***Diseño y desarrollo de un sistema de información para el control de asistencia de la ficha 683019 en el SENA***

***Natalia Pulido***

***Esneider Rojas***

***Jennifer Romani***

***Juan Saboyá***

***Cindy Saldaña***

***ADSI 683019***



***ADSI 683019***

***Jenn***

*2015*

**DISEÑO Y DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA EL CONTROL DE ASISTENCIA DE LA FICHA 683019 EN EL SENA**

**JESSICA NATALIA PULIDO LEÓN**

**ARLINSON ESNEIDER ROJAS LUNA**

**JENNIFER ANDREA ROMANI JAMAICA**

**JUAN DAVID SABOYÁ JIMÉNEZ**

**CINDY JINETH SALDAÑA GUTIÉRREZ**

**SERVICIO NACIONAL DE APRENDIZAJE (SENA)**

**CENTRO DE DISEÑO Y METROLOGÍA**

**ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**FICHA 683019**

**BOGOTÁ D.C.**

**2015**

*Contenido*

[1. Introducción 10](#_Toc429667191)

[2. Planteamiento del problema 11](#_Toc429667192)

[3. Justificación 12](#_Toc429667193)

[4. Objetivos 13](#_Toc429667194)

[4.1 Objetivo general: 13](#_Toc429667195)

[4.2 Objetivos específicos: 13](#_Toc429667196)

[5. Estado del arte 14](#_Toc429667197)

[6. Marco teórico 22](#_Toc429667198)

[6.1 Herramientas web para uso académico 22](#_Toc429667199)

[6.2 Tecnologías de desarrollo 25](#_Toc429667200)

[6.3 Lenguaje de programación 33](#_Toc429667201)

[7. Metodologías de desarrollo de software 37](#_Toc429667202)

[7.1 Planificación de requisitos 37](#_Toc429667203)

[7.1.1 Requerimientos funcionales y no funcionales 38](#_Toc429667204)

[7.1.1.1 Requerimientos funcionales 38](#_Toc429667205)

[7.1.1.2 Requerimientos no funcionales 39](#_Toc429667206)

[7.1.2 Planeación 40](#_Toc429667207)

[7.2 Diseño 49](#_Toc429667208)

[7.2.1 Modelo entidad relación (MER) 49](#_Toc429667209)

[7.2.2 Diccionario de datos 50](#_Toc429667210)

[7.3 Construcción 51](#_Toc429667211)

[7.3.1 Base de datos 51](#_Toc429667212)

[7.4 Implementación 53](#_Toc429667213)

[7.4.1 Diagrama de despliegue 53](#_Toc429667214)

[8. Ingeniería del producto 54](#_Toc429667215)

[8.1 Diagramas de distribución 54](#_Toc429667216)

[8.2 Servicios web 55](#_Toc429667217)

[8.2.1 Diagrama de clases 55](#_Toc429667218)

[8.2.2 Diagramas casos de uso 56](#_Toc429667219)

[8.2.3 Diagramas de colaboración 77](#_Toc429667220)

[8.4 Aplicación web - base de datos 88](#_Toc429667221)

[8.5 Aplicación cliente 88](#_Toc429667222)

[8.5.1 Diagramas de actividades 88](#_Toc429667223)

[8.5.2 Diagramas de secuencia 100](#_Toc429667224)

[8.5.3 Diagramas de estado 111](#_Toc429667225)

[9. Recomendaciones 114](#_Toc429667226)

[10. Conclusiones 115](#_Toc429667227)

[11. Referencias 116](#_Toc429667228)

[12. Webgrafía 117](#_Toc429667229)

[13. Anexos 117](#_Toc429667230)

**LISTA DE ILUSTRACIONES**

[Ilustración 1: Gráfica de Componentes de una base de datos. 26](#_Toc428433860)

[Ilustración 2: Entidades, atributos, relaciones 29](#_Toc428433861)

[Ilustración 3: Gráfica de Ejemplo Modelo Entidad-Relación (E-R). 30](#_Toc428433862)

[Ilustración 4: MER 46](#_Toc428433863)

[Ilustración 5: Diccionario de datos 47](#_Toc428433864)

[Ilustración 6: Diagrama de despliegue 50](#_Toc428433865)

[Ilustración 7: Diagrama de distribución 51](#_Toc428433866)

[Ilustración 8: Diagrama de clases 52](#_Toc428433867)

[Ilustración 9: Caso de uso general administrador 53](#_Toc428433868)

[Ilustración 10: Caso de uso general Instructor 53](#_Toc428433869)

[Ilustración 11: Caso de uso general Aprendiz 54](#_Toc428433870)

[Ilustración 12: Caso de uso 001, 002, 003 54](#_Toc428433871)

[Ilustración 13: Caso de uso 004, 005, 006 56](#_Toc428433872)

[Ilustración 14: Caso de uso 007 58](#_Toc428433873)

[Ilustración 15: Caso de uso 008 59](#_Toc428433874)

[Ilustración 16: Caso de uso 009 61](#_Toc428433875)

[Ilustración 17: Caso de uso 010 63](#_Toc428433876)

[Ilustración 18: Caso de uso 011, 012 65](#_Toc428433877)

[Ilustración 19: Caso de uso 013 67](#_Toc428433878)

[Ilustración 20: Caso de uso 014 69](#_Toc428433879)

[Ilustración 21: Caso de uso 71](#_Toc428433880)

[Ilustración 22: Caso de uso 016 73](#_Toc428433881)

[Ilustración 23: Diagrama de colaboración Registro de ingreso 75](#_Toc428433882)

[Ilustración 24: Diagrama de colaboración Generar reporte 76](#_Toc428433883)

[Ilustración 25: Diagrama de colaboración Ingresar al sistema 77](#_Toc428433884)

[Ilustración 26: Diagrama de colaboración Generar reporte aprendiz 78](#_Toc428433885)

[Ilustración 27: Diagrama de colaboración Modificar usuario 79](#_Toc428433886)

[Ilustración 28: Diagrama de colaboración Registrar usuario 80](#_Toc428433887)

[Ilustración 29: Diagrama de colaboración Inicio de clase 81](#_Toc428433888)

[Ilustración 30: Diagrama de colaboración Restaurar contraseña 82](#_Toc428433889)

[Ilustración 31: Diagrama de colaboración Cambiar contraseña 83](#_Toc428433890)

[Ilustración 32: Diagrama de colaboración Consultar usuario 84](#_Toc428433891)

[Ilustración 33: Diagrama de colaboración Eliminar usuario 85](#_Toc428433892)

[Ilustración 34: Diagrama de actividades Registro de ingreso 86](#_Toc428433893)

[Ilustración 35: Diagrama de actividades Generar reporte 87](#_Toc428433894)

[Ilustración 36: Diagrama de actividades Ingreso al sistema 88](#_Toc428433895)

[Ilustración 37: Diagrama de actividades Generar reporte aprendiz 89](#_Toc428433896)

[Ilustración 38: Diagrama de actividades registrar usuario 91](#_Toc428433897)

[Ilustración 39: Diagrama de actividades Inicio de clase 92](#_Toc428433898)

[Ilustración 40: Diagrama de actividades Restaurar contraseña 93](#_Toc428433899)

[Ilustración 41: Diagrama de actividades Cambiar contraseña 94](#_Toc428433900)

[Ilustración 42: Diagrama de actividades Consultar usuario 95](#_Toc428433901)

[Ilustración 43: Eliminar usuario 96](#_Toc428433902)

[Ilustración 44: Diagrama de secuencia Registro de ingreso 97](#_Toc428433903)

[Ilustración 45: Diagrama de secuencia Generar reporte 98](#_Toc428433904)

[Ilustración 46: Diagrama de secuencia Ingreso al sistema 99](#_Toc428433905)

[Ilustración 47: Diagrama de secuencia Generar reporte aprendiz 100](#_Toc428433906)

[Ilustración 48: Diagrama de secuencia Modificar usuario 101](#_Toc428433907)

[Ilustración 49: Diagrama de secuencia Registrar usuario 102](#_Toc428433908)

[Ilustración 50: Diagrama de secuencia Inicio de clase 103](#_Toc428433909)

[Ilustración 51: Diagrama de secuencia Restaurar contraseña 104](#_Toc428433910)

[Ilustración 52: Diagrama de secuencia Cambiar contraseña 105](#_Toc428433911)

[Ilustración 53 Consultar usuario 106](#_Toc428433912)

[Ilustración 54 Eliminar usuario 107](#_Toc428433913)

[Ilustración 55: Diagrama de estado Administrador 108](#_Toc428433914)

[Ilustración 56: Diagrama de estado Instructor 109](#_Toc428433915)

[Ilustración 57: Diagrama de estado Aprendiz 110](#_Toc428433916)

**LISTA DE ANEXOS**

**Anexo 1** Resultados encuesta a instructores SENA

**Anexo 2** Entrevista Ing. Marina Casallas Silva (líder programas ADSI).

**Anexo 3** Pruebas realizadas a la aplicación

**Anexo 4** Manual de usuario

**Anexo 5** Manual técnico

# 1. INTRODUCCIÓN

La tecnología es una de las herramientas actuales más importantes, que permite realizar distintas actividades necesarias para el ser humano como la comunicación, educación, entre muchas otras., hoy día se puede aplicar en todo lo que hacemos, ayudando y facilitando el diario vivir, desarrollando nuevas capacidades y a la vez nuevas tecnologías que sirven para brindar soluciones a las empresas, entidades y personas.

