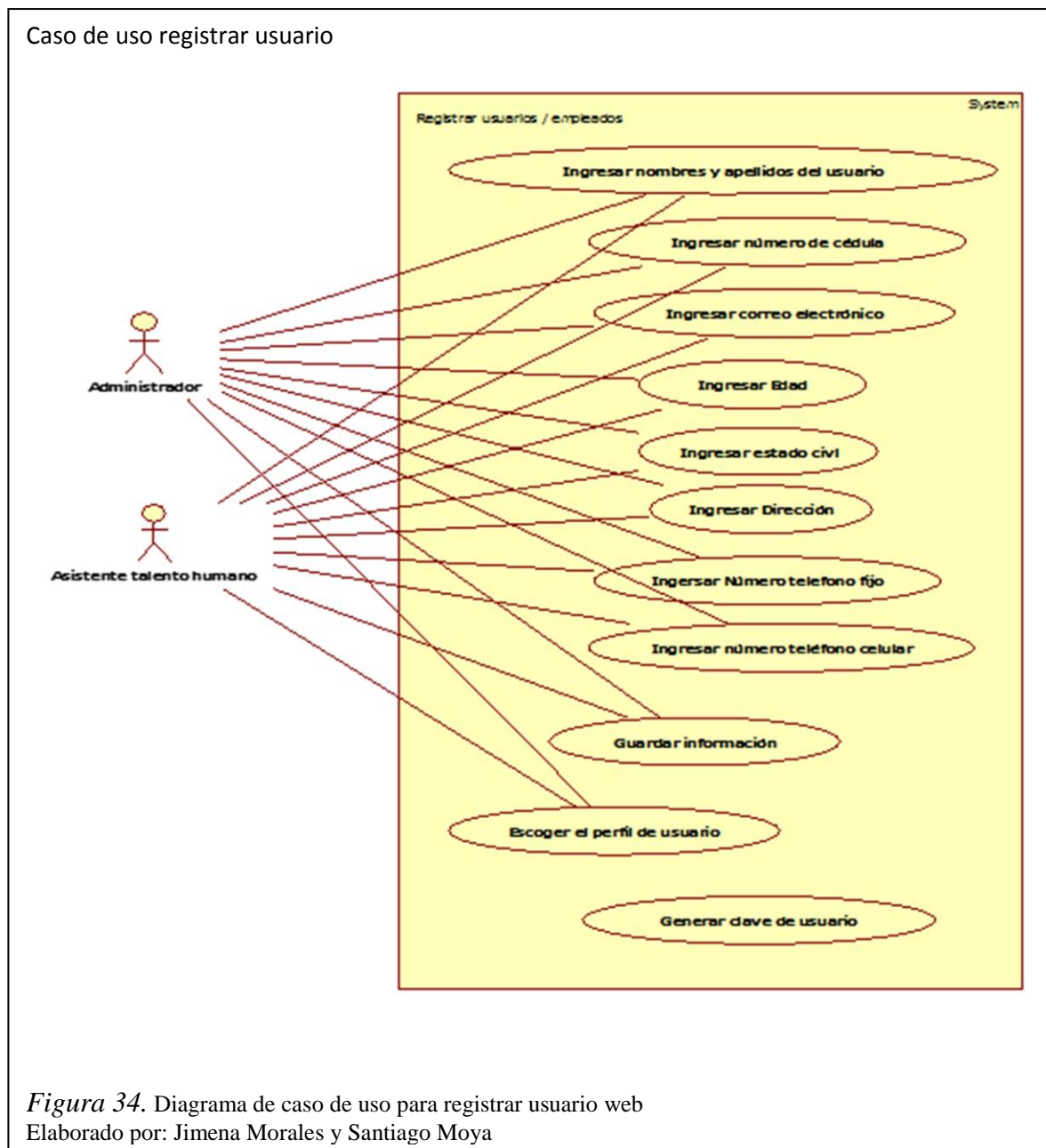


Tabla 7.
Detalle de caso de uso para registro de usuario web

Nombre	Registrar usuarios
Autor(es)	Jimena Morales / Santiago Moya
Fecha	1 de diciembre de 2013.
Descripción	
Permite el ingreso y registro de usuarios y empleados al sistema, solicita nombres y apellidos, número de cédula, correo electrónico al guardar genera una clave que se guarda encriptado en la base de datos, finalmente se le asigna un perfil al usuario para que pueda navegar en el programa.	
Actor(es)	
Administrador o Asistente de Talento Humano	
Precondiciones	
Debe estar creado previamente los perfiles El Administrador y Jefe de personal debe estar registrado o logueado previamente.	
Flujo normal	
<ol style="list-style-type: none"> 1. El administrador o jefe de personal ingresa registro de usuarios para crear un nuevo usuario. 2. El administrador o jefe de personal ingresa información del usuario/empleada: <ul style="list-style-type: none"> • Nombre y apellido del usuario • Cédula de identidad. • Mail de usuario • Edad. • Estado civil • Dirección. • Teléfono fijo. • Número de celular. 3. El administrador o jefe de personal da clic sobre el botón Guardar. 4. El código de usuario/clave es autogenerado por el sistema 5. El sistema valida la información ingresada según la asignación de registro de datos, desplegando un mensaje de ingreso de datos OK. 6. El administrador o jefe de personal escoge el perfil del usuario. 7. El Administrador da clic sobre el botón Grabar. 	
Flujo alternativo	
<ol style="list-style-type: none"> 5.1. El sistema muestra un mensaje de error si la información ingresada no pasa la validación de cada uno de los campos. 5.2. El sistema encripta la clave del usuario. 5.3. El usuario será asignado con el estado de nuevo. 	
Post condiciones	
Se registra en la base de datos los registros del usuario/empleada ingresado.	

Nota: Detalle da caso de uso para ingreso de datos de usuario en sistema web.

Elaborado por: Jimena Morales y Santiago Moya

Al concluir con el análisis considerado como la primera etapa de la fase de elaboración, se ha seleccionado la arquitectura para trabajar sobre sus conceptos además alinear y obtener los casos de uso del software guiándose en la especificación de requisitos, trabajando directamente con las necesidades del usuario, ratificando y corrigiendo en el proceso los requerimientos de cada uno de los actores, a la vez manteniendo la visión evolutiva de los requisitos incluso en la etapa de pruebas del software.

Finalmente se posee un panorama claro y racional de los aspectos y expectativas del usuario, identificando las reglas con las que los desarrolladores trabajarán en la construcción del software; continuando con la secuencia se diseña la propuesta inicial de la base de datos, sus relaciones y distribución usando la información adquirida, posterior a la actividad se pasará a la segunda etapa de la elaboración y presentar los diagramas con los que el equipo de desarrollo construirá el software.

2.4. Diseño (Segunda etapa fase elaboración)

La etapa de diseño en el desarrollo del proyecto se fundamenta en la información del análisis, utilizando el sustento de los casos de uso se elaboraron los diagramas y artefactos propuestos por RUP, para presentar de manera secuencial, ordenada y práctica los datos requeridos por el equipo de trabajo y desarrollar el software. Utilizando esta estructura el programador implementa módulos los mismos que pueden trabajar de forma independiente cumpliendo así con uno de los requisitos de la metodología que plantea proporcionar entregables al cliente, las propuestas iniciales del análisis se han mejorado en el proceso de diseño los cambios principalmente obedecen a la estructura que requiere el usuario al interactuar con el software. Los diagramas propuestos a continuación son mapas o guías para conseguir el objetivo principal del proyecto entregar un producto que resuelva la forma de registrar la asistencia del usuario y la información recopilada sea tratada de manera remota. En la segunda etapa de la fase de elaboración se diseña y muestra la construcción de la base de datos asimismo la información es expuesta y ordenada en los diagramas de clases, actividades, secuencias, navegación e interfaces que finalmente serán programadas en la etapa de construcción del sistema.

2.4.1. Base de datos

La base de datos del proyecto está desarrollada en Microsoft SQL 2012, este contenedor de información es el punto de partida para almacenar y controlar datos importantes como necesarios en donde interactúan el sistema de escritorio y la aplicación web mediante un intérprete (web service). Las tablas y campos del contenedor se construyeron bajo las necesidades de las clases creadas e identificadas en el proceso de análisis, en todas las etapas del proyectos se realizaron cambios y ajustes implementados en cada iteración hasta llegar al diseño final de la base de datos. En la figura 35 se presenta el modelo conceptual de la base de datos después de haber pasado el proceso de análisis, diseño e implementación; en la figura 36 se presenta el modelo físico de la base de datos en donde se visualizan los atributos de cada uno de las tablas.

Modelo conceptual de base de datos

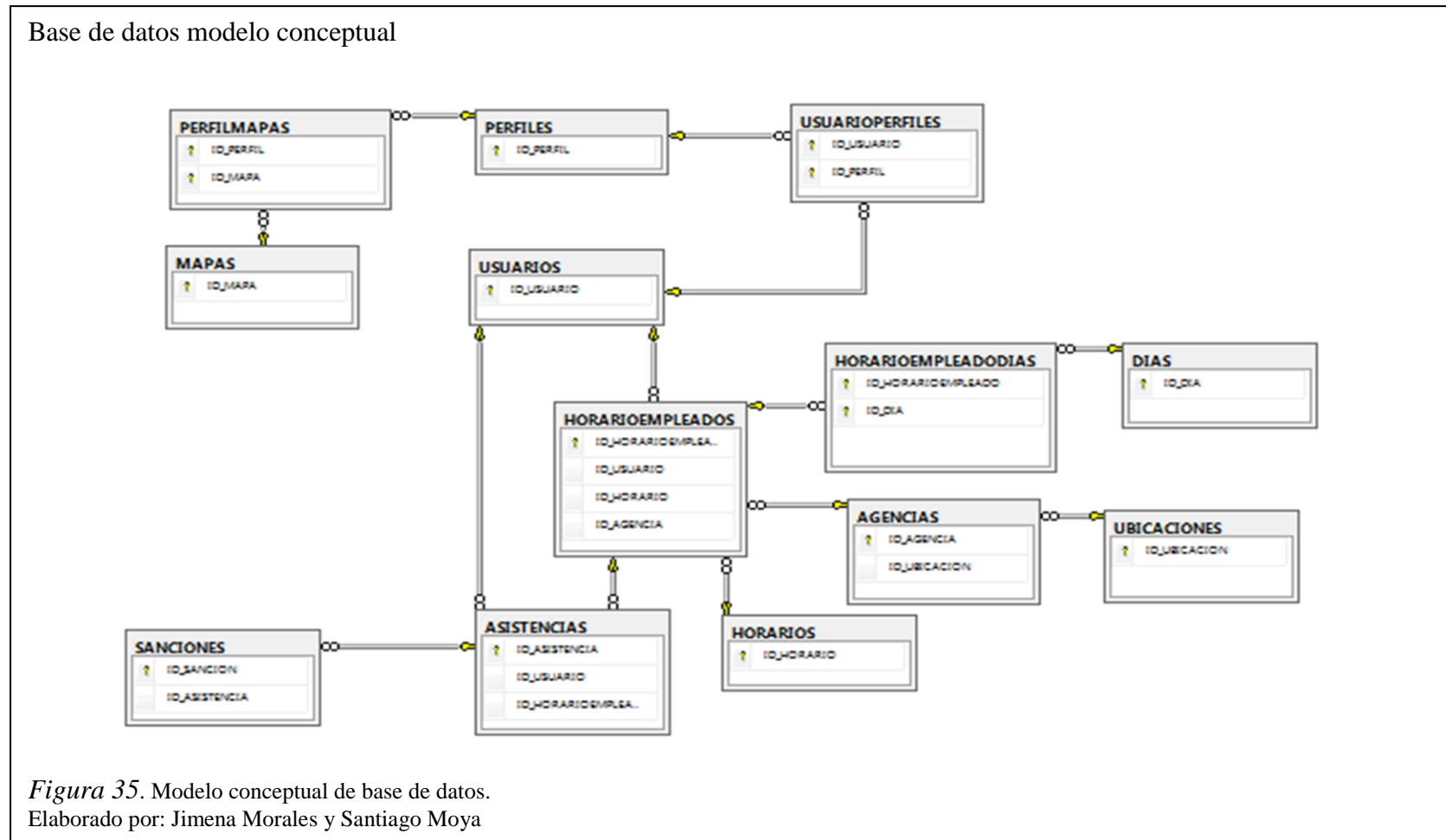


Figura 35. Modelo conceptual de base de datos.
Elaborado por: Jimena Morales y Santiago Moya

Modelo físico de base de datos

Base de datos modelo físico



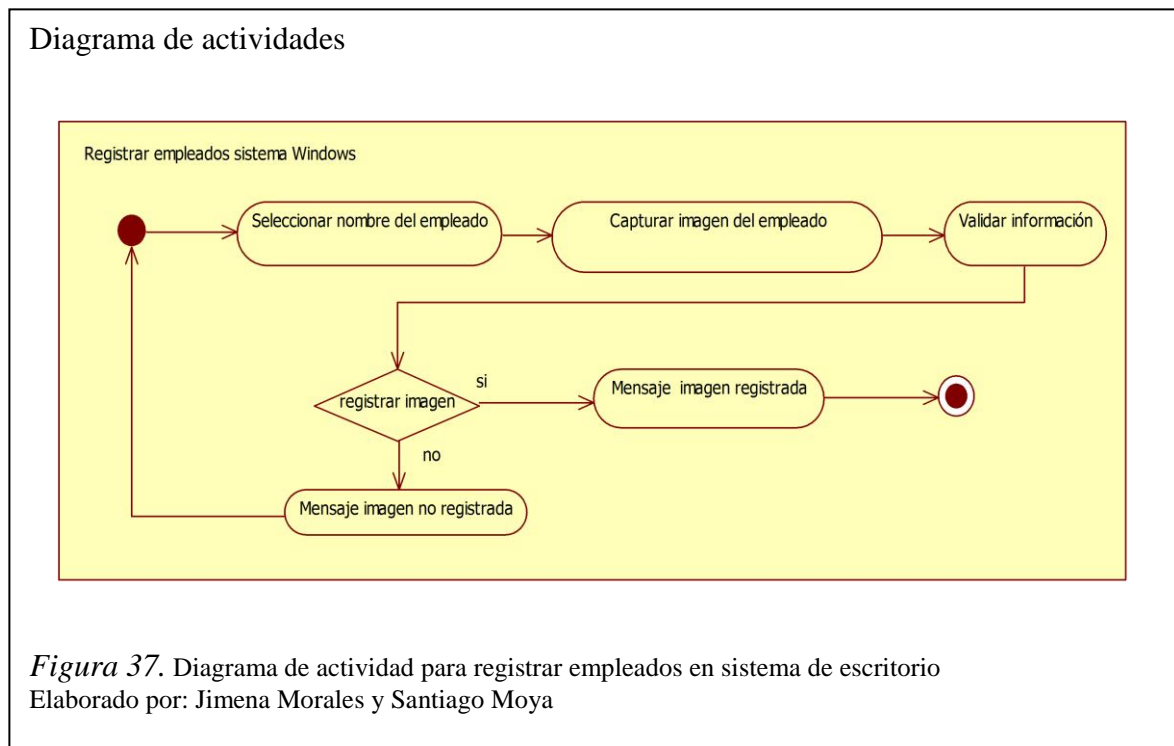
Figura 36. Modelo físico de la base de datos
Elaborado por: Jimena Morales y Santiago Moya

En el siguiente apartado se propone los diagramas de: actividades, clases, secuencias, navegación e interfaces, divididas por cada una de las aplicaciones.

2.4.2. Sistema de escritorio

2.4.2.1 Diagramas de actividades

El diagrama de actividades demuestra la serie de acciones que deben ser realizadas por caso de uso y las distintas rutas que pueden tomarse dentro de la actividad propuesta (Markmonitor, 1991). En la figura 37 se muestra un ejemplo del diagrama propuesto para registrar empleados usando el sistema de escritorio, el Anexo 10 evidencia todos los diagramas de actividades generados para el sistema de escritorio.



2.4.2.2 Diagrama de clases

El diagrama permite visualizar las relaciones entre las clases que involucran el sistema identificando la asociación, herencia y uso de contenido, parte fundamental de la arquitectura orientada a objetos (Chile, 2014). En la Figura 38 se propone el diagrama de clases utilizado para construir el software de escritorio.

Diagrama de Clases

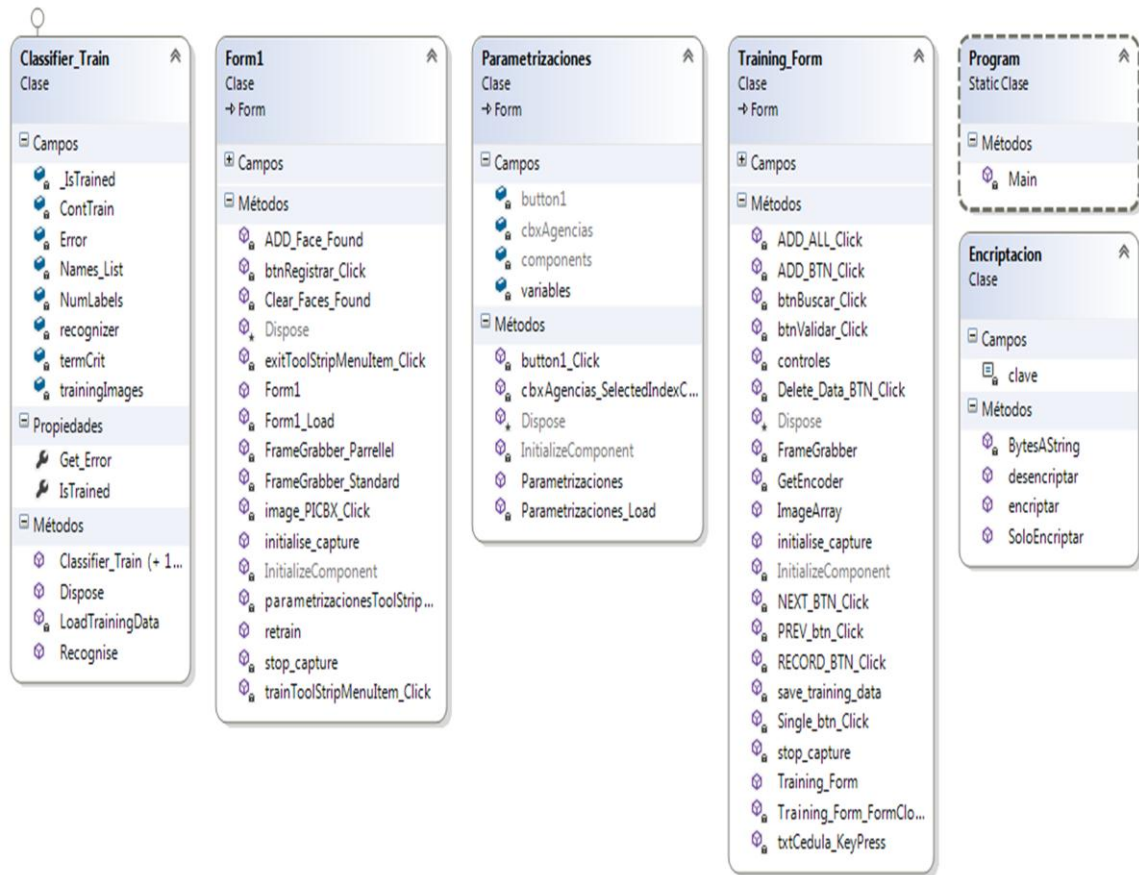


Figura 38. Diagrama de clases para aplicación de escritorio
Elaborado por: Jimena Morales y Santiago Moya

2.4.2.3 Diagrama de Secuencia

El diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación o sistema informático en una línea de tiempo de manera ordenada (Macrkmonitor, 1991). En la figura 39 se propone un ejemplo de diagrama de secuencia para registrar al empleado por reconocimiento de rostro. El Anexo 10 muestra la propuesta final de los diagramas de secuencias para la aplicación de escritorio.

Diagrama de secuencia

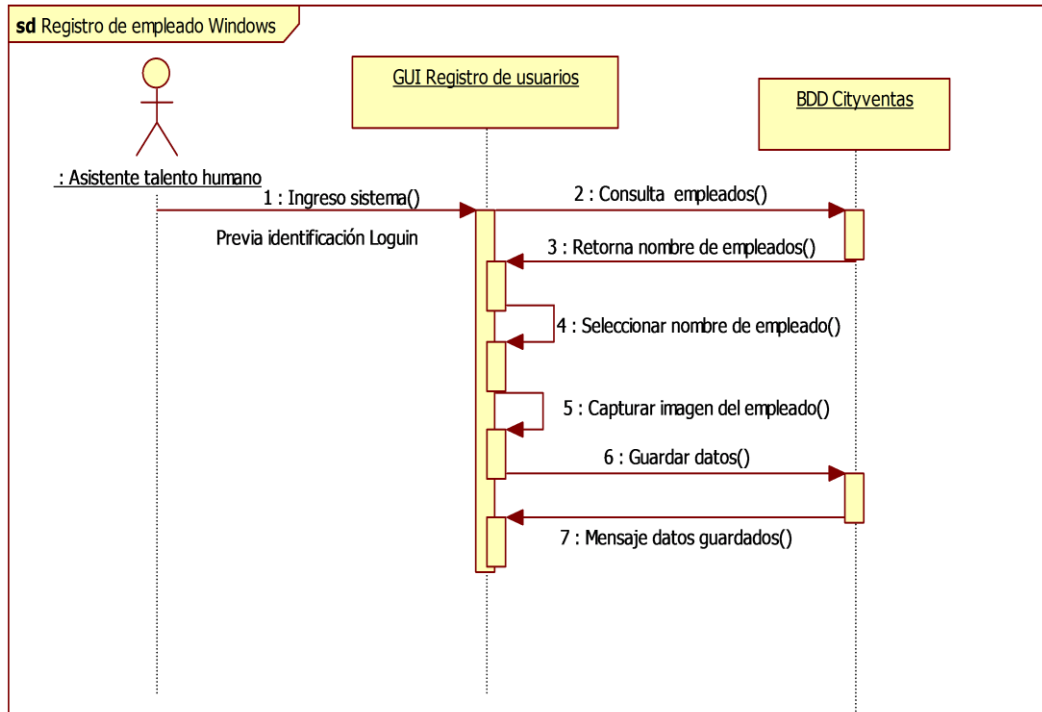
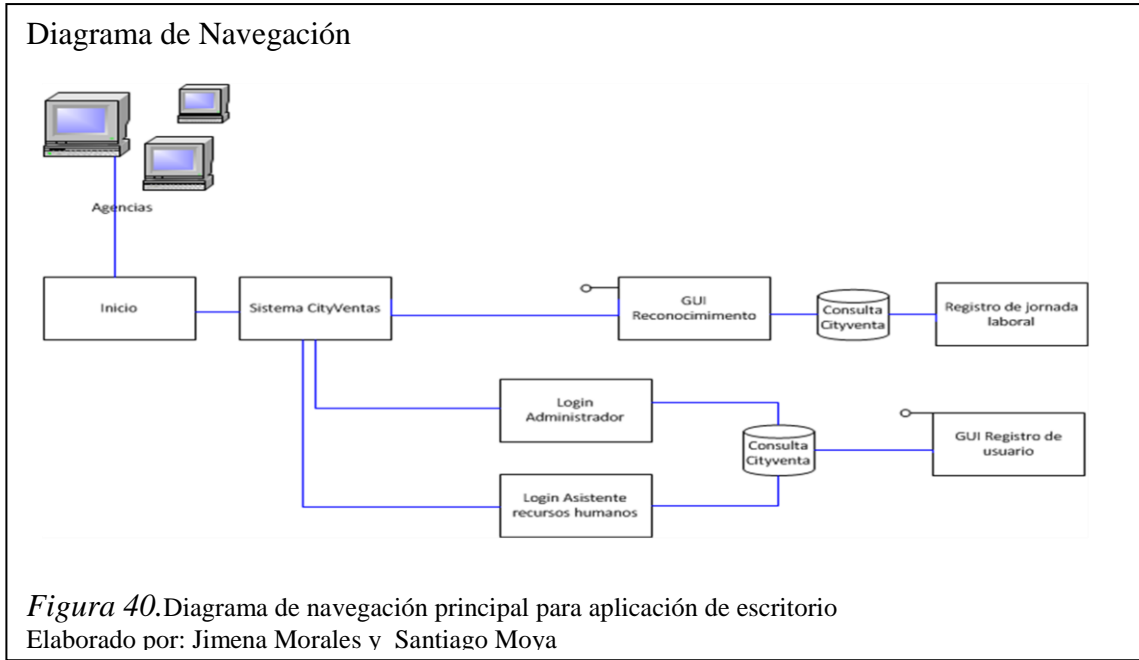


Figura 39. Diagrama de secuencia para registrar empleados en sistema de escritorio
Elaborado por: Jimena Morales y Santiago Moya

2.4.2.4 Diagrama de navegación

El diagrama de navegación expresa todas las relaciones de jerarquía y secuencia además permite elaborar escenarios de comportamiento de los usuarios (Blogspot, 2000). En la Figura 40 se muestra la distribución de navegación principal para la aplicación de escritorio. El Anexo 11 del documento propone los diagramas de navegación de la aplicación de escritorio.



2.4.2.5 Diagrama de Interfaces

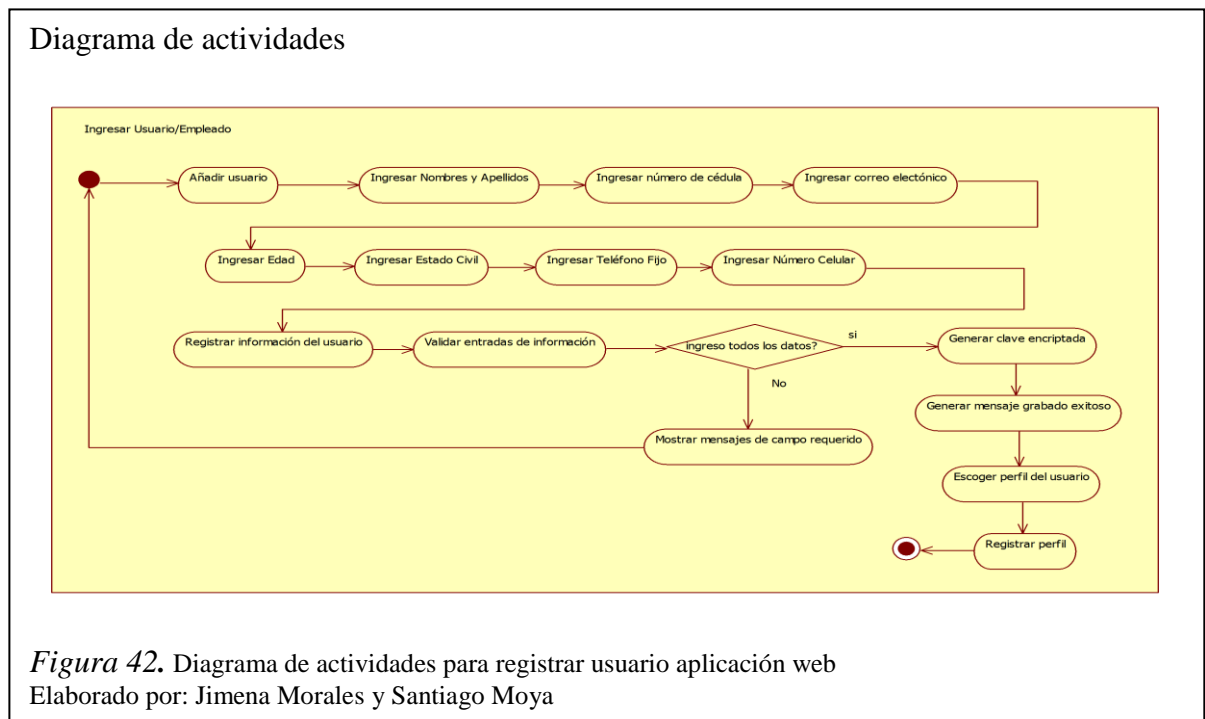
Es la representación visual disponible en donde interactúa usuario con el software (Blogspot, 2000), es decir, un entorno gráfico dispuesto para trabajo del usuario en donde se obtiene reconocimiento de rostro y registro de horarios. En la Figura 41 se muestra la propuesta del interface para reconocimiento de rostro del usuario (Nota: la imagen del usuario será incluida si previamente el usuario se encuentra registrado en la aplicación web), en el Anexo 11 se puede observar la presentación total de los diagramas de interface.



2.4.3. Aplicación web

2.4.3.1 Diagrama de actividades

El diagrama de actividades es la representación gráfica del proceso del control de información, en donde se plantea la raíz del diseño de la aplicación web. En la Figura 42 se propone un ejemplo para el diagrama de actividad del registro de usuario en la aplicación web; el Anexo 12 se muestra todos los diagramas de actividades propuestos.



2.4.3.2 Diagrama de clases

Es un diagrama estático que muestra las clases que componen el sistema además de los métodos que lo conforman, relacionando de manera eficiente cada uno de los atributos. En la Figura 43 y 44 muestran el gráfico de clases finales del proyecto después de haber pasado por los cambios propuestos por la metodología y aplicando la arquitectura MVC.

Diagrama de Clases para aplicación web.

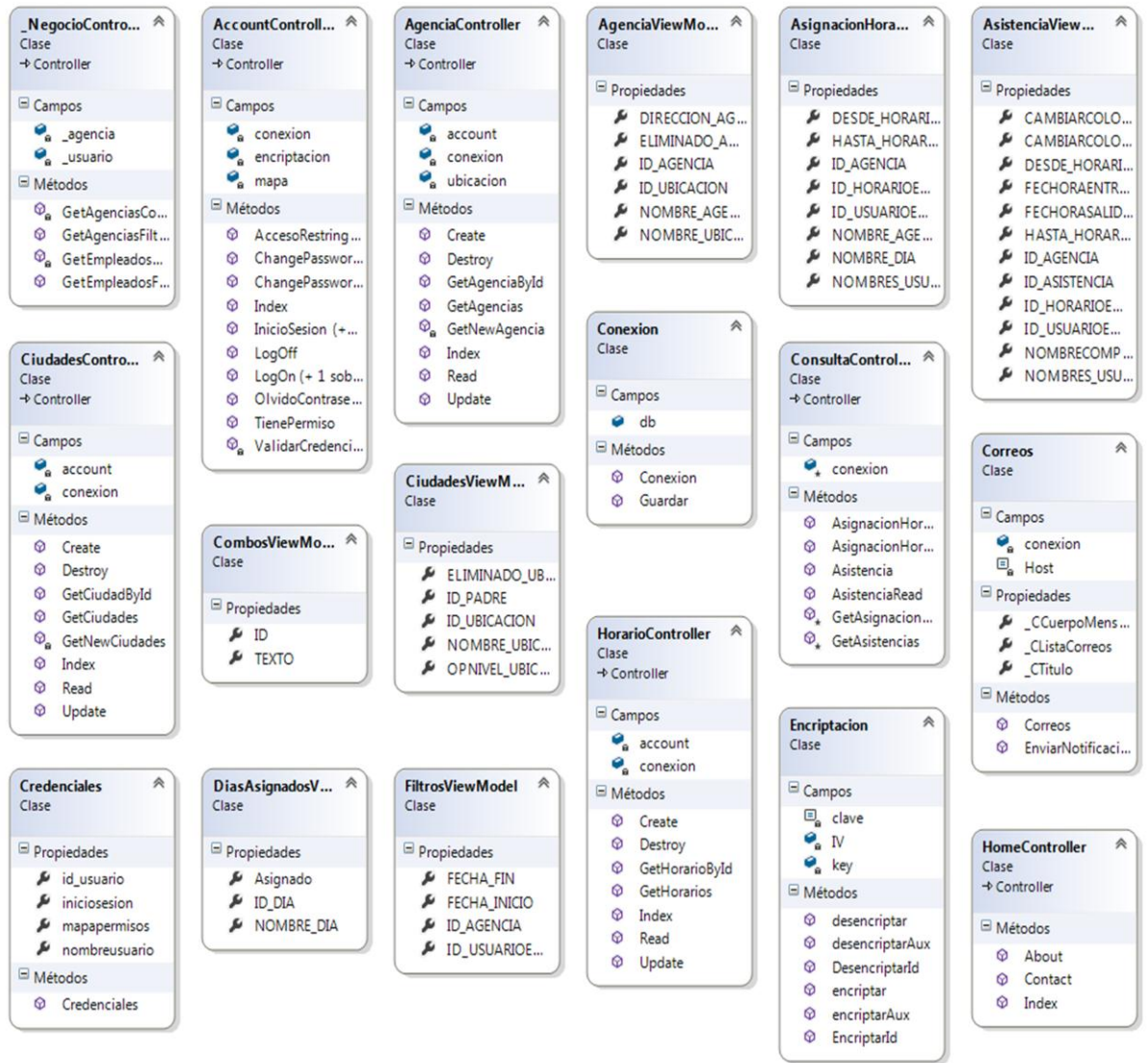


Figura 43. Primera parte de diagrama de clases para aplicación web
Elaborado por: Jimena Morales y Santiago Moya.

Diagrama de clases para aplicación web.

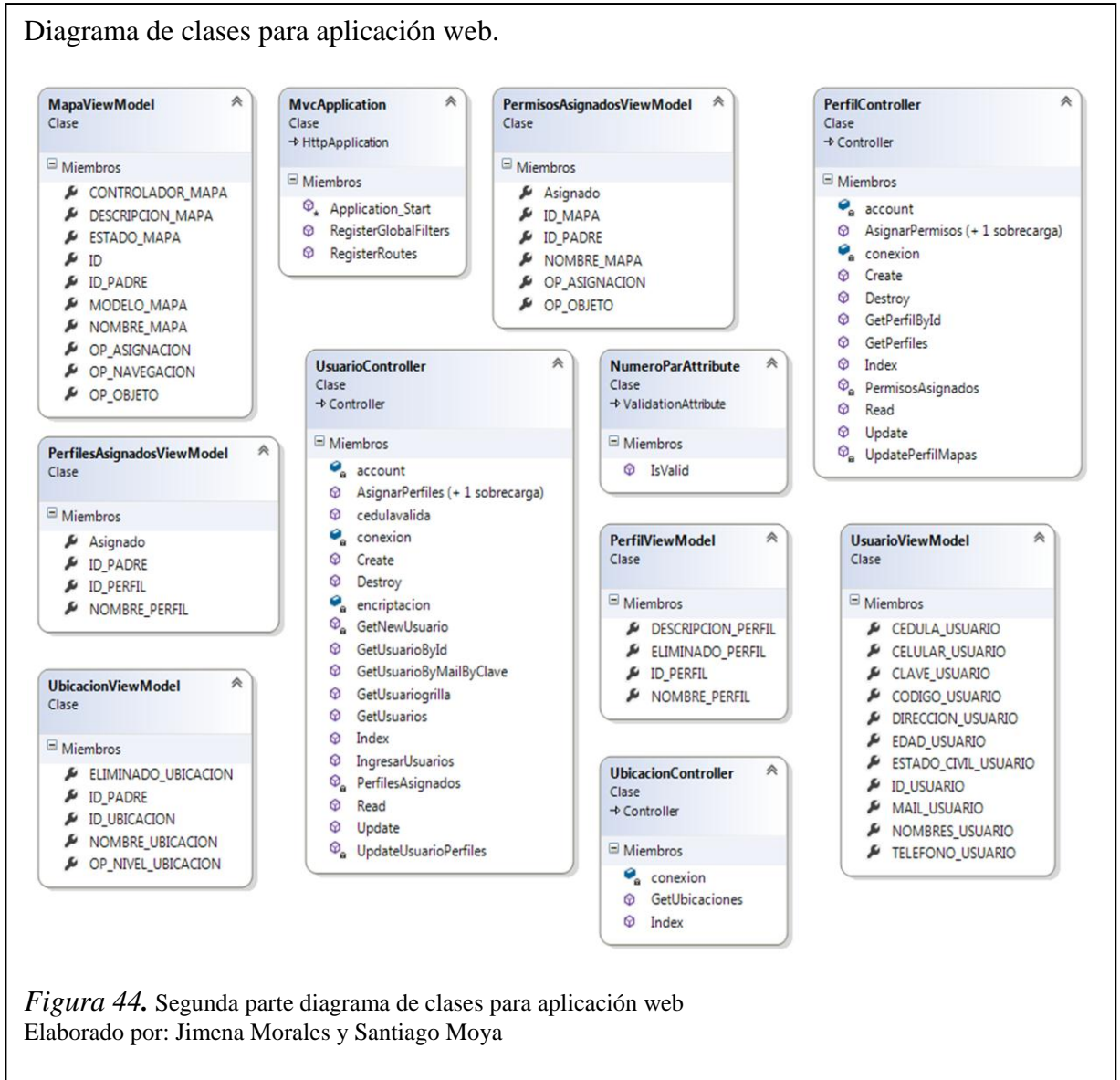


Figura 44. Segunda parte diagrama de clases para aplicación web
Elaborado por: Jimena Morales y Santiago Moya

2.4.3.3 Diagrama de secuencia

El diagrama de secuencia nos sirve para mostrar (modelar) la interacción entre los objetos del sistema en secuencia cronológica de las acciones a realizar en el sistema y aclarar la forma de acción de los casos de uso. A continuación se muestra en la Figura 45 el gráfico para el ingreso de información del usuario, para explorar la propuesta final de los diagramas revise el Anexo 12 del documento.

Diagrama de secuencia

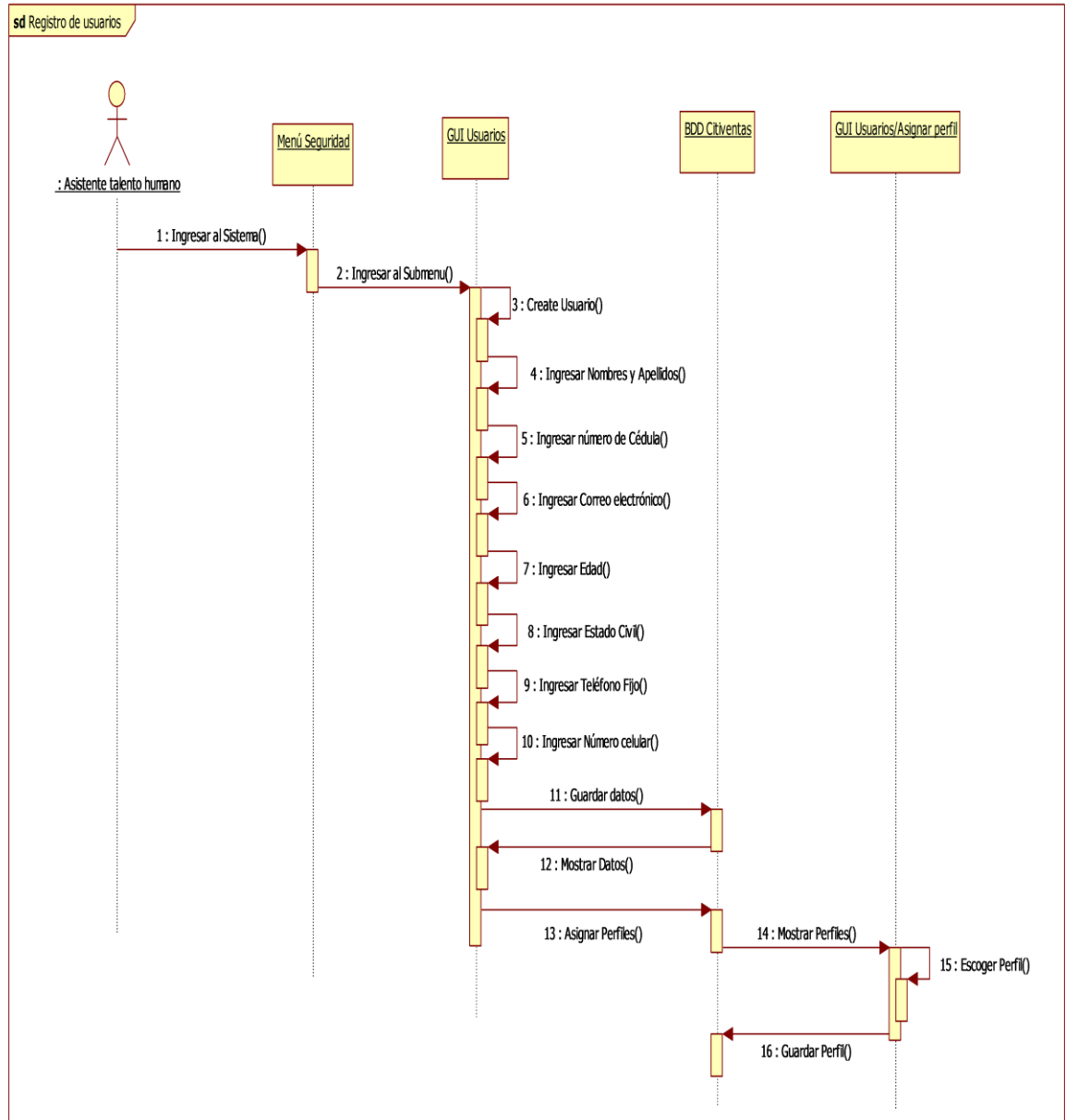
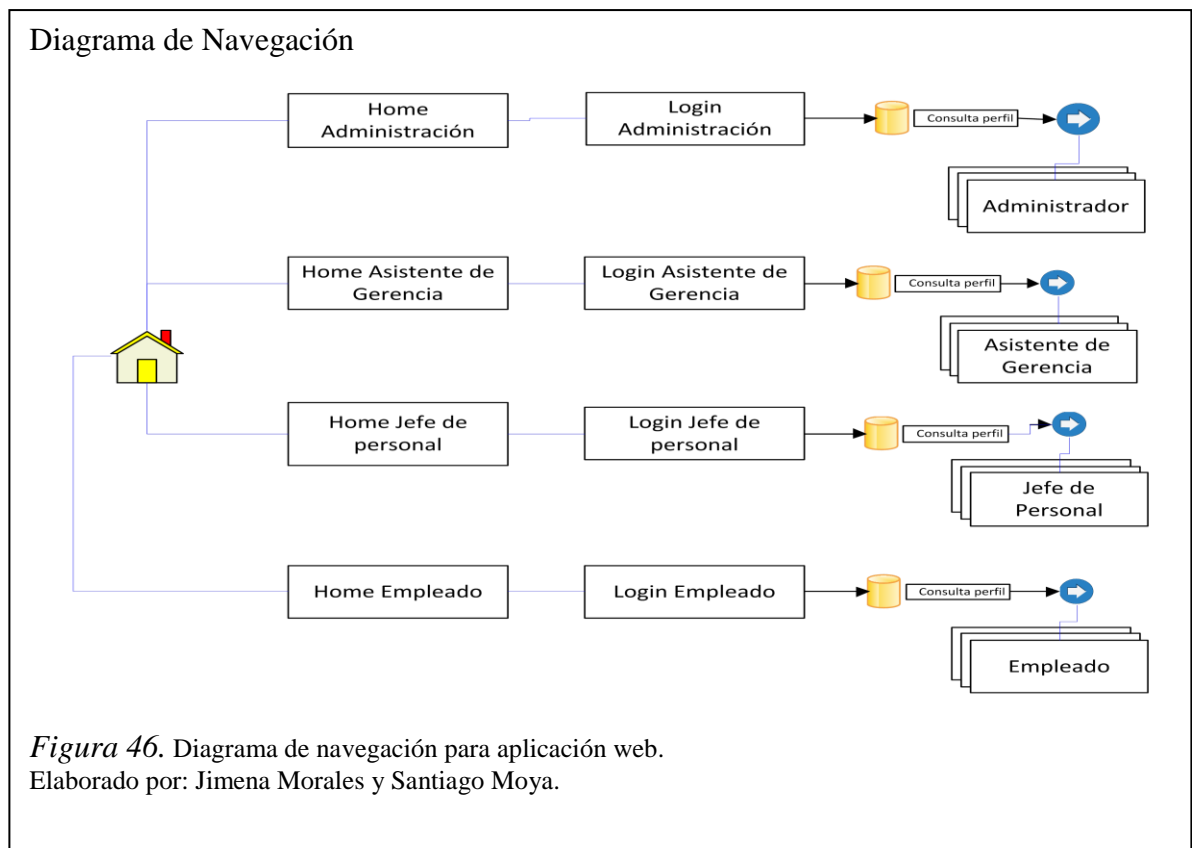


Figura 45. Diagrama de secuencia para registro de usuario
Elaborado por: Jimena Morales y Santiago Moya

2.4.3.4 Diagrama de navegación

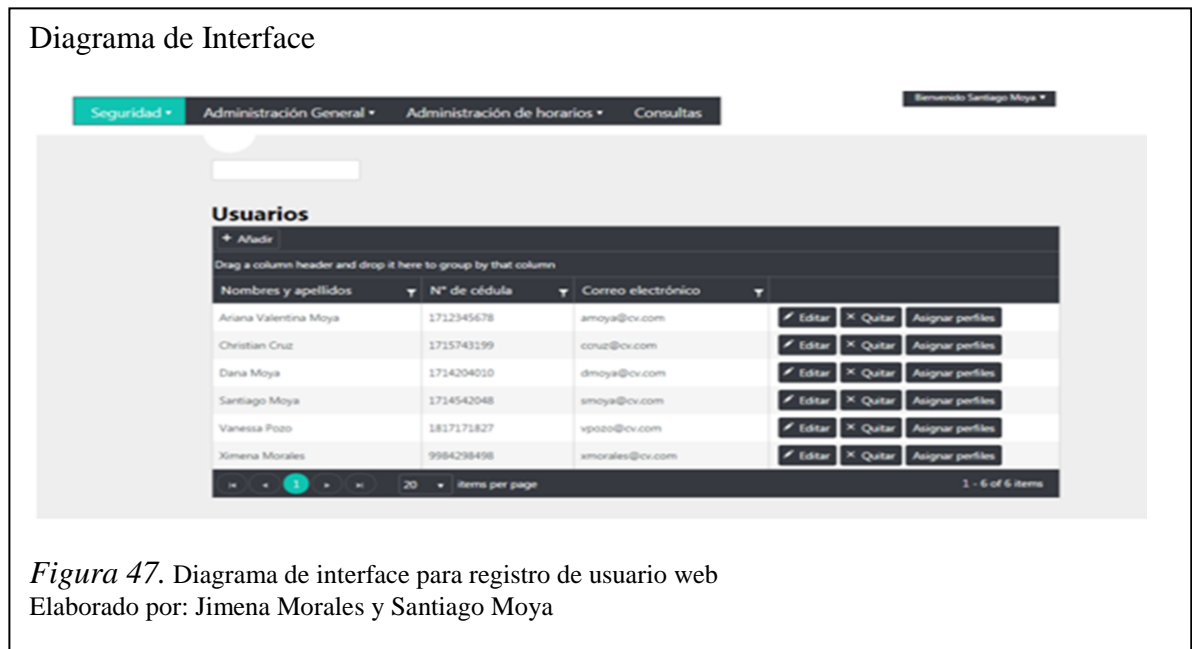
El diagrama de navegación muestra la organización del contenido del sitio Web, mediante una estructura jerárquica desde el Home (inicio) hasta cualquier contenido enlazado. En la Figura 46 se muestra el gráfico para navegación principal de la aplicación web, el Anexo 13 muestra todos los diagramas de la aplicación.