El propósito de este proyecto es elaborar un sistema de información que permita llevar el control de asistencia de los aprendices por medio del código de barras del carnet estudiantil, específicamente en el Servicio Nacional de Aprendizaje (SENA).

En el presente documento describe un problema, tomándolo como punto de partida para el análisis, desarrollo e implementación del proyecto. Consecutivamente se definen las fases del proyecto, entre esas se encuentra la fase de análisis donde se plantea y se define la solución más óptima a partir de estas se inicia una construcción del software en la fase de diseño, en la que se construyen los modelos del lenguaje de modelado visual UML, se termina con la etapa de implementación y pruebas.

Este documento muestra el desarrollo del sistema de información que ayuda al mejoramiento de los procesos, mediante un conjunto de módulos.

# 2. PLANTEAMIENTO DEL PROBLEMA

De las distintas experiencias en el proceso de formación con los aprendices ¿? durante 2 trimestres de estudio mediante la observación con respecto al control de asistencia de los aprendices, han mostrado que ésta se realiza de diferentes maneras. Por lo que se requiere analizar específicamente la manera adecuada del control de asistencia en el SENA, se realizó una encuesta a varios instructores. Los resultados muestran que la mayoría de los instructores entregan un formato a los aprendices para firmar, pero esto no garantiza que el aprendiz que firma se encuentre en el ambiente de formación, porque se puede presentar suplantación de identidad. Algunos, ingresan al aplicativo Sena Sofía, hacen el llamado de asistencia de forma verbal y en algunas ocasiones hasta 2 veces en el transcurso de la formación y esto hace que el instructor gaste mucho tiempo, haga doble trabajo y además el sistema tiene un procesamiento lento.

También se evidencia en la encuesta que una vez realizado el llamado de asistencia, la información se guarda de diferentes maneras. Algunos instructores guardan la información en un portafolio o carpeta, otros en memoria USB y otros en plataforma virtual lo cual demuestra que no hay una unificación en el proceso.

* Pregunta de investigación: ¿Cómo desarrollar un aplicativo web, que permita a los instructores realizar un control de asistencia sobre los aprendices en el SENA?

# 3. JUSTIFICACIÓN

Este proyecto nace dada la necesidad de sistematizar un proceso manual que se realiza actualmente en la institución SENA, en el que se realiza el control de asistencia de los aprendices, pero este proceso tiene algunas deficiencias como situaciones en las que se puede presentar suplantación de identidad, además se pierde mucho tiempo en este proceso.

El sistema Program Art consiste en un software que permite realizar el control de asistencia de los aprendices por medio del código de barras del carnet estudiantil.

El proyecto planteado es factible, porque el proceso de control de asistencia será dinámico, rápido y eficaz, como herramienta que se acopla a las necesidades del usuario lo cual se verá reflejado en lo siguiente:

• Los instructores tendrán un mayor control de los aprendices en cuanto a la asistencia.

• Manejo de la información de forma eficiente sobre las inasistencias de los aprendices.

• Existencia de una base de datos que podrá ser actualizada por trimestre, debido al ingreso de nuevos estudiantes se realiza cada 3 meses.

• Contribuye con la preservación del medio ambiente, porque se reduciría el uso del papel.

• Generar reportes de asistencias e inasistencias.

El perfil profesional, el conocimiento, la capacidad de dar soluciones, el hecho de estar involucrados en la problemática y la disposición de mejorar dichas carencias en este campo, además de conocer el enfoque que tiene el programa de Análisis y Desarrollo de sistemas de información (ADSI) respecto al estudio de los problemas y las necesidades presentes en el SENA, hacen de los estudiantes de ADSI las personas más idóneas para corregir las deficiencias de investigación y desarrollo tecnológico presentadas dentro de las instalaciones.

# 4. OBJETIVOS

## 4.1 Objetivo general:

Diseñar y desarrollar un aplicativo web, que permita a los instructores realizar un control de asistencia sobre los aprendices de la ficha 683019 en el SENA.

## 4.2 Objetivos específicos:

**Sociales**

* Analizar las dificultades que se presentan a la hora de llevar el control de asistencia en los aprendices al momento de iniciar la formación y de las tecnologías viables para diseñar la solución.

**Tecnológicos**

* Diseñar una aplicación web que se ajuste a las necesidades del usuario y brinde una solución eficaz de acuerdo a las dificultades analizadas anteriormente y en función de las tecnologías existentes en el mercado escogidas para dicha solución.
* Desarrollar la aplicación web de acuerdo con el diseño elaborado.
* Realizar pruebas al software para determinar información objetiva e independiente sobre la calidad del software.
* Adaptar al aplicativo los estándares de calidad relacionados a los procesos de desarrollo de software.

# 5. ESTADO DEL ARTE

**1) Control de acceso con verificación de identidad por medio de código de barras.**

Realizado por los estudiantes Iván Augusto Martínez y Carlos Enrique Montero de la universidad Javeriana en Bogotá D.C. en el año 2010. En este trabajo se encuentra una teoría sobre sistemas de control de acceso y los elementos con los que se trabajaba en ese momento, no solo con algo que la persona al identificarse posea (carné, tarjeta inteligente, etc.), sino con alguna característica física de las personas para ingresar a la institución como huella dactilar, ADN, tono de voz, etc. En ese momento estaban en auge las tarjetas inteligentes.

El termino tarjeta inteligente se usa de forma genérica para definir una serie de circuitos, cada uno de los cuales puede transportar datos en un dispositivo de reducidas dimensiones físicas, de forma que pueda ser fácilmente llevado en el bolsillo o en la cartera (Monsó,1994,p.157).

Por lo tanto, al pasar la tarjeta por el lector, éste debe verificar la huella dactilar como característica biométrica que se usa para identificar personas, dado que existe la suficiente investigación al respecto, además los dispositivos que se usan para ello tienen un costo económico.

En este proyecto se usó IDE de Visual Basic. Ramírez (2001) afirma: “El IDE de Visual Basic le permite diseñar la interfaz que utilizará para realizar (...). Esto lo logra agregando objetos a los formularios, estableciendo propiedades a los objetos, agregando código, etcétera.” (p.27). Este lenguaje tiene las facilidades necesarias para una integración óptima con las rutinas usadas por el lector de huellas dactilares y por la fácil interacción con el programa de manejo de base de datos, para lo cual se determinó Microsoft Access, que permite editar, mostrar y actualizar la información de una manera eficiente. De esta manera, soporta el tamaño requerido para la aplicación porque tiene el mismo motor de bases de datos que el lenguaje de programación.

Para las fotografías se maneja un tamaño de 240 x 320 pixeles y se trabaja en formato JPEG, el cual utiliza un espacio de memoria apropiado.

En consecuencia, el sistema automático de control de acceso multipunto, en el 2003 actualiza este proyecto. Desarrollo conjunto, de software y hardware, para controlar los puntos de acceso a la Facultad de Ingeniería.

Sin embargo, la información y los desarrollos obtenidos en dicho trabajo serán de importancia en cuanto que existe la posibilidad de no recibir las bases de datos de Registro Académico y de Recursos Humanos como archivos planos, sino en el formato originalmente dispuesto por ellos, Oracle. Finalmente, el programa en el que fue desarrollado el software también fue en el lenguaje Visual Basic.

**2) Proyecto: Análisis y diseño del software para el control de asistencia de estudiantes (STEEL).**

El proyecto “Análisis y diseño del software para el control de asistencia de estudiantes STEEL”tiene como propósito la elaboración de una aplicación web que permite realizar registro de datos de estudiantes, registro de asistencia y pagos ya sea por mensualidad, clase única o clase especial de la escuela de capoeira NATIVOS, permitiendo con manejo de la información de una manera óptima mejorando los procesos administrativos y ayudando a la toma de decisiones.

Realizado por el estudiante de la Corporación Universitaria Minuto de Dios Juan Carlos Alonso Bernate, en Bogotá D.C. en el año 2012.

Gracias a la implementación del sistema de información STEEL, la escuela tendrá una herramienta útil que le permite estar a la vanguardia y un paso delante de las demás escuelas de capoeira apoyando su administración y la tomas de decisiones.

El software STEEL está basado en la metodología MÉTRICA V3, es una Metodología de Planificación útil, para el Desarrollo y Mantenimiento de Sistemas de información. Promovida por el Ministerio de Administración Pública del gobierno español para la sistematización de actividades del ciclo de vida de los proyectos de software. Moliner (2005) afirma: “La metodología MÉTRICA Versión 3 ofrece a las organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software (...).” (p.71). Por medio de esta metodología se pueden definir las estrategias que se utilizan para el logro del objetivo propuesto; de esta manera también éste permite establecer satisfacer las necesidades y requerimientos de los usuarios, haciendo uso del estándar Información Tecnología / Software LifeCycleProcess (ISO/IEC 12207) en donde se incluyen las actividades que son aplicadas desde la definición de los requisitos, la adquisición y configuración del sistema hasta la culminación del mismo, proporcionando de esta manera una estructura adecuada tanto para los usuarios, programadores, desarrolladores entre otros individuos que interactúan con el sistema.

El desarrollo de éste proyecto nace de la necesidad de poder ejercer un control y un apoyo a la escuela de capoeira NATIVOS en los procesos que cotidianamente se llevan a cabo, como lo son el pago de mensualidad, registro de asistencia y datos de los alumnos entre otros. En los ya mencionados procesos se han evidenciado fallas graves en el seguimiento de éstos, tales como la perdida de información constante, la escuela no cuenta con una herramienta tecnológica que permita un control y gestión en los procesos que realiza la administración, perdida del registro de asistencia de estudiantes a clases, afectando el buen servicio a los estudiantes y la sostenibilidad de la escuela.

La norma ISO/IEC 15504 SPICE Software Process Improvement Capability Determination, (determinación de la Capacidad de Mejora del Proceso Software) es un modelo para la evaluación y mejora de los diferentes procesos de desarrollo y mantenimiento de los sistemas de información de productos de software. Por lo tanto, el desarrollo del software STEEL se basa en etapas claves que posee su metodología, debido a que está estructurada principalmente en el desarrollo y construcción de proyectos de software.

**3) Estudio de factibilidad para el control de acceso biométrico en una empresa empleando lectores de huella digital.**

La tecnología actualmente brinda herramientas para establecer distintos niveles de seguridad en industrias, contra fenómenos naturales, seguridad informática e identificación y acceso de personal. De ahí, que el siguiente proyecto denominado Estudio de factibilidad para el control de acceso biométrico en una empresa empleando lectores de huella digital, desarrollado por estudiantes de la Universidad de la Salle en Bogotá D.C,se enfoca a la identificación y acceso de personal, esto porque las compañías actualmente están implementando sistemas que facilitan el acceso a partir de información propia de cada usuario. Dentro de este campo existen varias alternativas que brindan soluciones para cada una de las necesidades, así por ejemplo cabe mencionar métodos como fingerprint (escaneo de la huella digital), escaneo del iris, reconocimiento de voz y reconocimiento a través de tarjetas magnéticas.

La identificación biométrica es uno de los avances más importantes dentro del control y reconocimiento de personal perteneciente a una entidad sin importar su actividad económica, por tal motivo, es necesario conocer a qué se refiere cuando se habla de biometría. Este concepto de ***biometría*** tiene su etimología en las palabras bio (vida) y metría (medida). Pinto (como se citó en la Revista InfoWeek, 2009) explica que, "la biometría dactilar tiene la particularidad de que si voy a realizar una transacción bancaria" (...). Ahora le pongo la barrera a alguien que quiera realizar un fraude". De este modo, la biometría es una tecnología basada en el reconocimiento de una característica de seguridad y en el reconocimiento de una característica física e intransferible de las personas, como por ejemplo la huella digital.

Actualmente, para las pequeñas empresas el control de acceso de personal es realizado manualmente o podría decirse lo que siempre ha venido siendo convencional, teniendo en la entrada, es decir en la puerta de ingreso a las instalaciones un vigilante; la cantidad de personas empleadas o visitantes que ingresan a sus instalaciones puede ser muy elevado en el día. Para que el vigilante pueda diferenciar qué personas tienen acceso autorizado o no a las instalaciones, el usuario debe presentar su carné siendo el reconocimiento visual la principal forma de control de acceso de los empleados. Para controlar la hora de ingreso de los mismos, existe el método de marcación de tarjeta que con un reloj mecánico marca la hora tanto de entrada como de salida del personal.

En consecuencia, este proceso de verificación es lento e inseguro, dado que la información está almacenada en planillas que pueden llegar a ser cientos de hojas en un mes y que no presenta ningún tipo de organización de datos, además que no es veraz que la hora de ingreso y salida plasmada en estás planillas no es confiable porque el personal encargado del control de acceso (vigilante) carece de la información detallada de los usuarios, para determinar quiénes tienen acceso y quiénes no. Cuando se producen los cambios de turno, y un vigilante o recepcionista es nuevo, éste no tiene la capacidad de saber qué personas tienen autorizado el ingreso y sólo con el tiempo puede dominar el ingreso de las personas a partir del reconocimiento visual. De tal manera, que el sistema actual no proporciona información detallada y organizada del personal que hace parte de la institución.

**4) Proyecto** CATEBus: control de asistencia en transporte escolar.

Proyecto realizado por el D. Javier Sanabria Fernández de la universidad Carlos III de Madrid en el año 2011.

Debido a la proliferación de centros educativos concertados/privados a las afueras de las ciudades, se ha hecho necesaria la utilización del transporte escolar como parte fundamental de la vida diaria de los alumnos. Este sistema de transporte, costeado en gran medida por los padres de los alumnos, permite que, parando en ciertos puntos de las ciudades (realizando trazados específicos denominados rutas), los alumnos puedan esperar a su llegada para poder ser transportados al centro educativo. De esta forma, se concretan diversas horas en las que deben pasar a recoger a los alumnos, teniendo en cuenta factores tales como la distancia al centro, el flujo de tráfico, etc.

Gracias a este sistema, los padres de los alumnos pueden delegar en el centro el transporte de sus hijos, debido tanto al nivel de vida como a las necesidades de las familias, muchos padres trabajadores no tienen a quien delegar el transporte de sus hijos a los centros. Con la aparición de las rutas, los padres pueden acompañar a sus hijos a las paradas de las mismas antes de dirigirse a sus puestos de trabajos, de forma que mantienen controlados a los mismos, con la seguridad de es transporte directo al centro, en el que estarán controlados.

Pero al tratarse del transporte de escolares, pueden surgir numerosos casos en los que el alumnado no pueda acudir a tomarx la ruta de transporte a la hora acordada. Dichos casos se deben a que los alumnos se retrasan, bien porque se duerman, se entretengan en sus actividades antes de acudir al centro (desayunar, vestirse, etc.). En estos casos, los padres son conscientes de que los alumnos no asistirán a clase en dichas ocasiones, o que se pueden retrasar en acudir al punto de parada de la ruta (dentro de un margen de tiempo antes de que empiecen las clases). En estos casos, los padres acuden a la parada de la ruta y comprueban que no hay ningún alumno más esperando, lo que significa que la ruta ya ha pasado, y que deben ser ellos quienes transporten a sus hijos al centro.

Otras situaciones en las que se pueden dar casos parecidos, es en aquéllos donde la ruta pasa por las paradas antes de la hora prevista (por condiciones de buen tráfico, por ejemplo). Si se pudiese realizar un control en el que los padres puedan avisar de estos hechos y que las rutas pudiesen esperar a estos casos, tanto los padres (que harían pleno uso del servicio de transporte) y el centro (que no tendría que controlar los alumnos que acuden al PROYECTO CATEBus).

**5) Proyecto:** Implantación de código de barras en un almacén de electrodomésticos en México D.F.

Este proyecto tiene como objetivo ser una guía para la implementación y resolución de problemas para el control de un almacén aplicable en la industria logística. La presente obra, fue elaborada por Ricardo Gabriel Guzmán Martínez, en México D.F. y realizada como parte del trabajo laboral dentro de Mabe S.A. de C.V. como líder de Proyecto de Código de Barras y la experiencia adquirida en la implementación y arranque del proyecto ha sido útil para plasmarlo en este documento.

El objetivo de esta obra aparte de describir la tecnología de código de barras, busca describir los puntos a considerar en la operación logística para el control de producto, propuestas para su mejor aplicación y las posibles áreas de oportunidad para lograr aprovechamiento.

Adicionalmente es una herramienta importante que sirve previo a la implementación de un control de almacén con recursos propios o con la colaboración de un proveedor especializado.

La metodología utilizada es en base a una mejora continua que consistió en hacer un análisis de la situación actual y proponer en base a una reingeniería, la conformación de equipos de trabajo donde se propusieron varios escenarios de mejora y requerimientos para el desarrollar el proyecto de Código de Barras. “Esto significa que esta mejora continua incluye procesos de medición y análisis que incluyen aquellos necesarios para medir y recopilar datos” (Pérez y Múnera, 2007, p.50).

Este trabajo se ha dividido en 3 capítulos y un caso práctico en donde se plantean aplicar las siguientes herramientas:

Por la complejidad de la operación en el almacén de electrodomésticos, se vuelve muy complicado tener controles manuales, información incompleta y reprocesos. Por ello se decide aplicar la utilización tecnológica de un sistema de código de barras, desarrollado en el capítulo 1 Donde de forma breve se abarca la historia, justificación, funcionamiento y la tendencia tecnológica de la utilización de estos equipos.

De igual manera una vez que se tenga adoptada una tecnología en la forma de trabajo es necesario definir un método y medición del trabajo. Para esto se plantea la elaboración de una reingeniería de procesos, desarrollado en el capítulo 2. Donde a partir de la elaboración de esta metodología, el objetivo es mostrar el cómo se hacen las actividades actualmente y con la implementación de mejoras; alcance el proceso un nivel óptimo que permite distribuir mejor los equipos de trabajo.