2.4.3.5 Diagrama de Interfaces

Al acceder a la aplicación web desde cualquier ordenador que se encuentre conectado a Internet, desplegará la solicitud de credenciales para acceso, esto permitirá que de acuerdo al perfil asignado al usuario se otorgue el menú de trabajo. Se decidió que la aplicación web sea desarrollada sobre una Grilla, la medida fue adoptada porque mayoría de actividades informáticas de CityVenta se realizan en macros de hojas de Excel, al ser el primer software informático de la empresa es relevante que sus

funcionarios se adapten de forma rápida y sencilla a la aplicación además crear un ambiente conocido pero con mayor funcionalidad. Kendo proporciona el template de una grilla que maneja de manera eficiente los datos de un contenedor, a pesar de que el template brinda beneficios al desarrollador sin el aporte conceptual, técnico y profesional de los programadores no se podría plasmar de manera efectiva los requerimientos que se están analizando y diseñando en este apartado. Para la concepción del producto se planteó un conjunto de botones o mandos que facilitarían las actividades del usuario; la primera fila un botón añadir para todas las presentaciones, en la segunda fila como propiedad exclusiva de la grilla un espacio para ordenar y agrupar el contenido desde la tercera en adelante la presentación de los datos del sistema; al costado derecho un conjunto de botones que ejecutarán las acciones que se proponen en los módulos. La grilla tiene como particularidad generar paginación de acuerdo a la selección del usuario y el uso de filtros para agrupamiento de información. En la Figura 47 se representa un ejemplo del diseño de pantalla que será mostrado al cliente para ingreso de usuarios como propuesta inicial del producto, siendo parte del primer ejecutable del software; toda la propuesta de diagramas se puede observar en el Anexo 13 del documento.



CAPÍTULO 3

CONSTRUCCIÓN Y PRUEBAS DEL SISTEMA.

3.1 Fase de construcción y pruebas del sistema.

Siguiendo con el proceso propuesto por la metodología RUP se ha llegado a la fase de construcción del producto después de pasar por las revisiones de la fase de elaboración en donde se ha definido claramente la arquitectura, el diseño además se ha propuesto los diagramas que son guías indispensables para la programación y puesta a punto del producto. La arquitectura MVC en el caso del desarrollo web incorpora normas propias de programación además es necesario aclarar las reglas de programación seleccionadas por los desarrolladores en la elaboración del software, generando organización y estructura. A continuación se menciona las normas seleccionadas.

3.2 Normas de programación

Las normas o estándares de programación son formas de entender el trabajo realizado por los desarrolladores además que las personas involucradas en el programa puedan comprender el código gracias a la definición de la estructura y organización del código fuente, es decir, se define la forma de declarar las variables, clases, comentarios; utilizando determinada nomenclatura. Las normas conocidas en el mercado para el desarrollo de programas son: Notación húngara, notación pascal casing y notación camelcasing (Enom, 2008).

3.2.1 Selección de las normas de programación

Para el desarrollo del programa se ha utilizado las consideraciones y recomendaciones de la notación pascal casing, la notación propone dentro del procedimiento de programación que la letra inicial del identificador debe ser mayúscula. La nomenclatura está compuesta por tantas palabras como sean necesarias. La primera letra de cada una

de las palabras irá siempre en mayúsculas, se obvia el uso de artículos (SRL, 2006). Además del uso para declaración de variables usando únicamente letras minúsculas y en el casos específicos separadas por el guión bajo (_).paquetes.

3.2.2 Estándares para paquetes

Los paquetes se declaran usando la norma de notación Pascal Casing. Ejemplos:

- Models.
- Views.
- Controlllers.

3.2.3 Estándares para creación de clases.

El nombre de la clase tendrá la referencia del nombre de la tabla que contiene la base de datos con la que se encuentra concatenada. Además por manejo de MVC se relacionará con el nombre del contenedor en donde se encuentre instanciada. Ejemplos:

- AgenciaController(); // clase que se encuentra dentro del paquete controladores y se relaciona a las agencias.
- AgenciaViewModel (); // clase que se encuentra en el paquete controlador vista y relaciona al manejo de agencias.

Se crearán atributos de la clase con el nombre de los campos de la tabla a los que se refiere, además el tipo de datos de estos atributos deben ser similares a los de los campos de la tabla a quien referencian. Ejemplo:

Ejemplo de Tabla

Column Name	Condensed Type	Nullable
ID_AGENCIA	int	No
ID_UBCACION	int	No
NOMBRE_AGENCIA	varchar(50)	No
DIRECCION_AGENCIA	text	Yes
ELIMINADO_AGENCIA	bit	No

Figura 48. Ejemplo de campos de la tabla agencia
Elaborado por: Jimena Morales y Santiago Moya

```
public int ID_AGENCIA { get; set; }
public int ID_UBICACION { get; set; }
public string NOMBRE_AGENCIA { get; set; }
public string DIRECCION_AGENCIA { get; set; }
public bool ELIMINADO_AGENCIA { get; set; }
public string NOMBRE_UBICACION { get; set; }
```

3.2.4 Estándares para creación de métodos.

- El nombre del método debe estar en verbo en infinitivo según el manejo global del software se puede usar palabras básicas del idioma inglés fusionadas con palabras en español respetando la estructura que MVC de Visual Basic .Net propone. (Add, Update, Create, Read, UpdateUsuarioPerfiles, etc.)

Ejemplo: `public ActionResult AsignarPerfiles(int id) { }`

- Las palabras que conforman el nombre del método deben estar unidas y con la primera letra de cada palabra usando mayúsculas.

```
private void UpdateUsuarioPerfiles(string[] PermisosSeleccionados,
    USUARIO ActualizarUsuario){}
```

3.2.5 Estándares para creación de variables

- Todo se declara y maneja en singular.
- Todas las declaraciones de variables se realizan al inicio de la función o clase.
- Se declara en español e inglés las palabras en inglés son básicas y argumentan contenido que tendrá la variable.
- En las clases las variables se las declaran al inicio de la clase.
- Las variables son declaradas como públicas y pueden ser usadas en el programa en donde sean requeridas usando referencias y punteros a las mismas.

3.2.6 Estándares para instancia de clases

Para la instancia de clases se utilizará el nombre de la instancia en español y con letras minúsculas sin tildes y se la instanciará con el nombre de la clase a la que se quiera referir: `nombreclase = new Clase();`

Ejemplo: `conexion = new Conexion();`

3.2.7 Estándares para ordenamiento y estructura

- Toda codificación poseerá sangrías y no se colocará llaves de abrir o cerrar en la misma línea de código, con la excepción de declaración de variables o en estructuras donde se deba respetar una sola línea como es el retorno de una variable `return({})`.

Ejemplo:

```
if ( .... )
    {
    conexion.Guardar();
    }
```

3.2.8 Estándares para comentar código

Se comentará eventos importantes dentro de la programación que se realice en el sistema y cada método tendrá una descripción de la utilidad y funcionalidad dentro del sistema.

Ejemplo:

```
//ESTE MÉTODO ASIGNA PERFILES A UN USUARIO
public ActionResult AsignarPerfiles(int id)
{
    var AsignarPerfiles = conexion.db.USUARIOS
}
```

3.3 Construcción (Tercera fase de RUP)

3.3.1 Construcción aplicación web.

Después de definir los estándares con los que se construirán el software, revisando los datos entregados en el análisis y diseño del producto, se decide elaborar la aplicación web, la aplicación genera los datos para que el sistema de escritorio pueda desenvolverse como recolector de información para el registro de la jornada laboral. Se crea el proyecto usando C# en la opción aplicación web de ASP.NET MVC4 con el nombre de “WebCityVenta”. Previamente con las consideraciones conceptuales del diseño del contenedor de datos se elabora la primera versión práctica de base de datos en SQL Server 2012. Usando la estructura del contenedor de datos concatenados con las

clases propuestas en el diseño creando la clase de conectividad entre la base de datos y el WebCityVenta ayudados de Linq.

El siguiente paso fue organizar y elaborar el primer módulo en este caso Parametrización que contiene el fundamento del negocio, es decir, el registro y asignación de horarios, para esto se debió crear la primera versión de ingreso de usuarios del sistema así como del registro de las agencias en donde funciona CityVenta. Siguiendo los procesos de la metodología se consigue el primer ejecutable que fue presentado al cliente para su revisión y pruebas las mismas que serán abordadas en el apartado 3.3 de este capítulo. De las pruebas realizadas se evidencia las primeras observaciones que el actor (Jefe de personal) identifica en cuanto a la asignación de horarios a los empleados, en primera instancia según los datos recopilados los empleados estaban asignados a un solo local para su jornada laboral y se frecuentaba el proceso por toda la semana salvo excepciones, realizando las pruebas con el cliente y gracias al control automatizado se solicita cambiar esta visión para que los empleados puedan ser asignados a varios locales en la semana y con la posibilidad incluso de trabajar en dos locales al día; este cambio da un giro a la lógica de programación en cuanto al cruce de horarios y la asignación de los locales. Solucionando utilizando un método para validar cruce de horas determinando consultas cruzadas cuando se asigne el día y el local comercial en donde trabajará el empleado.

```
private HORARIO EMPLEADOS ValidarCruceHoras(HORARIOS horario, int idUsuario, int idDia,
int idHorarioEmpleado)
{
    var validación = from hor in conexion.db.HORARIOS
from horemp in conexion.db.HORARIO EMPLEADOS
from hed in conexion.db.HORARIO EMPLEADODIAS
where hor.ID_HORARIO == horemp.ID_HORARIO
&& horemp.ID_HORARIO EMPLEADO == hed.ID_HORARIO EMPLEADO
&& horario.DESDE_HORARIO >= hor.DESDE_HORARIO
&& horario.DESDE_HORARIO <= hor.HASTA_HORARIO
&& horemp.ELIMINADO_HORARIO EMPLEADO == false
&& horemp.ID_USUARIO == idUsuario
&& hed.ID_DIA == idDia
select horemp;
return horarioEmpleado.FirstOrDefault();
}
```

Otro punto importante que se verificó en las pruebas fue la necesidad de recolectar mayor cantidad de datos del empleado, inicialmente se ingresaba nombres, cédula y

email finalmente se optó por solicitar datos como estado civil, números de teléfono, fecha de nacimiento; por la relevancia de los datos para el trabajo diario de los actores de mayor jerarquía, todos los datos llevan su respectiva validación de ingreso.

```
[DisplayName("Correo electrónico")]
[Required(ErrorMessage = "El campo {0} es requerido")]
[MaxLength(50, ErrorMessage = "Número de caracteres permitidos = 50")]
[RegularExpression(@"\w+([-+.]\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*",
                ErrorMessage = "Ingrese un correo válido")]
[DataType(DataType.EmailAddress)]
publicstring MAIL_USUARIO { get; set; }

[DisplayName("Estado civil")]
[Required(ErrorMessage = "El campo {0} es requerido")]
[RegularExpression(@"^\d{2}$", ErrorMessage = "Ingrese fecha correcta")]
publicint? EDAD_USUARIO { get; set; }

[DisplayName("Celular")]
[Required(ErrorMessage = "El campo {0} es requerido")]
[RegularExpression(@"^[0-9]{2,3}-?[0-9]{6,7}$",
                ErrorMessage = "Ingrese número celular correcto ")]
publicstring CELULAR_USUARIO { get; set; }
```

En cuanto a las agencias se solicita el ingreso de la ciudad en donde se ubicará el local comercial, en el planteamiento inicial internamente se daría opción a seleccionar las ciudades principales del Ecuador, pero se visualiza la posibilidad de llegar a lugares que no se podrían considerar en un contenedor estático por esta razón la necesidad de ingresar las ciudades al contenedor de datos para posterior ser asignadas. Con la actividad antes mencionada se concluye con el primer módulo cumpliendo con el objetivo de asignación de horarios en el cual el entregable fue puesto a consideración del cliente en tres sesiones y las observaciones realizadas fueron atendidas para pasar al siguiente módulo; se cumple de esta manera con los requisitos fundamentales de la metodología realizar módulos entregables y generar iteraciones de refinamiento.

El siguiente módulo abordado en la construcción de la aplicación web fue Seguridad y Administración de Usuario, en el cual se crean perfiles, se realiza refinamiento para ingreso de usuarios y accesos al sitio web mediante usuario y contraseña; se decide utilizar el correo electrónico y una contraseña inicial única para todos los usuarios además que mediante los perfiles se puedan seleccionar los sitios (vistas) en donde podrá

navegar el cliente cumpliendo de esta manera con los requisitos planteados en el análisis y diseño del sistema. En cuanto al uso de los perfiles se generan mapas de navegación desde los datos del contenedor de esta manera en la vista y en la programación del controlador se realizarán la llamada a estos datos dependiendo de la selección que se genere en cada perfil, es decir, al asignar al usuario un perfil cuando ingresa su correo y contraseña accede a una página personalizada para navegación.

```
List<WebCityVentas.ViewModels.MapaViewModel> mapas = credenciales.mapapermisos.ToList();  
mapas = mapas.Where(m =>m.OP_NAVEGACION == 3).ToList();
```

Se realizan dos sesiones de pruebas con los usuario antes de dar por concluido este módulo, de esta manera se genera un segundo entregable gracias a la información generada en el proceso de levantamiento de información, se cuenta con la intervención y verificación del departamento de talento humano en las pruebas de construcción avalando los perfiles de los colaboradores que interactúan con el sistema.

Finalmente se elabora el módulo de Consultas utilizando la información recopilada en los módulos iniciales en base a sentencias cruzadas en el controlador de consultas usando Linq.

```
varquery = from q inconexion.db.HORARIOEMPLEADODIAS  
selectnewAsignacionHorariosViewModel  
{  
    ID_HORARIOEMPLEADO = q.ID_HORARIOEMPLEADO,  
    ID_USUARIOEMPLEADO = q.HORARIOEMPLEADOS.ID_USUARIO,  
    ID_AGENCIA = q.HORARIOEMPLEADOS.ID_AGENCIA,  
    NOMBRES_USUARIO =  
        q.HORARIOEMPLEADOS.USUARIOS.NOMBRES_USUARIO,  
    NOMBRE_AGENCIA =  
        q.HORARIOEMPLEADOS.AGENCIAS.NOMBRE_AGENCIA,  
    NOMBRE_DIA = q.DIAS.NOMBRE_DIA,  
    DESDE_HORARIO = Convert.ToDateTime(  
        q.HORARIOEMPLEADOS.HORARIOS.DESDE_HORARIO.ToString()),  
    HASTA_HORARIO = Convert.ToDateTime(  
        q.HORARIOEMPLEADOS.HORARIOS.HASTA_HORARIO.ToString()),  
};
```

Las consultas de los horarios en el caso del perfil de empleado en la etapa inicial fue concebida de tal forma que el empleado según su ingreso de credenciales pueda consultar su jornada laboral, pero la organización a nombre del Jefe de personal solicita que también pueda visualizar el horario de sus compañeros esto por motivos de apoyo,

en caso de no contar con una conexión de internet y de haber olvidado consultar su horario en el local comercial. En el caso de la consulta del registro de la jornada laboral en la primera etapa no se puede verificar aun el resultado de la consulta ya que se debe elaborar el sistema de escritorio para que este recepte la información y pueda ser almacenada en el contenedor de datos. Después de concluir con las pruebas de funcionamiento de la aplicación web, al verificar con el director de proyecto el impacto visual y de presentación del software se decide modificar el template generando una capa de exposición amigable para los usuarios, hay que recordar que los colores iniciales fueron seleccionados por el cliente de acuerdo a los colores del logo de Vanguardia marca que CityVenta distribuye. Entre los retos que se debió sortear mediante investigación fue la forma de agregar gráficos a los botones de la grilla e insertar un evento realizado en un script para mostrar la función del botón al pasar el puntero en el objeto. A continuación el código realizado en la vista:

```

if (mapas.Where(m => m.OP_ASIGNACION == 16).Count() == 1)
command.Edit().Text("...").HtmlAttributes(new { title = "EditarUsuario" });
if (mapas.Where(m => m.OP_ASIGNACION == 18).Count() >= 1)
command.Custom("Asignarperfil").Click("AsignarPerfiles").Text(" ").HtmlAttributes
(new { title = "Asignar Perfiles a Usuario"
});

```

Ejecutada la presentación gráfica se realiza las pruebas de funcionamiento y adaptabilidad de los usuarios, gracias a estas pruebas se verifica que es necesario incluir en la pantalla de login ayuda para el usuario que olvido su contraseña y a la vez una opción de cambio de contraseña lo que hará que el sitio se personalice de manera segura.

```

public ActionResult OlvidoContraseña(USUARIOS _Usuario)
{
conexion = new Conexion();
Correos correo = new Correos();
encriptacion = new Encriptacion();
var RecuperarCorreo = from Query in conexion.db.USUARIOS
where Query.MAIL_USUARIO == _Usuario.MAIL_USUARIO
select Query;
}

```

En el Anexo 14 del documento se expone un cuadro con las clases y métodos utilizados para la ejecución del software, evidenciando una información completa del contenido, ubicación y funcionalidad.

3.3.2 Construcción aplicación de escritorio.

La construcción del sistema de escritorio se basó en la investigación previa de los métodos adecuados para el reconocimiento de rostros, autores señalan y coinciden que la mejor manera de reconocimiento es el procesamiento digital de las imágenes como se lo muestra en el capítulo I del documento, es decir, transformar una imagen a un sistema sencillo mediante conversiones matemáticas que puedan ser comparadas de manera efectiva al pasar por un filtro de parámetros comunes. La mejor forma de obtener buenos resultados es transportar la imagen a datos de 0 y 255 (blanco o negro), aunque al realizar esta operación se pierde información de la imagen pero se consigue destacar los rasgos más importantes necesarios para el reconocimiento. Al iniciar con el proyecto se realiza pruebas de las técnicas más utilizadas para el tratamiento de imagen, como la binarización, histograma, HSI, se verifica en cada uno los tiempos de conversión y procesamiento presentando promedios desde 15 hasta 95 segundos en la ejecución de algoritmo, con estas cifras sumado al tiempo de ejecución de la consulta refleja que las técnicas anteriores representa una solución poco eficiente para el registro del personal; en la búsqueda de una mejor gestión la bibliografía menciona el uso de librerías de código abierto (Open Source) desarrollados para el tratamiento de imágenes y especialmente para reconocimiento de rostros nuevamente en el capítulo I se menciona la evolución y aplicación de Emgu CV, librería que a continuación se aprovechará para obtener el resultado esperado en cuanto al reconocimiento de rostros. Se debe reiterar que a pesar de tener la librería a disposición es imprescindible investigar cómo se puede y debe aplicar cada opción para conseguir el resultado apropiado; en la organización para construir el software se usó la siguiente secuencia: Inicialización y reconocimiento de la cámara web, enfoque y recuadro para rostros, captura de imagen, conversión (escala de grises de la imagen), comparación de imágenes para obtención de reconocimiento.

Revisando los ejemplos planteados en la web sobre el uso y aplicación de la librería Emgu CV se identifica un ejemplo que expone código abierto para el reconocimiento de rostro “Face reconigtion”. A partir de este aplicativo y descargando las librerías adecuadas se adapta a las necesidades del control de asistencias. Usando la secuencia anteriormente planteada se presenta el proceso:

Inicialización y reconocimiento de cámara

Para realizar el reconocimiento e inicialización de la cámara que se encuentra incorporada como hardware al ordenador (debe estar instalado drives y realizar prueba de enlace con el software de la cámara), es necesario utilizar la clase Emgu.CV, Emgu.CV.Structure, Emgu.CV.CvEnumy generar la siguiente sentencia:

```
//INICIALIZA CÁMARA
    Frame = captura.QueryFrame().Resize(320, 240,
                                         Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
// VISUALIAZA EL CONTENIDO EN UN PICTUREBOX
    PictureBox1.Image = Frame;
```

Se obtiene como resultado:



Enfoque y recuadro para rostros

Con la instancia de las clases de la librería de Emgu.CV se utiliza los métodos de la clase para dibujar un recuadro cuando se reconoce un rostro dentro de la imagen con el siguiente código:

```
MCvAvgComp[][] rostros detectados = Frame.DetectHaarCascade(rostro,1.2,10,  
Emgu.CV.CvEnum.HAAR_DETECTION_TYPE.DO_CANNY_PRUNING,  
new Size(20, 20));
```

En donde MCvAvgComp[][] parte de la librería reconocimiento que genera una matriz para la detección de rostros dentro del Frame que creado para mostrar el video.

```
ImagenRespaldo = Frame.Copy(f.rect).Convert<Gray, byte>().Resize(100, 100,  
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);  
Frame.Draw(f.rect, new Bgr(Color.Red), 2);
```

Se respalda la imagen capturada para luego mostrarla y se dibuja un rectángulo de color rojo. Obteniendo como resultado:



Captura y conversión a escala de grises de la imagen.

Se captura la imagen y se la guarda para luego ser mostrada en un ImageBox (herramienta propia de Emgu CV), finalmente esta imagen podrá ser almacenada para el reconocimiento. Se utiliza el siguiente código:

```
imagenRespaldoDos = ImagenRespaldo.Resize(100, 100,  
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
```

```
imageBox1.Image =imagenRespaldoDos;
```

Para convertir una imagen a tonos grises para realizar comparación utilizamos el siguiente código:

```
grises = captura.QueryGrayFrame().Resize(320, 240,  
Emgu.CV.CvEnum.INTER.CV_INTER_CUBIC);
```

Como resultado obtendremos:

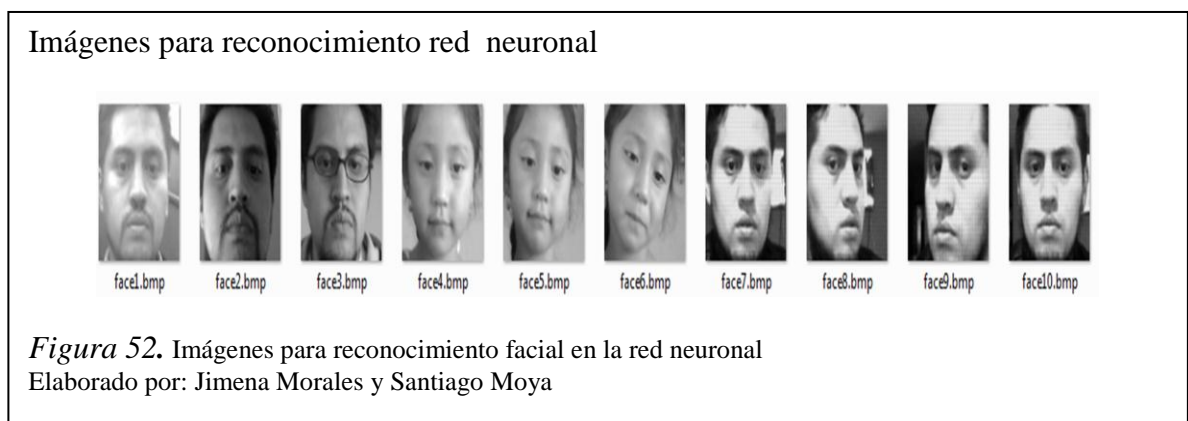


Comparar imágenes para reconocimiento.

Finalmente se debe comparar las imágenes para generar reconocimiento de rostros, Emgu CV propone que se deben tomar una cantidad considerable de muestras para que el reconocimiento sea adecuado estas imágenes se guardan en un contenedor; bajo un entrenamiento la librería propone el componente MCvAvgComp el mismo que utiliza una red neuronal básica para el entrenamiento. El código es el siguiente:

```
for (inti = 1; i<entrenamiento.ToArray().Length + 1; i++) {  
    entrenamiento.ToArray()[i - 1].Save(Application.StartupPath +  
        "/ContenedorRostros/rostro" + i + ".bmp");  
}  
foreach (MCvAvgComp f in RostroDetectado[0])  
{  
    entrenamiento = Frame.Copy(f.rect).Convert<Gray, byte>();  
    break;  
}
```

Obtenemos como resultado:



Comparando se muestra la respuesta del software:



Verificando el funcionamiento adecuado del procedimiento anterior, se aplica el enlace a la aplicación web mediante un Web Service, mismo que será utilizado como puente además de seguridad para manipular de manera segura la información del contenedor de datos, con esta información se podrá autenticar a los usuarios y generar los permisos para los que les fueron asignados dentro del sistema.

En el Anexo 15 del documento se muestra las clases y métodos propuestos para la construcción de la aplicación Windows.

3.3.3 Diccionario de datos

Para la presentación del diccionario de datos se elaboró una tabla que consta de los siguientes campos:

- Código: es el identificador único de la tabla, se compone de las siglas TBL y el número asignado a la tabla, hay que recordar que los datos deben ser del mismo tipo que el de la base de datos. Ejemplo: TBL-0001.
- Nombre: nombre asignado a la tabla en la base de datos.
- Campos: son las columnas que se tienen en cada tabla, en esta se identificará que campo es catalogado como clave principal o atributo en especial.
- Descripción: se registra observaciones importantes del desempeño de la tabla.

En la siguiente tabla se muestra el diccionario de datos.

Tabla 8.
Diccionario de Datos

CÓDIGO	NOMBRE		CAMPOS	DESCRIPCIÓN
TBL-0001	AGENCIAS	PK FK	<ul style="list-style-type: none"> • ID_AGENCIA (int, NOT NULL) • ID_UBICACION (int, NOT NULL) • NOMBRE_AGENCIA (varchar(50), NOT NULL) • DIRECCION_AGENCIA(text, NULL) • ELIMINADO_AGENCIA(bit, NOT NULL) 	Almacena información de las agencias.
TBL-0002	ASISTENCIAS	PK FK FK	<ul style="list-style-type: none"> • ID_ASISTENCIA(int, NOT NULL) • ID_USUARIO(int, NOT NULL) • ID_HORARIOEMPLEADO(int, NOT NULL) • FECHORAENTRADA_ASISTENCIA(datetime, NOT NULL). 	Almacena datos de la asistencia de un empleado.

			<ul style="list-style-type: none"> • FECHORASALIDA_ASISTENCIA(datetime, NOT NULL). • ELIMINADO_ASISTENCIA(nchar(10), NULL) 	
TBL-0003	ATRASOS	PK FK	<ul style="list-style-type: none"> • ID_ATRASOS • ID_ASISTENCIA • ENTRADA_ATRASO • SALIDA_ATRASO 	Almacena datos de atrasos de un empleado
TBL-0004	DIAS	PK	<ul style="list-style-type: none"> • ID_DIA(int, NOT NULL) • NOMBRE_DIA(varchar(50), NOT NULL) • SIGLAS_DIA(varchar(5), NOT NULL) • ELIMINADO_DIA(bit, NOT NULL) 	Almacena información de los días de la semana.
TBL-0005	HORARIOEMPLEADODIAS	PK, FK PK,FK	<ul style="list-style-type: none"> • ID_HORARIOEMPLEADO (int, NOT NULL) • ID_DIA (int, NOT NULL) 	Almacena información de los días asignados a un empleado.
TBL-0006	HORARIOEMPLEADOS	PK FK FK FK	<ul style="list-style-type: none"> • ID_HORARIOEMPLEADO(int, NOT NULL) • ID_USUARIO(int, NOT NULL) • ID_HORARIO(int, NOT NULL) • ID_AGENCIA(int, NOT NULL) • ELIMINADO_HORARIOEMPLEADO(bit, NOT NULL) 	Almacena información de los horarios de un empleado.
TBL-0007	HORARIOS	PK	<ul style="list-style-type: none"> • ID_HORARIO(int, NOT NULL) • NOMBRE_HORARIO (varchar(50), NOT NULL) • DESDE_HORARIO(datetime, NOT NULL). • HASTA_HORARIO(datetime, NOT NULL). • ELIMINADO_HORARIO(bit, NOT NULL) 	Almacena información de los horarios ingresados.
TBL-0008	MAPAS	PK	<ul style="list-style-type: none"> • ID_MAPA (int, NOT NULL) • OP_ASIGNACION(int, NOT NULL) 	Almacena información para la

			<ul style="list-style-type: none"> • ID_PADRE(int, NULL) • OP_NAVEGACION (int, NOT NULL) • NOMBRE_MAPA(varchar(50), NOT NULL) • OP_OBJETO(int, NULL) • MODELO_MAPA(varchar(50), NULL) • CONTROLADOR_MAPA(varchar(50), NULL) • DESCRIPCION_MAPA(text, NULL) • ELIMINADO_MAPA(bit, NOT NULL) 	navegación de los usuarios para la aplicación Web.
TBL-0009	PARAMETRIZACIONES	PK	<ul style="list-style-type: none"> • ID_PARAMETRIZACION(int, NOT NULL) • NOMBRE_PARAMETRIZACION(varchar(50), NOT NULL) • VALOR_PARAMETRIZACION (decimal(8,2), NOT NULL) • DESCRIPCION_PARAMETRIZACION(text, NULL) • ELIMINADO_PARAMETRIZACION(bit, NOT NULL) 	Almacena información de las condiciones para cobro de multas sanciones.
TBL-0010	PERFILES	PK	<ul style="list-style-type: none"> • ID_PERFIL(int, NOT NULL) • NOMBRE_PERFIL(varchar(50), NOT NULL) • DESCRIPCION_PERFIL(text, NULL) • ELIMINADO_PERFIL (bit, NOT NULL) 	Almacena información de los perfiles que se van a asignar a un usuario.
TBL-0011	PERFILMAPAS	PK,FK PK, FK	<ul style="list-style-type: none"> • ID_PERFIL (int, NOT NULL) • ID_MAPA (int, NOT NULL) 	Almacena información de los perfiles y en donde pueden navegar según asignación de los mapas.

TBL-0012	UBICACIONES	PK	<ul style="list-style-type: none"> • ID_UBICACION (int, NOT NULL) • ID_PADRE (int, NULL) • NOMBRE_UBICACION(varchar(50), NULL) • OPNIVEL_UBICACION (int, NOT NULL) • ELIMINADO_UBICACION(bit, NOT NULL) 	Almacena información de la ubicación de los almacenes.
TBL-0013	USUARIOPERFILES	PK,FK PK,FK	<ul style="list-style-type: none"> • ID_USUARIO (int, NOT NULL) • ID_PERFIL (int, NOT NULL) 	Almacena información de los perfiles asignados a un usuario.
TBL-0014	USUARIOS	PK	<ul style="list-style-type: none"> • ID_USUARIO (int, NOT NULL) • NOMBRES_USUARIO(varchar(100), NOT NULL) • CODIGO_USUARIO(varchar(10), NOT NULL) • CEDULA_USUARIO(varchar(10), NOT NULL) • MAIL_USUARIO(varchar(50), NULL) • CLAVE_USUARIO(varchar(50), NOT NULL) • NUEVO_USUARIO (bit, NOT NULL) • ACTIVO_USUARIO (bit, NOT NULL) • ELIMINADO_USUARIO(bit, NOT NULL) • EDAD_USUARIO (int, NULL) • ESTADO_CIVIL_USUARIO(varchar(20),NULL) • DIRECCION_USUARIO(varchar(50),NULL) • TELEFONO_USUARIO(varchar(20),NULL) • CELULAR_USUARIO(varchar(20),NULL) 	Almacena información de los usuarios del sistema.

Nota: Se muestra las tablas y cada uno de los campos que conforman el contenedor de datos.

Elaborado por: Jimena Morales y Santiago Moya

3.3.4 Construcción Web Service

El Web Service es un componente de una aplicación para ejecución remota accesible a través de la red utilizando protocolos estándar de internet. Dentro de visual studio C# por defecto están habilitados los protocolos SOAP que permite invocar métodos de servicios; XML que permite enviar y recibir mensajes desde y hasta los servicios además de los HTTP y SMTP para manejo de las capas de transporte tiene la particularidad de no poseer un interface gráfico. En el caso del software de CityVenta se utilizará los servicios web para realizar enlace y consultas a la base de datos, los servicios serán consumidos por el software de escritorio (Windows Form) para enviar y recibir datos. Usando sentencias Linq para realizar las consultas, por ejemplo:

```
//VALIDAR CREDENCIALES PARA INGRESO A REGISTRO DE IMÁGENES
```

```
public bool ValidarCredenciales(string mail, string clave)
```

```
{  
    if (clave.Trim().Length > 0)  
        var Query = from query in db.USUARIOS  
                    where query.MAIL_USUARIO == mail.Trim()  
                    && query.CLAVE_USUARIO == clave  
                    && query.ELIMINADO_USUARIO == false  
                    select query;  
    if (Query.Count() == 0) return false;  
    else return true;  
}
```

```
//CONSULTAR LOS USUARIOS POR CÉDULA
```

```
public USUARIOS GetUsuarioByCedula(string cedula)
```

```
{  
    var query = from u in db.USUARIOS  
                where u.CEDULA_USUARIO.Trim() == cedula.Trim()  
                && u.ELIMINADO_USUARIO == false  
                select u;  
    if (query.Count() >= 1)  
        usuario = query.FirstOrDefault();  
    return usuario;  
}
```

```
//CONSULTAR ASISTENCIAS POR USUARIO ESPECÍFICO
```

```
public ASISTENCIAS GetAsistenciaById(int id)
```

```
{  
    var query = from q in db.ASISTENCIAS
```

```

whereq.ID_ASISTENCIA == id
select q;
returnquery.FirstOrDefault();
}

```

Además dentro del Web Service tenemos la lógica del negocio para registrar la asistencia.

```

publicstringregistrarAsistencia(stringsNumeroCedula, int tipo, intidAgencia)
{
if (tipo == 1)// entrada
{

// 1. Verificar si la asistencia del usuario tiene relación con su horario de trabajo EN
HORA DE ENTRADA
varquery = from q in db.HORARIOEMPLEADODIAS
whereq.HORARIOEMPLEADOS.ID_USUARIO == id_usuario
&&q.ID_DIA == Convert.ToInt16(fechaActual.DayOfWeek)
&&q.HORARIOEMPLEADOS.ID_AGENCIA == idAgencia
&&q.HORARIOEMPLEADOS.HORARIOS.DESDE_HORARIO.Hours
                                     >= fechaActual.AddHours(-1).Hour
&&q.HORARIOEMPLEADOS.HORARIOS.DESDE_HORARIO.Hours
                                     <= fechaActual.AddHours(1).Hour

select q;
if (query.Count() == 0) // no está dentro del horario laboral
return "ENTRADA No puede registrar asistencia porque no está en horario de trabajo";
// verifico si existe atraso en la hora de entrada
TimeSpanHorarioEntrada= new TimeSpan();
HorarioEntrada = query.Select(m
=>m.HORARIOEMPLEADOS.HORARIOS.DESDE_HORARIO).FirstOrDefault();
HorarioEntrada= HorarioEntrada.Add(tolerancia_atrasos);
if (HorarioEntrada<hora_registro)
{
TimeSpandif= hora_registro- HorarioEntrada.Add(-tolerancia_atrasos);
atraso = new ATRASOS
{
    ENTRADA_ATRASO = dif,
    SALIDA_ATRASO = null
};
asistencia.ATRASOS.Add(atraso);
}
db.ASISTENCIAS.InsertOnSubmit(asistencia);
}
else
return "Ya tiene registrada la hora de entrada";
}
}

```

3.4 Pruebas (Cuarta fase de RUP: Transición)

3.4.1 Pruebas de la aplicación Web.

Las pruebas de software son ejecutadas para identificar y solucionar posibles problemas en el desempeño de la programación realizada, en el inicio del proceso se realizarán pruebas respaldados de las introducciones teóricas mismas que serán abordadas a continuación y que fueron realizadas en una máquina local en la etapa de construcción en pruebas realizadas con el cliente además en el transcurso del proceso se propone en el Anexo 6 del documento un cuadro de pruebas realizadas para cada uno de los módulos de la aplicación web mismas que se realizaron con cada uno de los actores cuando el software y base de datos fueron levantados al dominio Web www.CityVenta.com .

Como se describe en el párrafo introductorio para realizar las pruebas iniciales de construcción se tomarán en cuenta los siguientes aspectos teóricos:

- Prueba de caja negra: se lleva a cabo sin tener conocimiento de la estructura y codificación del sistema que se ha desarrollado.
- Prueba de caja blanca: en la que se utiliza datos del código de software para realizar análisis de su composición.

Se presentará las pruebas usando tablas la que tendrán el siguiente contenido:

Tabla para pruebas de caja Negra

Para la prueba de caja negra se utilizarán los siguientes campos:

- Código: identificador de la prueba de caja negra usando las siglas PCN adicionando el número en secuencia. Ejemplo: PCN-0001.
- Entrada: información que generará el disparo del método. Ejemplo: `encriptar(string "dato")`.
- Salida esperada: resultado de la prueba.
Ejemplo: `d/p3mXeczn8XmoluxjtV8POd3WYjyAR9mIJG66tzag`.
- Observaciones: detalles importantes en la ejecución del método.

Tabla para pruebas de caja blanca.

Para la prueba de caja blanca se utilizarán los siguientes campos.

- Código: identificador de la prueba de caja negra usando las siglas PCB adicionando el número en secuencia. Ejemplo: PCB-0001.
- Entrada: información que generará el disparo del método. Ejemplo: GetUsuarioById(3)
- Salida esperada: resultado de la prueba. Ejemplo: “Santiago Moya”
- Observaciones: detalles importantes en la ejecución del método.

En el caso de la aplicación web se han seleccionado determinados eventos (casos de uso) de acuerdo a la demanda de ejecución de los mismos, es decir se realizarán las pruebas en los de mayor demanda de ejecución. De esta lista tenemos:

- Ingreso de usuarios.
- Asignación de Jornada Laboral.
- Asignar perfiles a usuario.
- Consulta de registros.
- Registro de horarios.

3.4.1.1 Cronograma de pruebas para aplicación web

Se ha generado un cronograma para llevar a cabo las pruebas de la aplicación Web la misma que se refleja en la siguiente tabla:

Tabla 9.
Cronograma para elaborar pruebas

CÓDIGO	CASO PRUEBA	TIPO PRUEBA	EJECUCIÓN
CP-0001	Ingreso de usuario.	Caja Negra Caja Blanca	30-10-2014
CP-0002	Asignar perfiles a usuario	Caja Negra Caja Blanca	30-10-2014
CP-0003	Registro de horarios	Caja Negra Caja Blanca	02-11-2014
CP-0004	Asignación de jornada laboral.	Caja Negra Caja Blanca	02-11-2014
CP-0005	Consulta de registros	Caja Negra Caja Blanca	25-11-2014

Nota: Se muestra un cronograma para la elaboración de pruebas
Elaborado por: Jimena Morales y Santiago Moya

3.4.1.2 Condiciones para realizar pruebas

Las pruebas serán realizadas de forma local en una PC portátil marca HP Intel Core i7 2GHZ, Sistema operativo Windows 7 Home Premium Service Pack 1 con 8 GHZ de memoria RAM.

La base de datos del sistema CityVenta está alojada en la máquina local (local host).

En el Anexo 16 del documento se propone las pruebas de caja negra y caja blanca para pruebas del trabajo de la aplicación, se muestra una tabla que organiza los resultados de las pruebas. Concluyendo con las pruebas básicas se decide alojar la aplicación web contratando un dominio con el nombre de la empresa para la que está elaborado el software en el Anexo 17 del documento se muestra las pruebas del sistema con la aplicación alojada en el servidor.

3.4.2 Pruebas del Sistema Windows.

Para realizar las pruebas del sistema de escritorio, el reto principal es obtener una identificación de rostro adecuada para bajo este lineamiento efectuar el registro de asistencia de los empleados. Por este motivo se han planteado una secuencia de pruebas siguiendo la estructura teórica de los textos que refieren: “para un buen reconocimiento es necesario calibrar los elementos que intervienen en el procesamiento de la imagen: entorno, iluminación adecuada y la distancia para el reconocimiento” (Ramírez, 2006).

Como herramientas para realizar estas pruebas se han utilizado Luxmeter un software App de equipos móviles inteligente que es un aplicativo utilizado para medir la luminosidad (LUX) del ambiente.

Para iluminación del ambiente y bajo la recomendación teórica (INC., 2000)se ha utilizado luz blanca representada por 4 focos ahorradores de 20 Watts de potencia distribuidos en un área de 8 metros cuadrados (es el área de un local comercial de CityVenta) y que fueron accionados de manera progresiva. La iluminación debe estar al frente de la persona a ser registrada y detrás de la cámara web según la disposición de reconocimiento.

En cuanto al fondo (Godaddy.com, 2009) se recomienda tres tonalidades: blanca, negra y una de color intermedio que servirán como base para las pruebas de reconocimiento y bajo sus conclusiones se podrá sugerir al usuario sobre la disposición de la cámara en los locales comerciales.

3.4.2.1 Condiciones para realizar pruebas de reconocimiento

Con las tres condiciones propuestas en la sección anterior se ha realizado un cuadro de pruebas referenciadas al comportamiento al variar la distancia, el cuadro se encuentra compuesto de distancia e intensidad generando como resultado enfoque y reconocimiento del usuario. Interpretamos los símbolos de la siguiente manera:

-.-.-	Intermitente calidad	mala	-.-	Muy Intermitente	-	Intermitente
+++	Excelente		++	Muy Bueno	+	Bueno
EN	Enfoque		RE	Reconoce		

En el Anexo 18 se encuentran las pruebas realizadas de los resultados obtenidos se puede concluir:

- El fondo negro es el de más bajo nivel de reconocimiento de rostros en el ambiente normal de una habitación.
- El mejor fondo para reconocimiento por los resultados obtenidos es el blanco o un color de tonalidad clara.
- La distancia ideal para reconocimiento se encuentra entre 600 y 750 mm de separación desde la cámara hasta el usuario; pasado los 1000mm (1metro) el reconocimiento no es el adecuado para el registro de usuarios.
- La mejor iluminación para reconocimiento se genera con 3 (19LX) y 4 (27LX) focos. Al tener únicamente la iluminación de la cámara web (2LX) no se genera reconocimiento alguno.

- Bajo las conclusiones mencionadas la recomendación para los locales comerciales es: apuntar la cámara web al fondo más claro posible, usar luminarias que emiten luz blanca y en el punto de posicionamiento del usuario generar de 19 a 40 luxes y deben estar al frente del usuario y detrás de la cámara, finalmente el posicionamiento ideal del usuario debe ser entre 600mm a 800mm como referencia de la cámara.

3.4.2.2 Cronograma de pruebas para aplicación de escritorio

Se ha generado un cronograma para llevar a cabo las pruebas de la aplicación de escritorio para el registro de asistencia la misma que se refleja en la siguiente tabla:

Tabla 10.

Cronograma de pruebas funcionamiento aplicación de escritorio

CÓDIGO	CASO PRUEBA	TIPO PRUEBA	EJECUCIÓN
CP-0001	Registro de usuario	Caja Negra Caja Blanca	15/01/2015
CP-0002	Registro de asistencia de entrada	Caja Negra Caja Blanca	23/01/2015
CP-0003	Registro de asistencia de salida	Caja Negra Caja Blanca	23/01/2015

Nota: Cronograma elaborada para realizar pruebas de funcionamiento de la aplicación de escritorio.
Elaborado por: Jimena Morales y Santiago Moya

En el Anexo 18 del documento se muestra la ejecución de las pruebas propuestas en el cronograma y los resultados mostrados en el orden de ejecución.

3.4.3 Pruebas de la base de datos.

Las pruebas de la base de datos se las representa en la siguiente tabla que consta de los campos:

- Código: identificador de la prueba se compone de las siglas PDB y el número de la prueba realizada. Ejemplo: PDB-0001.
- Consulta: el código SQL a ejecutarse.
- Utilidad: fin por el que se realizó la consulta SQL.
- Resultado: la respuesta esperada de la consulta SQL.
- Tiempo: respuesta en segundos de la consulta SQL.
- Observaciones: comentarios importantes de la consulta.