# 6. MARCO TEÓRICO

## 6.1 Herramientas web para uso académico

**UMLet**

Es un código abierto de Java basado en la herramienta UML, diseñada para la enseñanza del Lenguaje Unificado de Modelado y para crear rápidamente diagramas UML. Es una herramienta de dibujo en lugar de una herramienta de modelado, debido a que no hay diccionario subyacente o directorio de objetos de diseño reutilizables. UMLet se distribuye bajo la Licencia Pública General GNU también conocida como “copyleft o fuente abierta“.

El término “fuente abierta” (open source) se refiere al código fuente del software de computadora, sin importar el sistema operativo, lenguaje o aplicación, e implica que cualquiera puede acceder al código. Sin embargo, esto no significa que el código esté disponible en forma gratuita o sea del dominio público. El concepto fuente abierta también se aplica a los contenidos, tales como documentación, datos y libros. (Longhorn, Henson-Apollonio y White, 2002, p.6).

UMLet tiene una interfaz de usuario simple que utiliza códigos de formato de texto para modificar las formas básicas con decoraciones y anotaciones, así que no hay bosques de iconos o lista de parámetros diálogos en el camino del usuario. Esto requiere que el usuario aprenda un nuevo lenguaje de marcas de texto, pero el esfuerzo es pequeño y el margen de beneficio obvio para el diseñador UML experimentado. Lara y Martínez (2006) afirman: “Un lenguaje de marcado es un conjunto de reglas que establecen qué tipo de marcas han de ser utilizadas, de qué modo se distinguirán (...). De forma genérica, se pueden distinguir dos tipos de lenguajes de marcado: Los lenguajes de marcado procedimentales y los descriptivos” (p.21).

Tiene una interfaz gráfica muy simple y utiliza un lenguaje de marcado para la creación y edición de los **elementos UML,** lo cual permite una ventaja para el **Modelado avanzado.**

**Balsamiq mockups**

Es una aplicación desarrollada con Flex y Adobe Air. La idea de ésta herramienta es facilitar el trabajo de armado de prototipos de las páginas web aunque podría usarse también para aplicaciones de escritorio. Guerrera (2011) refiere que Balsamiq Mockups nos provee de representaciones de todos los elementos utilizados para la construcción de una web, como pantallas de navegadores, títulos, menues, imágenes, videos, etc.

Balsamiq Mockups es una herramienta que puede ser usada tanto por clientes como por desarrolladores, con el objetivo de acordar aspectos importantes del alcance de una solución con una mínima inversión de tiempo dedicada.

Los clientes pueden hacer uso de Balsamiq Mockups sin tener ningún tipo de conocimiento técnico especial. Gracias a ello, pueden comunicar de una manera más eficiente sus ideas y necesidades al grupo de trabajo que realizará las implementaciones.

**GitHub**

Es una plataforma de**desarrollo colaborativo de software** para alojar proyectos utilizando el sistema de control de versiones [Git](http://es.wikipedia.org/wiki/Git). Asi mismo, Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente.

GitHub aloja el repositorio de código y brinda **herramientas** muy útiles para el **trabajo en equipo**, dentro de un proyecto. Además de eso, un usuario puede **contribuir a mejorar el software de los demás**. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un **fork** y solicitar **pulls**. Castillo (2012) afirma:

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

Tambien ofrece varias herramientas útiles para el **trabajo en equipo.** Entre ellas, cabe destacar:

* Una wiki para el mantenimiento de las distintas versiones de las páginas.
* Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
* Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un commit específico.
* Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

[**MySQL Workbench**](https://www.mysql.com/products/workbench/)

Es una herramienta unificada para los arquitectos de bases de datos, desarrolladores y administradores de bases. Ofrece un modelado de datos, desarrollo de SQL y herramientas completas de administración de configuración del servidor, administración de usuarios, copia de seguridad, y mucho más. Está disponible en Windows, Linux y Mac OS X. Jaque (2007) afirma que: “Es una aplicación para el diseño de bases de datos con MySQL. A fecha de hoy se encuentra todavía en fase beta, (…). Tiene, más o menos, la misma funcionalidad que DBDesigner” (p.35).

Tambien, permite a un Administrador de base de datos (DBA), desarrollador o arquitecto de datos generar y gestionar bases de datos. Incluye todo lo que un modelador de datos necesarios para la creación de modelos ER complejas, hacia adelante y la ingeniería inversa, y también ofrece características clave para la realización de las tareas difíciles de gestión del cambio y de la documentación que normalmente requieren mucho tiempo y esfuerzo.

## 6.2 Tecnologías de desarrollo

**¿Qué es una base de datos?**

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

A veces se utiliza DB, de *database* en inglés, para referirse a las bases de datos.

Los componentes de una base de datos son:

* Los datos: son el componente fundamental debido a que están interrelacionados entre sí, formando un conjunto con un mínimo de redundancia.
* El software: para que los datos puedan ser utilizados por diferentes usuarios y diferentes aplicaciones, deben estar estructurados y almacenados de forma independiente de las aplicaciones. Para ello se necesita un software o programa que actúa de interfaz entre los datos y las aplicaciones. A este software se le denomina un Sistema Gestor de Base de Datos (SGBD), el cual crea y organiza la base de datos y además atiende todas las solicitudes de acceso hechas a la base de datos tanto por usuario como por aplicaciones.

En la siguiente gráfica se pueden evidenciar los componentes de una base de datos, como lo son, los usuarios, un sistema de gestión de base de datos, un sistema operativo y los datos.

***Fuente: Elaboración propia.***

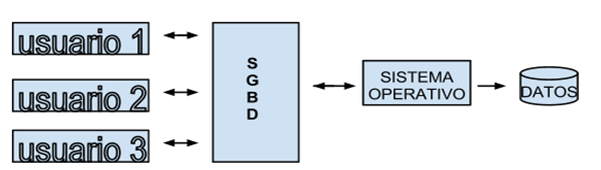


Ilustración 1: Gráfica de Componentes de una base de datos.

**Ventajas de una Base de Datos**

* Independencia de los datos, programas y procesos; es decir que permite modificar los datos sin modificar el código de las aplicaciones.
* Menor redundancia; quiere decir que no hace falta tanta repetición de datos, solo se indica la forma en la que se relacionan los datos.
* Integridad de los datos; hace referencia a que es menos probable que se pierdan los datos o que se realicen incoherencias entre ellos.
* Coherencia de los resultados; al recogerse y almacenarse la información una sola vez, en los tratamientos se utilizan siempre los mismos datos, por lo que los resultados coherentes.
* Hay mayor seguridad en los datos, porque al permitir limitar el acceso a los usuarios, cada tipo de usuario podrá acceder a algunas cosas, según corresponda.
* Acceso a los datos más eficientemente: gracias a la organización de los datos, se produce un resultado más óptimo en rendimiento.
* Reducción del espacio de almacenamiento; esto se da gracias a una buena estructuración de los datos.
* Acceso simultáneo a los datos: es más fácil controlar el acceso de usuario de forma concurrente.

**Desventajas de una Base de datos**

* Instalación costosa, de modo que el control y la administración de una base de datos requiere de un software y hardware de elevado coste, Además de la adquisición y el mantenimiento del sistema gestor de base de datos (SGBD).
* Implantación larga y fácil; debido a los puntos anteriores, se podría concluir que la adaptación del personal es mucho más complicada y requiere de bastante tiempo.
* Ausencia de estándares reales: significa una excesiva dependencia hacia los sistemas comerciales del mercado. Aunque, hoy en día, una buena parte de esta tecnología está aceptada como estándar de hecho.
* Falta de rentabilidad a corto plazo; esto debido al coste del personal y equipos, además del tiempo que tarda en estar operativa.

**¿Qué es el modelo entidad-relación (MER)?**

Es el modo lógico de datos más extendido para el diseño conceptual de una base de datos, porque siendo un modelo relativamente simple y fácil de emplear, posee una gran capacidad expresiva. Las técnicas que este modelo emplea, sirven básicamente para especificar las necesidades de información de una organización.

El modelo entidad-relación en su forma más simple implica identificar los asuntos de importancia dentro de una organización (entidades), las propiedades de los asuntos (atributos) y cómo se relacionan entre sí (relaciones), pero esto tiene valor solamente dentro del contexto de lo que se realiza en la empresa y en la forma de actuar de estas funciones de gestión sobre el modelo de información.

El modelo entidad-relacion es una técnica de ingeniera de la información que se utiliza para desarrollar un modelo de datos de alta calidad. El modelo de datos ofrece una forma estándar de definir los datos y las relaciones entre estos para todos los sistemas de información. Esto mejora enormemente la calidad del sistema e incrementa la productividad del software.

(Barker, 1990, p.6)

El siguiente gráfico representa los componentes de un modelo entidad-relación, a los cuales se les llama: entidades (nombre de una tablas, ej.: “Alumnos”), atributos (la información que requiere cada entidad, ej.: “Código alumno, Nombre alumno”) y relaciones (es la relación que hay entre entidades, ej.: entidad Alumno relacionada con entidad grado que cursa).

***Fuente: Elaboración propia.***

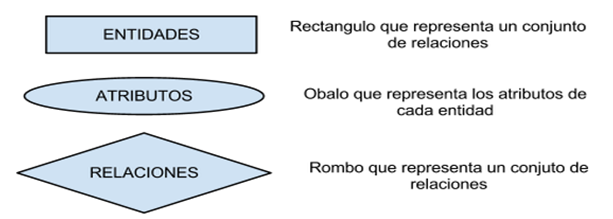


Ilustración 2: Entidades, atributos, relaciones del MER

Los objetivos más fundamentales que tiene el modelo entidad-relación, son dos:

* Ofrecer un modelo que refleje fielmente las necesidades de información de una organización, el cual será usado como base para el desarrollo de un sistema.
* Ofrecer un modelo independiente del posterior almacenamiento de los datos y sus métodos de acceso, lo que permitirá tomar decisiones objetivas acerca de la implementación más idónea.

En la siguiente gráfica se representa un ejemplo del modelo entidad-eelación, el cual da una idea más clara de cómo trabaja el modelo junto con sus componentes (Entidades, Atributos y Relaciones), siendo los rectángulos las entidades, los óvalos los atributos y el rombo la relación.

***Fuente: Elaboración propia.***

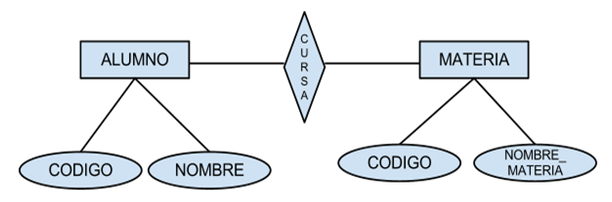


Ilustración 3: Gráfica de Ejemplo Modelo Entidad-Relación (E-R).

**¿Qué es un Sistema Gestor de Bases de Datos (SGBD)?**

Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto. Nevado (2010) afirma:

No debemos confundir una Base de Datos con un Sistema Gestor de Base de Datos. Una base de datos es la información almacenada, que cumple una serie de características y restricciones, pero para que la información peuda ser almacenada y el acceso a la misma satisfaga las características exigidas a una base de datos, es necesario que exista una serie de procedimentos, un sistema software que sea capaz de llevar a cabo tal labor. (p.32).

Un SGBD permite:

* Definir los datos a los distintos niveles de abstracción (físico, lógico y externo).
* Manipulación de los datos en la abse de datos, permitiendo insertar, modificar, borrar y consultar los datos.
* Mantenimiento de la integridad de la base de datos.
* Control de la privacidad y seguridad de los datos en la base de datos.

**Objetivos de un SGBD**

* Garantizar la independencia de los datos.
* Diseño y utilización orientado al usuario. Los datos y aplicaciones deben ser accesibles de la manera más amigable posible. Para ello, los SGBD han de fundamentarse en un modelo de datos teórico coherente y proporcionar lenguajes que permitan definir la estructura de la base de datos y acometer la generación, mantenimiento y acceso a los datos en términos que resulten apropiados.
* Centralización: los datos deben gestionarse de forma centralizada e independiente de las aplicaciones. Para satisfacer este objetivo un SGBD habrá de proporcionar una serie de utilidades que faciliten la administración de este fondo común.
* Evitar la redundancia y gestionar la concurrencia. Como un SGBD provee un fondo de datos común que puede ser compartido por varias aplicaciones, por lo tanto no es preciso duplicar datos. Sin embargo, puesto que varias aplicaciones pueden acceder a los mismos datos al mismo tiempo, el SGBD debe disponer de los mecanismos adecuados para gestionar esas concurrencias.
* Mantener la integridad semántica de los datos; un SGBD debe y puede proveer de mecanismos para evitar las alteraciones originadas por los datos.
* Mantener la seguridad. Un SGBD debe proveer un mecanismo que evite que algún usuario puede acceder a información no autorizada, utiliza mecanismos de identificación y gracias a ellos se podrá definir qué usuario accede a que recurso y cual no.
* Mantener la fiabilidad del sistema; algunos sistemas tienen mecanismos redundantes a nivel de hardware y de software para garantizar que el fallo de un componente sea asumido por otro sin que se pierda la información ni la disponibilidad de la base de datos.

**Microsoft SQL Server**

Es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft.

SQL nace de un lenguaje de cómputo llamado SEQUEL creado por IBM, que fue diseñado específicamente para consulta de base de datos. “SQL SERVER es el producto emblema de motor de base de datos de Microsoft que esta generando una enorme cantidad de interés en el mercado.” (Vergel, p.7). Microsoft se ha comprometido a invertir grandes cantidades de dinero en apoyo a la comercialización del producto y cuenta con que SQL SERVER se convierta en el principal motor de base de datos en la industria de cómputo para la plataforma WINDOWS XP aunque también puede operar en Windows 95, Windows 98, Vista…etc.

**Caracteristicas:**

* **Portabilidad:** las bases de datos pueden desarrollarse fácilmente ya sea en un equipo mainframe o una mini computadora, sin importar su sistema operativo. Cuenta con soporte de SUN, lo que permite una mayor comunicación entre servidores.
* **Compatibilidad:** los DBMS o SGBD (DataBase Management System, Sistema de Gestión de Bases de Datos) se pueden ejecutar ya sea en computadoras personales, microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo así como en distintas arquitecturas de hardware y software sin tener la necesidad de cambiar una sola línea de código.
* **Seguridad:** permite verificaciones de usuarios, mantener clientes diferentes en una misma base de datos y a señalar que a ciertos datos solo podrán acceder determinados usuarios así como la codificación de información privada. El Administrador puede programar permisos por tabla, columna o fila. Por medio del sistema operativo se permite la restricción de los movimientos que pudieran hacerse con los archivos, así como controlar los accesos con cuentas, a esto se le conoce como autenticación. La seguridad comprende en protección y codificación de tablas de datos, columnas y filas, así como las transferencias de datos entre un cliente y un servidor como auditorias que identifican violaciones a la seguridad.
* **Administración:** aporta funcionalidades de administración y tunning de la memoria, del CPU y de disco, de manera que se reduce el tiempo para la administración. Primera base de datos que soporta la configuración automática y la auto-optimización por medio del servicio Agente SQL Server.
* **Rendimiento:** permiten una alta disponibilidad de aplicaciones sin necesidad de una reconfiguración de datos.

## 6.3 Lenguaje de programación

El proyecto Program-Art se elaboró con el siguiente lenguaje de programación y entorno de desarrollo integrado:

**ASP.NET**

ASP.NET es una plataforma web que proporciona todos los servicios necesarios para compilar aplicaciones web empresariales basadas en servidor. ASP.NET está compilado en .NET Framework, por lo que todas las características de .NET Framework están disponibles en las aplicaciones ASP.NET. Las aplicaciones se pueden escribir en cualquier lenguaje que sea compatible con Common Language Runtime (CLR), incluido Visual Basic y C#.

Para crear aplicaciones web ASP.NET, puede usar Visual Studio. Además, hay disponible un producto gratuito independiente, Visual Studio Express para web, que incluye el conjunto básico de características de diseño web de Visual Studio.

**Requisitos de desarrollo de ASP.NET**

Para desarrollar aplicaciones ASP.NET, es necesario tener:

* Un editor de código.
* .NET Framework.
* Un servidor Web como IIS (Servicios de Internet Information Server)

Giardina afirma: “Adicionalmente a la programación web es necesario en muchos casos contar con algún soporte para el almacenamiento de datos. Lo ideal es contar con un motor de base de datos (SQL Server, Oracle, etc.)”.

**Visual Web Developer Express**

Es un entorno fácil de usar para el desarrollo dinámico de aplicaciones web. Visual Web Developer Express presenta una interfaz simplificada que se centra en proporcionar las herramientas necesarias para crear aplicaciones web. Todas las características de Visual Web Developer Express están también disponibles en Microsoft Visual Studio 2010, y los sitios web que cree con Visual Web Developer Express son compatibles con Visual Studio. Puede compartir páginas y sitios web entre Visual Web Developer Express y Visual Studio sin ningún problema de conversión ni pérdida de funcionalidad.

Todas las características de Visual Web Developer Express están también disponibles en Microsoft Visual Studio 2010, y los sitios web que cree con Visual Web Developer Express son compatibles con Visual Studio. Puede compartir páginas y sitios web entre Visual Web Developer Express y Visual Studio sin ningún problema de conversión ni pérdida de funcionalidad. (Microsoft, p.1).

Visual Web Developer 2010 Express forma parte de la familia de Visual Studio. Visual Studio 2010 es un entorno de desarrollo completo de Microsoft para crear aplicaciones web así como aplicaciones cliente (para Windows), servicios de Windows, componentes, controles y otros tipos de aplicaciones. Visual Web Developer Express, en cambio, presenta una interfaz simplificada que se centra en proporcionar las herramientas necesarias para crear aplicaciones web. Por tanto, Visual Web Developer Express es más compacto y resulta más fácil trabajar con él que con Visual Studio.

Todas las características de Visual Web Developer Express están también disponibles en Visual Studio, y los sitios web que cree con Visual Web Developer Express son compatibles con Visual Studio. Puede compartir páginas y sitios web entre Visual Web Developer Express y Visual Studio sin ningún problema de conversión ni pérdida de funcionalidad. (Microsoft, p.2).

# 7. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

## 7.1 Planificación de requisitos

A partir del planteamiento del problema, el cual se determinó en la descripción del proyecto, se iniciará una investigación que debe ir acompañada de una revisión bibliográfica, consulta a expertos y recolección de datos.