Tabla 11.
Pruebas de base de datos

CÓDIGO	CONSULTA	UTILIDAD	RESULTADO	TIEMPO (seg.)	OBSERVACIÓN
PDB-0001	select NOMBRES_USUARIO from USUARIOS where MAIL_USUARIO='smoya@cv.com' and CLAVE_USUARIO='d/p3mXeczn8Xmoluxjt V8POd3WYjyAR9mIJG66tzag='	Consulta si el usuario está registrado	Datos esperados	0.20	Resultado OK
PDB-0002	select ID_PERFIL,NOMBRE_PERFIL, DESCRIPCION_PERFIL FROM PERFILES WHERE ELIMINADO_PERFIL=0	Consulta de los datos de nombre y descripción de perfiles ingresados	Datos esperados	0.20	Resultado OK
PDB-0003	select ID_USUARIO FROM USUARIOS WHERE NOMBRES_USUARIO='Santiago	Consulta del ID de usuario sabiendo el	Datos esperados	0.20	Resultado OK

	Moya'	nombre del usuario			
PDB-0004	select * FROM PERFILES WHERE ID_PERFIL=1	Consulta de un perfil sabiendo su ID	Datos esperados	0.20	Resultado OK
PDB-0005	var Query = from query in conexion.db.USUARIOS wherequery.MAIL_USUARIO == "smoya@cv.com" && query.CLAVE_USUARIO == "ccc123" && query.ELIMINADO_USUARIO == false	Consulta el usuario se encuentra registrado mientras no haya sido eliminado el registro.	Datos esperados	0.20	Resultados OK
PDB-0006	var query = from q in conexion.db.PERFILES where q.ID_PERFIL == "2" select q;	Consulta perfil específico.	Datos esperados	0.20	Resultados OK
PDB-0007	varvalidacionInicio = fromhor in conexion.db.HORARIOSwherehor.ID_HORARIO == horemp.ID_HORARIO&& horemp.ID_HORARIOEMPLEADO== hed.ID_HORARIOEMPLEADO &&horario.DESDE_HORARIO>= hor.DESDE_HORARIO&&horario.DESDE_HORARIO<=hor.HASTA_HORARIO&& horemp.ELIMINADO_HORARIOEMPLEADO == false && horemp.ID_USUARIO==idUsuario&&hed.ID_DIA==idDia selecthoremp;	Consulta si un horario de entrada fue asignado a un empleado	Datos esperados	0.20	Resultados OK

Nota: Prueba de sentencia SQL para base de datos.
Elaborado por: Jimena Morales y Santiago Moya

CONCLUSIONES

- A través del uso de herramientas adecuadas, tanto para desarrollo web como para reconocimiento de rostros, después de haber pasado por un proceso de investigación, pruebas y demás procedimientos necesarios, se ha logrado cumplir el objetivo general del proyecto: generar un producto de software con el cual el personal de CityVenta pueda registrar su asistencia a la jornada laboral y que esta información pueda ser tratada en tiempo real por quienes están autorizados para realizar este trabajo.
- En la búsqueda de información sobre “sistemas informáticos híbridos”, se determina que no hay datos, ensayos o libros relevantes que sustenten este concepto, pero al tomar en cuenta las características teóricas de la palabra híbrido (elementos de distinta naturaleza que se asocian para entregar un producto final) se considera como un término apropiado para definir el desempeño del software que se está entregando.
- Al presentarse la necesidad de automatizar un proceso es indispensable que los desarrolladores se involucren en las tareas o actividades que se llevan a cabo para crear procedimientos con cada uno de los actores, de esta manera se consolidará de forma efectiva aquellos que se consideran los principales requerimientos del cliente. No siempre la visión de los desarrolladores del proyecto es la adecuada y puede estar sesgada a su criterio, de esta manera se evitará caer en falsas interpretaciones que limiten el proceso y dificulten los tiempos de entrega del producto.
- La selección de la metodología para el diseño de software es fundamental para alcanzar el objetivo de un proyecto, con RUP se ha podido documentar de manera adecuada cada proceso del proyecto y tomar decisiones en cuanto a la elaboración del producto. Al ser iterativo permite canalizar el proceso para cumplir el objetivo con alta calidad esto gracias a la mejora continua que se da en el transcurso del ciclo

de vida para elaborar el software. A pesar de tener una proyección clara en la etapa de análisis y diseño del producto a elaborarse, en la etapa de construcción existen cambios solicitados por el cliente que son atendidos de manera progresiva, esto gracias a la modulación y apertura de entregables del producto que hacen que se tomen correcciones en la marcha con la participación y aprobación activa del usuario.

- Cada iteración que propone la metodología, regula, refina y genera calidad, en este caso reduce notablemente gastos innecesarios de recursos de no conformidad; retira o incorpora datos en el proceso y lleva un control estricto frente a eventuales riesgos que son mitigados desde la concepción del proyecto.
- RUP tiene estricta rigurosidad en la documentación del proyecto, los autores recomiendan que se la debería utilizar en proyectos grandes (más de 3 desarrolladores) sin limitar su uso en proyectos menores, por este motivo realizar la documentación en este tipo de proyecto de tesis (considerado pequeño), genera la posibilidad de aplicar los conocimientos adquiridos en la carrera además de utilizar la experiencia alcanzada en nuevos proyectos.
- Al utilizar diagramas en el diseño del software hace que los involucrados en el desarrollo identifiquen claramente las necesidades del cliente y se ejecuten prácticas adecuadas al momento de programar; estos diagramas representan un mapa (arquitectura) de respaldo que permite enfocarse en las necesidades del producto final. En esta etapa se define con claridad los módulos o entregables del producto y se incorpora una visión ordenada de lo que se requiere ejecutar.
- Se ratifica que la programación y trabajo del desarrollador es transparente para el usuario final que su principal necesidad es que el resultado se cumpla, por esta razón el programador utiliza sus conocimientos para ejecutar de manera eficaz los requisitos funcionales y no funcionales para satisfacer los requerimientos del cliente.

- Los parámetros y lenguaje en el que se encuentra escrito los requerimientos del cliente debe ser revisados y entendidos por los participantes en el desarrollo, además de ser almacenados en los documentos de apoyo propuestos para el levantamiento de información, esto hará que conjuntamente con los diagramas que propone la metodología y arquitectura se edifique correctamente la estructura del software para cumplir los requerimientos.
- Las pruebas realizadas de manera parcial, es decir, después de la construcción de cada módulo hacen que las probabilidades de falla en cuanto a la concepción total del producto sean mínimas y el cliente pueda acoplarse sin problemas cuando se realiza la entrega total del producto. Involucrar al cliente en todos los procesos de construcción de software hacen que el desarrollador tome experiencia en cuanto al reconocimiento de los perfiles de los actores y pueda anticipar sus acciones en beneficio del cliente y de sí mismo además de no caer en reproceso innecesarios generados por la falta de comunicación entre las partes.
- Al comparar rostros es necesario definir el fondo al que se encuentra expuesto la imagen así como la luminosidad del ambiente, resolución del receptor y distancia de enfoque, al controlar los aspectos antes mencionados el efecto del procesamiento será eficiente y los resultados los esperados.
- La mejor forma de generar cultura en el ser humano es contrarrestar sus paradigmas usando los recursos teóricos y especialmente prácticos que evidencien que el cambio al usar tecnología en sus procesos ayudan a mejorar y sobre todo pueden generar ganancias y/o ahorros significativos.
- Se debe escoger las herramientas adecuadas para diagramar el modelo de negocios, la Web argumenta y aconseja varios tipos, lo importante es entender el funcionamiento y aplicar la herramienta de forma adecuada al modelo teórico.

- Es importante modelar los procesos a programar conociendo los requerimientos del sistema a desarrollar, con esto la construcción del programa será secuencial y ordenada.
- Al analizar un proceso que se desea automatizar no se debe obviar ninguna parte del manejo manual que se hace para efectuarlo, con esto se podrá tomar el núcleo de este proceso y entregar un producto adecuado que cumpla con los requisitos planteados.

RECOMENDACIONES

- Para la aplicación de escritorio se debe cumplir con los recursos mínimos de hardware y software expuestos en la etapa de pruebas del documento (procesador 2.5 Gb) para que la aplicación pueda ejecutarse sin contratiempos.
- Es necesario realizar el mantenimiento de la cámara ligada al reconocimiento de usuarios; cada 6 meses se debe verificar el enfoque, emisión de luz, además de verificar el procesamiento que efectúa; en caso de que la cámara se encuentre con un retardo notorio del procesamiento es necesario reinstalar el controlador de la misma, este control periódico ayudará para un correcto funcionamiento de la aplicación.
- El uso de este tipo de sistemas informáticos deberían ser adoptados por las empresas que tengan visión de automatización de sus procesos, con este software se tendrá un control personalizado de la asistencia con datos reales que evite pérdidas tanto para empresa como empleados.
- Al ser un sistema escalable es posible adaptar nuevos módulos que ayuden con el proceso administrativo de la empresa, como en el área contable para el manejo de nómina, además en procedimientos de seguridad con los controles de acceso.
- Sería conveniente que una unidad académica interna de esta u otra universidad considere la posibilidad de definir y describir el término “Sistemas informáticos híbridos” de modo que la comunidad académica y profesional mundial relacionada con Tecnologías de la Información pueda usar el mencionado término de una forma más cotidiana y sin confusiones.
- Se debe tener un control adecuado del ambiente (iluminación y fondo) en los que se utilizarán los sistemas de identificación ya que un mal manejo de los mismos causan problemas en el reconocimiento y mal rendimiento de la aplicación.

LISTA DE REFERENCIAS

- 1471-2000, I. (12 de 10 de 2009). *blogspot.com*. Recuperado el 05 de 11 de 2014, de [blogspot.com](http://arquitecturasoftware.blogspot.com/): <http://arquitecturasoftware.blogspot.com/>
- telerik.com*. (31 de 03 de 2001). Recuperado el 28 de 01 de 2014, de [telerik.com](http://docs.telerik.com/kendo-ui/introduction): <http://docs.telerik.com/kendo-ui/introduction>
- microsoft*. (05 de 02 de 2003). Recuperado el 19 de 11 de 2013, de [microsoft.com](http://technet.microsoft.com/es-es/library/ms166352(v=sql.90).aspx): [http://technet.microsoft.com/es-es/library/ms166352\(v=sql.90\).aspx](http://technet.microsoft.com/es-es/library/ms166352(v=sql.90).aspx)
- microsoft*. (09 de 10 de 2012). Recuperado el 10 de 03 de 2014, de [microsoft.com](http://msdn.microsoft.com/es-es/library/gg606533(v=vs.100).aspx): [http://msdn.microsoft.com/es-es/library/gg606533\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/gg606533(v=vs.100).aspx)
- julitogtu.com*. (21 de 07 de 2013). Recuperado el 25 de 06 de 2014, de [julitogtu.com](http://www.julitogtu.com/2013/05/14/asp-net-mvc-utilizando-el-grid-mvc-paso-a-paso/): <http://www.julitogtu.com/2013/05/14/asp-net-mvc-utilizando-el-grid-mvc-paso-a-paso/>
- Abelló, A. (2006). *Diseño y Administrador de bases de datos*. Barcelona: Edicions UPC.
- Adams, J. (2013). *Learning Kendo UI Web development*. USSA: Packt Publishing.
- Americas-Puebla, U. d. (01 de 01 de 1990). *udlap.mx*. Recuperado el 25 de 09 de 2014, de [udlap.mx](http://catarina.udlap.mx/u_dl_a/tales/documentos/meie/osorio_s_a/capitulo2.pdf) : http://catarina.udlap.mx/u_dl_a/tales/documentos/meie/osorio_s_a/capitulo2.pdf
- Andalucía, U. I. (15 de 10 de 2010). *unia.es*. Recuperado el 06 de 06 de 2014, de [unia.es](http://ocw.unia.es/ciencias-tecnologicas/tecnologia-del-ocio/materiales-basicos-folder/html/B2_U3/historia_de_opencv.html/skinless_view): http://ocw.unia.es/ciencias-tecnologicas/tecnologia-del-ocio/materiales-basicos-folder/html/B2_U3/historia_de_opencv.html/skinless_view
- arsys internet, S. D. (20 de 02 de 2003). *http://www.nicline.com*. Recuperado el 29 de 12 de 2014, de <http://www.nicline.com>: http://www.eqmed.com/como-automatizar-procesos-de-negocio/#.VQJQP06G_4A
- Artigas, J. M. (2002). *Tecnología del color*. Valencia: Maite Simón.
- aulaClic, C. d. (2013). *Curso de SQL Server*. Aula Clic S.L.
- Aumaille, B. (2002). *J2EE Desarrollo de aplicaciones Web*. Barcelona: ENI.
- Batiuk, A. (17 de 11 de 2012). *FINSLAB.COM*. Recuperado el 25 de 10 de 2014, de [FINSLAB.COM](http://finslab.com/dineroblog/modelos-de-negocio/que-tres-beneficios-traeria-rup-a-una-organizacion.php): <http://finslab.com/dineroblog/modelos-de-negocio/que-tres-beneficios-traeria-rup-a-una-organizacion.php>
- Berzal, F. (2003). *Desarrollo profesional de aplicaciones Web con ASP.Net*. Ikor Consulting.

- Betaniatech.com. (22 de 07 de 2014). *http://www.godaddy.com*. Recuperado el 15 de 10 de 2014, de *http://www.godaddy.com*:
http://betaniatech.com/SmallProjects/core.base_rup/guidances/concepts/tailoring_a_process_for_a_small_project_21481E2E.html
- Blogspot. (31 de 06 de 2000). *blogspot.com*. Recuperado el 25 de 11 de 2014, de *blogspot.com*: *http://basketenciernes.blogspot.com/2012/04/diagrama-de-navegacion-o-estructura-de.html*
- Bolivar, E. U. (1999). *Aplicaciones de las redes de Neuronas en supervisión diagnosis y control de procesos*. Caracas: Equinoccio.
- Bradski, G. (2008). *Open CV Computer Visión with OpenCv Library*. USSA: O'REILLY.
- BRANDON GRAY INTERNET SERVICES, I. D. (07 de 03 de 2012). *www.markmonitor.com*. Recuperado el 25 de 10 de 2013, de *www.markmonitor.com*:
http://metodologiasdelainvestigacion.wordpress.com/2012/03/07/introduccion-general-a-la-metodologia-de-la-investigacion/
- Bustio, J. M. (1994). *Sistemas de identificación y control automáticos*. Barcelona: Productica Marcombo S.A.
- Calero, C. (2010). *Calidad del producto y Proceso del Software*. Paracuellos de Jarama, Madrid: Ra-Ma.
- Catalunya, C. d. (25 de 02 de 2000). *www.maia.ub.es*. Recuperado el 24 de 03 de 2015, de *www.maia.ub.es*: *http://www.maia.ub.es/~sergio/linked/julian08.pdf*
- Chadwick, J. (2011). *Programming Razor*. USA: O'REALLY.
- Chávez, J. L. (2010). *Tratamiento Digital de Imágenes Multiespectrales*. México D.F.: UNAM.
- Chile, U. d. (22 de 01 de 2014). *uchile.cl*. Recuperado el 11 de 11 de 2014, de *uchile.cl*:
http://users.dcc.uchile.cl/~psalinas/uml/modelo.html
- Debrauwer, L. (2013). *UML2 Iniciación, ejemplos y ejercicios corregidos*. Barcelona: Ediciones ENI.
- dimages. (25 de 03 de 2006). *dimages.es*. Recuperado el 30 de 10 de 2013, de *dimages.es*: *http://www.dimages.es/Tutorial%20A.I/segmentacion/binaria.htm*

- Dollon, J. (2012). *Visual Studio 2010 Desarrollo aplicaciones Web con C#*. Barcelona: ENI.
- Domains, T. (08 de 02 de 2008). <http://www.tucowsdomains.com>. Recuperado el 10 de 11 de 2014, de <http://www.tucowsdomains.com>: <http://oa.upm.es/215/1/10200404.pdf>
- Elizondo, J. J. (2005). *Fundamento de procesamiento de imágenes*: Universidad Autónoma de Baja California.
- Elizondo, J. J. (2005). *Fundamentos de procesamiento de Imágenes*: Universidad Atonoma de Baja California.
- Enom. (27 de 02 de 2008). <http://www.enom.com>. Recuperado el 25 de 01 de 2014, de <http://www.enom.com>: <http://blog.unreal4u.com/2011/03/sobre-convenciones-y-notaciones-hungara-camelcase-etc/>
- Falgeras, B. C. (2003). *Ingeniería del software*. Aragón-Barcelona: Editorial Uoc.
- Falgueras, B. C. (2013). *Ingeniería del Software*. Aragón, Barcelona: Editorial UOC.
- Fernando, A. A. (2005). *INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE MODELOS DE DESARROLLO DE SOFTWARE*. Las Rosas- Madrid : Delta Publicaciones Universitarias.
- Freeman, A. (2010). *Pro LINQ Language Integrated Query in C#2010*. USSA: Apress.
- Freund, J. (2013). *BPMN 2.0 Manual de referencia y guía práctica*. Santiago de Chile: Edición Hispana.
- Gabillaud, J. (2013). *SQL Server 2012 Diseño y creación de base de datos*. Baelona: ENI.
- GODADDY.COM, L. (21 de 11 de 2009). <http://registrar.godaddy.com>. Recuperado el 24 de 01 de 2015, de <http://registrar.godaddy.com>: <http://www.blogdelfotografo.com/fotografia-retrato/>
- Guadalajara, U. d. (31 de 12 de 1989). udg.mx. Recuperado el 25 de 12 de 2014, de udg.mx: http://defis.cucea.udg.mx/sites/default/files/EstudioCaso_Davila_Galaviz_MariaCecilia.pdf
- Gutierrez, E. (2009). *Java Script conceptos básicos y Avanzados*. Barcelona: Ediciones ENI.

- Hitpass, B. (2012). *BPMN Fundamentos y conceptos de implementación*. Santiago de Chile: BHH Ltda.
- Holdings, I. (09 de 07 de 2009). <http://www.directnic.com>. Recuperado el 15 de 01 de 2015, de <http://www.directnic.com>: <http://www.docirs.com/uml.htm>
- IEEE. (01 de 11 de 2011). IEEE 1471-2000 Arquitectura de software. Estados Unidos.
- Ilce. (03 de 11 de 2008). ilce.edu.mx. Recuperado el 25 de 10 de 2013, de ilce.edu.mx: <https://www.library.cornell.edu/preservation/tutorial-spanish/technical/technicalB-04.html>
- Inc, G. (31 de 07 de 2000). blogspot.com. Recuperado el 28 de 10 de 2014, de blogspot.com: <http://arquitecturasoftware.blogspot.com/>
- INC., M. (03 de 03 de 2000). <http://www.markmonitor.com>. Recuperado el 23 de 01 de 2015, de <http://www.markmonitor.com>: <https://fotografiaperfecta.wordpress.com/2010/10/07/tutorial-iluminacion-para-el-retrato-y-colocacion-de-las-luces/>
- Innova. (2001). *Internet Java Script*. Málaga: Innovación y Cualificación S.L.
- Janover, J. (26 de 11 de 2008). definicion.de. Recuperado el 16 de 11 de 2013, de <http://definicion.de>: <http://definicion.de/ingenieria-de-software/>
- konradlorenz. (06 de 12 de 2010). www.cointernet.com.co. Recuperado el 13 de 12 de 2014, de www.cointernet.com.co: http://www.konradlorenz.edu.co/images/stories/suma_digital_industrial/bpm.pdf
- Kruchten, P. (2004). *The Rational Unified Process an introduction*. Boston: Pearson Education, Inc.
- López, E. (2003). *Especificación de sistemas de software en UML*. Barcelona: Politext.
- López, O. P. (2000). *Gestión de Base de datos*. Valencia: Reproval, S.L.
- Mackenzie, D. (2003). *Aprendiendo Visual Basic .Net*. Ciudad de México: Pearson Educación.
- Macrkmonitor. (21 de 05 de 1991). <http://www.markmonitor.com>. Recuperado el 23 de 11 de 2014, de <http://www.markmonitor.com>: <https://msdn.microsoft.com/es-es/library/dd409377.aspx>
- Markmonitor. (21 de 05 de 1991). <http://www.markmonitor.com>. Recuperado el 25 de 11 de 2014, de <http://www.markmonitor.com>: <https://msdn.microsoft.com/es-es/library/dd409360.aspx>

- Martin, F. (1999). *UML gota a gota (versión español de UML Distilled)*. Ciudad de México: Adison Wesley.
- mcgraw-hill. (22 de Abril de 1999). *hr.cl*. Recuperado el 03 de 11 de 2013, de <http://www.hr.cl/febrero.htm#¿Cómo se calcula el índice de ausentismo laboral?>
- microsoft.com. (05 de 02 de 2003). *MarkMonitor, Inc.* Recuperado el 16 de 11 de 2013, de MarkMonitor, Inc.: [http://msdn.microsoft.com/es-es/library/ee671793\(v=vs.100\).aspx](http://msdn.microsoft.com/es-es/library/ee671793(v=vs.100).aspx)
- Molina, H. G. (2006). *Avances en informática y Sistemas Computacionales Tomo I*. Tabasco: Conais.
- Monterrey, D. d. (20 de 10 de 2007). *ElMetododeProyectoscomoTecnicaDidactica*. Recuperado el 20 de 10 de 2013, de ElMetododeProyectoscomoTecnicaDidactica.: http://www.rsu.uninter.edu.mx/doc/marco_conceptual/ElMetododeProyectoscomoTecnicaDidactica.pdf
- msdnmicrosoft. (11 de 01 de 2009). Recuperado el 03 de 11 de 2014, de [https://msdn.microsoft.com/es-es/library/cc483497\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/cc483497(v=vs.71).aspx)
- Nacional, I. P. (30 de 04 de 1995). *ipn.mx*. Recuperado el 25 de 11 de 2014, de ipn.mx: http://tesis.bnct.ipn.mx/dspace/bitstream/123456789/6749/1/RECONOCIMROS_TROS.pdf
- Nacional, I. P. (30 de 04 de 1995). *ipn.mx*. Recuperado el 21 de 01 de 2015, de ipn.mx: <http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/8628/Archivo%20que%20incluye%20portada,%20%C3%ADndice%20y%20texto.pdf?sequence=1>
- Paul, K. (2009). *LINQ UNLEASHED for C#*. USSA: SAMS.
- Paz, J. R. (2010). *Pro Telerik ASP.NET and Silverlight Controls*. USA: Apress.
- Prisma, E. (19 de 02 de 2000). *ELPRISMA.COM*. Recuperado el 18 de 11 de 20013, de http://www.elprisma.com/apuntes/administracion_de_empresas/controldepersona1/default3.asp
- Ramírez, D. L. (2006). Conceptos básicos del procesamiento digital de imagenes. *Escuela de Física Universidad Nacional de Colombia*, 51 -60.
- Ramirez, F. J. (2007). *Aprenda practicando Visual Basic 2005*. Ciudad de México: Pearson Educación.

- Rodriguez, O. P. (2007). *Iniciación a la programación en C# un enfoque práctico*. Madrid: Delta.
- s.a.c., u. p. (23 de 02 de 2012). *upn.edu.pe*. Recuperado el 15 de 10 de 2014, de *upn.edu.pe*: <https://refi.upn.edu.pe/index.php/refi/article/view/8/24>
- S.L., A. N. (05 de 01 de 2013). *AVANZE.NET*. Recuperado el 15 de 11 de 2013, de *AVANZE.NET*: <http://www.avanze.net/>
- Sala, J. J. (2003). *Introducción a la programación teoría y práctica*. Alicante: ECU.
- Santos, A. C. (27 de 03 de 2014). *unam.com*. Recuperado el 11 de 01 de 2015, de *unam.com*: <http://www.mcc.unam.mx/~cursos/Objetos/Cap8/cap8.html>
- Sevilla, U. d. (21 de 11 de 2007). *us.es*. Recuperado el 05 de 11 de 2013, de *us.es*: <http://alojamientos.us.es/gtocom/pid/tema1-2.pdf>
- Shi, S. (2013). *Emgu CV Essentials*. Birmingham: Packt Publishing Ltd.
- slideshare.net. (04 de 04 de 2006). *slideshare.net*. Recuperado el 18 de 11 de 2013, de <http://www.slideshare.net/Tonymx/introduccion-a-linq>
- Sonora, U. d. (01 de 01 de 1994). *INFO.COM*. Recuperado el 25 de 03 de 2014, de *INFO.COM*: http://info.fisica.uson.mx/arnulfo.castellanos/archivos_html/quesonredneu.htm
- SRL, D. (18 de 08 de 2006). <http://www.dattatec.com>. Recuperado el 30 de 01 de 2014, de <http://www.dattatec.com>: <http://www.ba-technology.net/index.php/metodologia/91?start=1>
- trabajo, O. i. (2005). *Horas de trabajo De lo Flexible a lo fijo?* Ginebra: OIT.
- wordpress.com. (03 de 03 de 2000). *wordpress.com*. Recuperado el 20 de 12 de 2014, de *wordpress.com*: <https://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>
- Yeta, Á. C. (2012). *Diseño y programación de base de datos*. Madrid: Visión y Libros.
- Zanuy, M. F. (2000). *Tratamiento digital de voz e imagen*. USSA: Marcombo.

ANEXOS

Anexo 1. Elementos de Bpmn

1. Objetos de flujo

Son los elementos principales dentro de BPMN en los que se encuentra eventos, actividades y compuertas (control de flujo).

Eventos: están representados por un círculo y describen algo que sucede o puede suceder durante el proceso, los eventos se clasifican también como capturado o lanzado.



Imagen 1.1 Eventos

- **Evento inicial:** es representado por un círculo con una línea muy delgada y dentro del círculo relleno con color verde, este evento permite capturar.
- **Evento intermedio:** es representado por un círculo de línea doble simple y dentro del círculo relleno con color naranja, este permite saber que sucede entre el evento inicial y el final, este evento permite capturar o lanzar.
- **Evento final:** es representado con un círculo de línea gruesa y dentro del círculo relleno con color rojo, este evento permite lanzar.

Actividades: están representadas por un rectángulo con sus vértices redondeados, describe el tipo de trabajo que será realizado.

- **Tarea:** representa una sola unidad de trabajo que no se puede dividir a un mayor nivel de detalle de procesos de negocio.



Imagen 1.2 Tarea

- **Subproceso:** se usa para ocultar o mostrar otros niveles de detalle de negocio.

- Cuando se minimiza un subproceso se representa con un signo más ubicado en la línea inferior del rectángulo

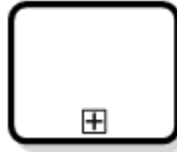


Imagen 1.3 Subproceso

- Cuando se expande permite mostrar todos los objetos de flujo, de conexión y artefactos.

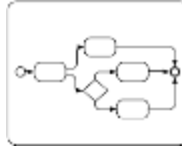


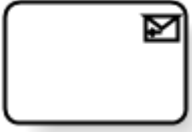



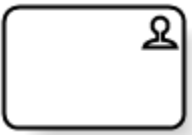
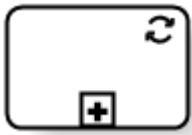
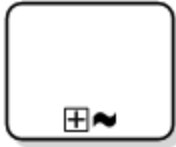

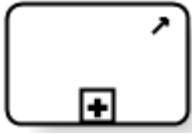
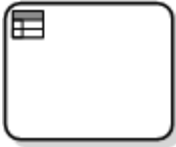



Imagen 1.4 Subproceso expandido

Tabla de marcadores vs tipo de tarea y marcadores vs tipos de subproceso

Marcador Tarea	Marcador Subproceso	Tipos de Tarea	Tipos de Subproceso
Los marcadores especifican el comportamiento particular de las actividades durante su ejecución.		Los tipos especifican la naturaleza de la tarea que se desea llevar a cabo.	
 Ciclo Estándar	 Ciclo Estándar	 Tarea	 Subproceso
Marcador Tarea	Marcador Subproceso	Tipos de Tarea	Tipos de Subproceso
 Tarea	 Subproceso	 Tarea	 Subproceso

Instancias Múltiples en Paralelo	Instancias Múltiples en Paralelo	Envío	Transacción
			
Instancias Múltiples en Secuencia	Instancias Múltiples en Secuencia	Recepción	Subproceso de Evento
			
Compensación	Compensación	Tarea de Usuario	Reusable
			
	Ad Hoc	Tarea Manual	Subproceso de Referencia
Marcador Tarea	Marcador Subproceso	Tipos de Tarea	Tipos de Subproceso
			
		Regla de Negocio	
			

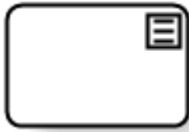
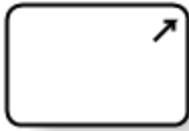



		Invocación de Servicio	
		 Ejecución de Script	
		 Actividad de Referencia	

Tabla 1.1 Cuadro de tipo de tareas y subprocesos (Bayard, 2011)

- **Transacciones:** es una forma de subproceso en la cual todas las actividades contenidas deben ser tratadas como un todo. Las transacciones se diferencian de los subprocesos expandidos por estar rodeando por un borde de doble línea.

Compuertas: estos se encuentran representados por un rombo que determinan si un proceso se separa o se conecta de acuerdo a las condiciones preestablecidas

Tipo de decisión	Definición	Notación	
Decisión Exclusiva	En un punto de bifurcación, selecciona exactamente un flujo de secuencia de entre las alternativas existentes. En un punto de convergencia, la compuerta espera a que un flujo incidente complete para activar el flujo saliente.		
Exclusiva Basada en Eventos (generadora de instancias)	En la ocurrencia de uno de los eventos subsecuentes se crea una nueva instancia del proceso.		






Decisión Basada en Evento	Esta compuerta siempre será seguida por eventos o tareas de recepción, y sólo activará un flujo saliente dependiendo del evento que ocurra en primer lugar.	
Decisión Inclusiva	En un punto de bifurcación, al menos un flujo es activado. En un punto de convergencia, espera a todos los flujos que fueron activados para activar al saliente.	
Decisión Compleja	Comportamiento complejo de convergencia/bifurcación no capturado por el resto de compuertas.	
Decisión Paralela	En un punto de bifurcación, todos los caminos salientes serán activados simultáneamente. En un punto de convergencia, la compuerta espera a que todos los flujos incidentes completen antes de activar el flujo saliente.	
Paralela Basada en Eventos (generadora de instancias)	En la ocurrencia de todos los eventos subsecuentes se crea una nueva instancia del proceso.	

Tabla 1.2 Cuadro de tipo de compuertas (Bayard, 2011)

2. Objetos de conexión

Permiten vincular a cada uno de los objetos de flujos con los demás elementos dentro de las distintas categorías. Estos objetos de conexión pueden ser de tres tipos:

Flujos de Secuencia: su función es el mostrar en qué orden las actividades serán realizadas. Están representados por una línea sólida en cuyo extremo sostiene una flecha



Imagen 2.1 Línea de Secuencia

Flujos de Mensaje: llamados también de datos, representa la iteración entre varios procesos, muestra señales o mensajes y no flujos de control, no especifica el orden de los mensajes. Están representados por una línea discontinua en cuyo inicio presenta un círculo vacío al inicio y la punta de una flecha al final

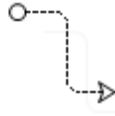


Imagen 2.2 Línea de mensaje

Asociaciones: se utiliza para asociar Artefactos o textos a un Objeto de Flujo, para asociar tareas de compensación. Están representados por medio de una línea punteada.



Imagen 2.3 Asociaciones

Asociaciones de datos: se utiliza para representar el movimiento de objetos de datos de entrada y salida de las actividades. Están representados por medio de una línea punteada y la punta de una flecha al final.

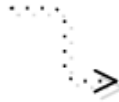


Imagen 2.4 Asociación de datos

3. Canales

Son elementos utilizados para organizar las actividades de flujos en diferentes categorías que representan áreas funcionales, roles o responsabilidades.

Pools: para colocar el nombre al pool se lo puede usar la del proceso o del participante, actúa como contenedor de un proceso y representa a un role, entidad o participante.

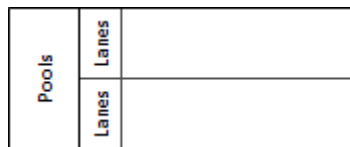


Imagen 3.1 Pools

Lanes: sirve para hacer subdivisiones del pool, representan los diferentes participantes al interior de una organización.



Imagen 3.2 Lanes

4. Artefactos

Sirven para permitir adicionar información necesaria para la legitimidad del diagrama o modelo de procesos desarrollado. Estos pueden ser:

Objetos de datos: están encargados de señalar al lector que información es requerida en una actividad, no afectan los flujos de secuencia o de mensajes. Está representador por una hoja.



Imagen 4.1 Objetos de datos

Grupos: están encargados de agrupar una serie de actividades, no afecta la secuencia flujo. Está representado por un rectángulo de esquinas redondeadas y líneas discontinuas.



Imagen 4.2 Grupos

Anotaciones: están encargados de dar información adicional necesaria sobre el proceso.



Imagen 4.3 Anotaciones

Anexo 2. Ejemplos básicos de programación del marco teórico

Ejemplo partes de un programa en C#

```
class Tesis {  
    //Ejemplo de variable  
    public string a= "salesiana";  
    public string nom_alum;  
    public double calif_alum;  
    //Ejemplo de método  
    public string ingenieria(string universidad) {  
        if (universidad= "salesiana") {return "sistemas";}   
        else { return "No existe"; }  
    }  
    //Ejemplo de objeto  
    Tesis (string alumno, int calificacion) {  
        nom_alum= alumno;  
        calif_alum= (calificacion*50);  
    }  
    public double resultado(){  
        double puntaje;  
        puntaje= (calif_alum/100);  
    }  
    //Cuerpo del programa  
    public void Main(){  
        string nombre="Santiago";  
        double puntaje= 20;  
        Tesis.ingenieria(a);  
        Tesis sistemas= new Tesis(nombre, puntaje);  
    }  
}
```

Ejemplo de la sintaxis de razor.

```
@{var texto= "EJEMPLO RAZOR PARA TESIS";}  
<html>  
<body>  
<h1> UNIVERSIDAD POLITÉCNICA SALESIANA</h1>
```

```

<h2>@texto</h2>
        Fecha: @ DateTime.Now
</body>
</html>

```

Ejemplo de Java Script

```

<HTML>
<HEAD>
<TITLE>EJEMPLO DE JAVA SCRIP PARA MARCO TEORICO.htm</TITLE>
<SCRIPT LENGUAJE="JavaScript">
alert("Tesis para obtener título de Ingeniero en Sistemas")
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LENGUAJE="JavaScript">
var nombre;
nombre=prompt(" Cuál es tu Nombre");
alert("BIENVENIDO"+ nombre);
</SCRIPT>
</BODY>
</HTML>

```

Ejemplo de código para uso de la Grilla de Kendo

El código que referencia y controla a la grilla lo podemos revisar a continuación:

```

<div>
//PROGRAMACIÓN RAZOR LLAMADA A LA GRILLA
@(Html.Kendo().Grid(Model)
    .Name("Grid")
//MANEJO DE COLUMNAS
    .Columns(columns =>
    {
columns.Bound(p => p.ID_USUARIO).Visible(false);
columns.Bound(p => p.NOMBRES_USUARIO);
columns.Bound(p => p.CEDULA_USUARIO);
columns.Bound(p => p.MAIL_USUARIO);

    }

```



```

//OPCIONES DE LA GRILLA

.Editable(editable => editable.Mode(GridEditMode.InLine))
.Filterable()
  .Sortable()
  .Selectable()
  .Pageable(m => m.PageSizes(new int[] { 10, 20, 50, 100 })))
.Groupable()

})

//LLAMADO DE CONTROLADORES DENTRO DE LA GRILLA PARA MANEJO DE BOTONES
.Create(update => update.Action("Create", "Usuario"))
  .Read(read => read.Action("Read", "Usuario"))
  .Update(update => update.Action("Update", "Usuario"))
  .Destroy(update => update.Action("Destroy", "Usuario"))
)
.Resizable(resize => resize.Columns(true))
)
}

</div>

```

Anexo 3. Descargar e instalar Kendo de Telerik

TELERIK dispone de una página para descargar una versión de prueba 30 días para el uso de las librerías (<http://www.telerik.com/products/aspnet-ajax/download.aspx>). El ejecutable ocupa un valor en disco de 3.24MB. Además de una versión paga.

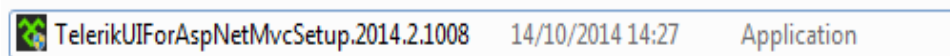


Figura 1 Vista de ejecutable de Telerik

Luego de la descarga se instala la librería que se acopla en el menú de Visual Basic .Net.

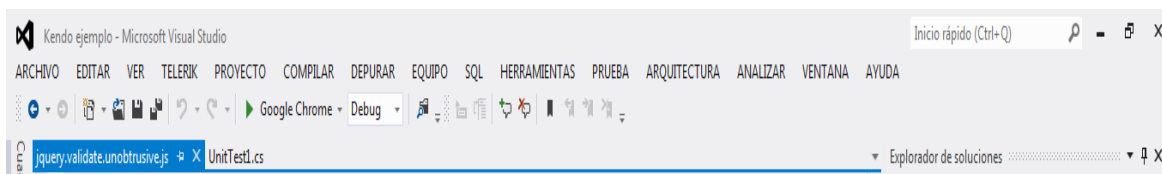


Figura 2 Vista de Telerik en Visual Basic .Net

Uso de Kendo con MVC

Para desarrollar un proyecto usando las utilidades de Kendo se debe seleccionar en el inicio del proyecto la ejecución de la aplicación Web con estos componentes y automáticamente se desplegará la solución con los accesos a las utilidades y herramientas de Telerik.

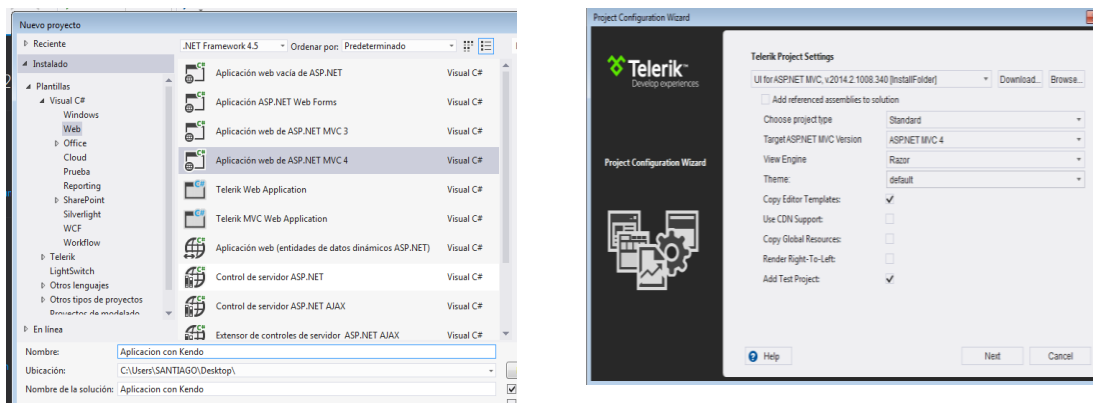


Figura 3 Vista crear nuevo proyecto usando TELERIK

Anexo 4. Entrevista para tesis

ANÁLISIS DE REQUERIMIENTOS SEGÚN LA IEEE830

Entrevista cliente de Software: Administrador

Datos representante legal

Nombre: Adm. Emp. Johan Schmollgruber
Cargo: Gerente Administrativo

Propuesta para desarrollo: Registrar horarios de trabajo para empleados de locales comerciales.

Cuántas personas laboran en la empresa: total 29
1 Asistente de Gerencia
1 Jefe de personal
27 empleados 10 en locales comerciales y
17 en logística y entrega de productos

Cuántos locales comerciales forman la cadena de distribución? total 4
Villa Cumbaya Cumbaya
Ventura Mall Tumbaco
CCI Quito - Av. amazonas
C.C El Bosque Quito - Av. del parque

Cómo se controla el horario de trabajo del personal?

El jefe de personal asigna los horarios a los empleados y las tareas a realizar son dos empleados por local comercial

Cómo gestiona la empresa los atrasos?

El jefe de personal informa a la Asistente de gerencia sobre los atrasos, llena un cuadro en documento excel gerente decide sanciones, llamadas de atención o liquidación

Se descuenta los atrasos?

Hay una norma de descuentos usado cálculo por minuto de atraso, se va a implementar un monto por atraso fijo de \$5 para evitar incumplimientos

El poco control es pérdida para la empresa?

Sí, pero no es posible contratar inspectores para cada centro comercial, no es rentable.

Como conoce la información del registro de sus empleados?

Solo por acceso a hoja de excel realizada por asistente de gerencia.

Que piensa sobre la automatización y el software?

Muy buenas, pero muy caro para empresas de recursos limitados.

Hay indicadores de pérdidas por atrasos de la empresa?

No se han realizado este tipo de análisis

Quién se encarga de la contratación?

La asistente de gerencia se encarga de la contratación, previa calificación de la carpeta por parte del Jefe de personal y aprobación del Gerente

En caso de ausencia quien se encarga de suplir al gerente?

El jefe de personal en temas de personal no pagos a proveedores

Entrevista cliente de Software: Jefe de personal

Nombre: Sra. Sara Calderón

Cargo: Jefe de personal

Cuáles son su actividades en la Empresa?

Control de personal, asistencia, asignación de horarios, revisión de cumplimiento de tareas, seleccionar nuevo personal, revisión de cuadernos de asistencias, recepción de llamadas de aviso de entrada del personal.

Cómo controla la asistencia a los Empleados?

Los empleados se registran en un cuaderno donde consta la hora de entrada de salida y con las llamadas telefónicas que cada empleado realiza a la hora de ingreso, esto es comparado con un cuaderno donde se tiene las asignaciones del personal cuando fue contratado.

Cómo asigna los horarios para el personal?

Este horario es asignado el momento de la contratación del personal, los empleados tienen un horario fijo para todo el tiempo que labore en la empresa

Cómo selecciona el sitio de trabajo para un empleado?

Para evitar atrasos se asigna el sitio de trabajo dependiendo el lugar de domicilio para el empleado, en caso de que el empleado viva en un lugar distante se al vierte en la contratación de las normas, de la toma de horarios, faltas, atrasos y sanciones

Cómo se gestiona multas para los empleados?

La determinación de multas para los empleados se realiza en conjunto con el Gerente determinando si existe reincidencia en el empleado, justificaciones de los imprevistos que hayan tenido, etc., donde se determina si tendrá un valor económico o una amonestación verbal.

Cómo se gestiona las sanciones para los empleados?

En la empresa no se trabaja con sanciones, pero si se desea implementar por motivo legal y de seguridad porque hay empleados que no cumple con las reglas de la empresa.

Cuál es el valor de las multas?

Este valor es calculado dependiendo el tiempo de atraso que tenga el empleado al día, su valor está establecido en rangos de tiempo. Las faltas se calcula el valor de un día de trabajo incluido un porcentaje adicional que puede ir entre el 5 al 10%

Cuántos empleados hay por local?

En cada local se designa 2 empleados por la extensión de horarios que tienen los centros comerciales, uno trabaja en el horario de la mañana y el otro trabaja en el horario de la tarde

Cómo distribuye los empleados por local?

La distribución de los empleados se realiza determinando la cercanía al domicilio del empleado para el centro comercial, siempre van a ser destinadas 2 personas a un local y el horario semanal a trabajar al momento de la contratación

Con qué tecnología cuenta cada local?

Cada local cuenta con un computador, impresora, cámaras de seguridad

Qué pasa con la distribución de horas por empleado?

Un empleado debe cumplir 40 horas a la semana y estas son distribuidas en los 7 días de trabajo de lunes a domingo por motivo de la atención de los centros comerciales, de lunes a viernes trabajan de 5 a 6 horas diarias y las faltantes son completadas entre sábado y domingo, en caso de exceder las cuarenta horas estas son canceladas en el rol como horas suplementarias o extraordinarias.

Si una persona falta cómo es remplazada?

El empleado da aviso sin importar la hora que sea sobre su falta y se utiliza a una persona de logística para hacer el remplazo.

Entrevista cliente de Software: Asistente de gerencia

Nombre: Sra. Doris Cabascango

Cargo: Asistente de gerencia

Cuáles son sus actividades en la organización?

Llevar la agenda del Gerente, contacto con proveedores (pagos y requerimientos) , contratos de personal, ingreso al IESS de documentos, manejo de nómina, pagos de personal, llevar registros de incumplimientos de personal y proveedores, registro de facturas, coordinación de eventos

Cómo contratan personal?

Como asistente toma las carpetas y las entrega al jefe de personal, nuevamente el jefe entrega las seleccionadas como aptas, las llama para que se entrevisten con el jefe de personal y el gerente finalmente decide la contratación

Cómo registran el personal contratado?

Primero toma los datos del nuevo empleado en cuaderno a mano, luego pasa esos datos a una hoja de excel para realizar el trámite en el seguro social del aviso de ingreso

Qué datos solicita al nuevo empleado?

Nombres completos	Dirección de correo electrónico
Cédula de identidad	Referencia Telefónicas
Fecha de nacimiento	Dirección de domicilio

Qué datos registra en el documento?

Nombres completos	Dirección de correo electrónico	Fecha de contrato
Cédula de identidad	Números Telefónicos	Fecha de ingreso al IESS
Fecha de nacimiento	Dirección de domicilio	

Solicita fotografía actualizada del empleado para su contratación?

No hay ningún registro

Se asigna las actividades del empleado en el contrato?

El empleado sabe para lo que es contratado, quien asigna las actividades es el Jefe de personal

Según el Administrador usted registra atrasos como lo hace?

Se registra en hoja de excel con los datos entregado por el jefe de personal, la información es compartida con el Gerente

Cómo se informa de las multas o sanciones a los empleados?

El Gerente y Jefe de personal se reúne y toman la decisión de enviar llamados de atención, multas o liquidación del empleado

Cómo compara o verifica las multas o sanciones del empleado?

Se controla mediante la hoja de excel, realizando por empleado y por fechas

Los registros de atrasos son frecuente y amplios?