El método de recolección de información inicial empleado fue la observación. Cabe mencionar que este método no tiene algún registro tangible, pero será de gran ayuda para para estudiar el problema tal y como se presenta en realidad. Luego, como método de medición ya mencionado anteriormente, se realizó una encuesta virtual a varios instructores del SENA por medio de la herramienta de Google Forms (Formularios), y mostró un análisis estadístico de los resultados de dicha encuesta. [(Ctrl+clic para ver resultados de la encuesta).](#_13._Anexos)

También, se realizó una entrevista a líder del Programa de Análisis y Desarrollo de Sistemas de Información (ADSI), la cual fue de gran importancia en la fase de análisis. [(Ctrl+clic para ver resultados de la entrevista).](#_13._Anexos)

### 7.1.1 Requerimientos funcionales y no funcionales

### 7.1.1.1 Requerimientos funcionales

|  |  |  |
| --- | --- | --- |
| Ref. # | Función | Rol |
| R1 | Los usuarios deben tener un usuario y una contraseña para poder acceder al sistema. | Administrador, Instructor y Aprendiz |
| R2 | Los usuarios podrán cerrar la sesión de su cuenta en el momento que deseen. | Administrador, Instructor y Aprendiz |
| R3 | Los usuarios podrán cambiar su contraseña de usuario. | Administrador, Instructor y Aprendiz |
| R4 | El sistema debe validar usuario y contraseña de los registros al ingreso de la aplicación. | El Sistema |
| R5 | El sistema debe registrar la información de los usuarios. | El Sistema |
| R6 | El sistema debe permitir que los usuarios registrados proporcionen el uso del sistema. | El Sistema |
| R7 | El sistema debe vincular toda información dada por los usuarios. | El Sistema |
| R8 | El sistema debe permitir al usuario instructor iniciar la clase. | El Sistema |
| R9 | El usuario Administrador podrá gestionar los usuarios del sistema | Administrador |
| R10 | El usuario Administrador podrá crear nuevos usuarios. | Administrador |
| R11 | El usuario Administrador podrá eliminar usuarios. | Administrador |
| R12 | El usuario Administrador podrá actualizar la información de los usuarios. | Administrador |
| R13 | El usuario Administrador podrá consultar la información de los usuarios. | Administrador |
| R14 | El usuario Instructor podrá crear la clase insertando datos como número de ficha, fecha, hora inicio, hora fin e iniciar un cronometro. | Instructor |
| R15 | El usuario Instructor podrá generar reportes a la fecha sobre las inasistencias de los aprendices en formato Pdf, Excel y/o Word. | Instructor |
| R16 | El usuario Aprendiz podrá generar reportes a la fecha sobre sus inasistencias en formato Pdf, Excel y/o Word. | Aprendiz |

### 7.1.1.2 Requerimientos no funcionales

* Conexión a internet para que se pueda ejecutar el sistema. Ancho de banda de mínimo 50Mb.
* El lenguaje de programación que se va a utilizar para el desarrollo del aplicativo será Visual Studio express 2010.
* El motor de base de datos que se va a utilizar para la creación y almacenamiento de los datos será SQL Server 2010.
* Navegador Mozilla, Google Chrome y/o Internet Explorer (versión 8 en adelante), para la conexión con el sistema.
* Capacidad de memoria RAM de mínimo 2Gb para el fácil acceso al sistema, y mejor rendimiento en el proceso de implementación.
* El diseño del software debe cumplir con los estándares adecuados según las normas de color.
* El diseño del software debe ser cómodo y amigable para el usuario, mejorando así su fácil manejo y entendimiento en la implementación.
* El software debe ser rápido, cargar en máximo 5 segundos, esto mejorará la calidad y agilizará el proceso de registro de asistencias.
* El sistema permitirá el ingreso de asistencia de los aprendices por lapso, el cual será asignado por el instructor.
* El sistema se dividirá en distintas páginas para acceso personalizado, lo cual permitirá tener control de distintas operaciones dependiendo el tipo de usuario.
* El sistema no debe tardar más de cinco segundos en mostrar los resultados de una búsqueda.