Los registros son llevados por el Jefe de personal para el control de asistencia, la información que se recibe es filtrada

DATOS REALES HORARIOS CITYVENTA

REGISTRO MES DE JUNIO

Día/emp	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27	
1	5	0	0	240	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	2	0	0	0	0	0	2	
3	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	0	0	0	0	2	
4	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	2	0	
5	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	2	0	
6	0	5	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	3	3	0	5	0	3	0	0	2	0	
7	0	0	0	0	0	0	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	
8	3	0	0	0	0	0	0	240	0	0	0	5	0	0	5	0	0	0	0	4	0	0	2	0	0	0	0	
9	0	0	5	0	0	0	0	240	0	0	0	5	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	
10	0	15	0	0	5	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	3	
11	0	0	0	0	5	0	10	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	4	0	0	2	
12	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	480	0	3	0	0	0	0	0	0	
14	15	0	0	0	0	240	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	10	0	0	0	0	0	
15	0	0	10	5	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	
16	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	240	0	0	0	0	0	0	0	0	5	0	0	0	0	0	2	0	0	0	0	0	3	0	0	0	0	
18	0	0	0	0	0	0	0	0	10	0	0	0	5	0	0	0	2	0	0	0	0	0	2	0	0	5	0	
19	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	
20	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	2	0	0	0	
21	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	
22	0	0	0	0	10	0	0	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	2	0
23	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	
24	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	2	4	2	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	
26	5	0	0	0	0	0	0	0	0	0	0	10	0	0	0	10	0	0	0	0	0	5	0	5	0	0	480	
27	0	0	15	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	3	0	0	0	0	0	15	0	0	
28	0	10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
TOTAL	46	270	45	500	20	250	15	730	25	245	22	500	25	480	9	15	13	18	493	19	12	25	14	20	15	16	489	
	RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM		RVM	

130

Nota: RVM revisar si hay justificación para las multas

# veces atrasos	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27
	6	4	5	4	3	2	2	4	3	2	4	4	4	1	3	2	5	4	5	6	4	4	6	6	1	6	5
\$5 por atraso	30	20	25	20	15	10	10	20	15	10	20	20	20	5	15	10	25	20	25	30	20	20	30	30	5	30	25
TOTAL VECES ATRASOS	105																										
TOTAL MULTAS POR ATRASO EMPLEADOS	525 \$																										

CÁLCULO ACTUAL DE MULTAS

Costo hora normal

1,33

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27
1,02	5,96	0,99	11,04	0,44	5,52	0,33	16,12	0,55	5,41	0,49	11,04	0,55	10,60	0,20	0,33	0,29	0,40	10,89	0,42	0,27	0,55	0,31	0,44	0,33	0,35	10,80
			LA				LA				LA		LA					LA								LA

Nota: LA enviar llamado de atención

Total multas mes de Junio **95,64**

Total minutos de atraso **4331**

Total horas de atraso **72**

representativo días **9**

TABLA REVISADA DE MES DE JUNIO

Día/emp	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27
1	5	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	2	0	0	0	0	0	2
3	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	3	0	0	0	0	2
4	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
5	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	2	0
6	0	5	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	3	3	0	5	0	3	0	0	2	0
7	0	0	0	0	0	0	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
8	3	0	0	0	0	0	0	0	0	0	0	5	0	0	5	0	0	0	0	4	0	0	2	0	0	0	0
9	0	0	5	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
10	0	15	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	3

11	0	0	0	0	5	0	10	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	4	0	0	2	
12	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	
14	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	0	0	0	10	0	0	0	0	0	
15	0	0	10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	
16	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	2	0	0	0	0	0	3	0	0	0	0	
18	0	0	0	0	0	0	0	0	10	0	0	0	5	0	0	0	2	0	0	0	0	2	0	0	5	0		
19	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	
20	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	2	0	0	0	0	
21	0	0	0	240	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	
22	0	0	0	0	10	0	0	0	0	0	5	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	
23	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	0	
24	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	2	4	2	0	0	0	0	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	0	0	5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	
26	5	0	0	0	0	0	0	0	0	0	0	10	0	0	0	10	0	0	0	0	0	5	0	5	0	0	0	
27	0	0	15	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	3	0	0	0	0	0	15	0	0	
28	0	10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	
TOTAL	46	30	45	260	20	10	15	10	25	245	22	20	25	480	9	15	13	18	13	19	12	25	14	20	15	16	9	
				RVM						RVM				RVM														

# veces atrasos	6	3	5	3	3	1	2	1	3	2	4	3	4	1	3	2	5	4	4	6	4	4	6	6	1	6	4	
\$5 por atraso	30	15	25	15	15	5	10	5	15	10	20	15	20	5	15	10	25	20	20	30	20	20	30	30	5	30	20	
TOTAL VECES ATRASOS																												
TOTAL MULTAS POR ATRASO EMPLEADOS																												

96

480 \$

CALCULO DE MULTAS

Costo hora normal

1,33

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27
1,02	0,66	0,99	5,74	0,44	0,22	0,33	0,22	0,55	5,41	0,49	0,44	0,55	10,60	0,20	0,33	0,29	0,40	0,29	0,42	0,27	0,55	0,31	0,44	0,33	0,35	0,20
													LA													

Nota: LA enviar llamado de atención

Total multas mes de Junio 32,04

CÁLCULO DE RELACIONES ATRASOS EFECTOS ECONÓMICOS

Datos Generales:

Número de empleados	30	horas laborables día	8
Días Laborables	365	horas laborables semana	40
Sueldo Básico	318		
Días laborables mes	30		

Cálculos:

Costo hora laboral mes	1,33 \$
Costo minuto laboral	0,02 \$

Referencia costos - atraso

Para un empleado

Atraso de un minuto a la semana	0,11 \$
Atraso de un minuto al mes	0,44 \$
Atraso de un minuto al año	5,74 \$

Para todo los empleados

Atraso de un minuto a la semana	3,31 \$
Atraso de un minuto al mes	13,25 \$
Atraso de un minuto al año	172,25 \$

Datos Resultantes

Para un empleado

Minutos atrasos al año	260 min	4,33 horas
Costo al año	5,74 \$	

Por todo los empleados

Minutos atrasos al año	7800 min	130 horas	16,25 días
Costo al año	172,25 \$		

Calculadora de costos – atrasos

Para un empleado por día

Ingrese minutos atrasos	1920	min/año	32	horas/año
Costo al año	42,4	\$	4	días/año

Réplica de atrasos para todos los empleados

Minutos atrasos al año	57600	min/año	960	horas/año
Costo al año	1272	\$	120	días/año
			4	mes/año

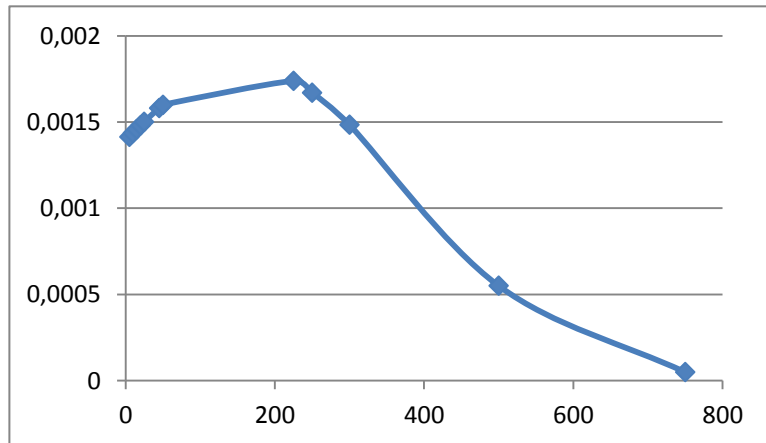
DISTRIBUCIÓN ESTADÍSTICA DE DATOS

Minutos de atraso totales del mes de Junio

Promedio : 147,346153 **valor mínimo:** 9

Desv. Estándar: 213,158897 **valor máximo:** 730

Datos	Distribución
5	0,001497511
10	0,001520735
15	0,001543469
20	0,001565682
25	0,00158734
45	0,00166781
50	0,001686236
225	0,001751411
250	0,001666653
300	0,001448231
500	0,00047627
750	3,43932E-05

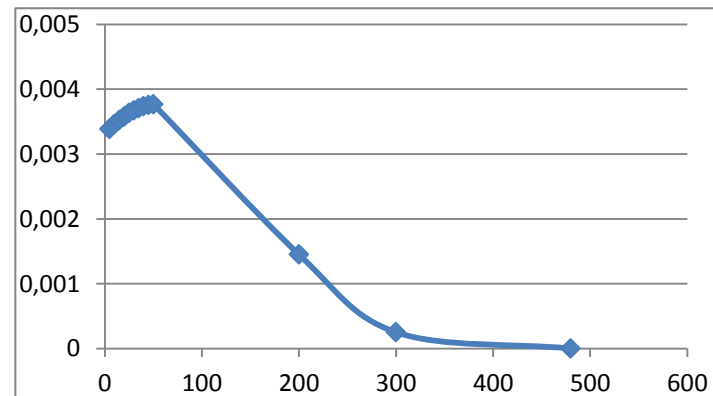


Minutos de atraso del personal mes de junio con revisión y justificación

Promedio: 53,7407407 **valor mínimo:** 9

Desv. Estándar: 105,848801 **valor máximo:** 480

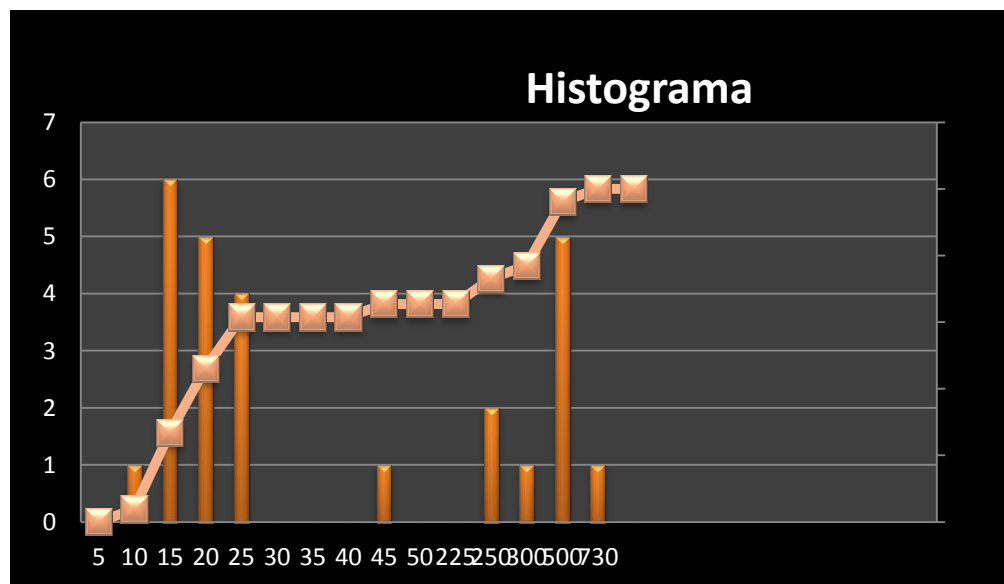
Datos	Distribución
5	0,003389852
10	0,003460532
15	0,003524811
20	0,003582282
25	0,003632575
30	0,003675364
35	0,003710369
40	0,003737359
45	0,003756154
50	0,00376663
200	0,001450856
300	0,000251694
480	1,13424E-06



ESTADÍSTICAS MINUTOS DE ATRASO

grupos		repetición
5	10	1
10	20	8
20	25	1
25	30	2
30	40	0
40	45	0
45	50	0
50	225	0
225	250	0
250	300	0
300	500	3
500	730	0

136



SUPUESTOS USANDO DATOS REALES DEL MES DE JUNIO 2013 DE COSTOS ANUAL PARA LA EMPRESA

Minutos perdidos	4331 min	empleados	30
Costos de atrasos	95,64 \$		

Si los tiempos de atraso se mantuvieran estables para todos los meses del año

Minutos perdidos en el año de todos los empleados	51972 min
---	-----------

Se deduce entonces:

Se pierde al año	866,20 horas
	108,28 días laborables 8 horas c/u
Por cada empleado	1732,40 minutos
	28,87 horas
	3,61 días laborables 8 horas c/u

Costos:

Total año de atraso	1147,72 \$
costo por empleado	38,26 \$

RECONOCIMIENTO DE ACTORES DEL PROCESO

ACTORES DEL PROCESO PARA AUTOMATIZAR

Cientes CITYVENTA

Actor	Cargo
Johan Schmollgrube	Gerente administrador
Sara Calderón	Jefe de personal
Doris Cabascango	Asistente de Gerencia
Magdalena Cabezas	Asistente de talento Humano
Alexandra Rubio	Pagaduria y contabilidad
Personal planta	Vendedores de almacen (10) Personal de logistica (5) Personal de producción (12)

DESARROLLADORES DE LA AUTOMATIZACIÓN DEL PROCESO

Proveedores Universidad Politécnica Salesiana

Actor	Cargo
Jimena Morales	Analistas y desarrolladores de
Santiago Moya	requerimientos para software

LOCALES COMERCIALES EN DONDE SE VA A AUTOMATIZAR EL PROCESO

Locales

Local	Ubicación
Villa Cumbaya	Cumbaya
Ventura Mall	Tumbaco
CCI	Quito - Av. amazonas
C.C El Bosque	Quito - Av. del parque

COSTOS DE DESARROLLO Y POSIBLE IMPLEMENTACIÓN

Hardware

Cant	Detalle	V. Unitario	V. Total
5	Computadores (core 2 duo -windows 7)	650	3250
1	Servidor	450	450
5	Cámara web	15	75

Software

1	Licencia de visual studio 2010 - 2013	799	799 *1
1	Licencia de SQL server 2012	898	898 *2
1	Alquiler de Hosting Web (anual)	55	55 *3

Desarrollo

2	Analista y Desarrollador de proyecto	1000	2000
---	--------------------------------------	------	------

Total **7527** \$

Posible Implementación Costo

Costo total implementación	7527
Hardware disponible en CITYVENTA	3325
TOTAL	4202 \$

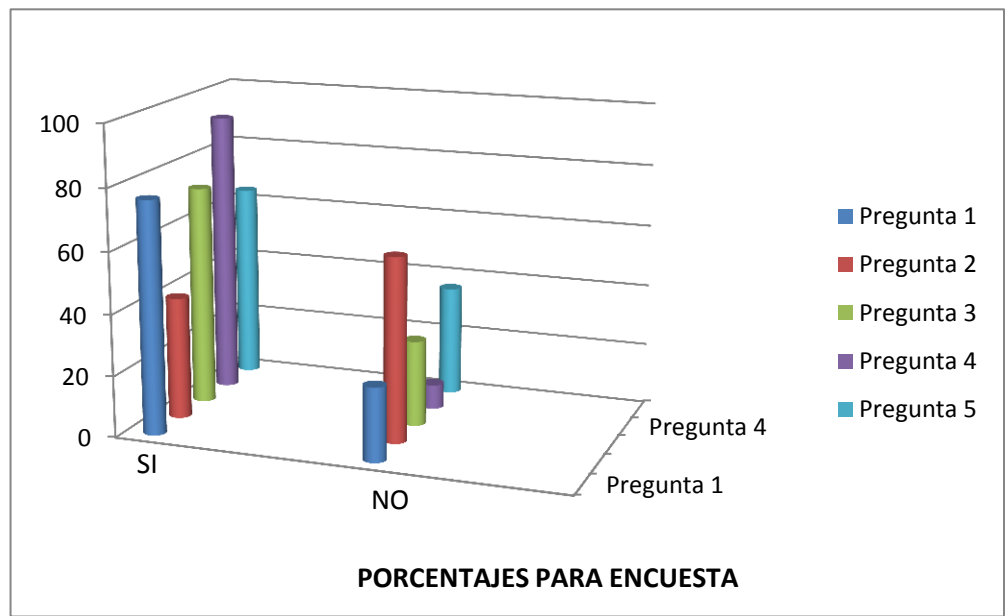
Anexo 5. Encuesta a empleados

ENCUESTA

- | | | | |
|-----|---|--------|--------|
| 1.- | Usted encuentra actualizado el cuaderno de asistencia cuando ingresa a laborar | SI () | NO () |
| 2.- | Está de acuerdo con el método de registro de asistencia que se está usando | SI () | NO () |
| 3.- | Cree Usted que la inasistencia o atraso de cualquier trabajador afecta su economía y la de la empresa | SI () | NO () |
| 4.- | Piensa Usted que un programa informático puede simplificar o mejorar las tareas diarias | SI () | NO () |
| 5.- | Estaría dispuesto a utilizar otra forma de registro de asistencia con tecnología informática | SI () | NO () |

RESULTADOS DE LA ENCUESTA																												
PREGUNTA	ENCUESTADOS																									SI	NO	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25			
1	0	0	0	X	0	0	X	X	0	0	0	0	X	X	X	0	0	0	0	0	0	0	0	0	0	0	19	6
2	0	0	X	X	X	X	0	0	0	X	X	X	X	0	0	X	X	X	X	X	X	0	0	0	X	10	15	
3	0	0	0	X	0	0	X	0	0	X	0	X	0	X	0	X	0	X	0	0	0	0	0	0	0	18	7	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	0	0	0	0	0	0	X	23	2	
5	X	X	X	0	0	0	0	0	X	X	0	0	0	0	0	0	X	X	X	0	0	0	0	0	X	16	9	

PORCENTAJE DE RESPUESTAS A CONSULTAS



PREGUNTA	% SI	% NO
1	76	24
2	40	60
3	72	28
4	92	8
5	64	36

Anexo 6. Cronograma para ejecución del proyecto

CRONOGRAMA PARA EJECUCIÓN DEL PROYECTO: ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Para la ejecución del proyecto y construcción del producto se ha seleccionado la metodología RUP, en los cronogramas se especifican las fechas de inicio - fin y el número de Iteraciones que se proponen para pasar cada una de las fases. En el cronograma específico se encuentra identificado los pasos definidos y la entrega de los artefactos requeridos en cada fase.

CRONOGRAMA GENERAL PARA FASES DE RUP			
FASES DE RUP	FECHA DE INICIO PROCESO	FECHA FIN PROCESO	ITERACIONES
INICIO O CONCEPCIÓN	13/10/2013	20/05/2014	4
ELABORACIÓN	25/04/2014	07/09/2014	6
CONSTRUCCIÓN	28/09/2014	28/01/2015	5
TRANSICIÓN	01/02/2015	26/02/2015	3

Anexo 7. Modelo de negocios

LEVANTAMIENTO DE MODELO DE NEGOCIO CITYVENTA PARA EL REGISTRO DE JORNADA LABORAL.

1. Modelo de contratación de personal

Se diagrama y muestra el modelo de contratación de personal de la siguiente manera:

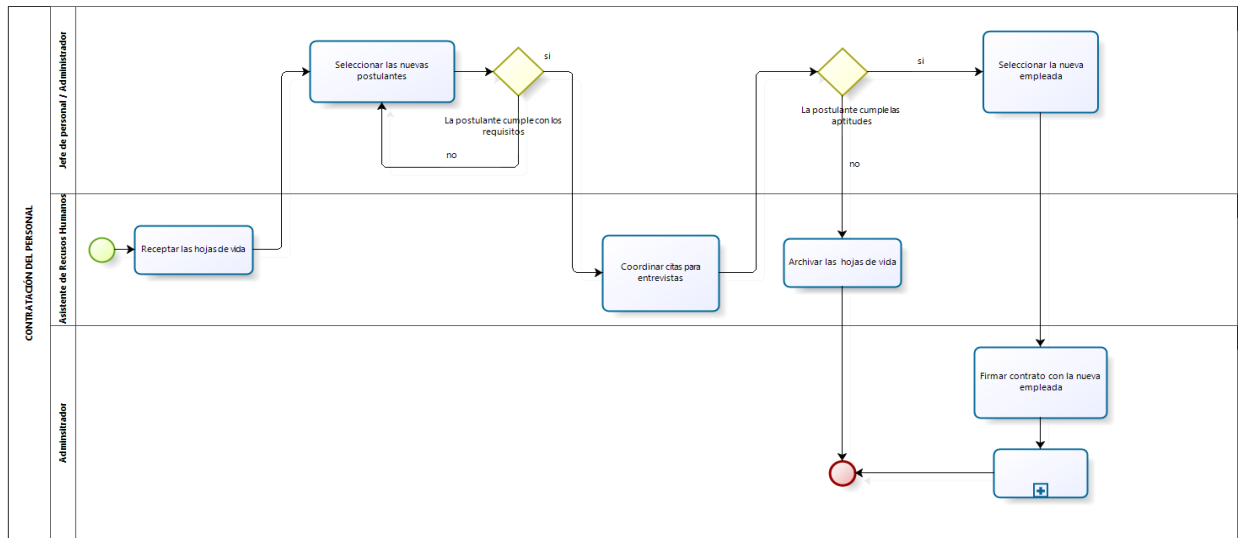


Diagrama 1.1 Modelo de negocios para contratación del personal

El modelo de negocios para la contratación de personal tiene un subproceso para el registro del personal:

- **Subproceso de registro de personal**

El subproceso se encarga de registrar al empleado con sus datos personales, las fecha de contrato y fecha de ingreso al Seguro social.

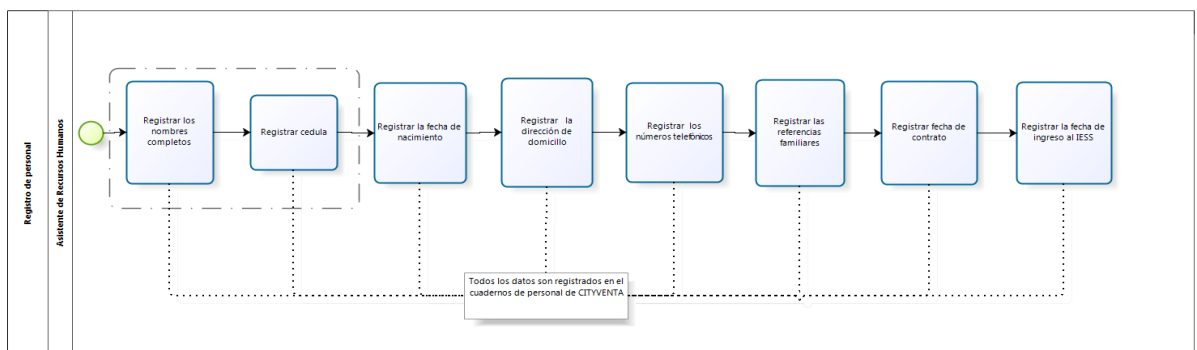


Diagrama 1.2 Subproceso de modelo de negocios para registro de personal

2. Modelo de negocio para registro de empleados

Se diagrama el modelo de negocio para registro de empleados en los locales comerciales.



Diagrama 2.1 Modelo de negocios para registro de empleado

El modelo de negocio para registro de empleados en los locales tiene dos subprocesos que definen la entrada y salida del personal:

- **Subproceso de registro de entrada del empleado**

El subproceso se refiere a la forma que los empleados registran su ingreso en los locales comerciales.

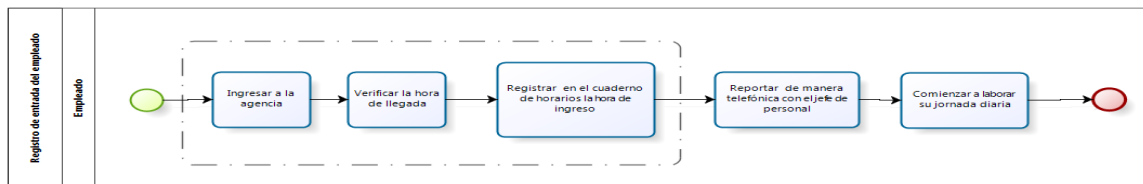


Diagrama 2.2 Subproceso de modelo de negocio para Registro de entrada del empleado

- **Subproceso de registro de salida del empleado**

El subproceso se refiere a la forma que los empleados registran su salida en los locales comerciales.

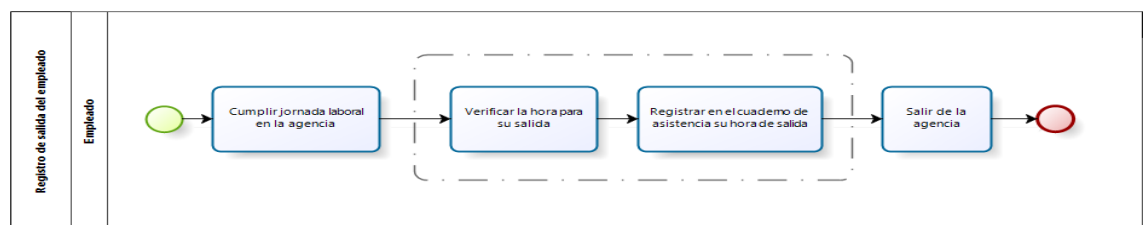


Diagrama 2.3 Subproceso de modelo de negocio para registro de salida del empleado

3. Modelo de negocio para asignación de horarios

El modelo de asignación de horarios se realiza de la siguiente manera:

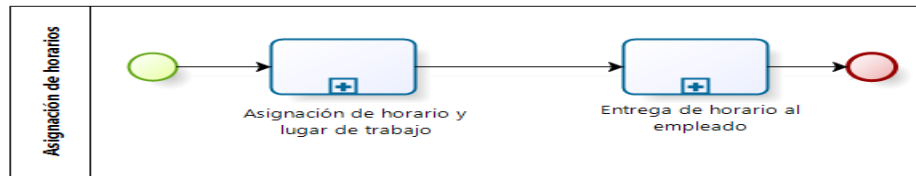


Diagrama 3.1 Modelo de negocio para asignación de horarios

El modelo de asignación de horarios tiene dos subprocesos en los cuales se asigna el horario y lugar de trabajo del personal:

- **Subproceso asignación de horario y lugar de trabajo del personal.**

El subproceso se encarga de la asignación de horarios dependiendo del centro comercial al que el empleado es enviado a laborar.

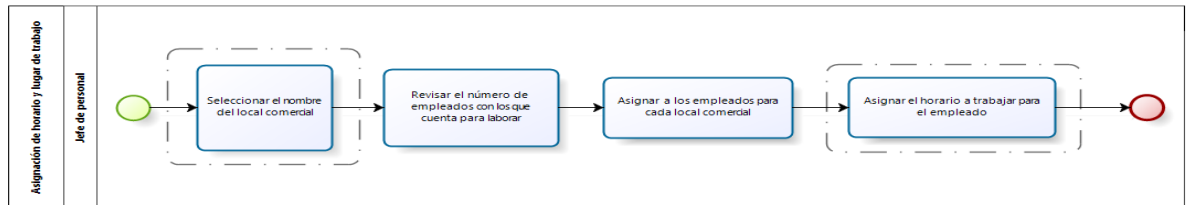


Diagrama 3.2 Subproceso modelo de negocio para asignación de horarios y lugar de trabajo del personal

- **Subproceso de entrega de horarios.**

El subproceso se encarga de difundir los horarios al personal, mediante una llamada telefónica y dicta el horario de trabajo de la semana al empleado que se encuentra registrado en el primer turno, los otros empleados verifican su horario en el cuaderno.

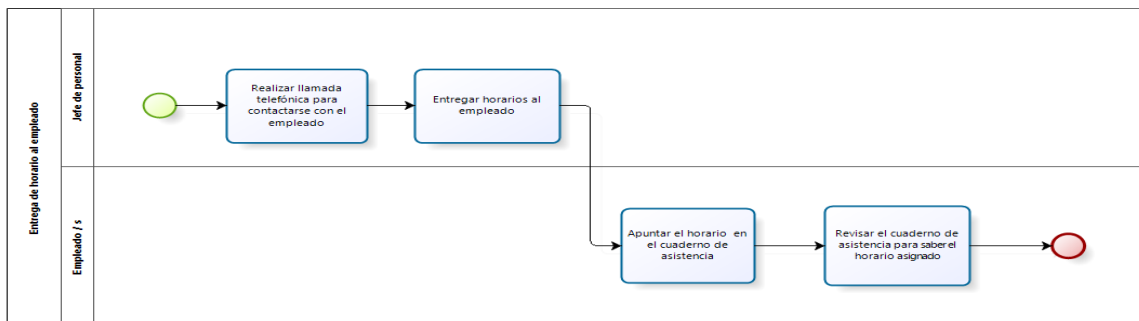


Diagrama 3.3 Subproceso de modelo de negocio para entrega de horario del empleado

4. Modelo de negocio para revisión de asistencia de empleados

El modelo de multas y sanciones se realiza de la siguiente manera:

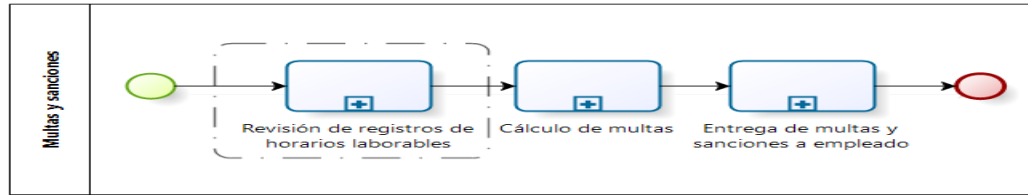


Diagrama 4.1 Modelo de negocio para revisión de asistencia para empleados

El modelo de multas y sanciones tiene tres subprocesos:

- **Subproceso de revisión de registros de horarios**

El subproceso se encarga de la recopilación de información registrada en cada local y comparación de coincidencias con las eventuales llamadas telefónicas realizadas para el registro.

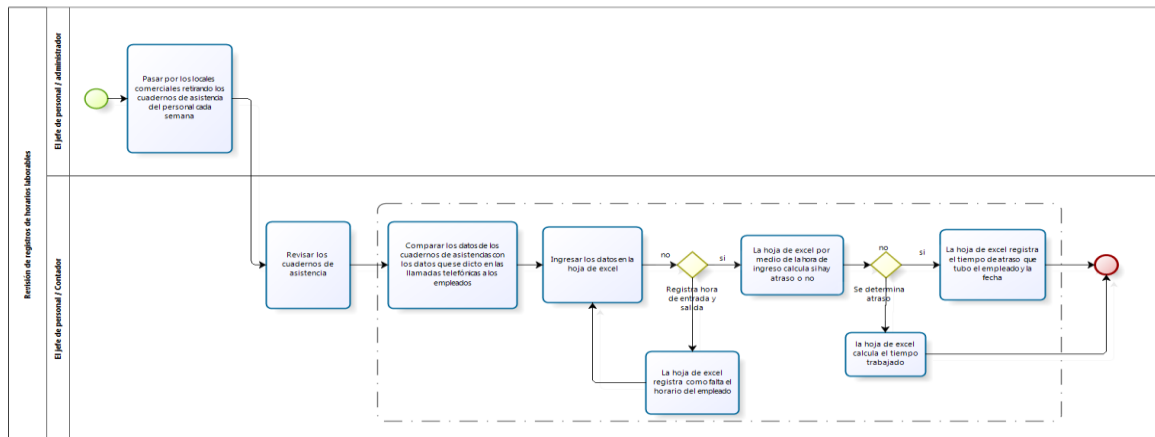


Diagrama 4.2 Modelo de negocio para revisión de registros de horarios

- **Cálculo de pagos**

El subproceso se encarga de realizar el cálculo de atrasos según registros, evalúa si no es justificado el atraso y se calcula el tiempo de atrasos para en su etapa final según políticas vigentes y las justificaciones generadas por los empleados.

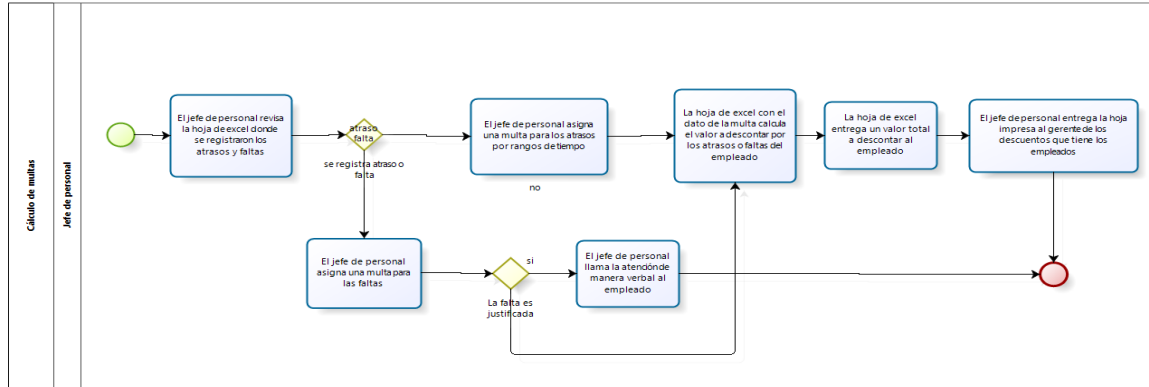


Diagrama 4.3 Subproceso modelo de negocio para cálculo de multas

- **Entrega de multas y sanciones a empleados.**

El subproceso se encarga del análisis de los atrasos y faltas, con el conocimiento de las multas al empleado, gerencia genera llamados de atención y se entrega por escrito las multas económicas.

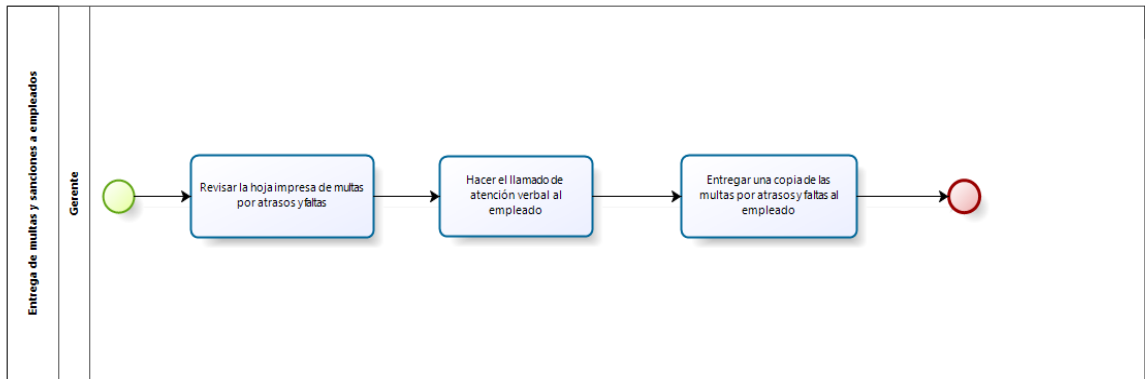


Diagrama 4.4 Subproceso modelo de negocio para entrega de multas y sanciones a empleados

Anexo 8. Casos de uso aplicación de escritorio

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Modelo de Casos de Uso para software aplicación de escritorio

Versión [4.0]

Historial de revisiones

Fecha	Versión	Descripción	Autor
[25/08/2014]	[1.0]	Casos de uso para software	Santiago Moya P.
[05/09/2014]	[2.0]	Casos de uso para software Reconocimiento.	Santiago Moya P.
[20/09/2014]	[3.0]	Casos de uso para software, control de asistencia	Santiago Moya P.
[29/09/2014]	[4.0]	Casos de uso para software revisión final	Santiago Moya P.

1. Diagrama de Casos de uso para aplicación de escritorio

Diagrama de caso de uso Autenticar usuarios

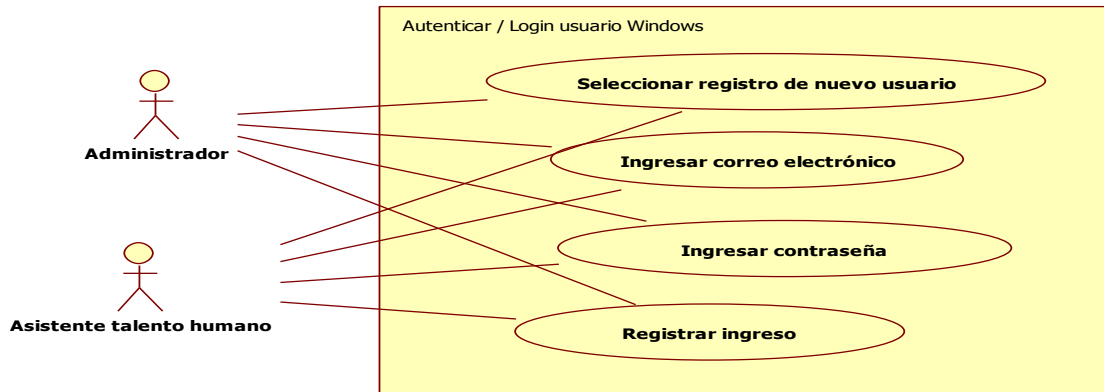


Diagrama 1.1 Caso de uso Autenticar usuario escritorio

Descripción del caso de uso

Nombre	Autenticar usuarios escritorio
Autor(es)	Jimena Morales / Santiago Moya
Fecha	22 de noviembre de 2013.
Descripción	Permite autenticar al usuario que tiene permisos para ingresar a la opción registro de usuario y realizar captura de imágenes.
Actores	Administrador, Asistente de talento humano
Precondiciones	El Administrador o la Asistente de talento humano deben estar registrados para acceder a la función. Deben estar creados perfiles de usuarios en la aplicación web.
Flujo normal	<ol style="list-style-type: none"> 5. La Asistente de talento humano debe seleccionar registro de nuevo usuario. 6. La Asistente de talento humano deben ingresar sus credenciales para acceder a las opciones de ingreso de datos del personal. 7. El sistema valida ingreso de información. 8. El sistema permite acceso a ventana de registro de empleado
Flujo alternativo	<ol style="list-style-type: none"> 6.1. El sistema muestra un mensaje de alerta si la información ingresada no pasa el filtro de validación.
Post condiciones	Se ingresa a la ventana de registro de empleado.

Tabla 1.1 Descripción caso de uso Autenticar usuarios en sistema de escritorio

Diagrama de casos de uso para registrar y reconocer usuarios.

- **Registrar empleado usando sistema de escritorio**

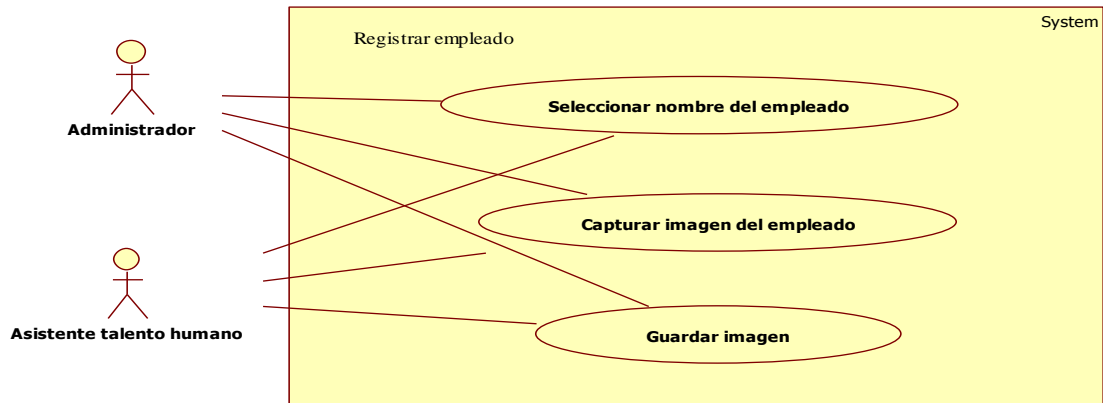


Diagrama 1.2 Caso de uso registrar empleado usando sistema de escritorio

Descripción del caso de uso

Nombre	Registrar empleado
Autor(es)	Jimena Morales / Santiago Moya
Fecha	22 de noviembre de 2013.
Descripción	Permite ingresar la imagen del usuario al sistema, para esto se selecciona el nombre del usuario y se captura su imagen (rostro del usuario).
Actores	Administrador, Asistente de talento humano
Precondiciones	El administrador o la asistente de talento humano deben estar registrados y logueados para acceder a la función. El empleado debe estar registrado en la base de datos mediante proceso de la aplicación web además de estar creados perfiles de usuarios. El sistema debe tener una cámara web.
Flujo normal	<ol style="list-style-type: none"> 1. La asistente de talento humano selecciona el nombre del empleado. 2. El administrador o la asistente de talento humano captura la imagen del rostro del usuario dando clic sobre el botón grabar 3. El sistema valida ingreso de información y consulta si desea guardar la imagen. 4. El Administrador o la Asistente de talento humano da clic en el botón aceptar. 5. El sistema muestra un mensaje de información registrada
Flujo alternativo	<ol style="list-style-type: none"> 3.1. El sistema muestra un mensaje de alerta si la información registrada no pasa el filtro de validación.
Post condiciones	Se registra la imagen del usuario en la base de datos con un archivo .jpg en el servidor.

Tabla 1.2 Descripción caso de uso registrar empleado usando el sistema de escritorio

- **Reconocer usuario usando sistema de escritorio**

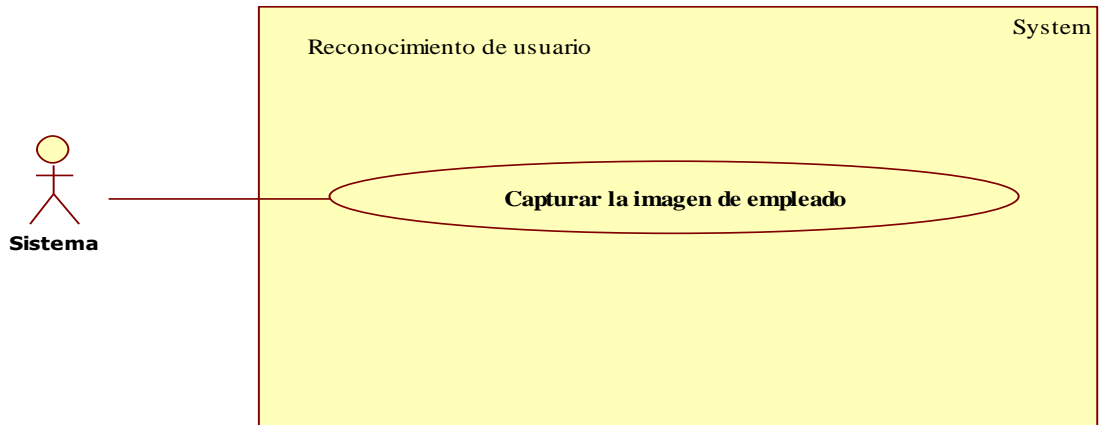


Diagrama 1.3 Caso de uso reconocer usuario usando sistema de escritorio

Descripción de caso de uso

Nombre	Reconocer usuario
Autor(es)	Jimena Morales / Santiago Moya
Fecha	30 de noviembre de 2013.
Descripción	Reconoce al usuario antes registrado en el sistema, mediante la captura automática de la imagen del rostro que es comparada con la imagen guardada en el registro del empleado.
Actor(es)	Sistema.
Precondiciones	El usuario debe estar ingresado previamente en la base de datos del sistema. La imagen del rostro del usuario debe estar guardada en el servidor y ligada a la base de datos del sistema. El sistema debe tener acceso a una cámara web.
Flujo normal	1. El empleado es reconocido por el sistema si se sitúa frente a la cámara web.
Flujo alternativo	2.1. El sistema muestra un mensaje de alerta si el rostro del usuario no coincide con alguna imagen guardada en el sistema.
Post condiciones	Los datos del sistema no son alterados.

Tabla 1.3 Descripción caso de uso reconocer usuario mediante sistema de escritorio

Diagrama de caso de uso de registrar horarios

- Registrar ingreso – salida de jornada laboral

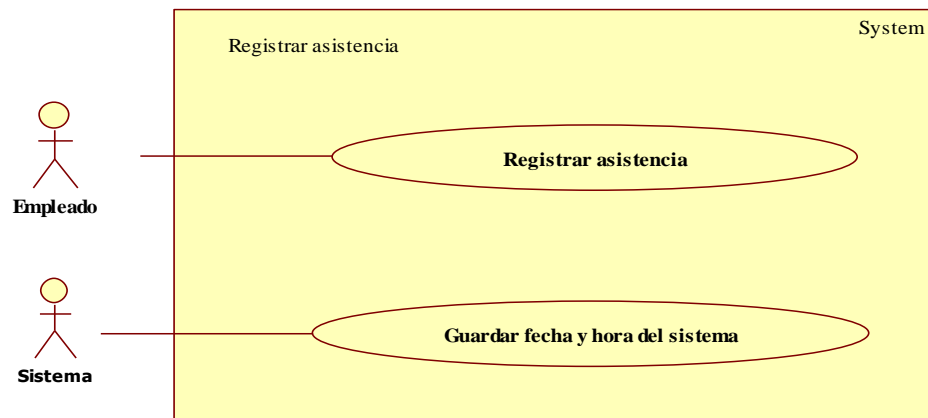


Diagrama 1.4 Caso de uso Registrar ingreso – salida de jornada laboral

Detalle de caso de uso

Nombre	Registrar ingreso salida de jornada laboral
Autor(es)	Jimena Morales / Santiago Moya
Fecha	30 de noviembre de 2013.
Descripción	Permite el Registro de la asistencia de los empleados que están en los diferentes locales comerciales.
Actores	Empleados, Sistema.
Precondiciones	El usuario debe estar ingresado previamente en la base de datos del sistema. La imagen del rostro del usuario debe estar guardada en el servidor y ligada a la base de datos del sistema. Debe estar seleccionado previamente el nombre del local. El sistema debe tener acceso a una cámara web.
Flujo normal	<ol style="list-style-type: none"> 1. El empleado es reconocido por el sistema si se sitúa frente a la cámara web. 2. El empleado da clic en el botón registrar asistencia y automáticamente el sistema captura la fecha, hora y la agencia en el que se realiza el evento. 3. El sistema muestra un mensaje de empleado registrado.
Flujo alternativo	<ol style="list-style-type: none"> 6.1. El sistema muestra un mensaje de error si el rostro del empleado no coincide con las imágenes guardadas en el sistema. 6.2. El sistema muestra un mensaje de Advertencia si el empleado: <ul style="list-style-type: none"> • No se encuentra asignado en el local. • Se encuentra retrasado del horario asignado. • No se encuentra su estado como activo en la base de datos.
Post condiciones	Se registra en la base de datos el horario.