### 7.1.2 Planeación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resultados de aprendizaje y actividades** | **Duración por horas** | **Inicio** | **Fin** | **Roles** | **Herramientas utilizadas** |
| **FASE 1 ANÁLISIS** | **520** | **21/04/14** | **28/06/14** |  |  |
| **Plantear diferentes alternativas de modelos tecnológicos de información empresarial, teniendo en cuenta la plataforma tecnológica de la empresa y las tendencias del mercado, para dar solución a las situaciones relacionadas con el manejo de la información.** | **60** | **21/04/14** | **27/04/14** | **5 analistas** | **Herramientas ofimáticas, Project** |
| Sugerir diferentes ideas de proyecto, para elegir la más conveniente y dar inicio a la solución del problema, de acuerdo a las necesidades del cliente. | 5 |  |  |  |  |
| Limitación del problema | 10 |  |  |  |  |
| Formato Sofía | 30 |  |  |  |  |
| Planeación del proyecto | 15 |  |  |  |  |
| **Aplicar las técnicas de recolección de datos, diseñando los instrumentos necesarios para el procesamiento de información, de acuerdo con la situación planteada por la empresa.** | **100** | **28/04/14** | **07/05/14** | **5 analistas** | **Formularios Google Drive, Herramientas ofimáticas** |
| Formular preguntas y realizar entrevista a la ing. Marina Casallas | 30 |  |  |  |  |
| Formular preguntas y realizar encuesta a los instructores del Sena. | 70 |  |  |  |  |
| **Interpretar el informe de requerimientos, para determinar las necesidades tecnológicas en el manejo de la información, de acuerdo con las normas y protocolos establecidos en la empresa.** | **100** | **08/05/14** | **18/05/14** | **5 analistas** | **Formularios Google Drive, Herramientas ofimáticas** |
| Analizar los requerimientos del cliente | 10 |  |  |  |  |
| Tabulación de datos recogidos. | 20 |  |  |  |  |
| Definir los requerimientos funcionales y no funcionales | 50 |  |  |  |  |
| Determinar la arquitectura a trabajar | 20 |  |  |  |  |
| **Elaborar mapas de procesos que permitan identificar las áreas involucradas en un sistema de información, utilizando herramientas informáticas y las Tic’s, para generar informes según las necesidades de la empresa.** | **60** | **19/05/14** | **25/05/14** | **5 analistas** | **Herramientas ofimáticas** |
| Crear el mapa de procesos correspondiente. | 40 |  |  |  |  |
| Elaborar el Modelo Entidad Relación (MER) | 20 |  |  |  |  |
| **Representa el bosquejo de la solución al problema presentado por el cliente, mediante la elaboración de diagramas de casos de uso, apoyado en el análisis del informe de requerimientos, al confrontar la situación con el usuario según normas y prototipos.** | **60** | **26/05/14** | **04/06/14** | **5 analistas** | **Herramientas ofimáticas, UmLet** |
| Elaborar los primeros diagramas de casos de uso de acuerdo a los requerimientos del cliente | 60 |  |  |  |  |
| **Construir el modelo conceptual del macrosistema frente a los requerimientos del cliente, mediante el uso e interpretación de la información levantada, representado en diagramas de clase, de interacción, colaboración y contratos de operación, de acuerdo** | **60** | **05/06/14** | **12/06/14** | **5 analistas** | **Herramientas ofimáticas, UmLet** |
| Elaborar un borrador del diagrama de clase de acuerdo a los requerimientos del cliente. | 20 |  |  |  |  |
| Elaborar un borrador del diagrama de interacción de acuerdo a los requerimientos del cliente. | 20 |  |  |  |  |
| Elaborar un borrador del diagrama de colaboración de acuerdo a los requerimientos del cliente. | 20 |  |  |  |  |
| **Valorar la incidencia de los datos en los procesos del macrosistema, tomando como referente el diccionario de datos y las mini especificaciones, para la consolidación de los datos que intervienen, de acuerdo con parámetros establecidos.** | **60** | **13/06/14** | **22/06/14** | **5 analistas** | **Herramientas ofimáticas** |
| Diccionario de datos | 60 |  |  |  |  |
| **Elaborar el informe de los resultados del análisis del sistema de información, de acuerdo con los requerimientos del cliente según normas y protocolos establecidos.** | **20** | **23/06/14** | **28/06/14** | **5 analistas** | **Herramientas ofimáticas** |
| Elaborar un informe final de la fase 1 | 20 |  |  |  |  |
| **FASE 2 DISEÑO** | **350** | **07/07/14** | **31/08/14** |  |  |
| **Elaborar el informe de diseño del sistema de información, de acuerdo con la selección de las herramientas, tanto de software como de hardware, requeridas para la solución informática.** | **25** | **07/07/14** | **13/07/14** | **2 diseñadores, 3 analistas** | **Herramientas ofimáticas, UmLet** |
| Indagar que software tiene el cliente para analizar su arquitectura. | 10 |  |  |  |  |
| Realizar un informe de diseño teniendo en cuenta el software y el hardware. | 15 |  |  |  |  |
| **Diseñar la arquitectura del software, mediante la interpretación de las clases, objetos y mecanismos de colaboración, utilizando herramientas tecnológicas de diseño, de acuerdo con las tendencias de las tecnologías de la información y la comunicación.** | **25** | **14/07/14** | **20/07/14** | **2 diseñadores, 3 analistas** | **Herramientas ofimáticas, UmLet** |
| Elaborar el diagrama de clase de acuerdo al diagrama analizado en la fase 1. | 10 |  |  |  |  |
| Elaborar el diagrama de colaboración de acuerdo al diagrama analizado en la fase 1. | 5 |  |  |  |  |
| Elaborar el diagrama de clase de acuerdo al diagrama analizado en la fase 1. | 5 |  |  |  |  |
| Elaborar los diagramas de casos de uso (continuación de la fase 1). | 5 |  |  |  |  |
| **Construir el prototipo del sistema de información, a partir del análisis de las características funcionales del sistema en relación con facilidad de manejo, funcionalidad y experiencia del usuario, apoyado en software aplicado según protocolos de diseño.** | **50** | **21/07/14** | **30/07/14** | **2 diseñadores, 3 analistas** | **Herramientas ofimáticas, UmLet, Software Mockup** |
| Diseñar el prototipo según la especificación de protocolos. | 30 |  |  |  |  |
| Elaborar diagrama de despliegue | 10 |  |  |  |  |
| Elaborar diagrama de distribución |  |  |  |  |  |
| Elaborar diagramas de secuencia | 10 |  |  |  |  |
| **Aplicar políticas y mecanismos de control en el diseño del sistema de información, mediante el análisis de la vulnerabilidad de la información, siguiendo los parámetros establecidos por la organización.** | **50** | **31/07/14** | **06/08/14** | **2 diseñadores, 3 analistas** | **Herramientas ofimáticas, UmLet** |
| Fijar los mecanismos y políticas de control por medio del análisis de los datos obtenidos. | 50 |  |  |  |  |
| **Diseñar la arquitectura tecnológica del sistema de información, mediante el reconocimiento de hardware y software, de acuerdo con la tecnología disponible en el mercado, el informe de análisis levantado y el diagrama de distribución.** | **100** | **08/08/14** | **17/08/14** | **2 diseñadores, 3 analistas** | **Herramientas ofimáticas, UmLet** |
| Elaborar diagramas de estado | 50 |  |  |  |  |
| Elaborar diagramas de actividades | 50 |  |  |  |  |
| **Diseñar la estructura de datos, a partir del modelo conceptual determinado en el análisis del sistema, utilizando herramientas tecnológicas de bases de datos, según las normas y estándares establecidos.** | **100** | **18/08/14** | **31/08/14** | **2 diseñadores, 3 analistas** | **Herramientas ofimáticas, UmLet** |
| Diseñar la base de datos correspondiente | 70 |  |  |  |  |
| Elaborar el Modelo Entidad Relación (MER) según especificaciones de la fase 1. | 30 |  |  |  |  |
| **FASE 3 DESARROLLO** | **600** | **01/10/14** | **12/12/14** |  |  |
| **Ejecutar y documentar las pruebas del software, aplicando técnicas de ensayo-error, de acuerdo con el plan diseñado y los procedimientos establecidos por la empresa.** | **100** | **01/10/14** | **12/10/14** | **2 desarrolladores, 3 analistas** | **Herramientas ofimáticas, Visual Studio** |
| Realizar pruebas de software. | 70 |  |  |  |  |
| Documentar las pruebas de software. | 30 |  |  |  |  |
| **Realizar la codificación de los módulos del sistema y el programa principal, a partir de la utilización del lenguaje de programación seleccionado, de acuerdo con las especificaciones del diseño.** | **50** | **13/10/14** | **19/10/14** | **2 desarrolladores, 3 analistas** | **Visual Studio** |
| Establecer la codificación de los módulos del sistema de acuerdo a las especificaciones del diseño. | 50 |  |  |  |  |
| **Interpretar el informe técnico de diseño, para determinar el plan de trabajo durante la fase de construcción del software, de acuerdo con las normas y protocolos establecidos en la empresa.** | **50** | **20/10/14** | **26/10/14** | **2 desarrolladores, 3 analistas** | **Herramientas ofimáticas** |
| Construir un plan de trabajo de acuerdo a las normas y protocolos establecidos. | 50 |  |  |  |  |
| **Elaborar el manual técnico de la aplicación, de acuerdo con la complejidad del aplicativo y según normas y procedimientos establecidos por la empresa.** | **50** | **27/10/14** | **02/11/14** | **2 desarrolladores, 3 analistas** | **Herramientas ofimáticas** |
| Crear el manual técnico de la aplicación de acuerdo a los protocolos del sistema. | 50 |  |  |  |  |
| **Construir la interfaz de usuario, apoyado en la evaluación del prototipo, determinando las entradas y salidas requeridas en el diseño y definiendo los lineamientos para la navegación, de acuerdo con las necesidades del usuario.** | **150** | **03/11/14** | **16/11/14** | **2 desarrolladores, 3 analistas** | **Visual Studio** |
| Crear la interfaz de usuario de acuerdo a los protocolos establecidos. | 150 |  |  |  |  |
| **Construir el programa de instalación del aplicativo, utilizando las herramientas de desarrollo disponibles en el mercado, según las características de la arquitectura de la solución.** | **150** | **17/11/14** | **30/11/14** | **2 desarrolladores, 3 analistas** | **Visual Studio** |
| Crear el programa de instalación de la aplicación utilizando el lenguaje de programación Visual Studio. | 150 |  |  |  |  |
| **Construir la base de datos, a partir del modelo de datos determinado en el diseño del sistema, utilizando sistemas de gestión de base de datos, según los protocolos establecidos en la organización.** | **150** |  |  | **2 desarrolladores, 3 analistas** | **SQL Server** |
| Crear la base de datos correspondiente utilizando herramientas como SQL Server. | 150 | **01/12/14** | **12/12/14** |  |  |
| **FASE 4 NEGOCIACIÓN TECNOLÓGICA** | **140** | **19/01/15** | **04/04/15** |  |  |
| **Participar en los perfeccionamientos de contratos informáticos, estableciendo cláusulas técnicas, que respondan a las necesidades de los actores de la negociación, de acuerdo con la ley de contratación.** | **50** | **19/01/15** | **08/02/15** | **5 analistas** | **Herramientas ofimáticas** |
| Establecer clausulas técnicas de acuerdo a las necesidades y a la ley de contratación. | 20 |  |  |  |  |
| Realizar investigación sobre contratos informáticos y licitaciones | 15 |  |  |  |  |
| Realizar investigación sobre pliego de condiciones. | 15 |  |  |  |  |
| **Definir estrategias para la elaboración de términos de referencia y procesos de evaluación de proveedores, en la adquisición de tecnología, según protocolos establecidos.** | **25** | **09/02/15** | **22/02/15** | **5 analistas** | **Herramientas ofimáticas** |
| Establecer estrategias de elaboración de términos de referencia. | 25 |  |  |  |  |
| **Elaborar el informe sobre el cumplimiento de los términos de referencia previstos en la negociación, de acuerdo con la participación de cada uno de los actores en relación con la satisfacción de los bienes informáticos contratados y recibidos, según norma** | **25** | **23/02/15** | **08/03/15** | **5 analistas** | **Herramientas ofimáticas** |
| Desarrollar el informe de acuerdo a la participación de los actores previstos en la negociación. | 25 |  |  |  |  |
| **Interpretar el diagnóstico de necesidades informáticas, para determinar las tecnológicas requeridas en el manejo de la información, de acuerdo con las normas y protocolos establecidos por la empresa.** | **40** | **09/03/15** | **04/04/15** | **5 analistas** | **Herramientas ofimáticas** |
| Deducir las necesidades para definir las tecnologías requeridas. | 40 |  |  |  |  |
| **FASE 5 CALIDAD** | **200** | **07/04/15** | **28/06/15** |  |  |
| **Aplicar los estándares de calidad involucrados en los procesos de desarrollo de software, siguiendo el plan establecido para mantener la integridad de los productos de trabajo definidos, según las prácticas de configuración establecidas por la empresa.** | **50** | **07/04/15** | **26/04/15** | **5 analistas** | **Herramientas ofimáticas** |
| Adaptar los estándares de calidad relacionados a los procesos de desarrollo de software. | 50 |  |  |  |  |
| **Elaborar instrumentos e instructivos, requeridos por el aseguramiento de la calidad, para documentar y evaluar los procesos de desarrollo de software, de acuerdo con las normas y procedimientos establecidas por la empresa.** | **25** | **27/04/15** | **10/05/15** | **5 analistas** | **Herramientas ofimáticas** |
| Crear manuales de calidad, para documentar los procesos de desarrollo de software. | 25 |  |  |  |  |
| **Identificar los puntos críticos de control en los procesos de desarrollo de software, para establecer las acciones a seguir, garantizando el cumplimiento de los estándares de calidad, siguiendo los lineamientos establecidos por la organización.** | **25** | **11/05/15** | **24/05/15** | **5 analistas** | **Herramientas ofimáticas** |
| Determinar los puntos de control en los procesos de desarrollo de software. | 25 |  |  |  |  |
| **Elaborar el informe final del proceso de gestión de calidad en el desarrollo de software, que consolide la información de las evidencias, hallazgos y novedades frente al seguimiento y control de los productos, según normas internacionales y protocolos de** | **25** | **25/05/15** | **07/06/15** | **5 analistas** | **Herramientas ofimáticas** |
| Elaborar un informe final sobre el desarrollo del software. | 25 |  |  |  |  |
| **Evaluar procesos y productos de desarrollo de software, documentar y concertar acciones a seguir, para garantizar el cumplimiento de las normas establecidas, de acuerdo con el plan definido y con los criterios de medición, métricas y políticas determinad** | **25** | **08/06/15** | **14/06/15** | **5 analistas** | **Herramientas ofimáticas** |
| Determinar los procesos de desarrollo de software y documentar. | 25 |  |  |  |  |
| **Identificar las características de los procesos de desarrollo de software, frente al referente de calidad adoptado por la empresa, ajustándolos a los resultados de las mediciones, evaluaciones y recomendaciones realizadas.** | **50** | **15/06/15** | **28/06/15** | **5 analistas** | **Herramientas ofimáticas** |
| Determinar las características de los procesos de desarrollo en la empresa. | 25 |  |  |  |  |
| **FASE 6 IMPLANTACIÓN** | **360** | **06/07/15** | **25/08/15** |  |  |
| **Capacitar a los usuarios del sistema, sobre la estructuración y el manejo del aplicativo, de acuerdo con el plan establecido, el perfil de los usuarios, según políticas de la organización.** | **100** | **06/07/15** | **19/07/15** | **5 analistas** | **Herramientas ofimáticas** |
| Capacitar a los usuarios sobre el manejo del aplicativo. | 25 |  |  |  |  |
| **Configurar el software de la aplicación para cliente y servidor, mediante la utilización del hardware disponible, ejecutándola en la plataforma tecnológica, según normas y protocolos establecidos por la empresa.** | **100** | **20/07/15** | **02/08/15** | **5 analistas** | **Herramientas ofimáticas** |
| Estructurar y configurar el software utilizando el hardware. | 100 |  |  |  |  |
| **Elaborar el informe administrativo, siguiendo los protocolos de la organización, basado en los planes de instalación, respaldo y migración del sistema de información, facilitando la operatividad y mantenimiento de la solución informática.** | **50** | **03/08/15** | **09/08/15** | **5 analistas** | **Herramientas ofimáticas** |
| Elaborar un informe de acuerdo a los protocolos de la empresa. | 50 |  |  |  |  |
| **Elaborar informes técnicos relacionados con la solución informática implantada, de acuerdo con las propuestas de alternativas aplicadas, teniendo en cuenta las técnicas de comunicación y según normas y protocolos establecidos** | **50** | **10/08/15** | **16/08/15** | **5 analistas** | **Herramientas ofimáticas** |
| Crear informes de acuerdo a las alternativas aplicadas. | 50 |  |  |  |  |
| **Definir estrategias para la validación de manuales de usuario y de operación, respondiendo a las necesidades y satisfacción del cliente, frente a la solución informática propuesta, según políticas de la organización.** | **60** | **17/08/15** | **25/08/15** | **5 analistas** | **Herramientas ofimáticas** |