Tabla 1.4 Descripción caso de uso registrar asistencia de jornada laboral

Anexo 9. Casos de uso aplicación web

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Modelo de Casos de Uso para software aplicación web

Versión [5.0]

Historial de revisiones

Fecha	Versión	Descripción	Autor
[25/10/2014]	[1.0]	Casos de uso para software web	Santiago Moya P.
[05/11/2014]	[2.0]	Casos de uso para software módulo parametrización	Santiago Moya P.
[15/11/2014]	[3.0]	Casos de uso para software, módulo seguridad y usuarios	Santiago Moya P.
[28/11/2014]	[4.0]	Casos de uso para software consultas	Santiago Moya P.
02/12/2014	[5.0]	Casos de uso para software revisión final	Santiago Moya P.

1. Diagrama de Casos de uso para aplicación web

Diagramas de caso de uso de seguridad/Administración de usuario

- **Autenticar usuarios.**

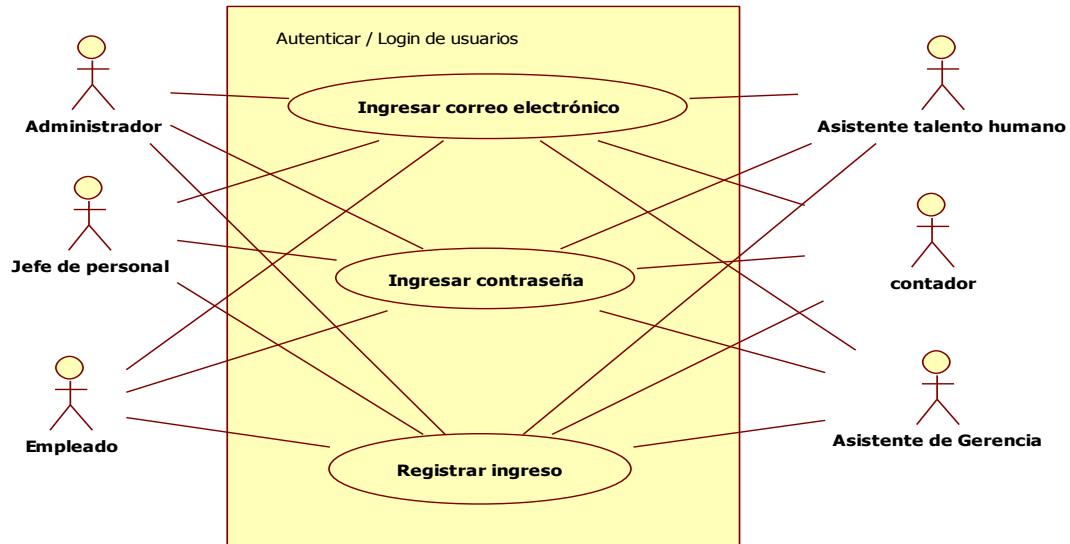


Diagrama 1.1 Caso de uso Autenticar Usuarios

Detalle de caso de uso

Nombre	Autenticar Usuarios
Autor(es)	Jimena Morales / Santiago Moya
Fecha	1 de diciembre de 2013.
Descripción	Permite acceso al sistema de los diferentes usuarios.
Actor(es)	Administrador, Asistente de talento humano, Asistente de gerencia, Contador, Jefe de personal, Empleado.
Precondiciones	Los usuarios deben estar registrados y asignados un perfil previamente.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario ingresa el correo electrónico asignado. 2. El usuario ingresa la contraseña asignada. 3. El usuario da clic en botón Iniciar Sesión.
Flujo alternativo	<ol style="list-style-type: none"> 3.1. El sistema muestra un mensaje de error si el usuario no se encuentra registrado.
Post condiciones	Los datos del sistema no son alterados.

Tabla 1.1 Descripción de caso de uso Login de Usuario

- Registrar perfiles de usuario.

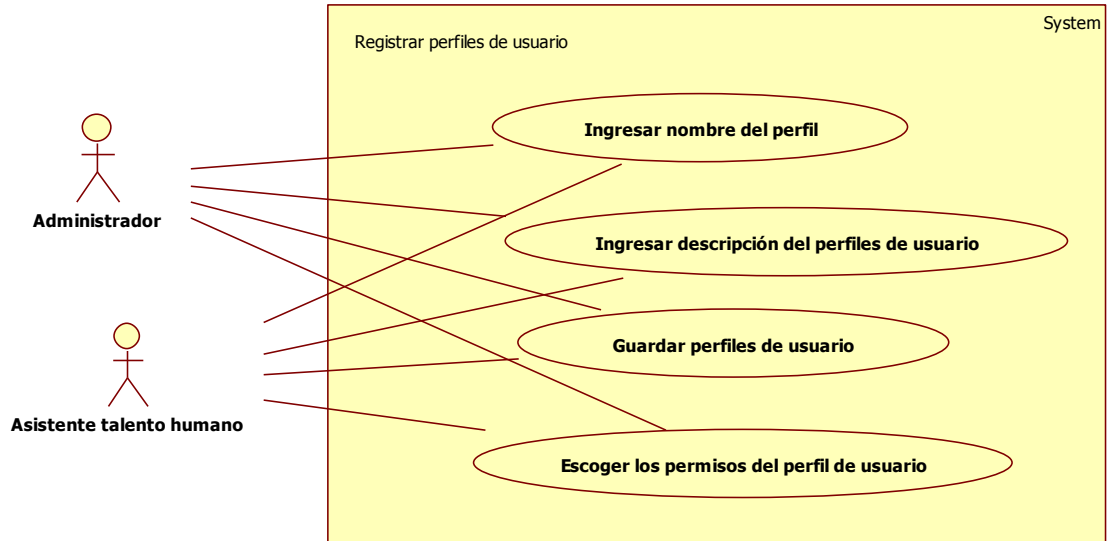


Diagrama 1.2 Caso de uso registrar perfiles de usuario

Detalle de caso de uso

Nombre	Registrar perfiles de usuario
Autor(es)	Jimena Morales / Santiago Moya
Fecha	1 de diciembre de 2013.
Descripción	Crea los perfiles de usuario para el acceso a las funciones de los usuarios en el sistema.
Actor(es)	Administrador, Asistente de talento humano
Precondiciones	El administrador o la asistente de talento humano deben estar registrados y logeados previamente.
Flujo normal	<ol style="list-style-type: none"> 1. El administrador o la asistente de talento humano ingresa el nombre del perfil. 2. El administrador ingresa la descripción del perfil. 3. El administrador da clic en el botón Grabar. 4. El sistema valida la información. 5. El administrador da clic en botón Asignar permisos. 6. El administrador escoge los permisos del perfil.
Flujo alternativo	<ol style="list-style-type: none"> 4.1. El sistema muestra un mensaje de error si la información validada no pasa el filtro de autorización.
Post condiciones	Se registra en la base de datos el perfil que se ha creado.

Tabla 1.2 Descripción caso de uso registrar perfiles de usuario

- Registrar usuarios

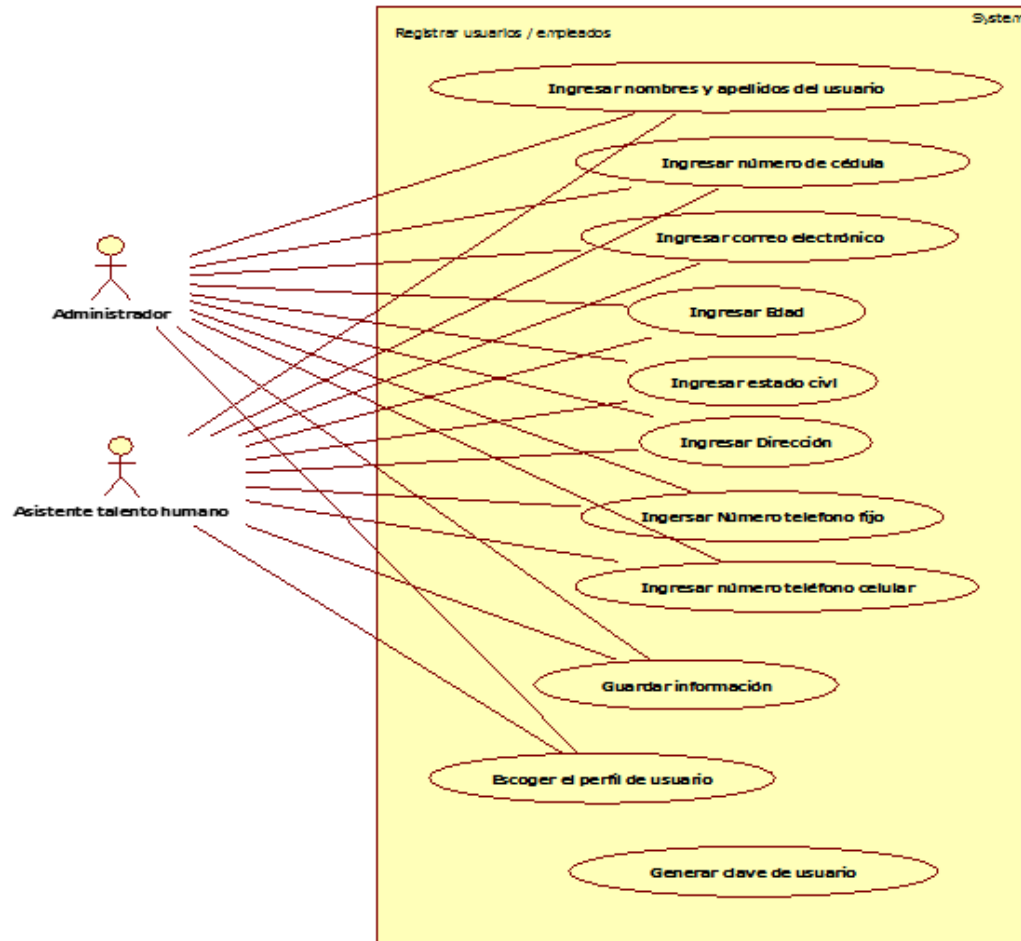


Diagrama 1.3 Caso de uso registrar usuario web

Detalle de caso de uso

Nombre	Registrar usuarios
Autor(es)	Jimena Morales / Santiago Moya
Fecha	1 de diciembre de 2013.
Descripción	Permite el ingreso y registro de usuarios y empleados al sistema, solicita nombres y apellidos, número de cédula, correo electrónico al guardar genera una clave que se guarda encriptada en la base de datos, finalmente se le asigna un perfil al usuario para que pueda navegar en el programa.
Actor(es)	Administrador o jefe de personal.
Precondiciones	Debe estar creado previamente los perfiles El Administrador y Jefe de personal debe estar registrado o logueado previamente.
Flujo normal	<ol style="list-style-type: none"> 8. El administrador o jefe de personal ingresa registro de usuarios para crear un nuevo usuario. 9. El administrador o jefe de personal ingresa información del usuario/empleada: <ul style="list-style-type: none"> • Nombre y apellido del usuario

<ul style="list-style-type: none"> • Cédula de identidad. • Mail de usuario • Edad. • Estado Civil • Dirección. • Teléfono Fijo. • Número de Celular. <p>10. El administrador o jefe de personal da clic sobre el botón Guardar.</p> <p>11. El código de usuario/clave es autogenerado por el sistema</p> <p>12. El sistema valida la información ingresada según la asignación de registro de datos, desplegando un mensaje de ingreso de datos OK.</p> <p>13. El administrador o jefe de personal escoge el perfil del usuario.</p> <p>14. El Administrador da clic sobre el botón Grabar.</p>
<p>Flujo alternativo</p> <p>10.1. El sistema muestra un mensaje de error si la información ingresada no pasa la validación de cada uno de los campos.</p> <p>10.2. El sistema encripta la clave del usuario.</p> <p>10.3. El usuario será asignado con el estado de nuevo.</p>
<p>Post condiciones</p> <p>Se registra en la base de datos los registros del usuario/empleo ingresado.</p>

Tabla 1.3 Descripción caso de uso registrar usuario usando sistema web.

Diagramas de caso de uso de parametrización

- **Registrar agencias**

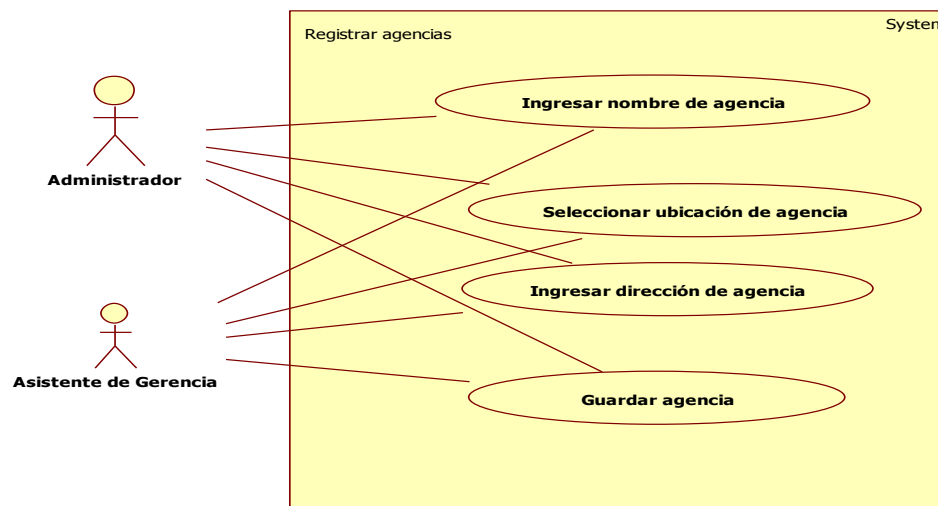


Diagrama 1.4 Caso de uso Registrar agencias

Detalle de caso de uso

Nombre	Registrar agencias
Autor(es)	Jimena Morales / Santiago Moya
Fecha	2 de diciembre de 2013.
Descripción	Registra en el sistema las agencias que serán parte del proyecto.
Actor(es)	Administrador, Asistente de gerencia.
Precondiciones	El administrador o la Asistente de gerencia deben estar registrados y logueado previamente, debe estar ingresada la ciudad en la que se encuentra la agencia.
Flujo normal	<ol style="list-style-type: none"> 1. El administrador o la asistente de gerencia ingresan el nombre de la agencia. 2. El administrador o la asistente de gerencia selecciona la ciudad en donde se encuentra ubicada la agencia. 3. El administrador o la Asistente de gerencia ingresan la dirección de la agencia 4. El administrador o la Asistente de gerencia dan clic en el botón guardar. 5. El sistema valida la información ingresada. 6. El sistema genera mensaje de agencia registrada
Flujo alternativo	<ol style="list-style-type: none"> 5.1. El sistema muestra mensaje de error si no se ingresa el nombre de la agencia. 5.2. El sistema muestra un mensaje de agencia existente si la agencia ya fue ingresada.
Post condiciones	Se registra en la base de datos la nueva agencia ingresada.

Tabla 1.4 Descripción caso registrar agencias

- **Registrar horarios**



Diagrama 1.5 Caso de uso registro horarios

Detalle de caso de uso

Nombre	Registrar horarios
Autor(es)	Jimena Morales / Santiago Moya
Fecha	2 de diciembre de 2013.
Descripción	Registra en el sistema los horarios que se asignarán a los empleados
Actor(es)	Jefe de personal
Precondiciones	El jefe de personal debe estar registrado y logueado previamente.
Flujo normal	<ol style="list-style-type: none"> 1. El jefe de personal ingresa o asigna un nombre al horario. 2. El jefe de personal selecciona la hora de inicio de la jornada laboral. 3. El jefe de personal selecciona la hora de finalización del horario de la jornada laboral. 4. El jefe de personal da clic en el botón guardar. 5. El sistema valida la información ingresada. 6. El sistema genera un mensaje de horario registrado.
	<ol style="list-style-type: none"> 5.1. El sistema muestra mensaje de error si no se ingresa el nombre del horario. 5.2. El sistema muestra un mensaje de horario existente si el horario se encuentra anteriormente ingresado.
Post condiciones	Se registra en la base de datos el nuevo horario.

Tabla 1.5 Descripción caso de uso registrar horarios

- **Asignar Jornada Laboral**

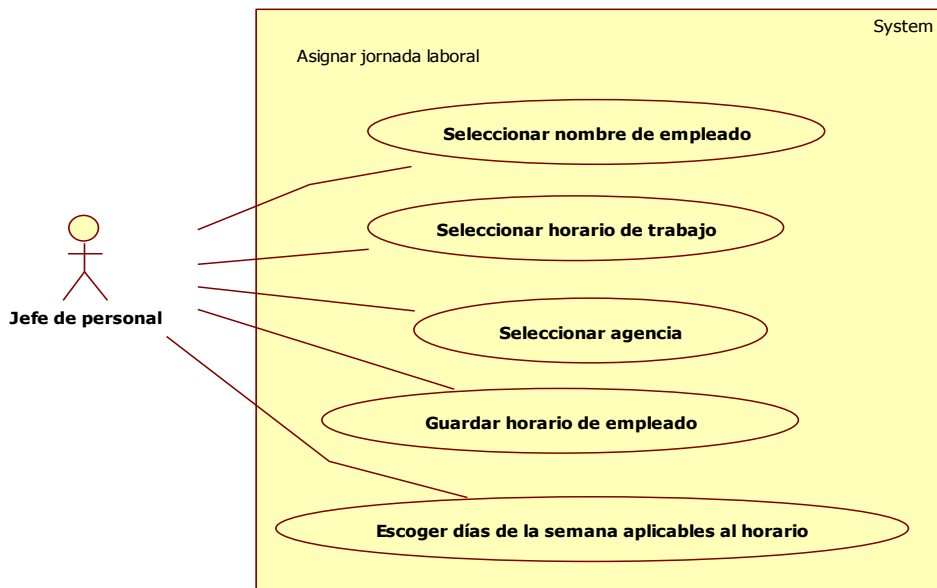


Diagrama 1.6 Caso de uso asignar Jornada Laboral

Detalle de caso de uso

Nombre	Asignar Jornada Laboral
Autor(es)	Jimena Morales / Santiago Moya
Fecha	2 de diciembre de 2013.
Descripción	Asigna los horarios a los empleados.
Actor(es)	Jefe de personal
Precondiciones	El jefe de personal debe estar registrado y logueado previamente. El empleado debe estar ingresado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El jefe de personal selecciona el nombre del empleado. 2. El jefe de personal selecciona el horario. 3. El Jefe de personal selecciona la agencia. 4. El jefe de personal da clic en el botón Guardar 5. El sistema asigna el horario al empleado y muestra un mensaje de asignación OK. <p>5.1. El sistema muestra mensaje de error si el empleado, agencia u horario se repiten o está dentro del rango de un horario que ya fue asignado al empleado.</p>
Post condiciones	Se registra en la base de datos el horario asignado al empleado.

Tabla 1.6 Descripción caso de uso asignar jornada laboral

Diagramas de caso de uso de consultas

- Consultar horarios asignados.

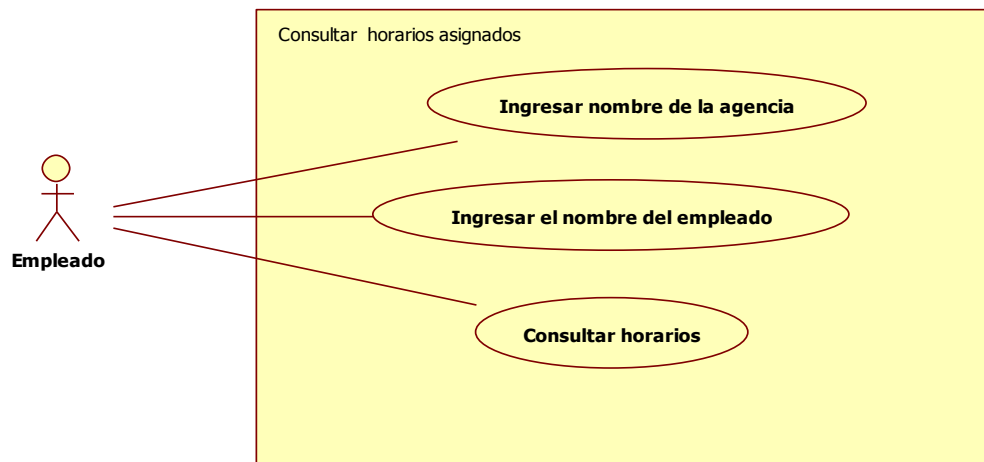


Diagrama 1.7 Caso de uso consultar horarios asignados

Detalle de caso de uso

Nombre	Consultar horarios asignados
Autor(es)	Jimena Morales / Santiago Moya
Fecha	4 de diciembre de 2013.
Descripción	Muestra los horarios y agencias en donde el empleado debe acudir para cumplir con su jornada laboral.
Actor(es)	Empleado.
Precondiciones	El Empleado debe estar registrado y logueado previamente. Deben estar creados horarios en el sistema Debe estar ingresada la distribución de horarios en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El empleado ingresa el nombre de la agencia que desea consultar o ingresar la palabra "Todos" para que se generalice la consulta a todas las agencias. 2. El empleado ingresa su nombre. 3. El empleado da clic en el botón buscar
Flujo alternativo	15. Si no se ingresa el nombre de la agencia o el nombre del empleado el sistema muestra un mensaje de campos requeridos.
Post condiciones	Se muestra los días, hora de entrada y salida del empleado que está realizando la consulta.

Tabla 1.7 Descripción caso de uso consultar horarios asignados

- **Consultar Asistencia**

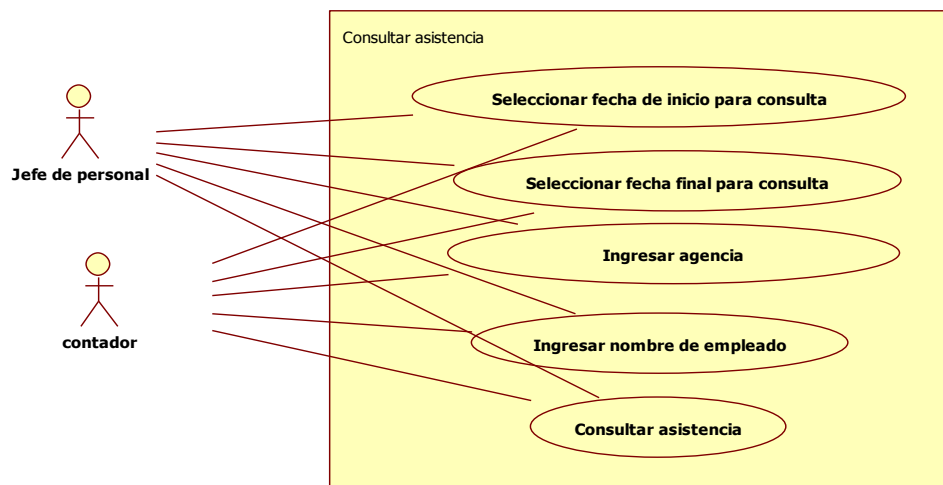


Diagrama 1.8 Caso de uso consultar Asistencia

Detalle de caso de uso

Nombre	Consultar Asistencia
Autor(es)	Jimena Morales / Santiago Moya
Fecha	4 de diciembre de 2013.
Descripción	Muestra la asistencia por agencia o por empleado que se seleccione en la consulta.
Actor(es)	Jefe de personal, contador
Precondiciones	El Jefe de personal o contador deben estar registrados y logueados previamente.
Flujo normal	<ol style="list-style-type: none">1. El Jefe de personal, contador selecciona la fecha de inicio de la consulta.2. El Jefe de personal, contador selecciona la fecha de fin de la consulta.3. El Jefe de personal, contador ingresa la agencia que desea consultar la asistencia, si requiere realizar consulta de todas las agencias ingresa la palabra "Todos".4. El Jefe de personal, contador ingresa el nombre del empleado a consultar, si requiere realizar consulta de todos los empleados ingresa la palabra "Todos", si requiere más de un empleado puede ingresar los nombres de los empleados.5. El Jefe de personal, contador da clic en el botón buscar
Flujo alternativo	<ol style="list-style-type: none">5.1 Si los rangos de las fechas no son correspondientes el sistema muestra un mensaje de error.5.2 Si no se selecciona el nombre de la agencia o del empleado el sistema muestra mensaje de campos requeridos.
Post condiciones	Muestra la asistencia del empleado en la agencia(s) que fueron ingresadas.

Tabla 1.8 Descripción caso de uso para consultar Asistencia.

Anexo 10. Diagrama de actividades y secuencias aplicación de escritorio

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Diagramas actividades y secuencias aplicación de escritorio

Versión [5.0]

Historial de revisiones

Fecha	Versión	Descripción	Autor
25/08/2014	1.0	Actividades y secuencia reconocer usuario	Santiago Moya P.
01/09/2014	2.0	Actividades y secuencia registrar usuario	Santiago Moya P.
10/09/2014	3.0	Actividades y secuencia registro de horario	Santiago Moya P.
15/09/2014	4.0	Revisión actividades y secuencia registro de horario	Santiago Moya P.
29/09/2014	5.0	Revisión Final Actividades y secuencia	Santiago Moya P.

1. Diagrama de actividades

Autenticar usuario sistema de escritorio

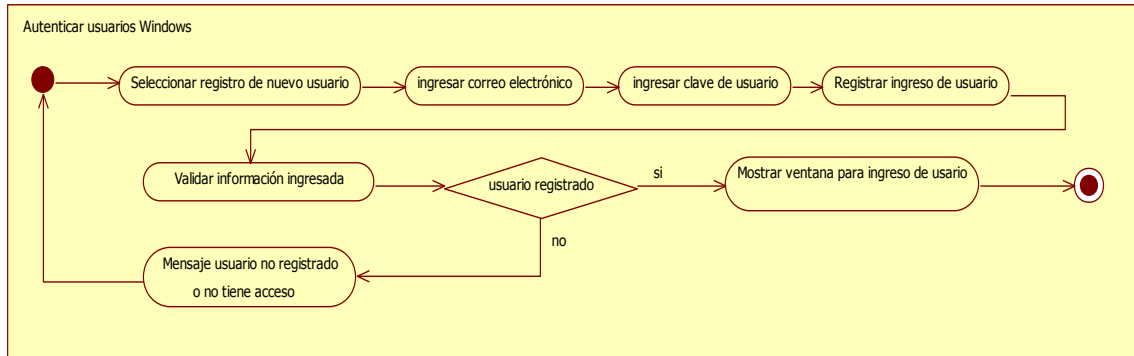


Diagrama 1.1 Actividades Autenticar usuario escritorio

Registrar y reconocer usuarios

- Registrar empleados sistema de escritorio

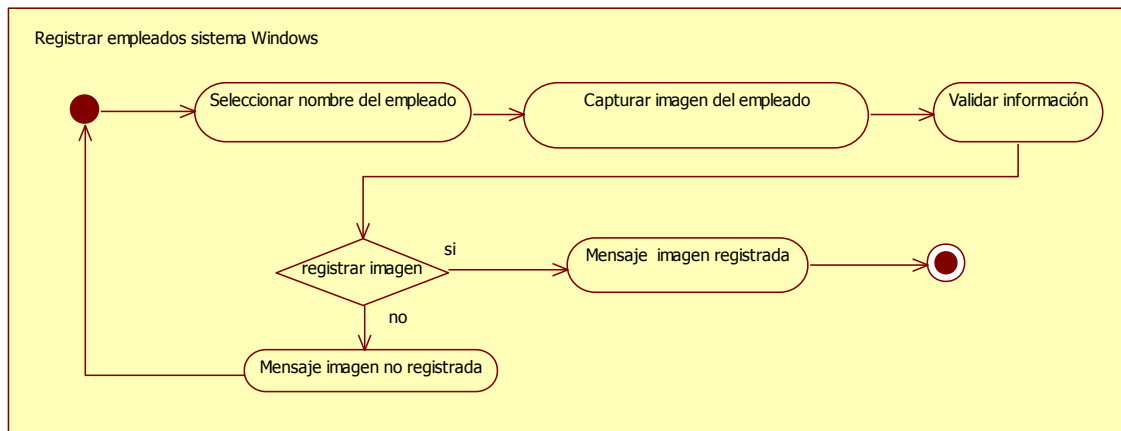


Diagrama 1.2 Actividades registrar empleados sistema de escritorio

- **Reconocer usuario usando sistema de escritorio**

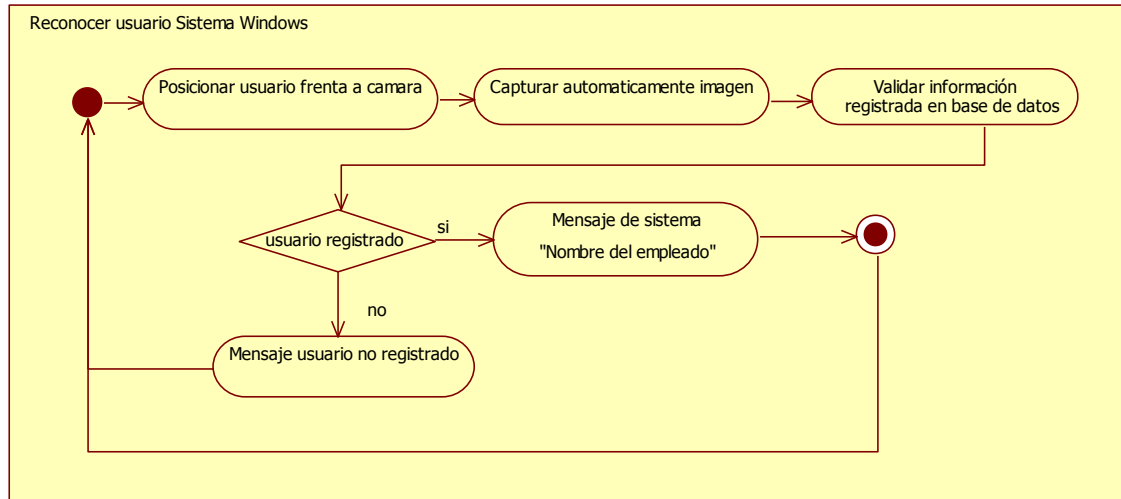


Diagrama 1.3 Actividades reconocer usuario sistema de escritorio

Registro de horario

- **Registrar ingreso – salida de jornada laboral sistema de escritorio**

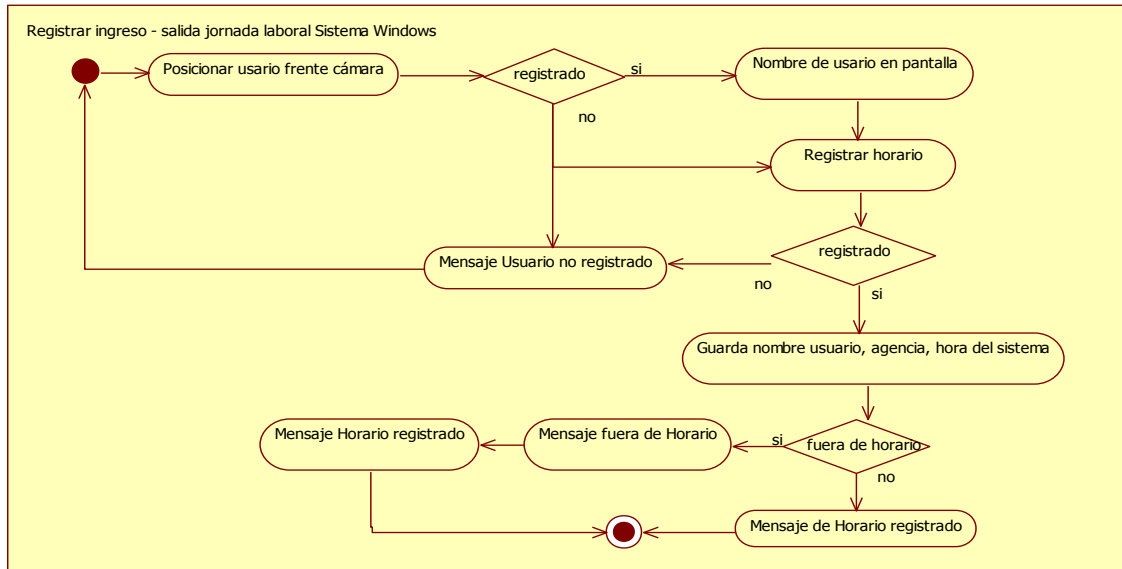


Diagrama 1.4 Actividades registrar ingreso – salida de jornada laboral sistema de escritorio

2. Diagrama de secuencias

Login de usuario sistema de escritorio

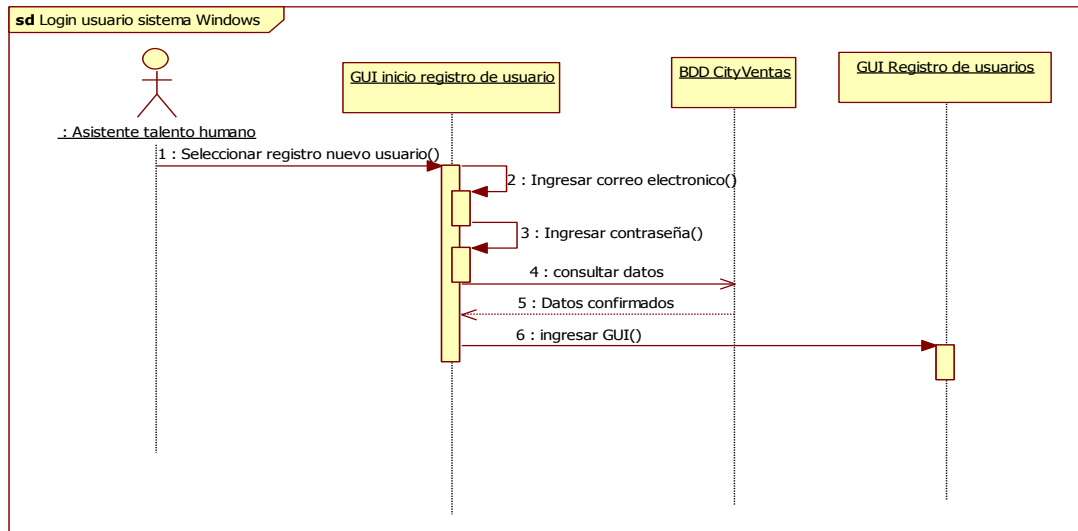


Diagrama 1.5 Secuencia Autenticar usuario escritorio

Reconocimiento de rostro

- Registrar empleados sistema de escritorio

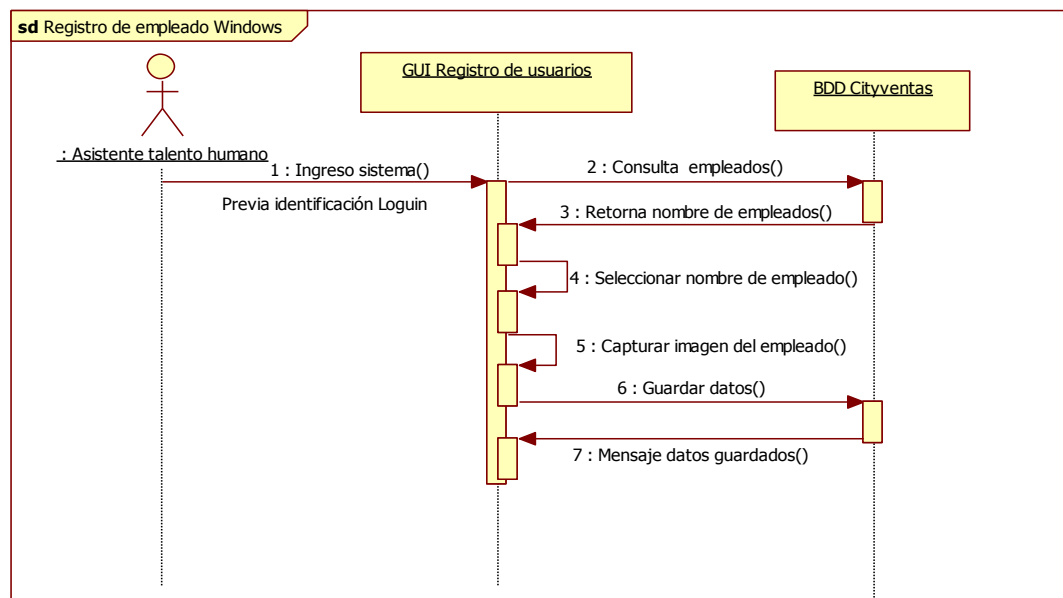


Diagrama 1.6 Secuencia registrar empleados sistema de escritorio

- **Reconocer usuarios sistema de escritorio**

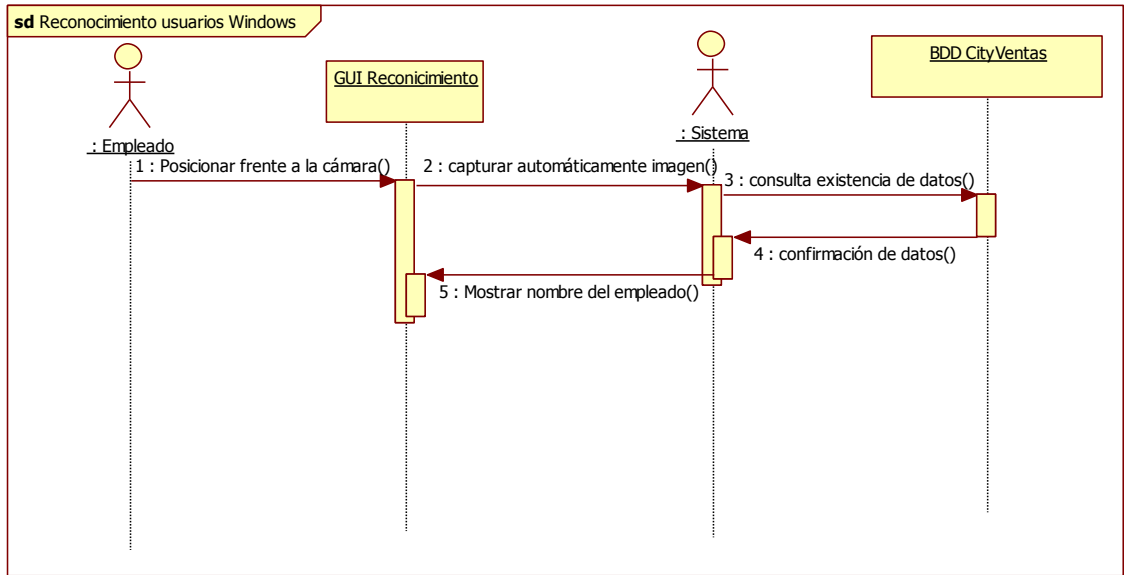


Diagrama 1.7 Secuencia reconocer usuarios sistema de escritorio

Registro de horario

- **Registrar ingreso - salida jornada laboral**

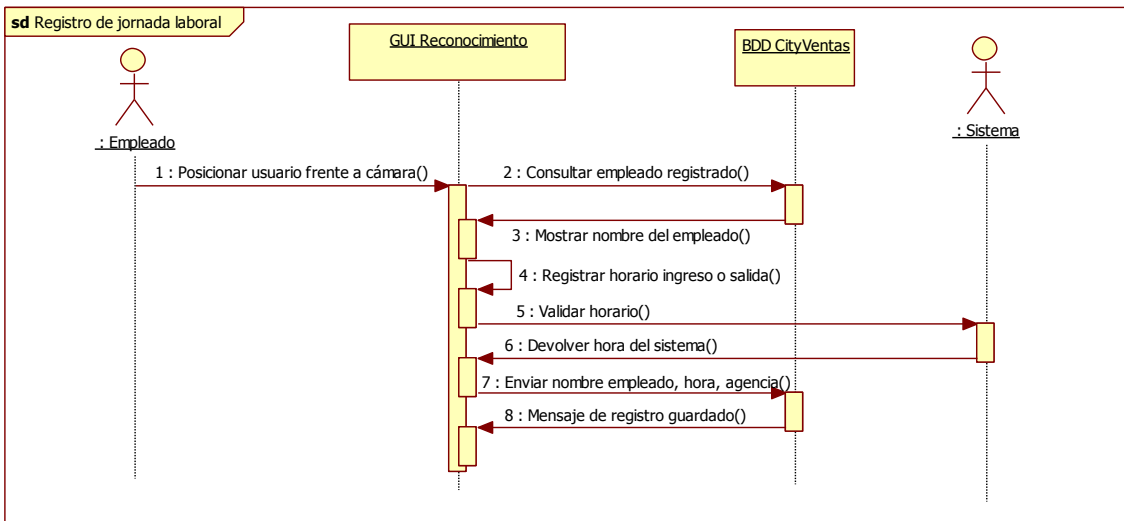


Diagrama 1.8 Secuencia registrar ingreso - salida jornada laboral

Anexo 11. Diagrama de navegación e interface para aplicación de escritorio

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Diagramas Navegación e Interface aplicación de escritorio

Versión [5.0]

Historial de revisiones

Fecha	Versión	Descripción	Autor
25/08/2014	1.0	Navegación e interfaces reconocer usuario	Santiago Moya P.
01/09/2014	2.0	Navegación e interfaces registrar usuario	Santiago Moya P.
10/09/2014	3.0	Navegación e interfaces registro de horario	Santiago Moya P.
15/09/2014	4.0	Revisión Navegación e interfaces registro de horario	Santiago Moya P.
29/09/2014	5.0	Revisión Final Navegación e interfaces	Santiago Moya P.

1. Diagrama Navegación.

Navegación Principal.

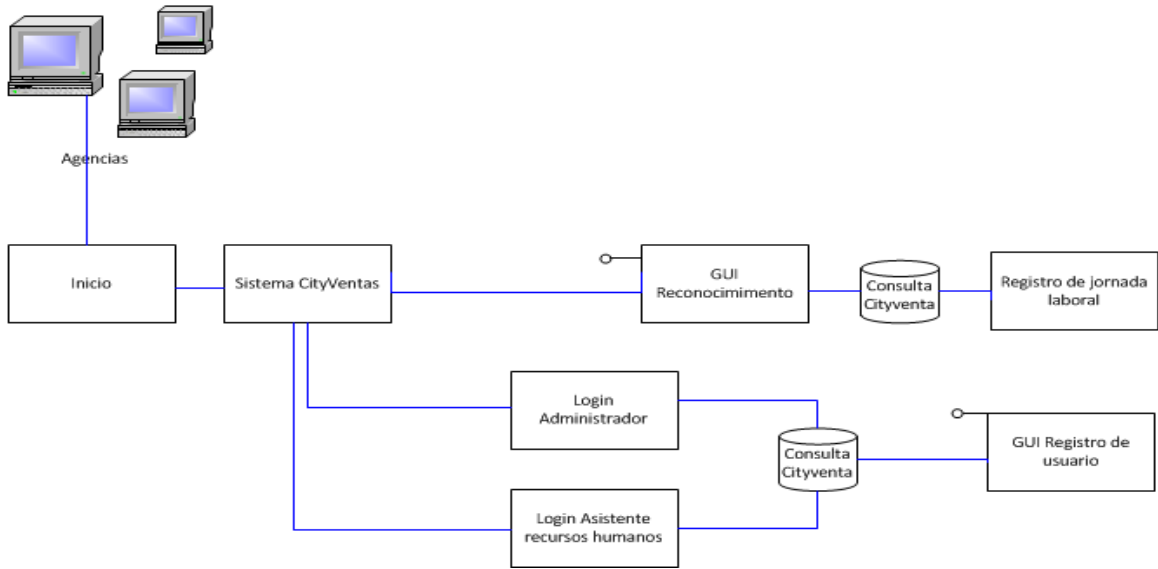


Diagrama 1.1 Navegación principal sistema de escritorio

Reconocimiento de Usuario.

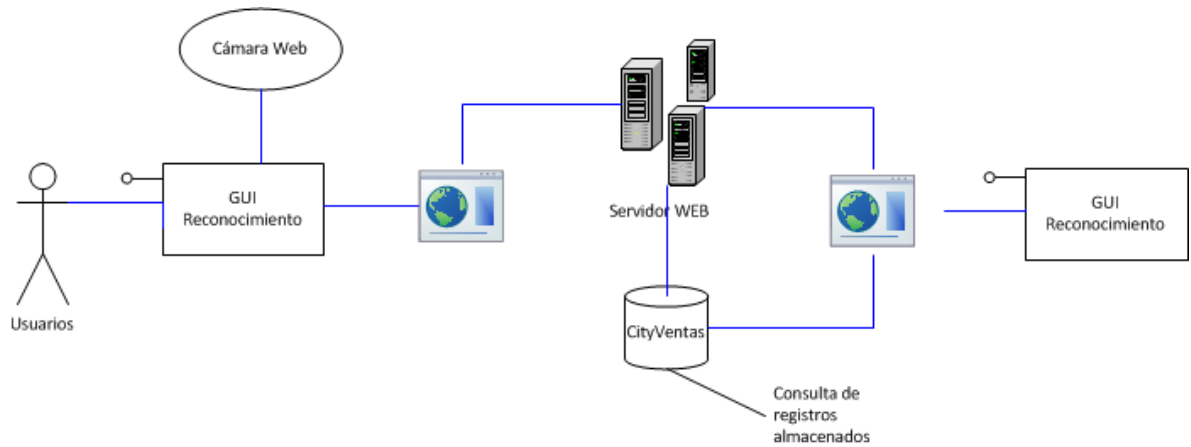


Diagrama 1.2 Navegación reconocimiento de Usuario

Registrar Usuarios.

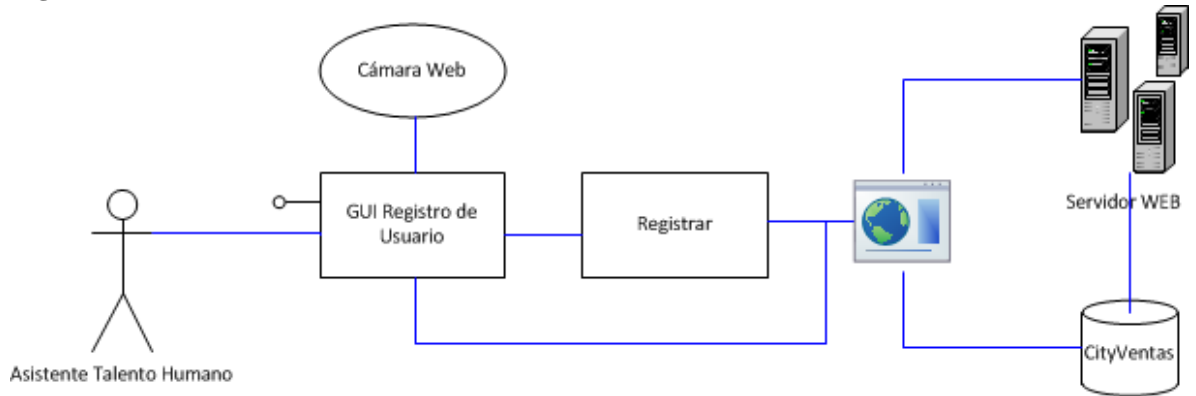


Diagrama 1.3 Navegación registrar usuarios

Registrar jornada Laboral.

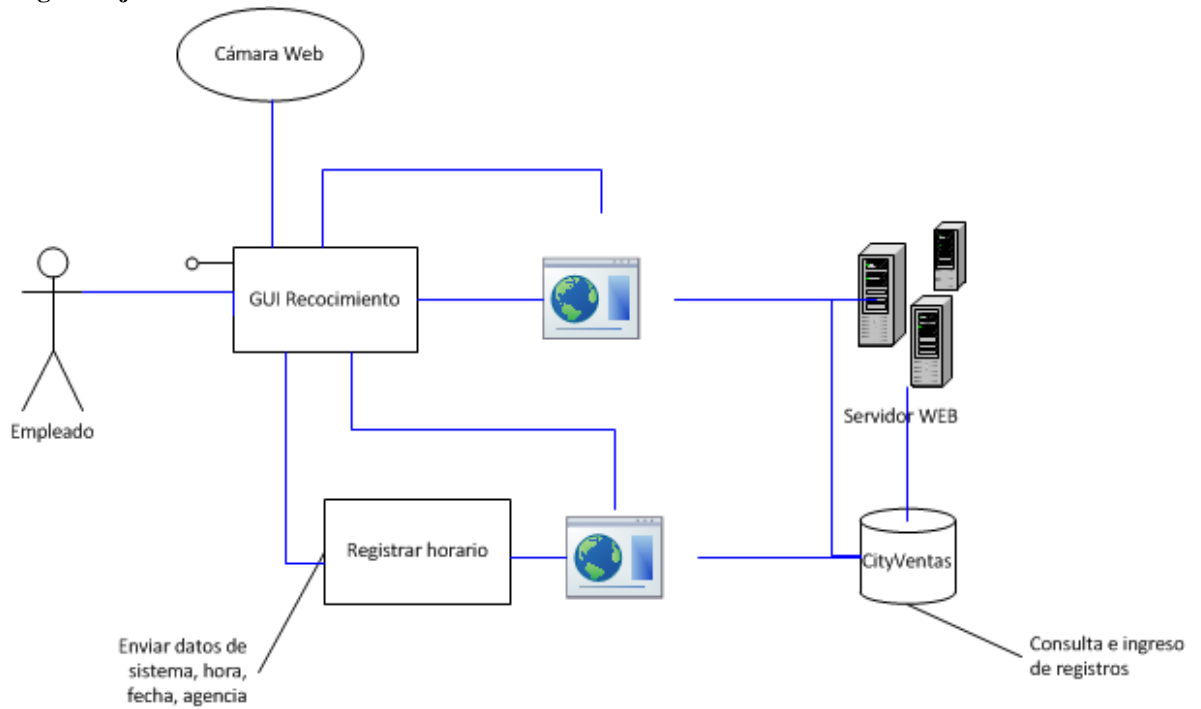
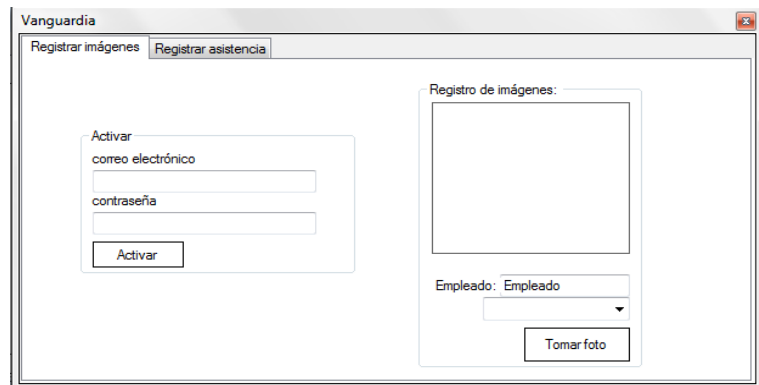


Diagrama 1.4 Navegación registrar jornada laboral

2. Diagrama de Interfaces

Autenticar Usuario

Se solicitará las credenciales del usuario para poder ingresar nuevos usuarios al sistema de escritorio, estos serán comparados con el registro de la base de datos de la aplicación web mediante un web service que se utilizará como enlace de los dos sistemas.




The screenshot shows a window titled 'Vanguardia' with two tabs: 'Registrar imágenes' (selected) and 'Registrar asistencia'. The 'Registrar imágenes' tab contains two main sections. On the left, under the heading 'Activar', there are two input fields labeled 'correo electrónico' and 'contraseña', followed by an 'Activar' button. On the right, under the heading 'Registro de imágenes:', there is a large empty rectangular box. Below this box is a dropdown menu labeled 'Empleado:' with 'Empleado' selected, and a 'Tomar foto' button.

Diagrama 1.5 Interface autenticar usuario

Registrar y reconocer usuario

- **Registrar Usuario**

Será incluido la imagen del usuario si previamente se encuentra registrado en la aplicación web.



The screenshot shows a window titled 'Detección de Rostro CityVenta'. It features a large video feed on the left showing a person's face with a red bounding box and the name 'Santiago Moya' overlaid. To the right of the video feed is a section titled 'Ingreso de rostro de Usuario:' containing a smaller image of the same person. Below this is an 'Ingreso nombre' field with 'Santiago Moya' entered and an 'Agregar usuario' button. Further right, under the heading 'Usuario detectado', there is a message: 'Nombre de la persona reconocida Santiago Moya,'.

Diagrama 1.6 Interface registrar usuario

- **Reconocer Usuario**

El usuario se posicionará frente a la cámara y será reconocido si previamente está registrado.



Diagrama 1.7 Interface reconocimiento de usuario

Registro de horarios

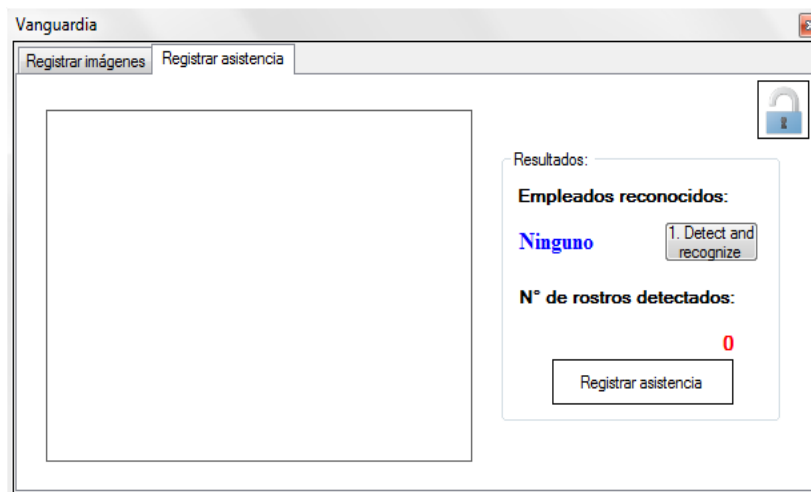


Diagrama 1.8 Interface registro de horarios

Anexo 12. Diagrama de actividades y secuencia aplicación web

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Diagramas actividades y secuencias aplicación web

Versión [5.0]

Historial de revisiones

Fecha	Versión	Descripción	Autor
25/08/2014	1.0	Actividades y secuencia para parametrización	Santiago Moya P.
01/09/2014	2.0	Actividades y secuencia para administración de usuario	Santiago Moya P.
10/09/2014	3.0	Actividades y secuencia consultas	Santiago Moya P.
15/09/2014	4.0	Revisión actividades y secuencia administración de usuario	Santiago Moya P.
29/09/2014	5.0	Revisión Final Actividades y secuencia	Santiago Moya P.

1. Diagrama de actividades

Autenticar Usuarios.

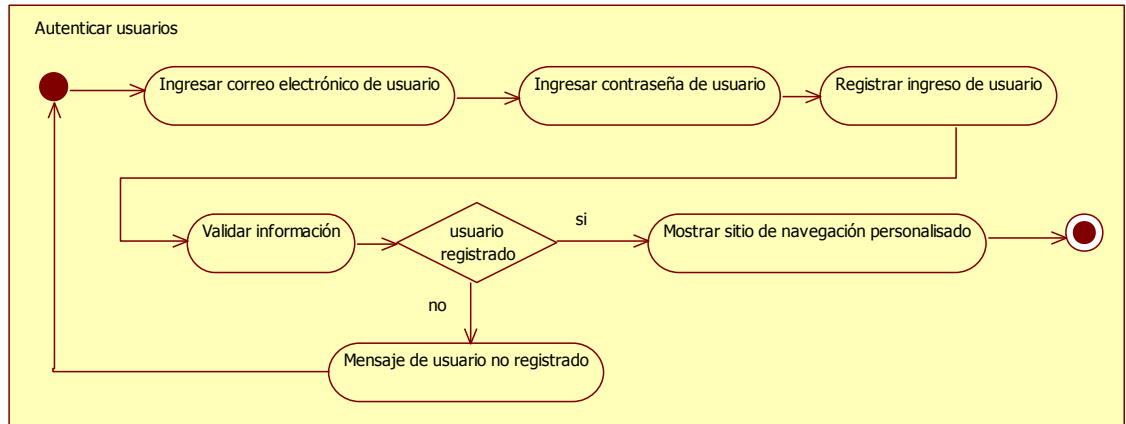


Diagrama 1.1 Actividades Autenticar usuarios

Administración de usuario

- Registrar perfiles de usuario.

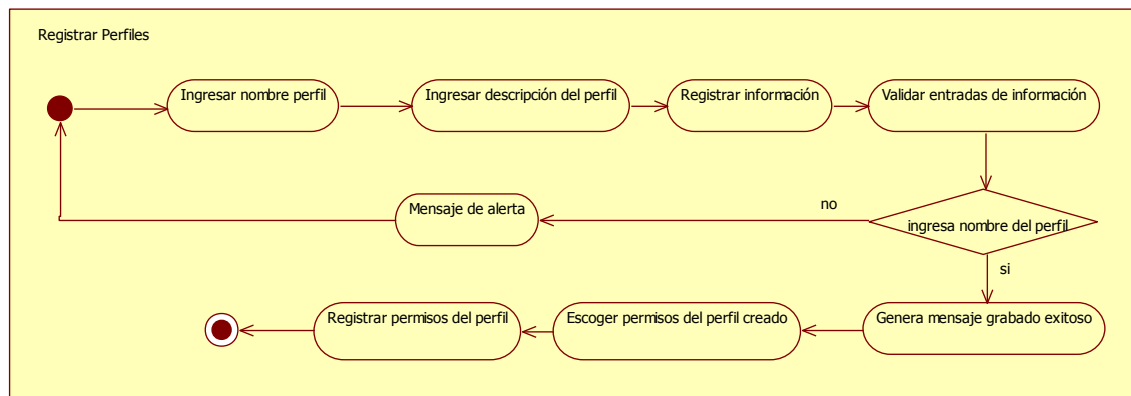


Diagrama 1.2 Actividades Registrar perfiles de usuario

- **Registrar Usuario/Empleado**

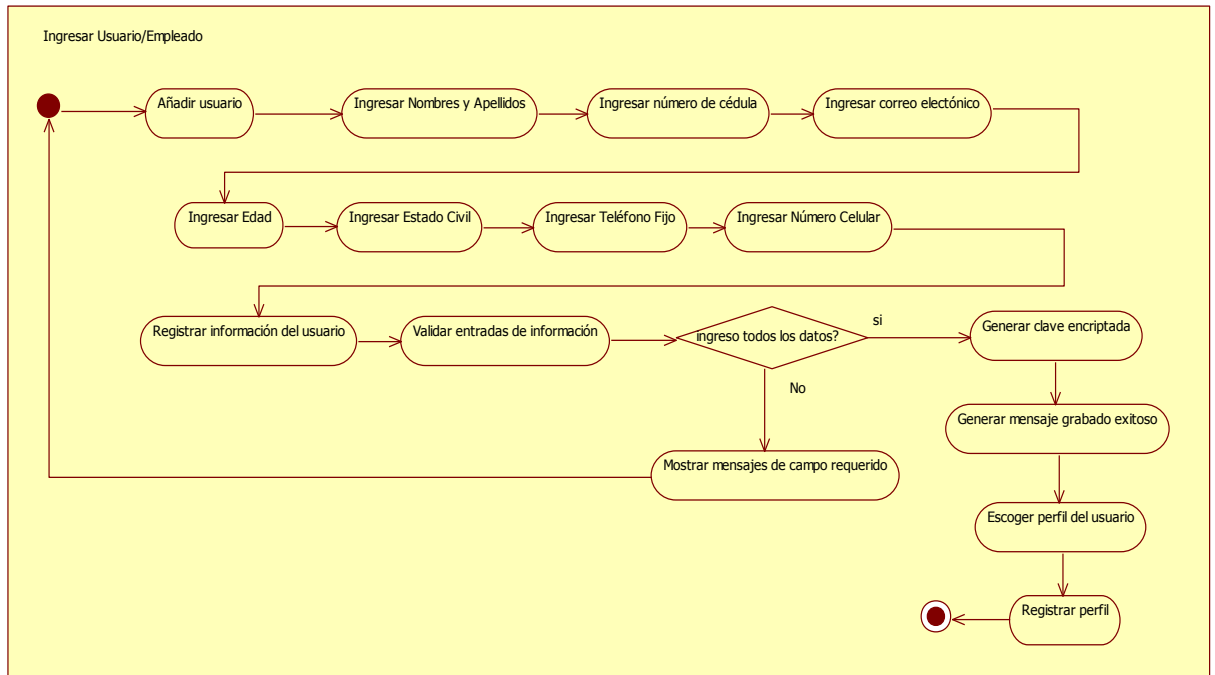


Diagrama 1.3 Actividades registrar usuarios

Parametrización

- **Registrar agencias**

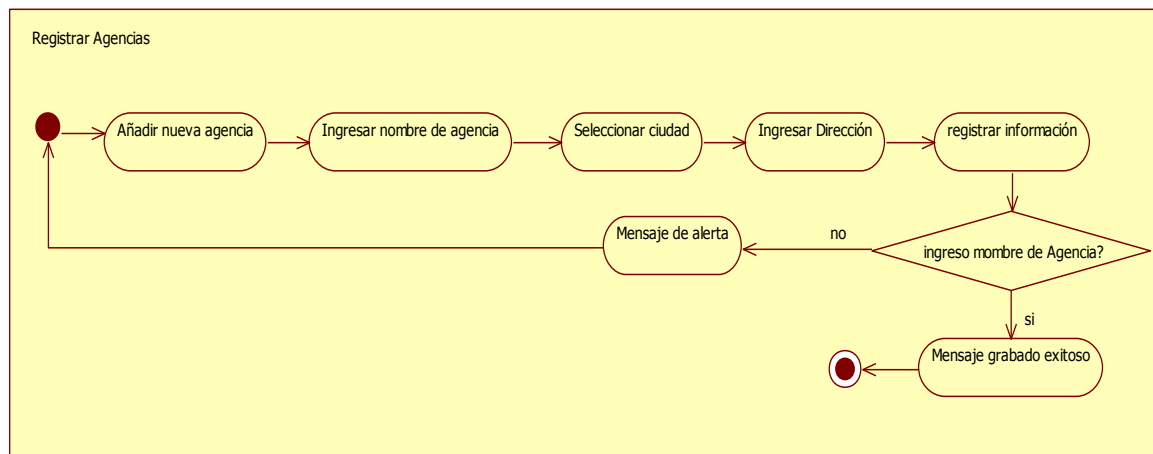


Diagrama 1.4 Actividades registrar agencias

- **Registrar horarios**

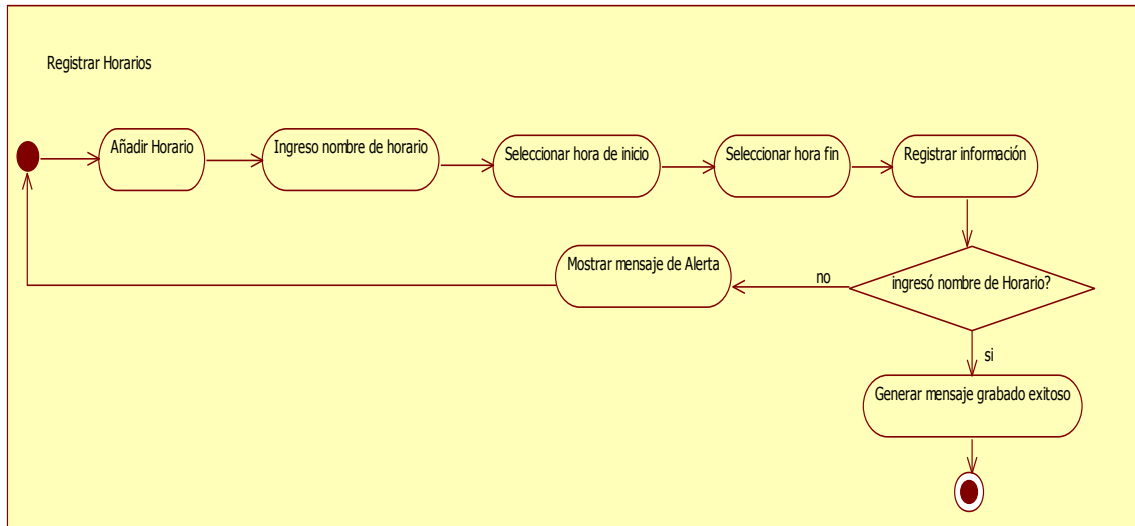


Diagrama 1.5 Actividades registrar horarios

- **Asignar jornada laboral**

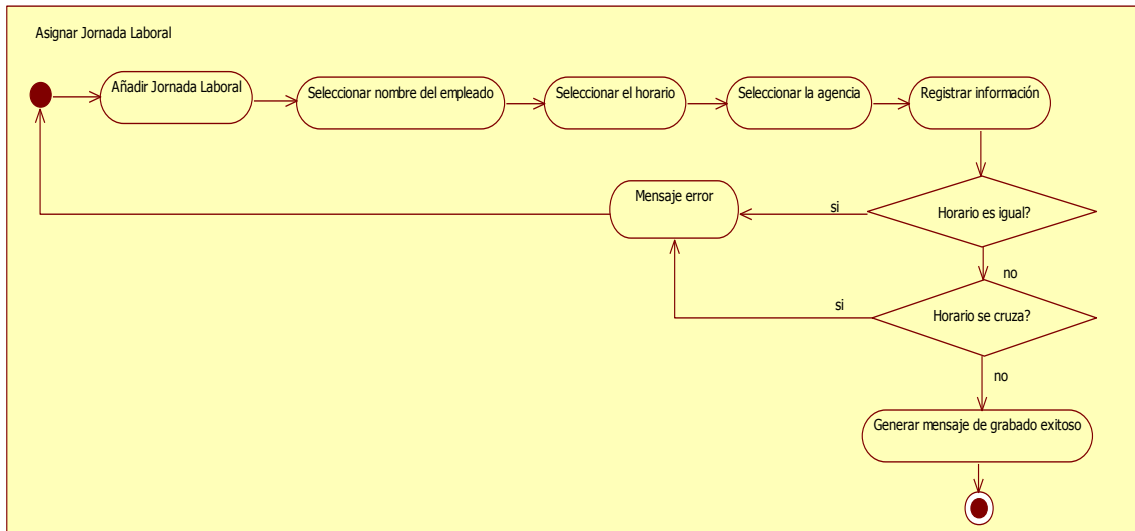


Diagrama 1.6 Actividades asignar jornada laboral

Consultas

- **Consultar horarios asignados.**

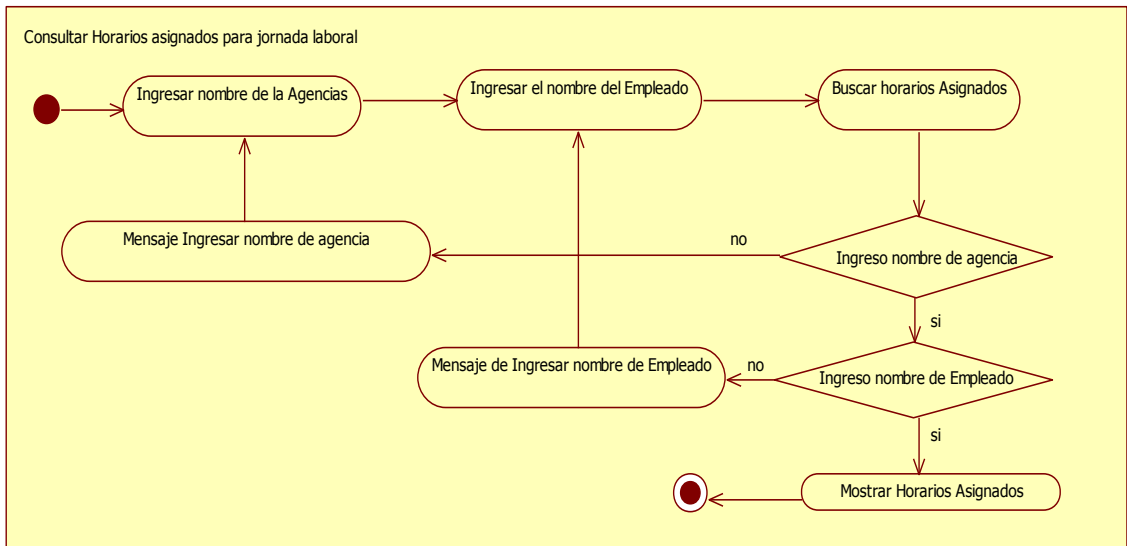


Diagrama 1.7 Actividad consultar horarios asignados para jornada laboral

- **Consultar Asistencias**

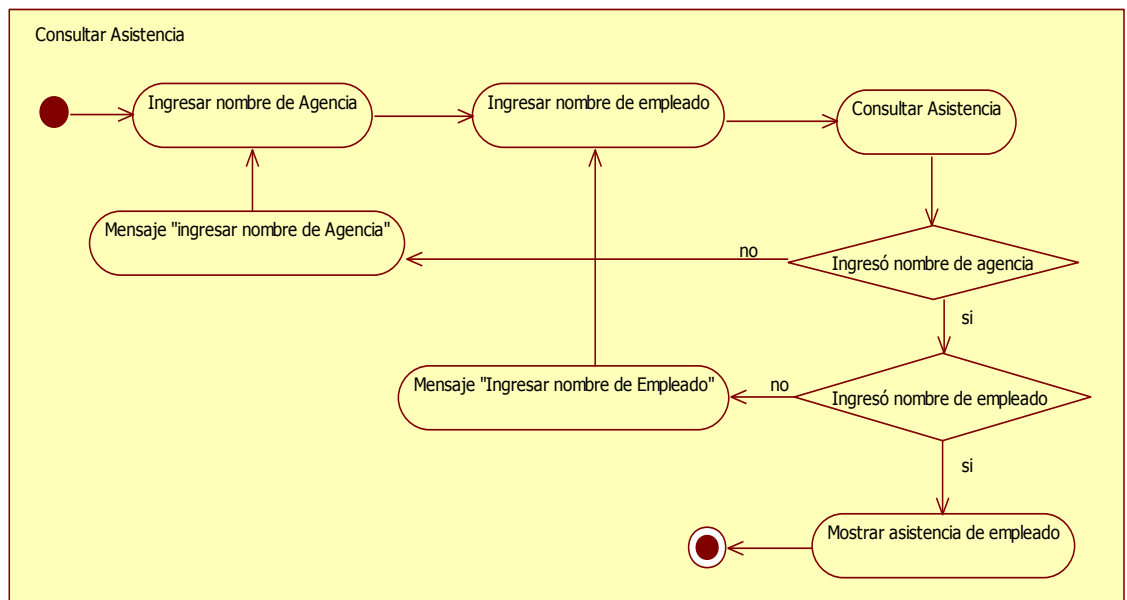


Diagrama 1.8 Actividad consultar Asistencia

2. Diagrama de Secuencias

Login de Usuario

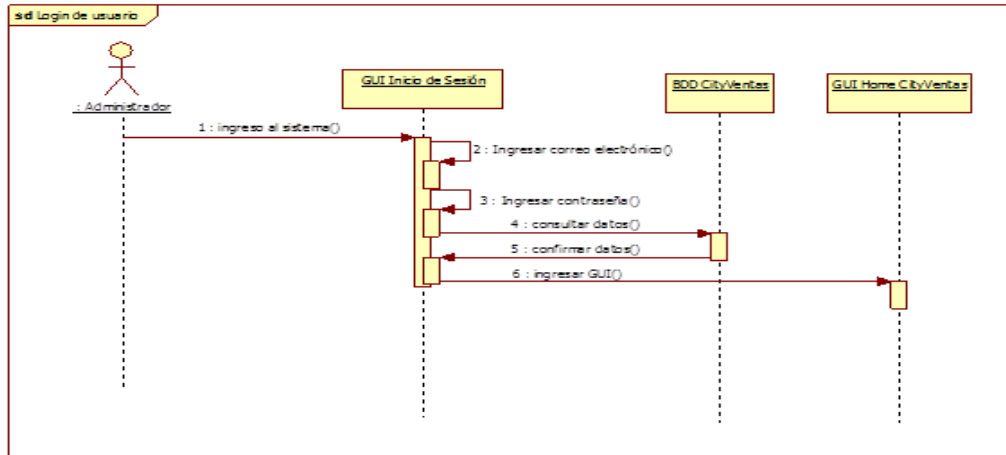


Diagrama 2.1 Secuencia de login de usuario

Administración de usuario

- Registro de perfiles de usuario

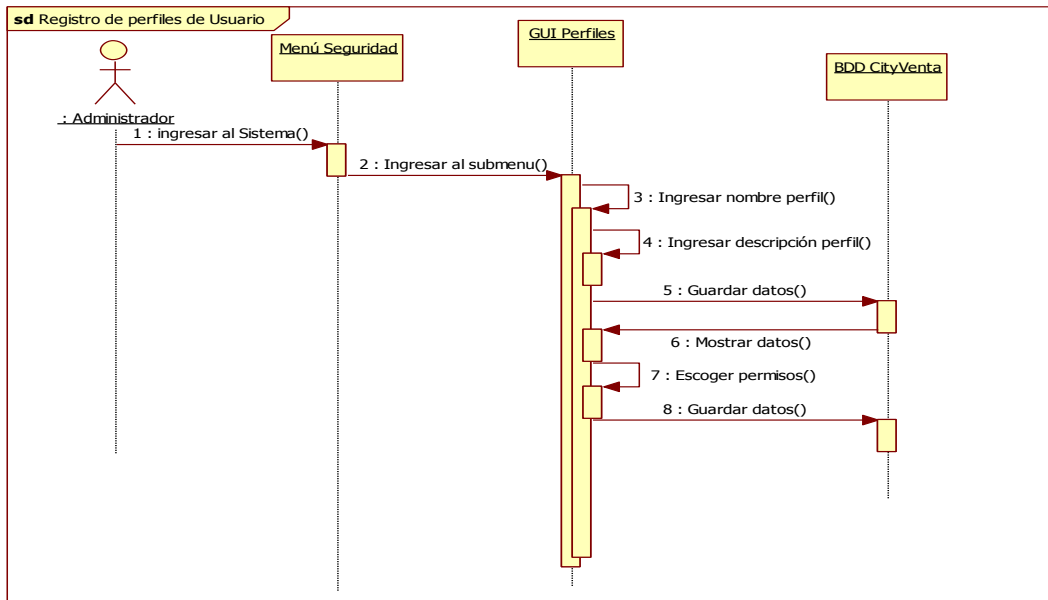


Diagrama 2.2 Secuencia registro de perfiles de usuario

- **Registro de usuario/empleado.**

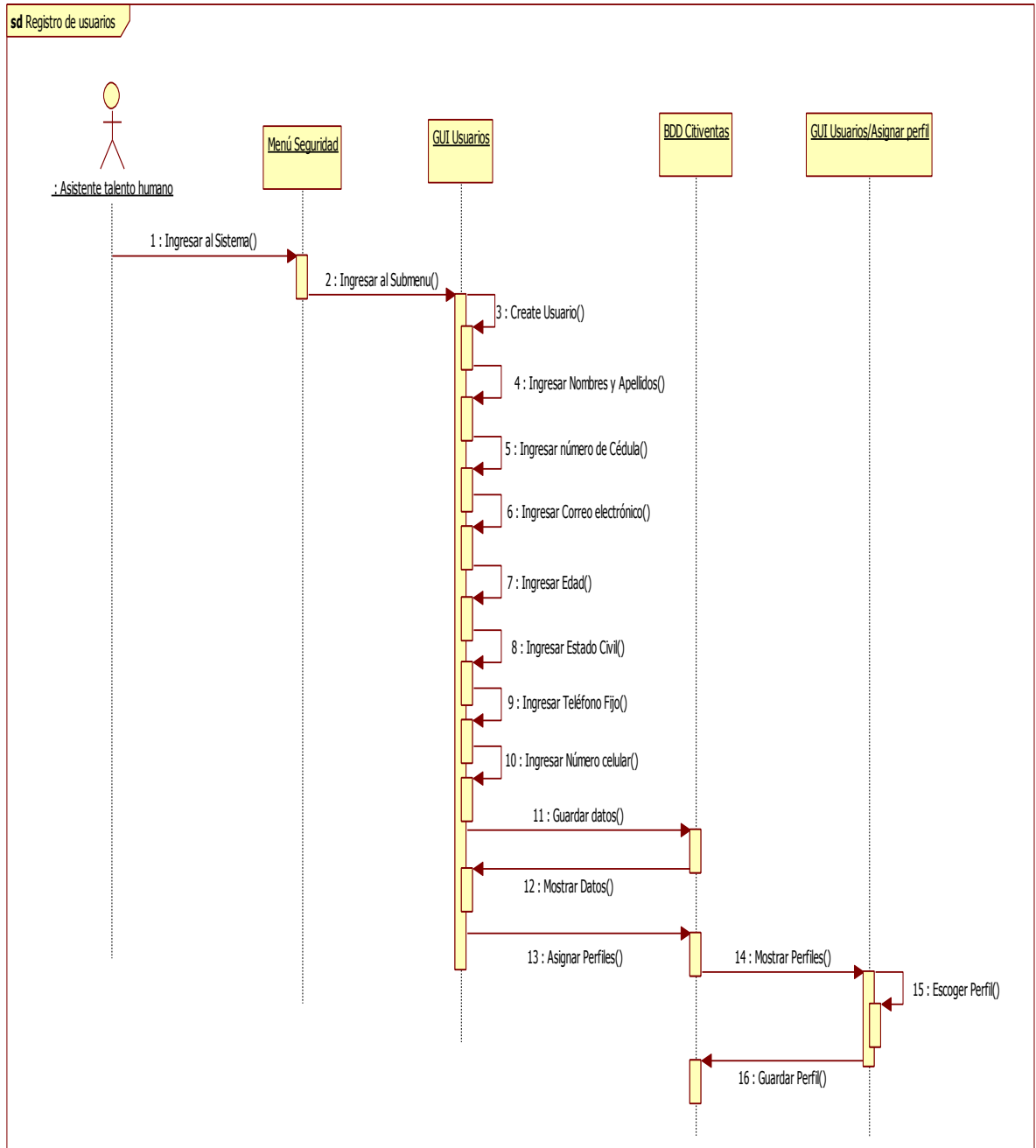


Diagrama 2.3 Secuencia registro de Usuario

Parametrización

- Registro de agencias

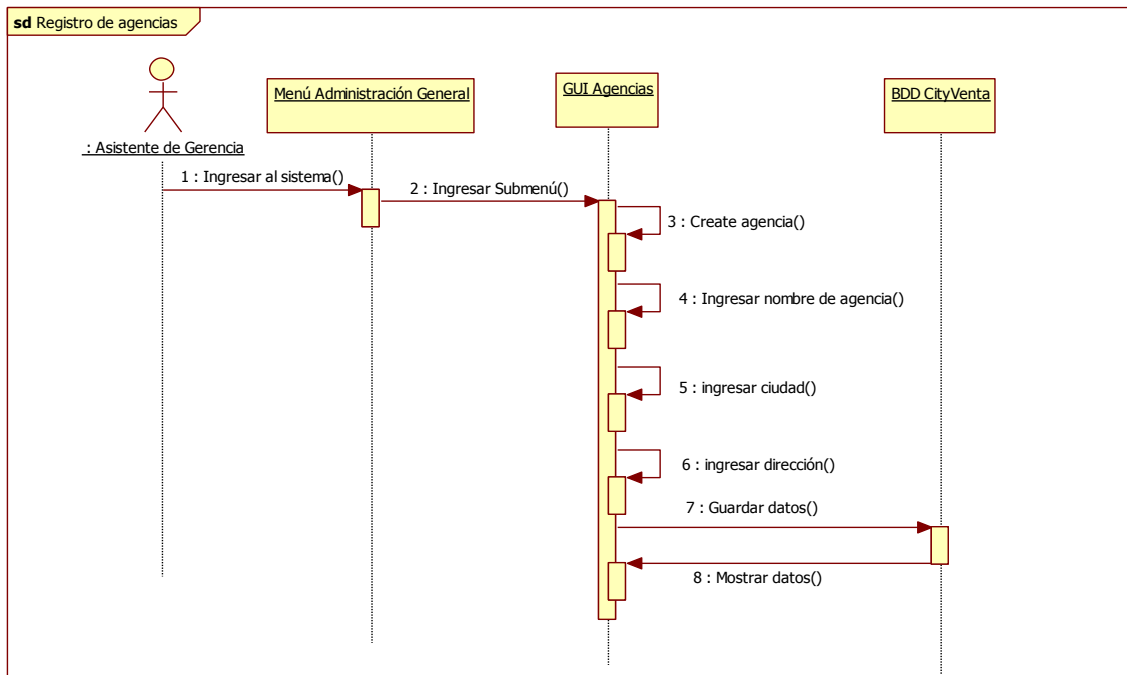


Diagrama 2.4 Secuencia registro de agencia

- Registro de horarios

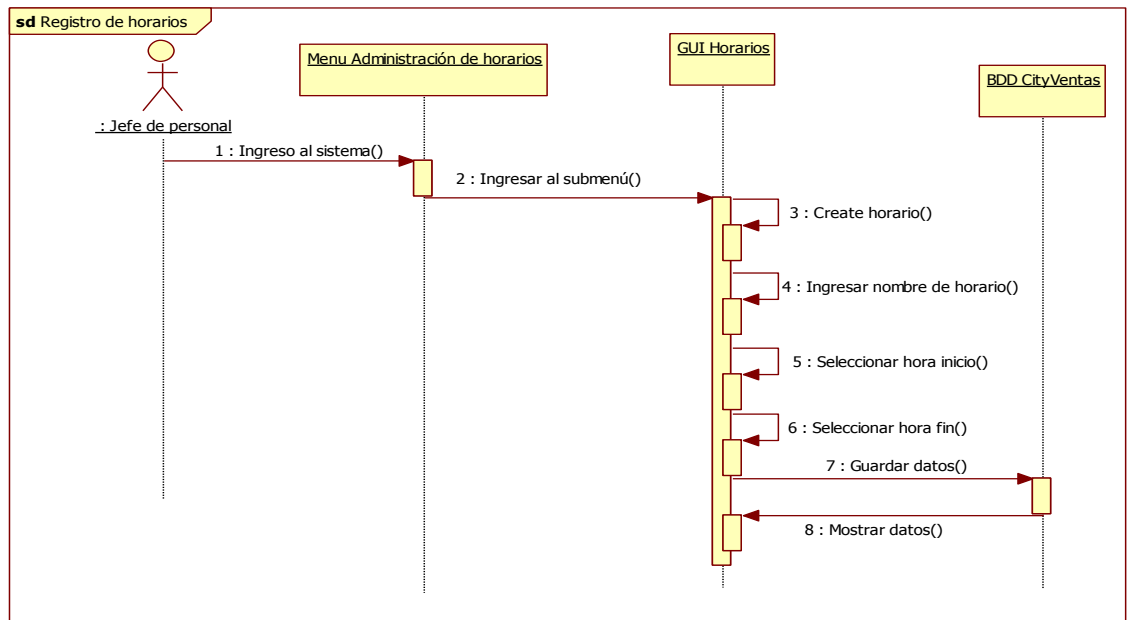


Diagrama 2.5 Secuencia de registro de horarios

- **Asignación de jornada laboral**

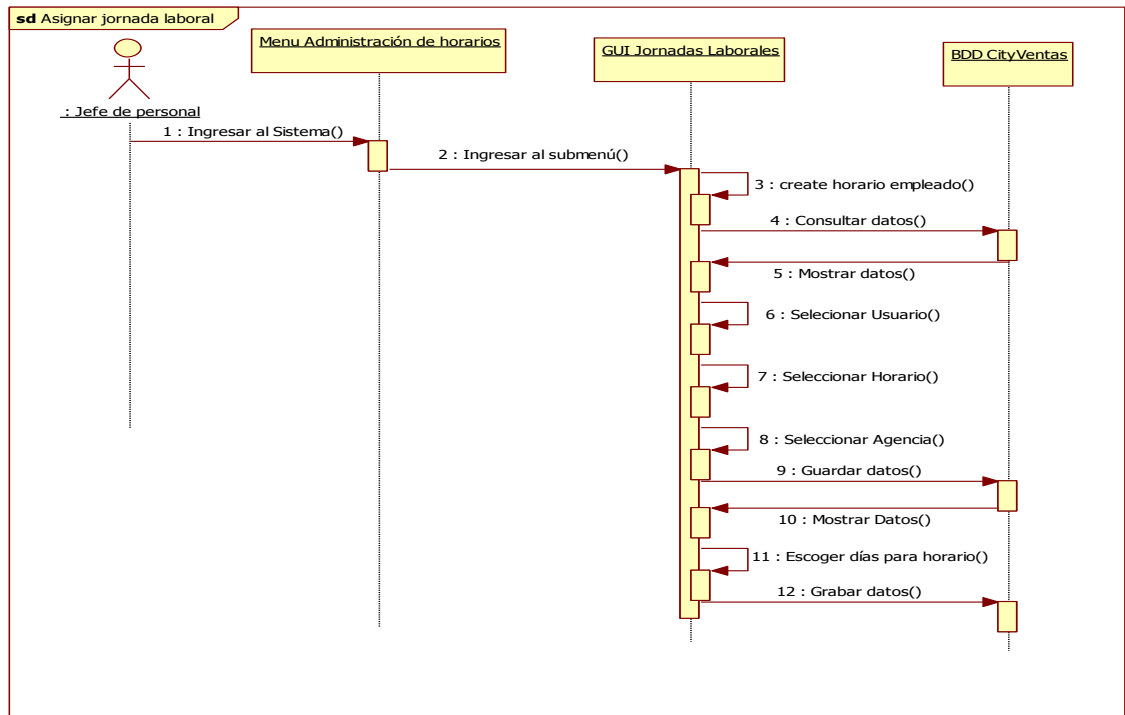


Diagrama 2.6 Secuencia Asignar jornada laboral

Consultas

- **Consultar horarios asignados**

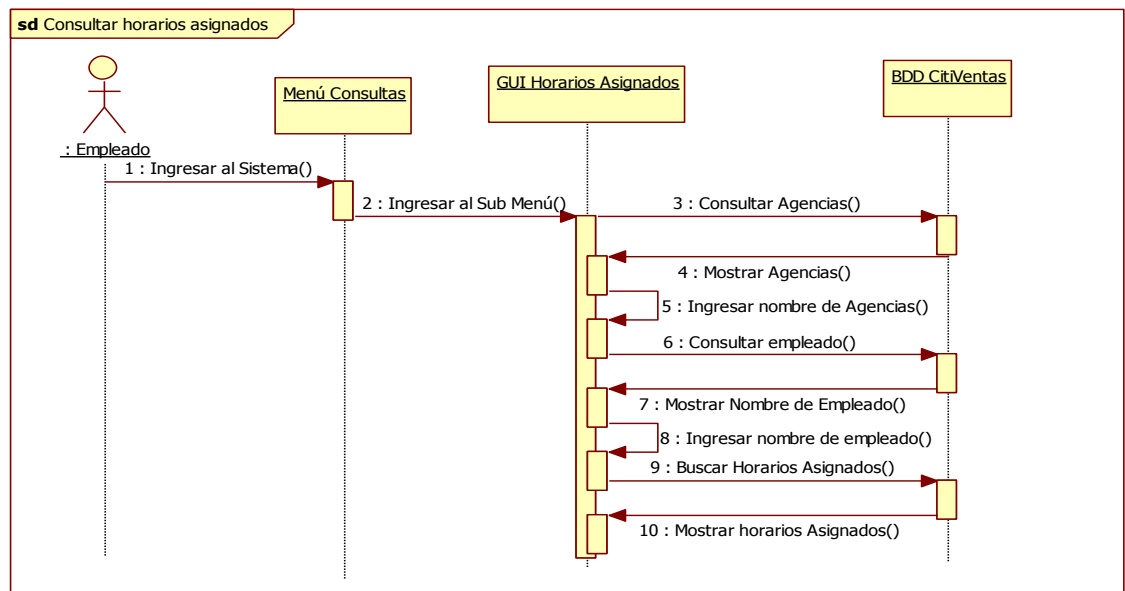


Diagrama 2.7 Consultar Horarios asignados

- Consultar asistencias

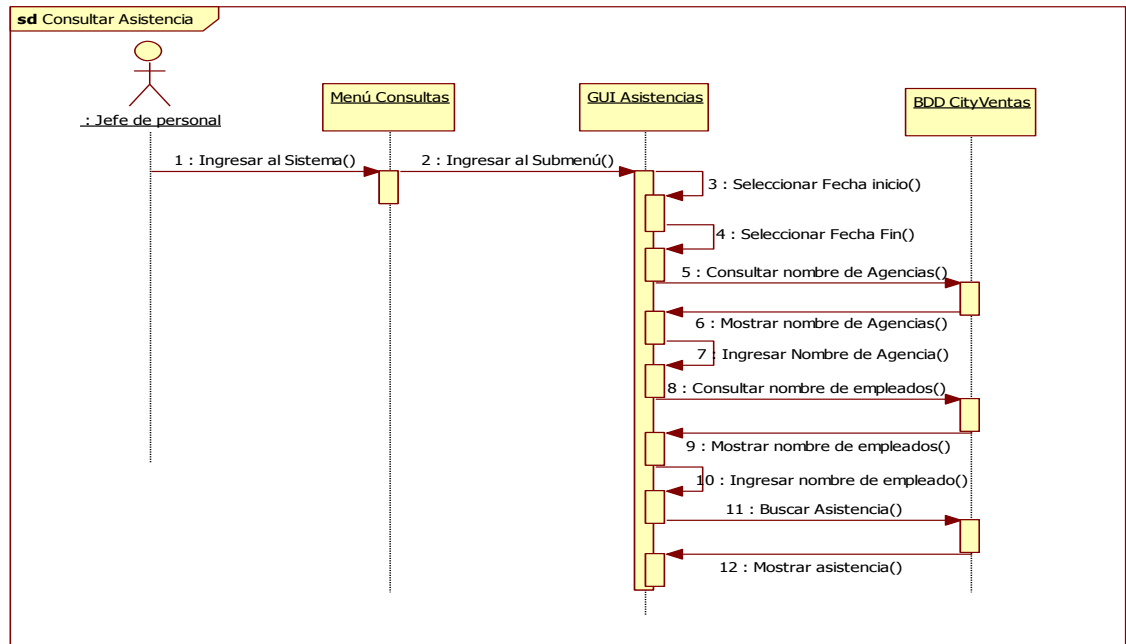


Diagrama 2.8 Secuencia consultar asistencia

Anexo 13. Diagramas de navegación e interface aplicación web

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Diagramas Navegación e Interface aplicación web

Versión [5.0]

Historial de revisiones

Fecha	Versión	Descripción	Autor
25/08/2014	1.0	Navegación e Interfaces para parametrización	Santiago Moya P.
01/09/2014	2.0	Navegación e Interfaces para administración de usuario	Santiago Moya P.
10/09/2014	3.0	Navegación e Interfaces de consultas	Santiago Moya P.
15/09/2014	4.0	Revisión Navegación e Interfaces administración de usuario	Santiago Moya P.
29/09/2014	5.0	Revisión Final Navegación e interfaces	Santiago Moya P.

1. Diagrama Navegación.

Navegación Principal.

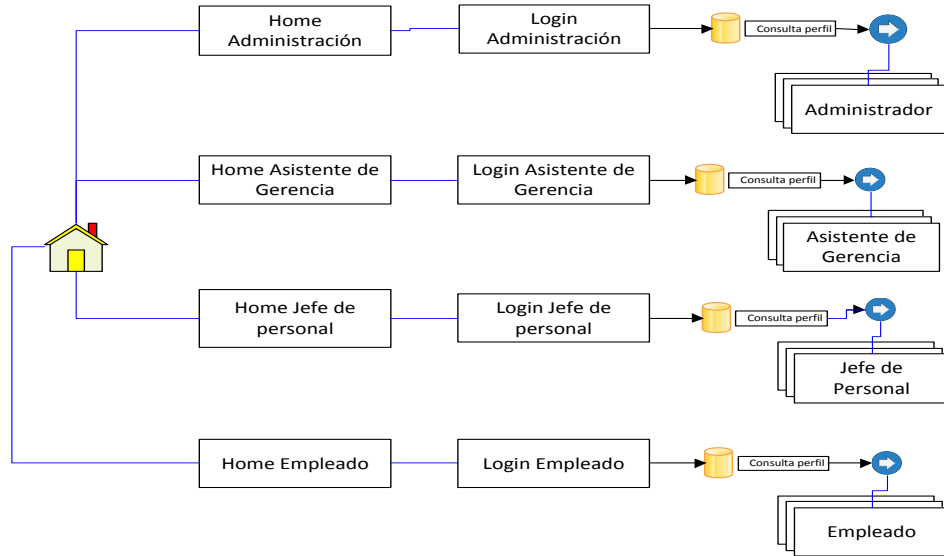


Diagrama 1.1 Navegación página inicio

Flujo de navegación para Administrador.

El administrador tiene acceso a todo el contenido del sitio Web por sus permisos para modificar, insertar y consultar.

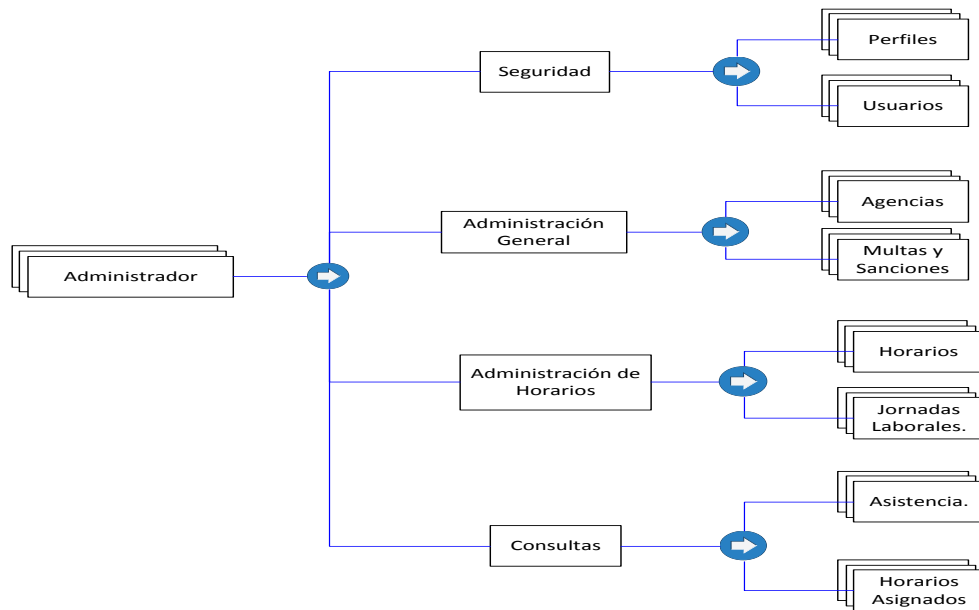


Diagrama 1.2 Navegación para Administrador

Navegación para Administración de perfiles.

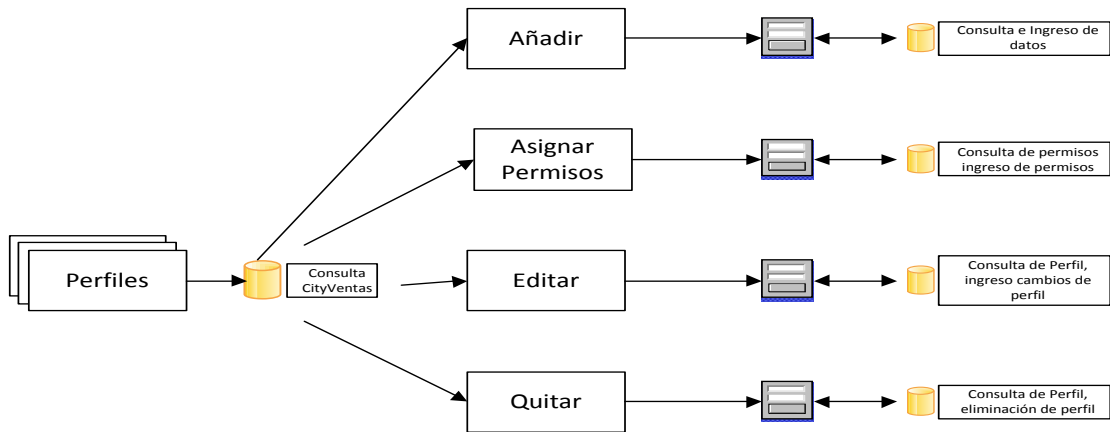


Diagrama 1.3 Navegación para administración de perfiles

Navegación para Administración de Usuarios

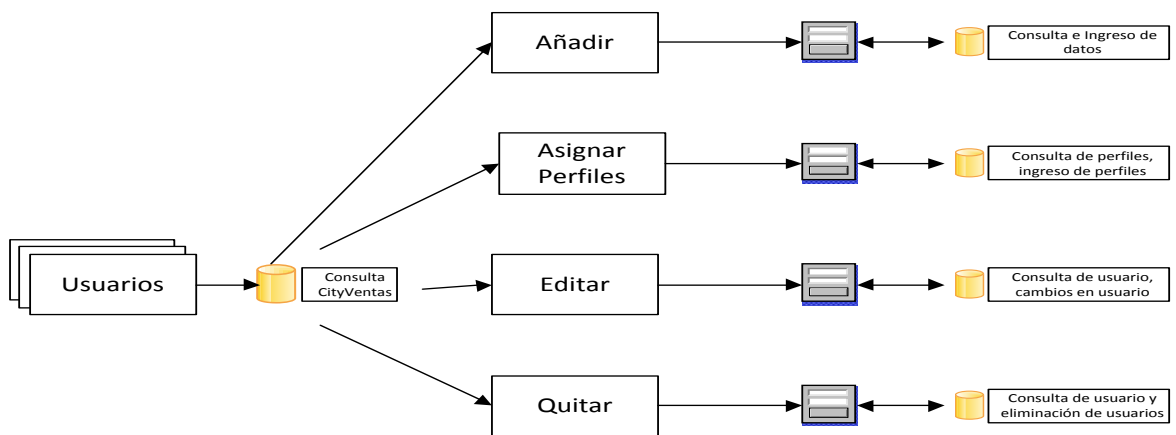


Diagrama 1.4 Navegación para administración de usuarios

Navegación para administración de agencias.

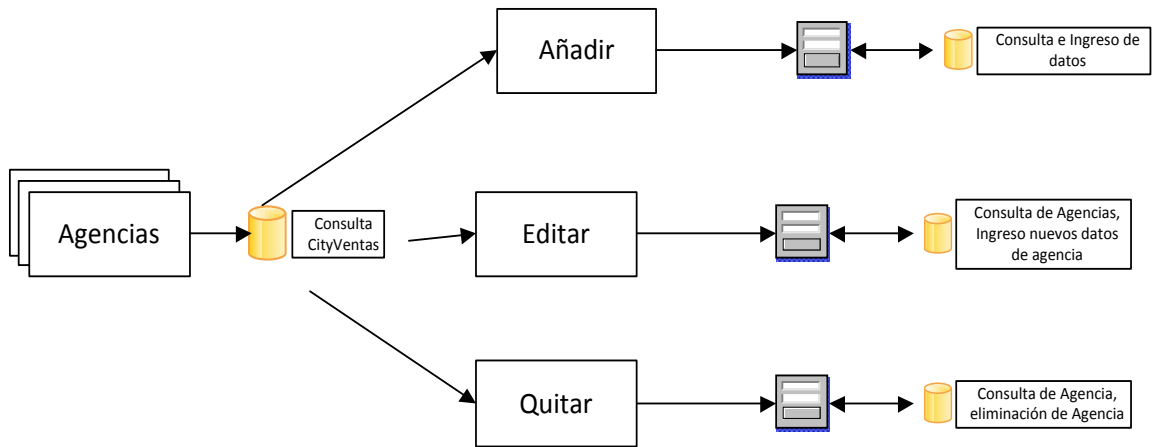


Diagrama 1.5 Navegación para administración de agencias.

Navegación para administración de horarios.

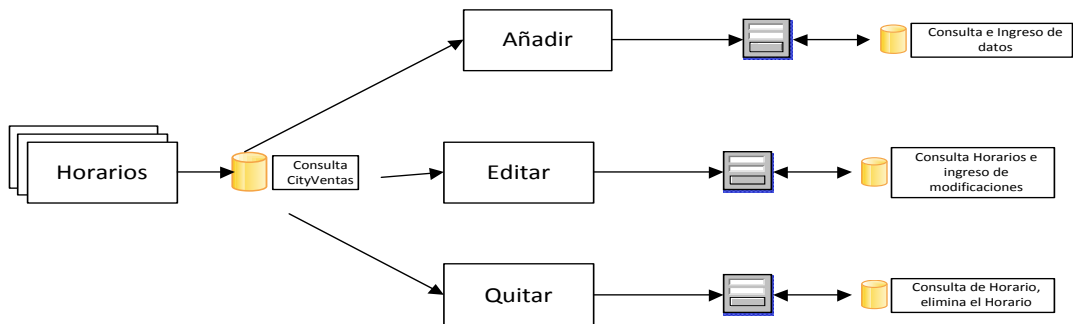


Diagrama 1.6 Navegación para administración de horarios

Flujo de navegación para Asistente de Gerencia.

El/ la asistente de gerencia tendrá accesos a parte de la aplicación web que se limita mediante la asignación de permisos en los perfiles de usuarios. El asistente podrá navegar en Seguridad – Usuarios, Administración de horarios – Horarios y Consultas.

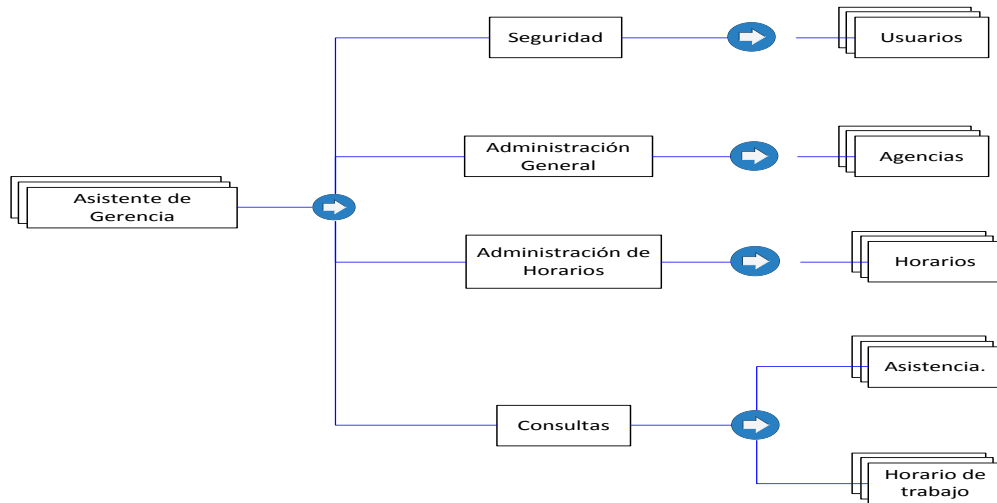


Diagrama 1.7 Navegación para asistente de gerencia

Navegación para consulta de asistencia.

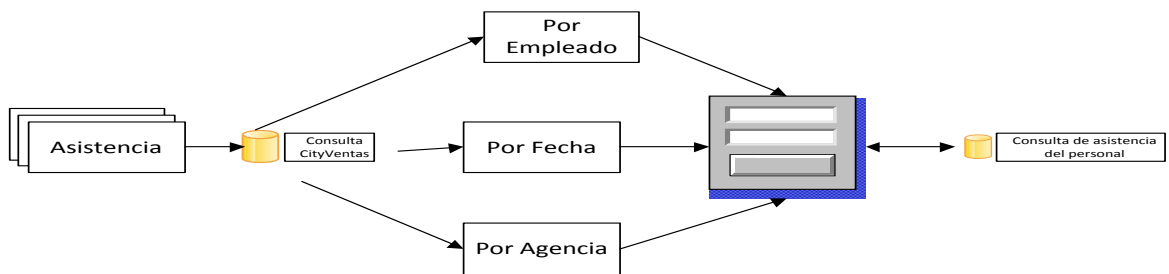


Diagrama 1.8 Navegación para consulta de asistencia.

Flujo de navegación para jefe de personal.

El Jefe de Personal, encargado de la distribución del talento humano es asignado en el sistema para el manejo de administración de horarios y consultas de asistencia en caso de falta o nueva asignación.

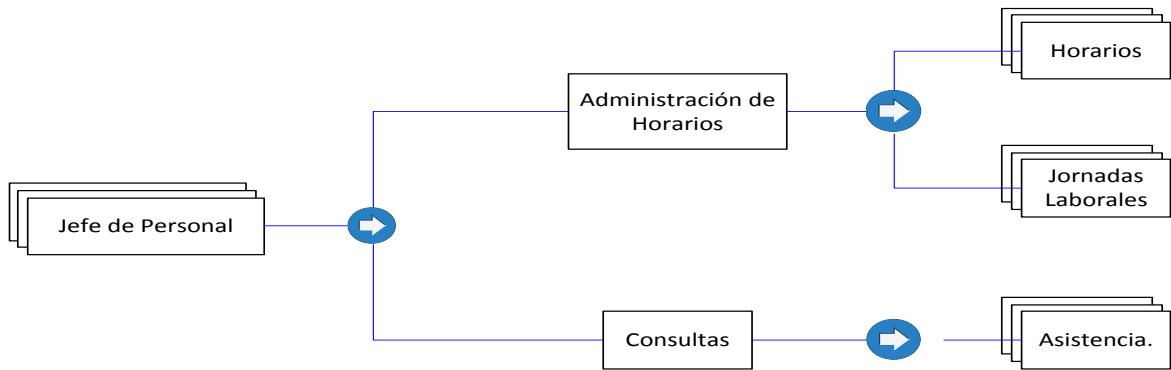


Diagrama 1.9 Navegación para Jefe de personal.

Navegación para asignación de jornadas laborales.

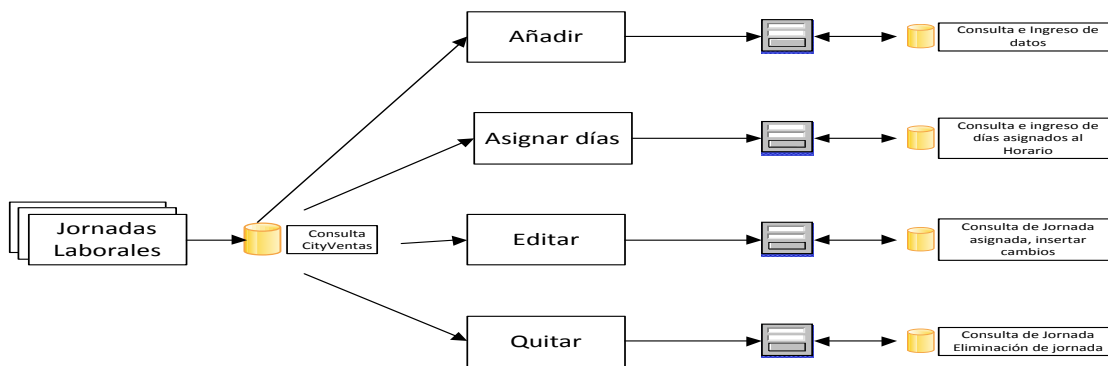


Diagrama 1.10 Navegación para asignación de jornadas laborales.

Flujo de Navegación para Empleado.

El empleado tiene acceso al sistema para realizar consulta sobre la asignación que se le entrega de las jornadas laborales y la agencia en donde deberá desempeñar su actividad.

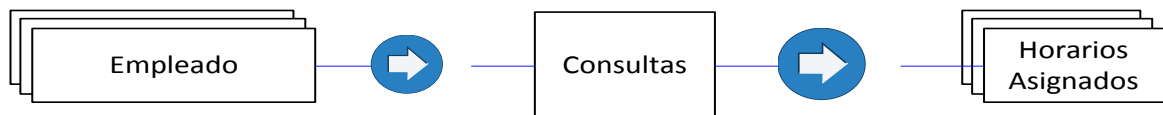


Diagrama 1.11 Navegación para empleado

Navegación para consulta de Horarios.

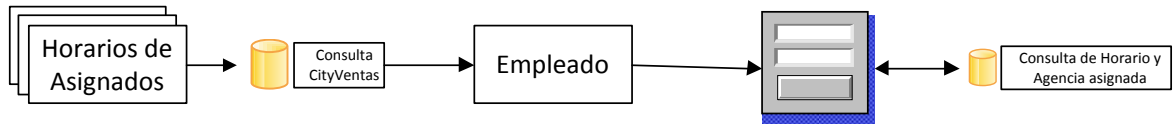


Diagrama 1.12 Navegación para consulta de Horarios.

2. Diagrama de Interface.

Inicio de Sesión

La Interface de inicio de sesión tendrá dos campos de texto para ingreso del correo además de la contraseña con la que fue asignado el usuario para loguearse. Se lo puede observar en el diagrama.



Diagrama 2.1 Interface login de usuario

Interface para Administrador del Sistema.

Después de haber ingresado a la navegación principal del sistema tendremos los siguientes enlaces:

- Seguridad.
- Administración General.
- Administración de Horarios.
- Consultas.



Diagrama 2.2 Interface administrador del sistema

Estos enlaces a su vez tienen sub enlaces asociados que se observarán a continuación

Interface para Seguridad del Sistema

La interface de Seguridad del Sistema tendrá dos sub enlaces: Perfiles y Usuarios



Diagrama 2.3 Interface seguridad del sistema

Interface para registro de Perfiles

La interface de perfiles estará asentada sobre una grilla en las filas y columnas muestran los perfiles ingresados y las opciones editar, quitar y asignar permisos.

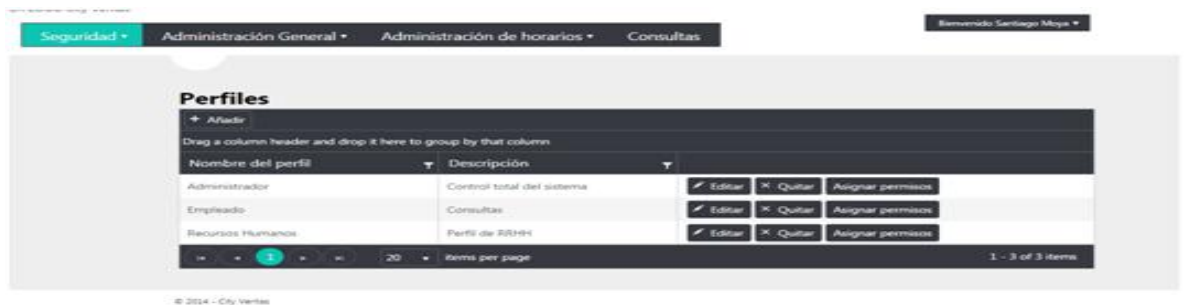


Diagrama 2.4 Interface registro de perfiles

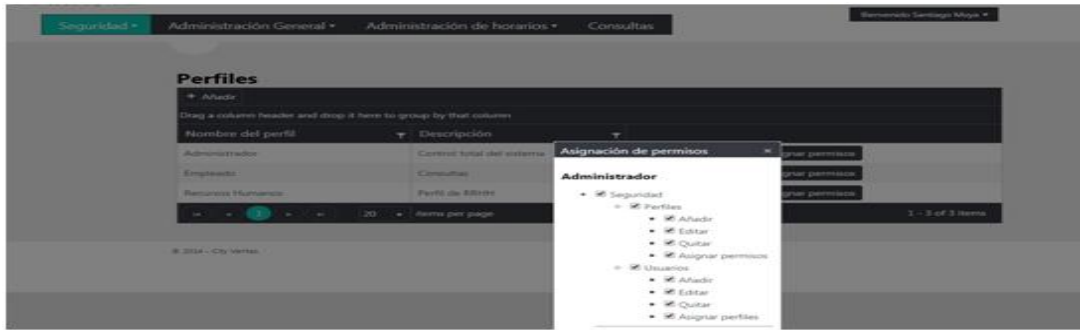


Diagrama 2.5 Interface asignación de permisos

Interface para registro de Usuarios

La interface para ingreso de usuario desarrollado sobre la grilla consta de un botón añadir y los campos para ingreso de nombre y apellido, número de cédula y el correo electrónico. Además de los botones editar, quitar y asignar perfiles.

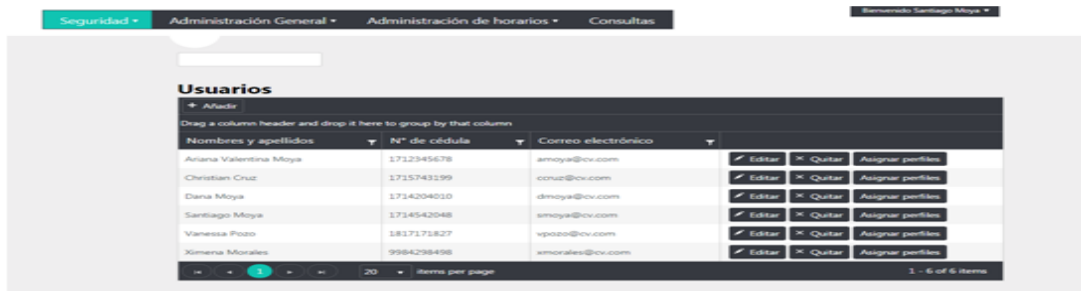


Diagrama 2.6 Interface registro de usuarios

Interface para Administración General

La interface Administración general tendrá el sub enlace Agencias



Diagrama 2.7 Interface administración general

Interface para registro de Agencias

La interface para agencias desarrollada sobre las propiedades de la grilla consta de un botón añadir, y los campos: nombre de agencia, ciudad, dirección. Además de los botones de editar y quitar.



Diagrama 2.8 Interface registro de agencias

Interface para Administración de Horarios

La interface para administración de horarios tiene dos sub enlaces: Horarios y Jornadas Laborales.

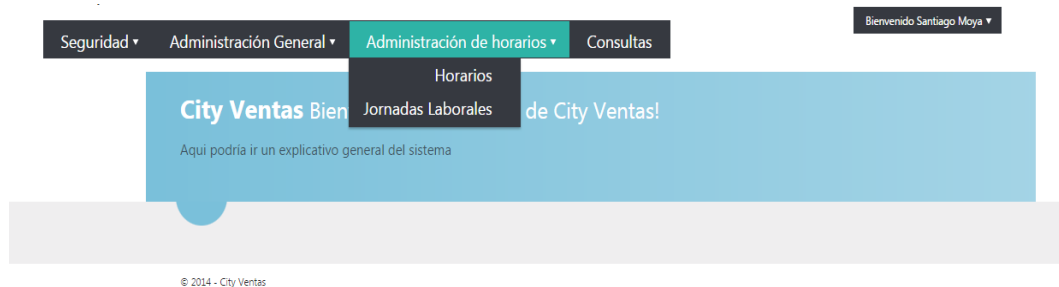


Diagrama 2.9 Interface administración de horarios

Interface de registro Horarios.

La interface para registrar horarios desarrollada en las propiedades de la grilla tiene un botón añadir, y campos como el nombre del horario, la hora de inicio y la hora de fin del horario. Los botones de Editar y Quitar.

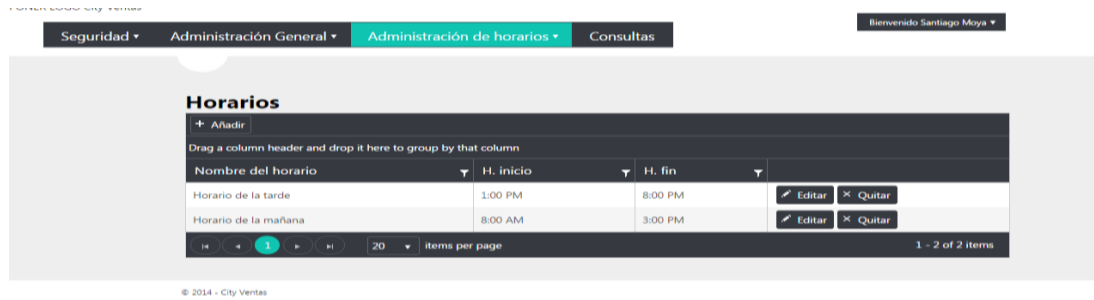


Diagrama 2.10 Interface registro de horarios

Interface de Jornadas Laborales.

La interface jornadas laborales desarrollada sobre las propiedades de la Grilla tiene como campos: nombre del empleado, horario de trabajo y agencia. Además de los botones de editar, quitar y el botón de asignar los días laborables del empleado en este horario.



Diagrama 2.11 Interface asignación de días a horario

Interface para consultas

Interface de consulta de horarios asignados

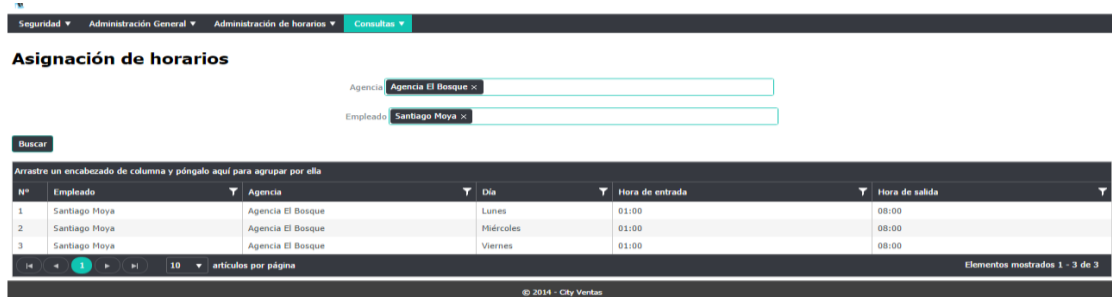


Diagrama 2.12 Interface consultar horarios asignados

Interface de consulta de Asistencias.



Diagrama 2.13 Interface Consultar Asistencia

Anexo 14. Tabla clases y métodos aplicación web

1. Tabla de Clases aplicación web.

La tabla de clases comprende de los campos:

- Código: Es el identificador de la clase, se compone de las siglas CLS (significa clase) más el número por secuencia de la clase; las siguientes letras identifican la aplicación Web (WEB). Ejemplo: CLSWIN-0001.
- Nombre: Nombre propuesto para la clase.
- Ubicación: El lugar (repositorio) en donde se almacena la clase.
- Descripción: Registra detalles importantes de la clase.

A continuación se proponen las clases generadas

TABLA DE CLASES Y MÉTODOS PARA EL DESARROLLO DE LA APLICACIÓN WEB

CÓDIGO	NOMBRE	UBICACIÓN	DESCRIPCIÓN
CLSWEB-0001	AccountController	Controller	Maneja el inicio de sesión para las cuentas de usuarios asignados.
CLSWEB-0002	AgenciaController	Controller	Ingresa, consulta, lee, actualiza, guarda y elimina datos de las agencias ingresadas en el sistema.
CLSWEB-0003	CiudadesController	Controller	Ingresa del nombre de las ciudades en donde se encuentra la empresa CityVenta
CLSWEB-0004	ConsultasController	Controller	Genera información de los datos recopilados en el sistema y los muestra al usuario.
CLSWEB-0005	HorarioController	Controller	Ingresa, consulta, lee, actualiza, guarda y elimina datos de los horarios ingresados en el sistema.
CLSWEB-0006	HorarioEmpleadoController	Controller	Ingresa, consulta, lee, actualiza, guarda y elimina horarios asignados a un empleado.
CLSWEB-0007	MapaController	Controller	Genera información de la navegación del sistema.
CLSWEB-0008	PerfilController	Controller	Ingresa, consulta, lee, actualiza, guarda y elimina perfiles de un usuario.
CLSWEB-0009	UbicacionController	Controller	Genera información de cómo se encuentra asignado el mapa de navegación del sistema.

CLSWEB-0010	UsuarioController	Controller	Ingresa, consulta, lee, actualiza, guarda y elimina información de usuarios del sistema.
CLSWEB-0011	NegocioController	Controller	Valida y muestra información de las consultas generadas por los usuarios del sistema de escritorio.
CLSWEB-0012	Conexión	Models	Genera conexión a la base de datos.
CLSWEB-0013	Correos	Models	Genera correo para usuarios que han perdido su contraseña.
CLSWEB-0014	Credenciales	Models	Genera información de los accesos que tiene el/los usuarios en el sistema.
CLSWEB-0015	Encriptacion	Models	Genera Encriptado de datos.
CLSWEB-0016	AgenciaViewModel	ViewModels	Validación de campos en el ingreso de agencias, declaración de variables.
CLSWEB-0017	CiudadesViewModel	ViewModels	Declaración y validación de campos para ingreso y consulta de ciudades.
CLSWEB-0018	DiasAsignadosViewModel	ViewModels	Declaración de variables para asignación de días.
CLSWEB-0019	HorarioEmpleadoViewModel	ViewModels	Valida campos y declara variables para entrega de horarios a los empleados.
CLSWEB-0020	HorarioViewModel	ViewModels	Valida campos y declara variables para ingreso de horarios.
CLSWEB-0021	LoginViewModel	ViewModels	Valida campos y declara variables para ingreso al sistema.
CLSWEB-0022	MapaViewModel	ViewModels	Declara variables para navegar en la página Web.
CLSWEB-0023	PerfilesAsignadosViewModel	ViewModels	Declara variables para asignar perfiles a usuarios ingresados.
CLSWEB-0024	PerfilViewModel	ViewModels	Valida y declara variables para el ingreso de nuevos perfiles.
CLSWEB-0025	PermisosAsignadosViewModel	ViewModels	Declara variables para los permisos asignados a cada usuario.
CLSWEB-0026	UbicacionViewModel	ViewModels	Valida campos y declara variables para ingreso de ubicación de locales comerciales.
CLSWEB-0027	UsuarioViewModel	ViewModels	Valida campos y declara variables para ingreso de nuevos usuarios.

Tabla 1.1 Tabla de clases aplicación web

2. Tabla de métodos para aplicación web.

La tabla de métodos comprende de los campos:

- Código: es el identificador de la clase, se compone de las siglas MTO (significa método) más el número por secuencia del método; las siguientes letras identifican la aplicación web (WEB). Ejemplo: MTOWEB-0001.
- Nombre: nombre propuesto para el método.
- Ubicación: el nombre de la clase lugar (repositorio) en donde se deposita el método.
- Descripción: registra detalles importantes del método.
- Estructura: forma del método con los atributos de entrada y salida.

CÓDIGO	NOMBRE	UBICACIÓN	DESCRIPCIÓN	ESTRUCTURA
MTOWEB-0001	InicioSesion	AccountController	Carga la pantalla de inicio de sesión	public ActionResult InicioSesion() { }
MTOWEB-0002	InicioSesion	AccountController	Toma y envía datos de inicio de sesión de usuario, ingreso de usuario al sistema.	public ActionResult InicioSesion (LoginViewModel model, string CORREO_USUARIO, string CLAVE_USUARIO, string returnUrl) { }
MTOWEB-0003	ValidarCredenciales	AccountController	Valida los datos ingresados en el inicio de sesión mediante consulta a la base de datos.	private bool ValidarCredenciales(string mail, string clave) { }
MTOWEB-0004	TienePermiso	AccountController	Devuelve valor verdadero o falso si el usuario se encuentra asignado para navegación del sistema.	public bool TienePermiso(int? id_acceso) { }
MTOWEB-0005	LogOn	AccountController	Si el usuario está ingresado devuelve la pantalla en donde puede navegar el usuario. (permisos asignados)	public ActionResult LogOn([DataSourceRequest] DataSourceRequest request, LoginViewModel model, string returnUrl) { }
MTOWEB-0006	LogOff	AccountController	Si el usuario no está registrado re direcciona a la pantalla de inicio.	public ActionResult LogOff() { }

MTOWEB-0007	AccesoRestringido	AccountController	Refresca la vista (pantalla) de inicio.	public ActionResult AccesoRestringido() { }
MTOWEB-0008	OlvidoContraseña	AccountController	Verifica y devuelve los datos del usuario que olvido su contraseña	public ActionResult OlvidoContraseña(USUARIOS _Usuario) { }
MTOWEB-0009	ChangePassword	AccountController	Permite cambiar la contraseña del usuario.	public ActionResult ChangePassword(LoginViewModel model) { }
MTOWEB-0010	GetAgencias	AgenciaController	Devuelve el nombre de las agencias ingresada en la base de datos.	public IEnumerable<AgenciaViewModel> GetAgencias() { }
MTOWEB-0011	GetAgenciaById	AgenciaController	Devuelve la agencia que contiene el Id ingresado.	public AGENCIA GetAgenciaById(int? id_agencia) { }
MTOWEB-0012	GetNewAgencia	AgenciaController	Carga los datos en la vista (pantalla) de la nueva agencia que fue ingresada.	private AgenciaViewModel GetNewAgencia(AGENCIA agencia) { }
MTOWEB-0013	Create	AgenciaController	Ingreso de nuevas agencias.	public ActionResult Create([DataSourceRequest] DataSourceRequest request, AGENCIA agencia) { }
MTOWEB-0014	Update	AgenciaController	Actualiza cambios realizados en las agencias existentes.	public ActionResult Update([DataSourceRequest] DataSourceRequest request, AGENCIA agencia) { }
MTOWEB-0015	Destroy	AgenciaController	Elimina Agencia de la base de datos.	public ActionResult Destroy([DataSourceRequest] DataSourceRequest request, AGENCIA agencia) { }
MTOWEB-0016	GetCiudades	CiudadesController	Devuelve el nombre de las ciudades ingresadas en la base de datos	public IEnumerable<CiudadesViewModel> GetCiudades() { }
MTOWEB-0017	GetCiudadById	CiudadesController	Devuelve el nombre de una ciudad consultada por el identificador	public UBICACIONES GetCiudadById(int? id_ubicacion) { }
MTOWEB-0018	GetNewCiudades	CiudadesController	Asigna las variables para los nuevos ingresos de las ciudades.	private CiudadesViewModel GetNewCiudades(UBICACIONES ciudades) { }

MTOWEB-0019	Create	CiudadesController	Ingresa nuevas ciudades a la base de datos.	public ActionResult Create([DataSourceRequest] DataSourceRequest request, UBICACIONES ciudades) { }
MTOWEB-0020	Update	CiudadesController	Actualiza los datos de las ciudades ingresadas	public ActionResult Update([DataSourceRequest] DataSourceRequest request, UBICACIONES ciudades) { }
MTOWEB-0021	Destroy	CiudadesController	Elimina ciudades que han sido ingresadas	public ActionResult Destroy([DataSourceRequest] DataSourceRequest request, UBICACIONES ciudades) { }
MTOWEB-0022	Index	CiudadesController	Muestra la vista principal de ciudades.	public ActionResult Index() { }
MTOWEB-0023	GetAsistencias	ConsultaController	Genera una lista de asistencias del usuario dependiendo de los rangos de fechas propuestos.	protected IEnumerable<AsistenciaViewModel> GetAsistencias(string fecha_inicio, string fecha_fin, string[] id_agencia, string[] id_usuarioempleado) { }
MTOWEB-0024	AsistenciaRead	ConsultaController	Toma los datos ingresados por el usuario para realizar la consulta.	public ActionResult AsistenciaRead([DataSourceRequest] DataSourceRequest request, string fecha_inicio, string fecha_fin, string[] id_agencia, string[] id_usuarioempleado) { }
MTOWEB-0025	GetAsignacionHorarios	ConsultaController	Genera un listado de los horarios dependiendo de los datos de agencia y usuario propuestos.	protected IEnumerable<AsignacionHorariosViewModel> GetAsignacionHorarios(string[] id_agencia, string[] id_usuarioempleado) { }
MTOWEB-0026	AsignacionHorariosRead	ConsultaController	Toma los datos ingresados por el usuario para la consulta de horarios.	public ActionResult AsignacionHorariosRead([DataSourceRequest] DataSourceRequest request, string[] id_agencia, string[] id_usuarioempleado) { }

MTOWEB-0027	Index	HomeController	Describe el mensaje de bienvenida al sistema. (Creado por la aplicación)	public ActionResult Index() { }
MTOWEB-0028	GetHorarios	HorarioController	Devuelve y enlista los horarios ingresados en la base de datos.	public IEnumerable<HorarioViewModel> GetHorarios() { }
MTOWEB-0029	GetHorarioById	HorarioController	Devuelve los datos de un horario específico consultado por su identificador.	public HORARIO GetHorarioById(int? id_horario) { }
MTOWEB-0030	Create	HorarioController	Ingresa nuevos horarios a la base de datos.	public ActionResult Create([DataSourceRequest] DataSourceRequest request, HORARIO horario) { }
MTOWEB-0031	Update	HorarioController	Actualiza los datos de los horarios ingresados.	public ActionResult Update([DataSourceRequest] DataSourceRequest request, HORARIO horario) { }
MTOWEB-0032	Destroy	HorarioController	Elimina horarios de la base de datos.	public ActionResult Destroy([DataSourceRequest] DataSourceRequest request, HORARIO horario) { }
MTOWEB-0033	GetHorariosEmpleados	HorarioEmpleadoController	Devuelve los horarios que se encuentran ingresados en la base de datos	public IEnumerable<HorarioEmpleadoViewModel> GetHorariosEmpleados() { }
MTOWEB-0034	GetHorarioEmpleadoById	HorarioEmpleadoController	Devuelve los datos específicos de un horario asignado a un empleado.	public HORARIOEMPLEADO GetHorarioEmpleadoById(int? id_horarioempleado) { }
MTOWEB-0035	GetNewHorarioEmpleado	HorarioEmpleadoController	Devuelve el nuevo horario que se le asignó a un empleado.	private HorarioEmpleadoViewModel GetNewHorarioEmpleado(HORARIOEMPLEADO horarioempleado) { }
MTOWEB-0036	ValidarCruceHoras	HorarioEmpleadoController	Valida el cruce de horarios de un empleado antes de registrar los datos.	private HORARIOEMPLEADOS ValidarCruceHoras(HORARIOS horario, int idUsuario, int idDia, int idHorarioEmpleado) { }
MTOWEB-0037	Create	HorarioEmpleadoController	Crea o asigna un nuevo horario a un empleado	public ActionResult Create([DataSourceRequest] DataSourceRequest request, HORARIOEMPLEADO horarioempleado) { }

MTOWEB-0038	Update	HorarioEmpleadoController	Actualiza un horario asignado a un empleado.	public ActionResult Update([DataSourceRequest] DataSourceRequest request, HORARIOEMPLEADO horarioempleado) { }
MTOWEB-0039	Destroy	HorarioEmpleadoController	Elimina el horario asignado a un empleado	public ActionResult Destroy([DataSourceRequest] DataSourceRequest request, HORARIOEMPLEADO horarioempleado) { }
MTOWEB-0040	AsignarDias	HorarioEmpleadoController	Asignación de días para un horario.	public ActionResult AsignarDias(int id) { }
MTOWEB-0041	DiasAsignados	HorarioEmpleadoController	Consulta de días asignados a un horario de un empleado.	private void DiasAsignados(HORARIOEMPLEADO horarioempleado) { }
MTOWEB-0042	AsignarDias	HorarioEmpleadoController	Guarda los datos de la nueva asignación de horarios de un empleado.	public ActionResult AsignarDias(int id, FormCollection formCollection, string[] DiasSeleccionados) { }
MTOWEB-0043	UpdateHorarioEmpleado	HorarioEmpleadoController	Actualiza el horario y días asignados a un empleado	private void UpdateHorarioEmpleado(string[] DiasSeleccionados, HORARIOEMPLEADO ActualizarHorarioEmpleado) { }
MTOWEB-0044	GetPermisosByIdUsuario	MapaController	Obtiene los permisos que fueron asignados al usuario y que parte de la aplicación puede navegar	public IEnumerable<MapaViewModel> GetPermisosByIdUsuario(int? id_usuario) { }
MTOWEB-0045	GetPerfiles	PerfilController	Obtiene los perfiles que fueron ingresados en la base de datos.	public IEnumerable<PerfilViewModel> GetPerfiles() { }
MTOWEB-0046	GetPerfilById	PerfilController	Obtiene los datos específicos de un perfil que está consultando.	public PERFILE GetPerfilById(int? id_perfil) { }
MTOWEB-0047	Create	PerfilController	Guarda los datos de un nuevo perfil que se ha ingresado.	public ActionResult Create([DataSourceRequest] DataSourceRequest request, PERFILE perfil) { }
MTOWEB-0048	Update	PerfilController	Actualiza los datos de un perfil ingresado previamente.	public ActionResult Update([DataSourceRequest] DataSourceRequest request, PERFILE perfil) { }

MTOWEB-0049	Destroy	PerfilController	Elimina un perfil seleccionado.	public ActionResult Destroy([DataSourceRequest] DataSourceRequest request, PERFILE perfil) { }
MTOWEB-0050	AsignarPermisos	PerfilController	Asigna permisos a un perfil	public ActionResult AsignarPermisos(int id) { }
MTOWEB-0051	PermisosAsignados	PerfilController	Consulta permisos que se le puede asignar a un perfil.	private void PermisosAsignados(PERFILE perfiles) { }
MTOWEB-0052	AsignarPermisos	PerfilController	Asigna permisos a un perfil específico.	public ActionResult AsignarPermisos(int id, FormCollection formCollection, string[] PermisosSeleccionados) { }
MTOWEB-0053	UpdatePerfilMapas	PerfilController	Actualiza los sitios en donde el usuario puede navegar según la asignación de permisos.	private void UpdatePerfilMapas(string[] PermisosSeleccionados, PERFILE ActualizarPerfil) { }
MTOWEB-0054	GetUbicaciones	UbicacionController	Obtiene los datos de las ubicaciones ingresadas en la base de datos.	public IEnumerable<UbicacionViewModel> GetUbicaciones() { }
MTOWEB-0055	GetUsuarios	UsuarioController	Obtiene los datos de los usuarios ingresados en la base de datos.	public IEnumerable<UsuarioViewModel> GetUsuarios() { }
MTOWEB-0056	GetUsuarioById	UsuarioController	Obtiene los datos de un usuario específico.	public USUARIO GetUsuarioById(int? id_usuario) { }
MTOWEB-0057	GetNewUsuario	UsuarioController	Instancia a un nuevo usuario que ha sido creado o modificado	private UsuarioViewModel GetNewUsuario(USUARIO usuario) { }
MTOWEB-0058	GetUsuarioByMailByClave	UsuarioController	Obtengo los datos del usuario por clave y mail.	public UsuarioViewModel GetUsuarioByMailByClave(string mail_usuario, string clave_usuario) { }
MTOWEB-0059	Create	UsuarioController	Guarda los datos de nuevos usuarios en la base de datos	public ActionResult Create([DataSourceRequest] DataSourceRequest request, USUARIO usuario) { }
MTOWEB-0060	Update	UsuarioController	Actualiza los datos de un usuario.	public ActionResult Update([DataSourceRequest] DataSourceRequest request, USUARIO usuario) { }
MTOWEB-	Destroy	UsuarioController	Elimina usuarios de la	public ActionResult

0061			base de datos.	Destroy([DataSourceRequest] DataSourceRequest request, USUARIO usuario) { }
MTOWEB-0062	PerfilesAsignados	UsuarioController	Obtiene los datos de los perfiles ingresados en la base de datos.	private void PerfilesAsignados(USUARIO usuario) { }
MTOWEB-0063	AsignarPerfiles	UsuarioController	Asigna perfiles seleccionados a un usuario.	public ActionResult AsignarPerfiles(int id, FormCollection formCollection, string[] PerfilesSeleccionados) { }
MTOWEB-0064	UpdateUsuarioPerfiles	UsuarioController	Actualiza los perfiles asignados a un usuario.	private void UpdateUsuarioPerfiles(string[] PermisosSeleccionados, USUARIO ActualizarUsuario) { }
MTOWEB-0065	CedulaValida	UsuarioController	Verifica si la cédula ingresada del usuario es válida	public bool CedulaValida(string cedula){ }
MTOWEB-0066	Conexion	Conexión	Instancia variable para nueva conexión a base de datos	public Conexion() { }
MTOWEB-0067	Guardar	Conexión	Registra los cambios realizados en la base de datos	public void Guardar() { }
MTOWEB-0068	Credenciales	Credenciales	Toma los datos que son ingresados para el registro del usuario.	public Credenciales() { }
MTOWEB-0069	Encriptar	Encriptacion	Encripta cadena de caracteres	public string encriptar(string cadena) { }
MTOWEB-0070	Desencriptar	Encriptacion	Desencripta cadena de caracteres.	public string desencriptar(string cadena) { }
MTOWEB-0071	EncriptarAux	Encriptacion	Convierte la cadena / clave en arreglo de bytes	public string encriptarAux(string cadena) { }
MTOWEB-0072	DesencriptarAux	Encriptacion	Convierte cadena/ clave en arreglo de 64 bits	public string desencriptarAux(string cadena) { }

Tabla 2.1 Tabla de métodos aplicación web

Anexo 15. Tabla de clases y métodos para desarrollo de la aplicación de escritorio

TABLA DE CLASES Y MÉTODOS PARA DESARROLLO DE LA APLICACIÓN DE ESCRITORIO

1. Tabla de Clases Sistema Windows

CÓDIGO	NOMBRE	UBICACIÓN	DESCRIPCIÓN
CLSWIN-0001	Classifier_Train	FaceRecognition	Reconocimiento de rostros, entrenamiento de neuronal, clasificación de rasgos para rostros
CLSWIN-0002	Form 1	FaceRecognition	Conversión imágenes en grises, registro de usuario, cuadros para detección de rostros, inicializar cámara Web, inicialización red neuronal.
CLSWIN-0003	Training_Form	FaceRecognition	Registro y almacenamiento de imágenes, Entrenamiento para reconocimiento de rostros

Tabla 1.1 Tabla de clases sistema Windows

2. Tabla de métodos para sistema Windows

CÓDIGO	NOMBRE	UBICACIÓN	DESCRIPCIÓN	ESTRUCTURA
MTOWIN-0001	ADD_Face_Found	Form1	Recuadro para detectar rostro.	void ADD_Face_Found(Image<Gray, Byte> img_found, string name_person){ }
MTOWIN-0002	btnRegistrar_Click	Form1	Registrar Horario de empleado.	Private void btnRegistrar_Click(object sender, EventArgs e){ }
MTOWIN-0003	Form1	Form1	Inicializar el Form	public Form1(int idAgencia)
MTOWIN-0004	FrameGrabber_Parrellel	Form1	Detección de imágenes Conversión escala de grises muestra de imagen	void FrameGrabber_Parrellel(object sender, EventArgs e){ }
MTOWIN-0005	FrameGrabber_Standard	Form1	Detección y procesamiento de imagines registradas	void FrameGrabber_Standard(object sender, EventArgs e){ }
MTOWIN-0006	initialise_capture	Form1	Detección e inicio de cámara Web	public void initialise_capture() { }
MTOWIN-0007	Retrain	Form1	Inicializar red neuronal	retrain() { }
MTOWIN-0008	stop_capture	Form1	Finalizar cámara web	private void stop_capture() { }
MTOWIN-0009	Classifier_Train	Classifier_Train	Iniciar proceso de reconocimiento	public Classifier_Train() { }

MTOWIN-0010	LoadTrainingData	Classifier_Train	Devolución de dato verdadero o falso si usuario reconocido.	Private bool LoadTrainingData(string Folder_location) { }
MTOWIN-0011	Recognise	Classifier_Train	Reconocimiento de imagen ingresada	public string[] Recognise(Image<Gray, byte> Input_image){ }
MTOWIN-0012	ADD_BTN_Click	Training_Form	Guardar imagen obtenida.	private void ADD_BTN_Click(object sender, EventArgs e){ }
MTOWIN-0013	btnBuscar_Click	Training_Form	Buscar usuario registrado en sistema Web	private void btnBuscar_Click(object sender, EventArgs e) { }
MTOWIN-0014	btnValidar_Click	Training_Form	Validar usuario que puede registrar rostros	private void btnValidar_Click(object sender, EventArgs e) { }
MTOWIN-0015	FrameGrabber	Training_Form	Detección de rostros, cuadro de reconocimiento facial	void FrameGrabber(object sender, EventArgs e) { }
MTOWIN-0016	ImageArray	Training_Form	Construcción de arreglo de imágenes	public byte[] ImageArray(Image Imagen) { }
MTOWIN-0017	save_training_data	Training_Form	Guardar datos de usuario registrado	private bool save_training_data(Image face_data) { }
MTOWIN-0018	Training_Form	Training_Form	Inicializar componentes y variables	public Training_Form(Form1 _Parent) { }

Tabla 2.1 Tabla de métodos para sistema Windows

Anexo 16. Prueba de caja negra y caja blanca aplicación web

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Ejecución casos de prueba de caja y negra y caja blanca para aplicación web

Versión [3.0]

1. Ejecución de casos de prueba.

Según el cronograma presentado se realiza las pruebas del sistema en la siguiente secuencia:

Caso de Prueba CP-0001 Ingreso de usuario.

- **Prueba de Caja Blanca.**

Método: Create (Ingresa un nuevo usuario)

Funcionalidad: Guarda los datos de nuevo usuario en la base de datos.

```
//SI EL USUARIO ES DIFERENTE DE NULO Y ESTA VALIDADO
//VERIFICAMOS QUE:
if (usuario != null && ModelState.IsValid)
    {
    //SI EL MAIL NO SE ENCUENTRA REGISTRADO
if (GetUsuarios().Where(m => m.MAIL_USUARIO.ToUpper() ==
usuario.MAIL_USUARIO.ToUpper()).Count() == 0)
    {
        //MUESTRA MENSAJE DE GRABADO EXITOSO
ModelState.AddModelError("", "Grabado exitoso");
    }
    //SI EL MAIL DEL USUARIO SE ENCUENTRA REGISTRADO
    Else
    //MENSAJE DE USUARIO YA REGISTRADO
ModelState.AddModelError("", "Error: \n Ya existe registrado el correo");
//RETORNAMOS LOS DATOS PARA QUE SE REFLEJEN EN LA VISTA
return Json(new[] {GetNewUsuario(usuario) } .ToDataSourceResult(request,
ModelState));
    }
    }
//SI NO SE PUEDE GUARDAR POR PROBLEMA DE CONEXIÓN
// ENVÍA MENSAJE DE ERROR.
catch (Exception name) { ModelState.AddModelError("", "Error: \n " + name.Message);
}
}
```

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	ViewModels y Controllars	Vallor Null (Press Update)	Mensajes de campos requeridos	catch (Exception name) { ModelState.AddModelError("", "Error: \n " + name.Message); }
PCB-0002	ViewModels y Controllars	Datos ingresados correctos (Press Update)	Mensaje de Grabado exitoso	if (GetUsuarios().Where(m => m.MAIL_USUARIO.ToUpper() == usuario.MAIL_USUARIO.ToUpp er()).Count() == 0) // MENSAJE DE GRABADO EXITOSO

Tabla 1.1 Pruebas de caja blanca CP-001

- **Prueba de Caja Negra.**

Prueba 1.- Entradas valor “ “ (NULL).

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Nombres y Apellidos	Null	El campo Nombres y apellidos son requeridos.	0.0002	[Required(ErrorMessage = "El campo {0} es requerido")]
	N° de cédula	Null	El campo N° de cédula es requerido	0.0002	[Required(ErrorMessage = "El campo {0} es requerido")] [MaxLength(10, ErrorMessage = "Número de caracteres permitidos = 10")]
	Correo electrónico	Null	El campo Correo electrónico es requerido	0.0002	[Required(ErrorMessage = "El campo {0} es requerido")] [MaxLength(50, ErrorMessage = "Número de caracteres permitidos = 50")] [RegularExpression(@"\w+([-+.] \w+)*@\w+([-+.] \w+)*\.\w+([-+.] \w+)*", ErrorMessage = "Ingrese un correo válido")]
	Press Update		Mensajes de campo requerido	0.0002	catch (Exception name) { ModelState.AddModelError("", "Error: \n " + name.Message); }

Tabla 1.2 Prueba 1 de caja negra para PC-0001

Prueba 2.- Entradas valores acordes a cada campo

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Nombres y Apellidos	“Santiago Paúl Moya Portilla”			
	N° de cédula	“1714542048”			
	Correo electrónico	“smoya@cv.com”			
	Press Update		Mensaje de Grabado exitoso	0.0002	catch (Exception name) { ModelState.AddModelError("", "Error: \n " + name.Message); }

Tabla 1.3 Prueba 2 de caja negra para PC-0001

Caso de Prueba CP-0002 Asignar perfiles a usuario.

- **Prueba de Caja Blanca.**

Método: AsignarPerfiles (Asigna perfiles a un usuario)

Funcionalidad: Asigna perfiles seleccionados a un usuario.

Ingreso al método AsignarPerfiles(int id):

```
try
{
conexion = new Conexion();
//CONSULTA SOBRE PERFILES INGRESADOS.
//ENVIA CONSULTA A MÉTODO PERFILESASIGNADOS
PerfilesAsignados(AsignarPerfiles);
//DEVUELVE EL NOMBRE DEL USUARIO QUE SE
//ASIGNARÁ EL PERFIL.
ViewBag.Nombre = AsignarPerfiles.NOMBRES_USUARIO;
//RETORNA LA VISTA
return PartialView(AsignarPerfiles);
}
```

Ingreso al método AsignarPerfiles(int id, FormCollection formCollection, string[] PerfilesSeleccionados):

```
try
{
//ENVÍO DATOS PARA LA FUNCIÓN ACTUALIZAR
UpdateUsuarioPerfiles(PerfilesSeleccionados, AsignarPerfil);
//GUARDA LOS CAMBIOS
conexion.Guardar();
}
```

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	Controllers	No escogemos ningún valor	No graba valores en la base de datos.	UpdateUsuarioPerfiles(PerfilesSeleccionados, AsignarPerfil);
PCB-0002	Controllers	Escogemos por lo menos un valor.	Graba en base de datos el perfil seleccionado.	UpdateUsuarioPerfiles(PerfilesSeleccionados, AsignarPerfil);

Tabla 1.4 Pruebas de caja blanca CP-0002

- **Prueba de Caja Negra.**

Prueba 1.- Entradas valor “no check”.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Administrador	No check	No guarda cambios	0.0001	No se asigna perfil.
	Talento Humano	No check	No guarda cambios.	0.0001	No se asigna perfil.
	Empleado	No check	No guarda cambios.	0.0001	No se asigna perfil.

Tabla 1.5 Prueba 1 de caja negra para PC-0002

Prueba 2.- Entrada escoge el perfil Empleado.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Administrador	No check	No guarda cambios	0.0001	No se asigna perfil.
	Talento Humano	No check	No guarda cambios.	0.0001	No se asigna perfil.
	Empleado	Check	Guarda cambios.	0.0001	Se asigna perfil.

Tabla 1.6 Prueba 2 de caja negra para PC-0002

Prueba 3.- Ingreso al sistema del usuario asignado perfil Empleado.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Correo electrónico	“bportilla@cv.com”			Ingreso del campo correo electrónico
	Contraseña	*****			Ingreso del campo contraseña.
	Clic Iniciar Sesión		Bienvenido Blanca Portilla.	0.0002	Vista a selección de consultas. No tiene acceso a otras pestañas.

Tabla 1.7 Prueba 3 de caja negra para PC-0002

Caso de Prueba CP-0003 Registro de horarios.

- **Prueba de Caja Blanca.**

Método: Create (Crear un nuevo Horario)

Funcionalidad: Ingresa nuevos horarios a la base de datos.

```
//SI EL HORARIO NO EXISTE Y LOS INGRESOS ESTÁN
//VALIDADOS
if (horario != null && ModelState.IsValid)
{
//SI NO EXISTE EL HORARIO YA INGRESADO
If (GetHorarios().Where(m => m.DESDE_HORARIO.ToString() ==
horario.DESDE_HORARIO.ToString()
&& m.HASTA_HORARIO.ToString() == horario.HASTA_HORARIO.ToString()).Count() ==
0)
{
//INGRESA Y GUARDA EL HORARIO EN LA BASE DE
//DATOS.
//MENSAJE DE GRABADO EXITOSO.
ModelState.AddModelError("", "Grabado exitoso");
}
Else
//SI EXISTE EL HORARIO SE DA MENSAJE DE HORARIO
ModelState.AddModelError("", "Error: \n Ya existe una horario con las mismas horas de inicio
y de fin");
}
```

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	ViewModels y Controllars	No se ingresa nombre de Horario. Press Update	El campo Nombre del horario es requerido.	[Required(ErrorMessage = "El campo {0} es requerido")]
PCB-0002	ViewModels y Controllars	Ingreso de nombre de Horario, selección hora de entrada, selección hora de salida. Press Update.	Grabado exitoso	ModelState.AddModelError("", "Grabado exitoso");
PCB-0003	ViewModels y Controllars	Ingreso de nombre de Horario, selección hora de entrada (duplicada), selección hora de salida. (duplicada) Press Update.	Ya existe una horario con las mismas horas de inicio y de fin	ModelState.AddModelError("", "Error: \n Ya existe una horario con las mismas horas de inicio y de fin");

Tabla 1.8 Pruebas de caja blanca para CP-0003

- **Prueba de Caja Negra.**

Prueba 1.- Entradas valor "NULL".

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Nombre de Horario	Null	El campo Nombre del horario es requerido.	0.0002	[Required(ErrorMessage = "El campo {0} es requerido")]
	Press Update		Mensajes de campo requerido	0.0002	catch (Exception name) { ModelState.AddModelError("", "Error: \n " + name.Message); }

Tabla 1.9 Prueba 1 de caja negra para CP-0003

Prueba 2.- Entradas valor "Horario de la mañana" en nombre Horario.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Nombre de Horario	"Horario de la mañana"			
	H. Inicio	Selección "8:00 AM"			
	H. Fin	Selección "03:00 PM"			
	Press Update		Grabado exitoso	0.0002	Ingreso de información en la base de datos.

Tabla 1.10 Prueba 2 de caja negra para CP-0003

Caso de Prueba CP-0004 Asignación de Jornada Laboral..

- **Prueba de Caja Blanca.**

Método: Create (Asignar un Horario a un empleado)

Funcionalidad: Asigna un horario a un empleado.

//SI EL HORARIO NO ES NULO Y LOS DATOS SON VALIDADOS

if (horarioempleado != null && ModelState.IsValid)

{

//SI EL HORARIO NO SE ENCUENTRA PREVIAMENTE

//REGISTRADO

if (GetHorariosEmpleados().Where(m => m.ID_USUARIO ==

horarioempleado.ID_USUARIO && m.ID_HORARIO ==

horarioempleado.ID_HORARIO).Count() == 0)

{

//GUARDAR EN LA BASE DE DATOS LA INFORMACIÓN

//Y MOSTRAR MENSAJE DE GRABADO EXITOSO

ModelState.AddModelError("", "Grabado exitoso");

}

// CASO CONTRARIO ENVIAR MENSAJE DE JORNADA LABORAL

//EXISTENTE

else

ModelState.AddModelError("", "Error: \n Ya existe una jornada laboral asignada al mismo empleado");

}

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	Views y Controllers	Primer empleado de la lista, primer horario de la lista, primera agencia de la lista consulta devuelve datos guardados de forma alfabética. Press Update	Ya existe una jornada laboral asignada al mismo empleado	ModelState.AddModelError("", "Error: \n Ya existe una jornada laboral asignada al mismo empleado");
PCB-0002	Views y Controllers	Selecciona nuevo empleado, selecciona horario de inicio y fin. Press Update	Grabado exitoso	ModelState.AddModelError("", "Grabado exitoso");

Tabla 1.11 Pruebas de caja blanca para CP-0004

- **Prueba de Caja Negra.**

Prueba 1.- Entradas valor consulta de base de datos.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Empleado	Consulta a base de datos	Ariana Moya	0.0002	columns.ForeignKey(p => p.ID_USUARIO, (System.Collections.IEnumerable)View

					Data["Usuarios"], "ID_USUARIO", "NOMBRES_USUARIO").Title("Empleado");
	Horario	Consulta a base de datos	Horario de la mañana(8:00 a 15:00)	0.0002	columns.ForeignKey(p => p.ID_HORARIO, (System.Collections.IEnumerable)ViewData["Horarios"], "ID_HORARIO", "NOMBRECOMPLETO_HORARIO").Title("Horario");
	Agencia	Consulta a base de datos	Agencia El Bosque	0.0002	columns.ForeignKey(p => p.ID_AGENCIA, (System.Collections.IEnumerable)ViewData["Agencias"], "ID_AGENCIA", "NOMBRE_AGENCIA").Title("Agencia");
	Press Update			0.0002	Mensaje: Ya existe una jornada laboral asignada al mismo empleado

Tabla 1.12 Prueba 1 de caja negra para PC-0004

Prueba 2.- Selección de nombre de empleado ingresado recientemente.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Empleado	Consulta a base de datos Selección de nombre	Blanca Eulalia Portilla Loachamín	0.0002	columns.ForeignKey(p => p.ID_USUARIO, (System.Collections.IEnumerable)ViewData["Usuarios"], "ID_USUARIO", "NOMBRES_USUARIO").Title("Empleado");
	Horario	Consulta a base de datos Selección Horario	Horario de la mañana(8:00 a 15:00)	0.0002	columns.ForeignKey(p => p.ID_HORARIO, (System.Collections.IEnumerable)ViewData["Horarios"], "ID_HORARIO", "NOMBRECOMPLETO_HORARIO").Title("Horario");
	Agencia	Consulta a base de datos	Agencia Quito Sur	0.0002	columns.ForeignKey(p => p.ID_AGENCIA, (System.Collections.IE

		Selección Agencia			numerable)ViewData["Agencias"], "ID_AGENCIA", "NOMBRE_AGENCIA").Title("Agencia");
	Press Update			0.0002	Grabado exitoso

Tabla 1.13 Prueba 2 de caja negra para CP-0004

Prueba 3.- Asignar días a Jornada Laboral.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Asignar día	Consulta a base de datos	Lun Mar Mié Jue Vie Sáb Dom	0.0002	var HorarioEmpleadosDias = new HashSet<int>(horarioempleado.HORARIOEMPLEADODIAS.Select(c => c.ID_DIA));
PCN-0002	Asignación de días	Escoger días Grabar	Datos ingresados en base de datos	0.0002	UpdateHorarioEmpleado(DiasSeleccionados, AsignarHorario);

Tabla 1.14 Prueba 3 de caja negra para CP-0004

Prueba 4.- Editar Jornada Laboral de un empleado.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Editar	Consulta a base de datos	Carga todos los horarios y agencias ingresados en la base de datos.	0.0002	
	Update		Grabado exitoso	0.0002	ModelState.AddModelError("", "Grabado exitoso")

Tabla 1.15 Prueba 4 de caja negra para CP-0004

Prueba 5.- Eliminar Jornada Laboral de un empleado.

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Press Quitar		Está seguro de eliminar el registro.	0.0002	
	Press Aceptar		Elimina el registro de la base de datos.	0.0002	vUsuario.ELIMINADO_HORARIOEMPLEADO = true;

Tabla 1.16 Prueba 5 de caja negra para CP-0004

Anexo 17. Pruebas para aplicación de escritorio

ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Ejecución casos de prueba blanca para aplicación de escritorio

Versión [3.0]

1. Ejecución de casos de pruebas de reconocimiento

Fondo color rosa

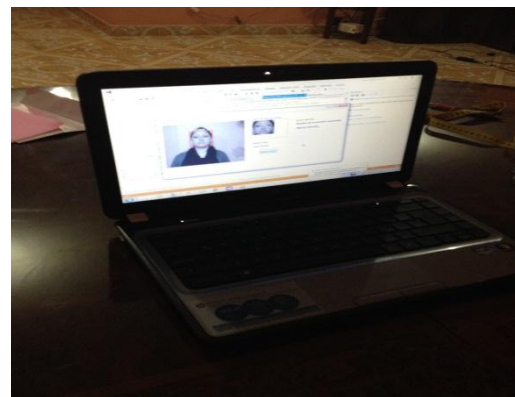
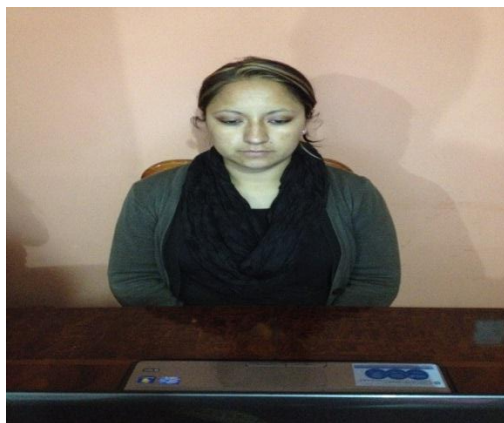


Imagen 1.1 Pruebas reconocimiento fondo rosa

Distancia(mm)	600	750	1000	1250	1500
Intensidad(Lux)					
2LX	EN+/NR	EN+/NR	EN-/NR	NE/NR	NE/NR
8LX	EN+/RE+	EN+/RE	EN/NR	EN-./NR	EN-./NR
19LX	EN+/RE++	EN+/RE++	EN/RE+	EN/RE-	EN-/NR
27LX	EN+/RE++	EN/RE++	EN/RE+	EN/RE-	EN/RE-.-

Tabla 1.1 Resultados pruebas reconocimiento usando fondo rosa

Fondo color Blanco

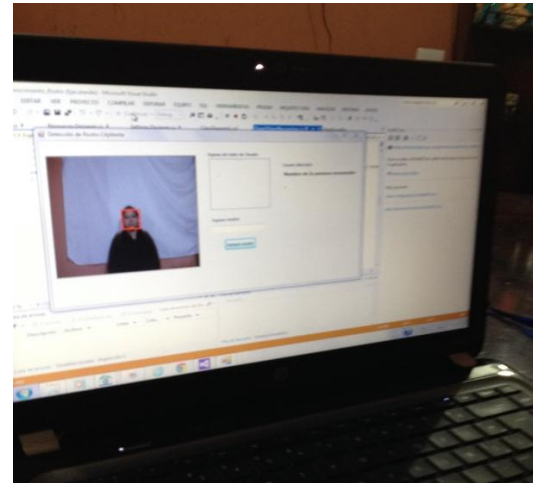
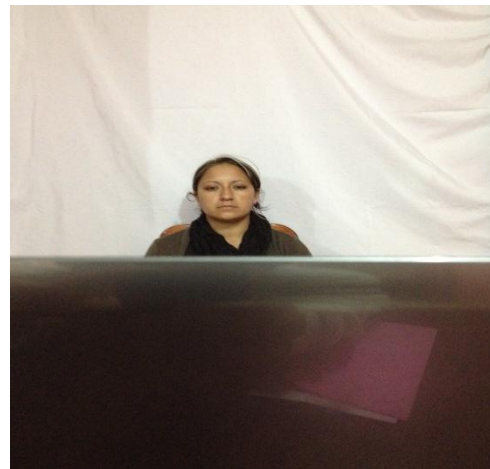


Imagen 1.2 Prueba de reconocimiento fondo blanco

217

Distancia(mm)	600	750	1000	1250	1500
Intensidad(Lux)					
2LX	EN+/NR	EN+/NR	EN/NR	EN-/NR	NE/NR
8LX	EN++/RE	EN++/RE	EN/RE-.-	EN/RE-.-	EN-./RE-.-.-
19LX	EN++/RE+	EN++/RE++	EN/RE	EN/RE-	EN-/RE-.-.-
27LX	EN++/RE+++	EN++/RE++	EN/RE+	EN/RE	EN/RE-.-

Tabla 1.2 Resultados pruebas de reconocimiento fondo blanco

Fondo Color Negro

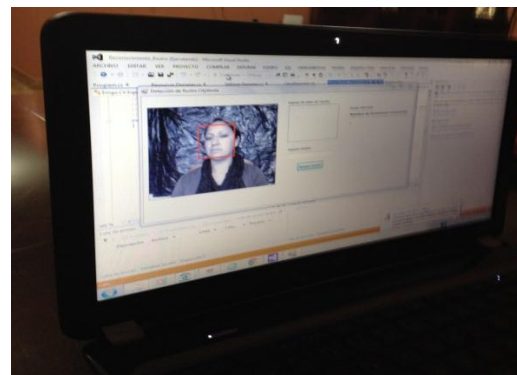
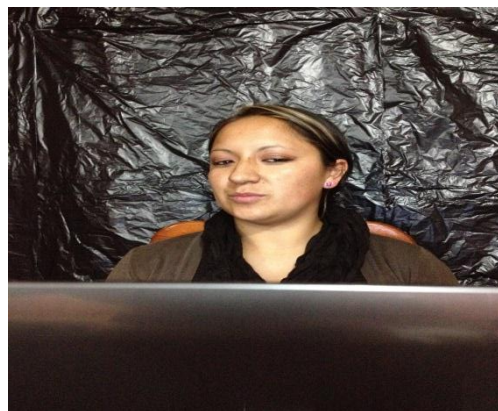


Imagen 1.3 Prueba reconocimiento fondo negro

218

Distancia(mm)	600	750	1000	1250	1500
Intensidad(Lux)					
2LX	EN/NR	EN/NR	EN-./NR	EN-./NR	NE/NR
8LX	EN/NR	EN/NR	EN-/NR	EN-./NR	EN-/NR
19LX	EN/RE-.-	EN/RE-.-	EN/RE-.-	EN-./RE-.-	EN/RE-.-
27LX	EN/RE-.-.-	EN/RE-	EN/RE-	EN/RE-	EN/NR

Tabla 1.3 Resultados pruebas de reconocimiento fondo negro

2. Ejecución de casos de prueba para registro de asistencia.

Según el cronograma presentado se realiza las pruebas del sistema en la siguiente secuencia:

Caso de Prueba CP-0001 Registro de usuario.

- **Prueba de Caja Blanca.**

Método: btnBuscar()

Funcionalidad: Verifica si el usuario está registrado en el sistema web

```
void btnBuscar_Click(object sender, EventArgs e) {
    variable = new VariablesInternas() // llamada al web Service;
    _usuario= variable.GetUsuarioByCedula(sCedula);
    if (_usuario.CEDULA_USUARIO != null)
    {
        txtNombres.Text = _usuario.NOMBRES_USUARIO;
        face_PICBX.Visible = true;
    }
    else
    {
        lblMensaje.Text = "No se encontró usuario registrado";
    }
}
```

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	Training Form	Vallor Null (Press Validar)	No encontró usuario registrado	lblMensaje.Text = "No se encontró usuario registrado";
PCB-0002	Training Form	Datos ingresados usuario registrado smoya@cv.com ccc123 (Press Validar)	Nombre de usuario Imagen conversión escala de grises	txtNombres.Text = _usuario.NOMBRES_USUARIO; face_PICBX.Visible = true;

Tabla 1.4 Pruebas de caja blanca CP-0001 sistema Windows

- **Pruebas de Caja Negra.**

Prueba1.- Entradas valores acordes a cada campo

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Correo electrónico	“smoya@cv.com”			
	contraseña	“ccc123”			
	Press Validar		Nombre de Usuario. Imagen escala de Grises	0.0002	txtNombres.Text = _usuario.NOMBRES_USUARIO; face_PICBX.Visible = true;

Tabla 1.5 Prueba 1 de caja negra para CP-0001 sistema Windows

Prueba2.- Registro de rostro

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Número de cédula de usuario	1714542048	“Santiago Moya” Imagen escala de Grises	0.002	
	Añadir imagen		Imagen registrada	0.002	face_data.Save(Application.StartupPath + "/TrainedFaces/" + facename, ImageFormat.Jpeg);

Tabla 1.6 Prueba 2 de caja negra para CP-0001 sistema Windows

Caso de Prueba CP-0002 Registro Asistencia de Horario entrada.

- **Prueba de Caja Blanca.**

Método: registrarAsistencia()

Funcionalidad: Ingreso de horario de entrada

```
string mensaje=variables.registrarAsistencia(sNumeroCedula,tipo, 1);

//MÉTODO DEL WEB SERVICE

public string registrarAsistencia(string sNumeroCedula, int tipo, int idAgencia) {
if (verificoHoraEntrada.Count() == 0)
{
// verifico si existe atraso en l hora de entrada
TimeSpan HorarioEntrada= new TimeSpan();
HorarioEntrada = query.Select(m =>
m.HORARIOEMPLEADOS.HORARIOS.DESDE_HORARIO).FirstOrDefault();
HorarioEntrada= HorarioEntrada.Add(tolerancia_atrasos);
if (HorarioEntrada < hora_registro)
{
TimeSpan dif= hora_registro- HorarioEntrada.Add(-tolerancia_atrasos);
atraso = new ATRASOS
{
ENTRADA_ATRASO = dif,
SALIDA_ATRASO = null
};
asistencia.ATRASOS.Add(atraso);
}
db.ASISTENCIAS.InsertOnSubmit(asistencia);
}
else
return "Ya tiene registrada la hora de entrada";
db.SubmitChanges();

return "Asistencia Registrada";
}
}
```

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	Web Service CityVenta	1714542048 08h00 (Press registrar)	Asistencia Registrada	MessageBox.Show(mensaje);
PCB-0002	Web Service CityVenta	1714542048 08h25 (Press registrar)	"Ya tiene registrada la hora de entrada";	MessageBox.Show(mensaje);

Tabla 1.7 Prueba caja blanca para CP-0002 sistema Windows

- **Pruebas de Caja Negra.**

Prueba1.- Usuario no asignado a horario de trabajo

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Seleccionar	Entrada 8h00 Domingo			El usuario tiene registro para lunes, miércoles y Viernes de 8h00 a 14:00
	Press button	Registrar asistencia	"ENTRADA No puede registrar asistencia porque no está en horario de trabajo";	0.002	MessageBox.Show

Tabla 1.8 Prueba 1 de caja negra para CP-0002 sistema Windows

Prueba2.- Usuario Atrasado

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Seleccionar	Entrada 8h15 miércoles			El usuario tiene registro para lunes, miércoles y Viernes de 8h00 a 14:00
	Press button	Registrar asistencia	"Horario registrado tiene un atraso"	0.002	MessageBox.Show

Tabla 1.9 Prueba 2 de caja negra para CP-0002 sistema Windows

Caso de Prueba CP-0003 Registro Asistencia de Horario salida.

- **Prueba de Caja Blanca.**

Método: registrarAsistencia()

Funcionalidad: Ingreso de horario de entrada

string mensaje=variables.registrarAsistencia(sNumeroCedula,tipo, 1);

//MÉTODO DE WEB SERVICE

```

public string registrarAsistencia(string sNumeroCedula, int tipo, int idAgencia) {

    TimeSpan HorarioSalida= new TimeSpan();
    HorarioSalida = query.Select(m =>
    m.HORARIOEMPLEADOS.HORARIOS.HASTA_HORARIO).FirstOrDefault();
    HorarioSalida= HorarioSalida.Add(-tolerancia_atrasos);
    if (HorarioSalida > hora_registro)
        {
            TimeSpan dif= HorarioSalida.Add(tolerancia_atrasos)-hora_registro ;
            atraso.SALIDA_ATRASO = dif;
            return " Registro su salida antes de tiempo ";
        }
    return "Asistencia Registrada";
}
    
```

CÓDIGO	UBICACIÓN	ENTRADAS	SALIDA ESPERADA	OBSERVACIÓN
PCB-0001	Web Service CityVenta	1714542048 14h00 (tomado automático del sistema) (Press registrar)	Asistencia Registrada	MessageBox.Show(mensaje);
PCB-0002	Web Service CityVenta	1714542048 13h25 (tomado automático del sistema) (Press registrar)	"Registro su salida antes de tiempo";	MessageBox.Show(mensaje);

Tabla 1.10 Prueba caja blanca para CP-0003 sistema Windows

- **Pruebas de Caja Negra.**

Prueba1.- Usuario no registra entrada y registra salida

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0001	Seleccionar	Salida 14h00 Domingo			El usuario tiene registro para lunes, miércoles y Viernes de 8h00 a 14:00, pero no registro hora de entrada.
	Press button	Registrar asistencia	"ENTRADA No puede registrar asistencia porque no registro entrada";	0.002	MessageBox.Show

Tabla 1.11 Prueba 1 de caja negra para CP-0003 sistema Windows

Prueba2.- Usuario registra hora antes de cumplir su horario

CÓDIGO	ENTRADAS		SALIDA ESPERADA	RESPUESTA (Tiemp. (seg))	OBSERVACIÓN
PCN-0002	Seleccionar	Salida 13h15 Viernes			El usuario tiene registro para lunes, miércoles y Viernes de 8h00 a 14:00
	Press button	Registrar asistencia	“Horario registrado tiene Usted salió antes de tiempo”	0.002	MessageBox.Show

Tabla 1.12 Prueba 2 de caja negra para CP-0003 sistema Windows

Anexo 18. Pruebas finales de software

**ANALIZAR, DISEÑAR Y CONSTRUIR UN SISTEMA
INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL
HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE
LA EMPRESA “CITYVENTA”, MEDIANTE
IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN
REMOTA DE INFORMACIÓN.**

Ejecución casos de pruebas finales de software

Versión [4.0]

Casos de Pruebas finales de Software

Elaborado por: Santiago Moya/ Jimena Morales

Proyecto: SISTEMA INFORMÁTICO HÍBRIDO, PARA EL MANEJO DEL HORARIO DE INGRESO Y SALIDA DEL PERSONAL DE LA EMPRESA “CITYVENTA”, MEDIANTE IDENTIFICACIÓN DE ROSTROS Y ADMINISTRACIÓN REMOTA DE INFORMACIÓN.

Ciclo de Pruebas: Pruebas finales de Software

Id	Caso de Prueba	Descripción	Fecha	Área Funcional / Sub proceso	Funcionalidad / Característica	Datos / Acciones de Entrada	Resultado Esperado
CP-001	Inicio de Sesión – registro	Ingresar usuario	12/01/2015	Todas las áreas	Ingreso al sistema	Datos en blanco	Mensaje de correo y contraseña requerida
CP-002	Inicio de Sesión – registro	Ingresar usuario	12/01/2015	Todas las áreas	Ingreso al sistema	Ingreso solo correo	Mensaje de contraseña requerida
CP-003	Inicio de Sesión – registro	Ingresar usuario	12/01/2015	Todas las áreas	Ingreso al sistema	Ingreso dato que no es correo	Mensaje ingrese correo válido
CP-004	Inicio de Sesión – registro	Ingresar usuario	12/01/2015	Todas las áreas	Ingreso al sistema	Ingresar contraseña inválida	Mensaje error inicio de sesión
CP-005	Inicio de Sesión – registro	Ingresar usuario	12/01/2015	Todas las áreas	Ingreso al sistema	Ingreso correo y contraseña valida	Ingreso a Home personalizado del sistema
CP-006	Olvido su contraseña	Usuario olvido su contraseña	12/01/2015	Todas las áreas	Recuperar contraseña	Ingreso correo inválido	Mensaje de correo no registrado
CP-007	Olvido su contraseña	Usuario olvido su contraseña	12/01/2015	Todas las áreas	Recuperar contraseña	Ingreso correo válido	Envía correo electrónico al usuario
CP-008	Cambio de contraseña	Cambio de contraseña	12/01/2015	Todas las áreas	Cambiar contraseña	Ingreso en blanco de datos	Mensajes de ingreso de datos

CP-009	Cambio de contraseña	Cambio de contraseña	12/01/2015	Todas las áreas	Cambiar contraseña	Ingreso solo contraseña actual los otros datos en blanco	Mensajes de ingreso de nueva contraseña
CP-010	Cambio de contraseña	Cambio de contraseña	12/01/2015	Todas las áreas	Cambiar contraseña	Ingreso contraseña actual e información de contraseña en un casillero no se ratifica la contraseña	Mensajes de confirme su nueva contraseña
CP-011	Cambio de contraseña	Cambio de contraseña	12/01/2015	Todas las áreas	Cambiar contraseña	Ingreso adecuado de datos para cambio de contraseña	Ingreso a Home personalizado del sistema
CP-012	Ingreso con contraseña modificada	Ingresar usuario con contraseña modificada	12/01/2015	Todas las áreas	Ingreso al sistema	Ingreso de nueva contraseña	Ingreso a Home personalizado del sistema
CP-013	Ingreso de nuevo perfil	Crear nuevo perfil	14/01/2015	Talento Humano	ingreso perfil	Datos en blanco	Mensaje "Campo nombre perfil requerido"
CP-014	Ingreso de nuevo perfil	Crear nuevo perfil	14/01/2015	Talento Humano	ingreso perfil	Ingreso de nombre de perfil	Mensaje "Grabado exitoso"
CP-015	Actualización de perfil	Actualizar perfil	14/01/2015	Talento Humano	Actualizar perfil	Ingreso de descripción de perfil	Mensaje "Grabado exitoso"
CP-016	Actualización de perfil	Actualizar perfil	14/01/2015	Talento Humano	Actualizar perfil	Ingreso de nombre de perfil existente para actualización	Mensaje "Ya existe un perfil con el mismo nombre"

CP-017	Ingreso de perfil existente	Crear perfil	14/01/2015	Talento Humano	Crear perfil	Ingreso de nombre de perfil existente	Mensaje "Ya existe un perfil con el mismo nombre"
CP-018	Eliminar perfil existente	Eliminar perfil	14/01/2015	Talento Humano	Eliminar perfil	Pulsar eliminar perfil	Mensaje "Está seguro de que quiere eliminar el registro"
CP-019	Asignar permisos al perfil	Asignar permisos	14/01/2015	Talento Humano	Asignar permisos	Seleccionar perfiles	Retorno a la vista de perfiles
CP-020	Prueba de Asignación permisos al perfil	Verificar permisos	14/01/2015	Usuario/ Asistente Gerencia	Prueba de permisos	Ingreso de perfil a usuario / Ingreso de usuario y contraseña	Vista personalizada del Usuario
CP-021	Prueba de Asignación permisos al perfil	Verificar permisos	14/01/2015	Usuario/ Empleado	Prueba de permisos	Ingreso de perfil a usuario / Ingreso de usuario y contraseña	Vista personalizada del Usuario
CP-022	Prueba de Asignación permisos al perfil	Verificar permisos	14/01/2015	Usuario/ Asistente de talento humano	Prueba de permisos	Ingreso de perfil a usuario / Ingreso de usuario y contraseña	Vista personalizada del Usuario
CP-023	Prueba de Asignación permisos al perfil	Verificar permisos	14/01/2015	Usuario/ Jefe de Personal	Prueba de permisos	Ingreso de perfil a usuario / Ingreso de usuario y contraseña	Vista personalizada del Usuario

CP-024	Prueba de Asignación permisos al perfil	Verificar permisos	14/01/2015	Usuario/ Personal contable	Prueba de permisos	Ingreso de perfil a usuario / Ingreso de usuario y contraseña	Vista personalizada del Usuario
CP-025	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	Datos en blanco	Mensaje de campos requeridos
CP-026	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	Ingreso de números campo Nombres y apellidos	Mensaje "Ingrese solo texto"
CP-027	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	Ingreso de caracteres especiales en campo Nombres y apellidos	No muestra mensaje acepta caracteres
CP-028	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	Ingreso de letras en campo cédula	Mensaje "Campo solo puede ingresar números"
CP-029	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso de datos sin @	Mensaje "ingrese un correo válido"
CP-030	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso de letras en edad	Mensaje "ingrese números"
CP-031	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso de edad menor a 18	Mensaje "La edad debe estar en rango de 18 a 65 años"
CP-032	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso de edad menor a 18	Mensaje "La edad debe estar en rango de 18 a 65 años"

CP-033	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso número telefónico de 3 dígitos	Mensaje "Ingrese un número telefónico correcto"
CP-034	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso número celular 4 dígitos	Mensaje "Ingrese un número celular correcto"
CP-035	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	ingreso cédula errónea	Mensaje "número cedula incorrecto"
CP-036	Ingreso nuevo usuario	Crear usuario	20/01/2015	Talento Humano	Ingreso de usuario	Ingreso datos adecuados	Mensaje "Grabado Exitoso"
CP-037	Editar datos de usuario	Editar usuario	20/01/2015	Talento Humano	Editar usuario	Ingreso de números campo Nombres y apellidos	Mensaje "Ingrese solo texto"
CP-038	Editar datos de usuario	Editar usuario	20/01/2015	Talento Humano	Editar usuario	Ingreso de letras en campo cédula	Mensaje "Campo solo puede ingresar números"
CP-039	Editar datos de usuario	Editar usuario	20/01/2015	Talento Humano	Editar usuario	Borrado de ".com" del email	Mensaje "Ingrese correo electrónico válido"
CP-040	Editar datos de usuario	Editar usuario	20/01/2015	Talento Humano	Editar usuario	Cambio de número en la cédula	Mensaje "número de cédula incorrecto"
CP-041	Asignar perfil a usuario creado	Asignar perfil	20/01/2015	Talento Humano	Asignar permisos de navegación	Selección perfil Empleado ingreso con correo y contraseña	Ingreso a Home personalizado del sistema para empleado

CP-042	Eliminar usuario	Quitar usuario	20/01/2015	Talento Humano	Eliminar usuario	Presionar botón eliminar	Mensaje "Está seguro de que quiere eliminar el registro"
CP-043	Ingresar Agencias	Crear agencia	23/01/2015	Asistente Gerencia	Crear nueva Agencia	Datos en blanco	Mensaje "El campo nombre es requerido"
CP-044	Ingresar Agencias	Crear agencia	23/01/2015	Asistente Gerencia	Crear nueva Agencia	Ingreso nombre de agencia	Mensaje "Grabado exitoso"
CP-045	Actualizar Agencias	Actualizar agencia	23/01/2015	Asistente Gerencia	Modificar Agencia	Cambio de ciudad	Mensaje "Grabado exitoso"
CP-046	Actualizar Agencias	Actualizar agencia	23/01/2015	Asistente Gerencia	Modificar Agencia	Cambio de dirección	Mensaje "Grabado exitoso"
CP-047	Eliminar Agencia	Eliminar agencia	23/01/2015	Asistente Gerencia	Eliminar Agencia	Presionar botón eliminar	Mensaje "Está seguro de que quiere eliminar el registro"
CP-048	Ingresar Ciudad	Crear Ciudad	23/01/2015	Asistente Gerencia	Crear nueva Ciudad	Datos en blanco	Mensaje "El campo ciudad es requerido"
CP-049	Ingresar Ciudad	Crear Ciudad	23/01/2015	Asistente Gerencia	Crear nueva Ciudad	Ingreso Ciudad	Mensaje "Grabado exitoso"
CP-050	Actualizar Ciudad	Actualizar Ciudad	23/01/2015	Asistente Gerencia	Actualizar Ciudad	Cambio nombre de ciudad	Mensaje "Grabado exitoso"
CP-051	Eliminar Ciudad	Eliminar Ciudad	23/01/2015	Asistente Gerencia	Eliminar Ciudad	Presionar botón eliminar/ ACEPTAR	Mensaje "Está seguro de que quiere eliminar el registro"
CP-052	Horarios-Ingreso	Crear Horario	26/01/2015	Jefe de Personal	Crear Horario	Nombre en blanco	Mensaje "El campo nombre de horario es requerido"

CP-053	Horarios-Ingreso	Crear Horario	26/01/2015	Jefe de Personal	Crear Horario	Ingreso nombre no selecciona Horario	Mensaje "El valor de hora inicio y fin no son válidos"
CP-054	Horarios-Ingreso	Crear Horario	26/01/2015	Jefe de Personal	Crear Horario	Ingreso nombre y selecciona Horario entrada y salida	
CP-055	Horarios-Atualizar	Actualizar Horario	26/01/2015	Jefe de Personal	Actualizar Horario	Cambio nombre de Horario	Mensaje "Grabado exitoso"
CP-056	Horarios-Atualizar	Actualizar Horario	26/01/2015	Jefe de Personal	Actualizar Horario	Cambio Horas	Mensaje "Grabado exitoso"
CP-057	Horarios-Eliminar	Eliminar Horario	26/01/2015	Jefe de Personal	Eliminar Horario	Presionar botón eliminar/ ACEPTAR	Mensaje "Está seguro de que quiere eliminar el registro"
CP-058	Jornada Laboral-Ingresar	Crear Jornada Laboral	26/01/2015	Jefe de Personal	Ingresar Jornada Laboral	Seleccionar Empleado y jornada laboral	Mensaje "Grabado exitoso"
CP-059	Jornada Laboral-Ingresar	Crear Jornada Laboral	26/01/2015	Jefe de Personal	Ingresar Jornada Laboral	Seleccionar mismos Empleado y jornada laboral	Mensaje "Ya existe una Jornada Laboral asignada al mismo empleado"
CP-060	Jornada Laboral-Ingresar	Crear Jornada Laboral	26/01/2015	Jefe de Personal	Ingresar Jornada Laboral	Seleccionar Horario que se Cruza en jornada Laboral y en la misma agencia	Mensaje "Grabado exitoso" (NOTA: Deja ingresar la jornada se verifica cruce cuando asigna días)

CP-061	Jornada Laboral- Ingresar	Crear Jornada Laboral	26/01/2015	Jefe de Personal	Ingresar días Jornada Laboral	Seleccionar día que se Cruza en jornada Laboral y en la misma agencia	Mensaje "Error: Existe un cruce de horarios en un o más día seleccionado favor revise nuevamente el ingreso"
CP-062	Jornada Laboral- Actualizar	Actualizar Jornada Laboral	26/01/2015	Jefe de Personal	Actualizar Jornada Laboral	Cambio de agencia mismos horario	Mensaje "Grabado exitoso"
CP-063	Jornada Laboral- Actualizar	Actualizar Jornada Laboral	26/01/2015	Jefe de Personal	Actualizar días Jornada Laboral	Cambio de días	Mensaje "Usted tiene un cruce de Horario"
CP-064	Jornada Laboral- Eliminar	Eliminar Jornada Laboral	26/01/2015	Jefe de Personal	Eliminar jornada laboral asignada	Presionar botón eliminar/ ACEPTAR	Mensaje "Está seguro de que quiere eliminar el registro"
CP-065	Asignación de horarios-Consultas	Consulta de Jornada Laboral	26/01/2015	Empleado	Consulta de Jornada laboral	Selección de Todas las agencias no selecciona empleado	Mensaje "Seleccione empleado que desea filtrar"
CP-066	Asignación de horarios-Consultas	Consulta de Jornada Laboral	26/01/2015	Empleado	Consulta de Jornada laboral	Selección de Todas las agencias y empleados	Despliega en orden Alfabético los usuarios y agencias asignadas
CP-067	Asignación de horarios-Consultas	Consulta de Jornada Laboral	26/01/2015	Empleado	Consulta de Jornada laboral	Selección de agencias "El bosque" y todos los empleados	Despliega en orden Alfabético los usuarios que están asignados a la Agencia
CP-068	Asignación de horarios-Consultas	Consulta de Jornada Laboral	26/01/2015	Empleado	Consulta de Jornada laboral	Selección nombre de empleado y todas las agencias	Despliega nombre empleado y agencias en donde fue asignado

CP-069	Asignación de horarios-Consultas	Consulta de Jornada Laboral	26/01/2015	Empleado	Consulta de Jornada laboral	Selección nombre de empleado y agencia "El bosque"	Muestra los horarios asignados en la agencia el bosque para el empleado.
--------	----------------------------------	-----------------------------	------------	----------	-----------------------------	--	--

Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Asignar Perfiles	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		
Google Chrome/ Firefox	Ninguno	Ninguno	Exitoso	OK		