**Notas:**

\*Los resultados de aprendizaje son los que aparecen en letra negrilla.

\*Las actividades son las que aparecen a continuación de los resultados de aprendizaje y no están en negrilla.

## 7.2 Diseño

### 7.2.1 Modelo Entidad Relación (MER)

***Fuente: elaboración propia***

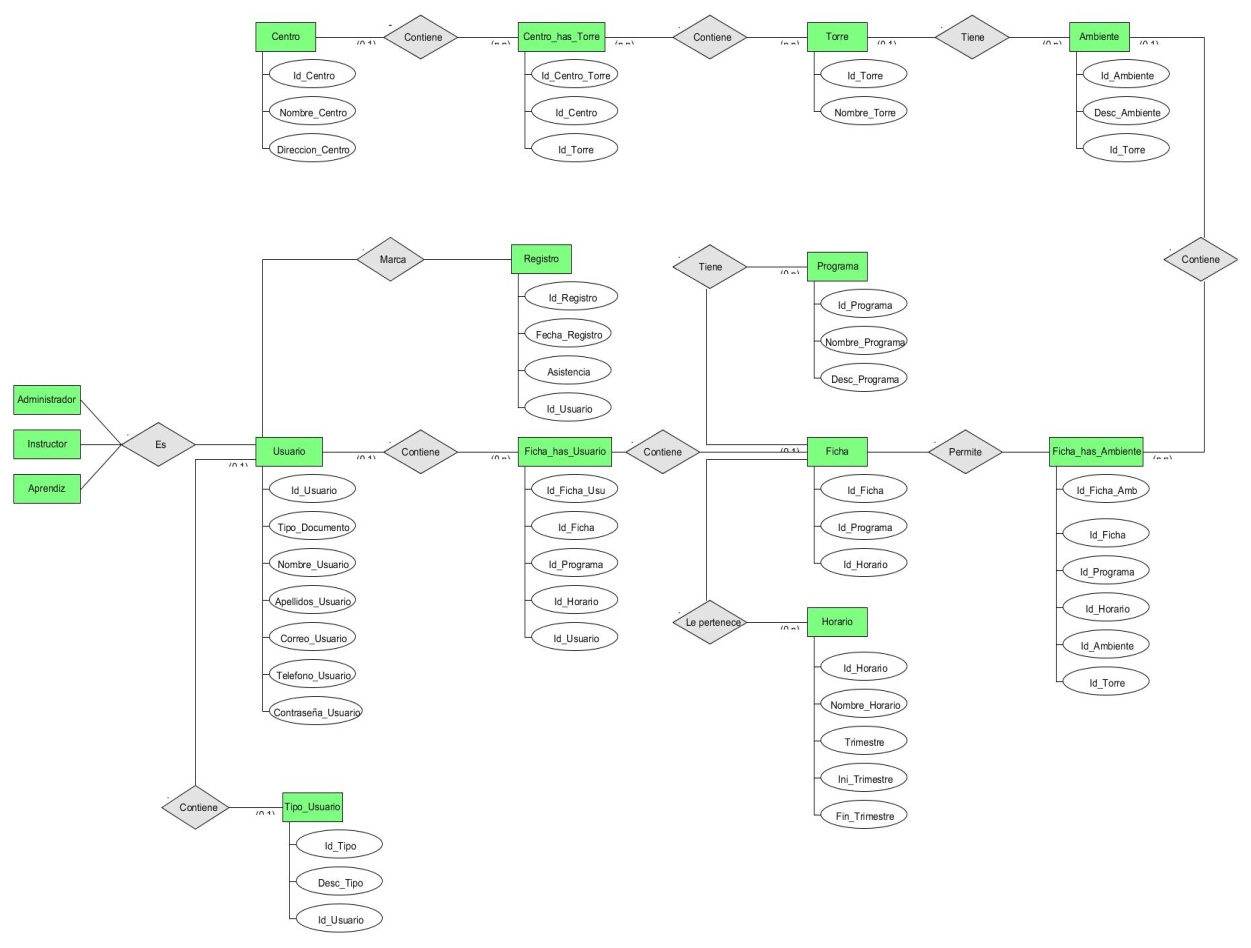
****

Ilustración 4: MER

### 7.2.2 Diccionario de datos

***Fuente: elaboración propia***

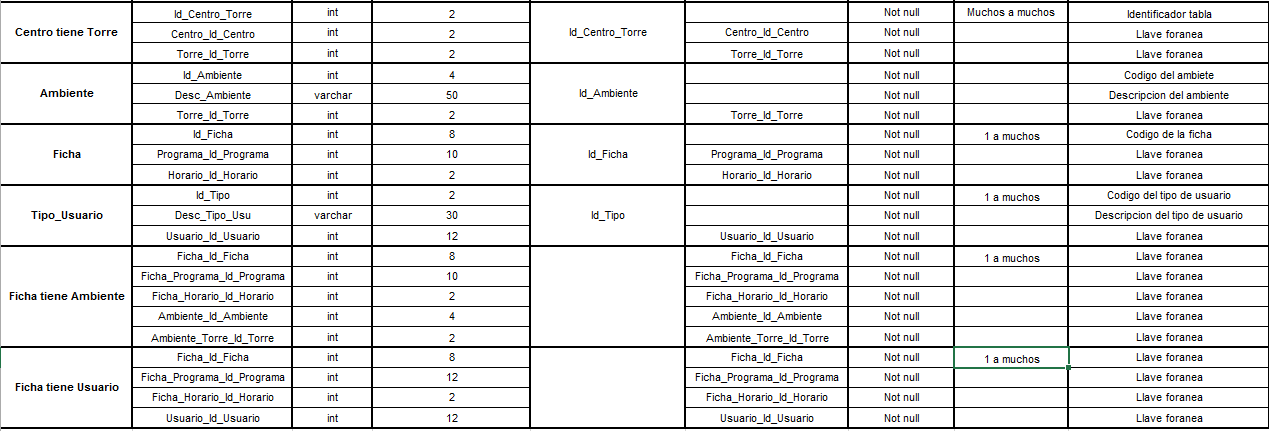
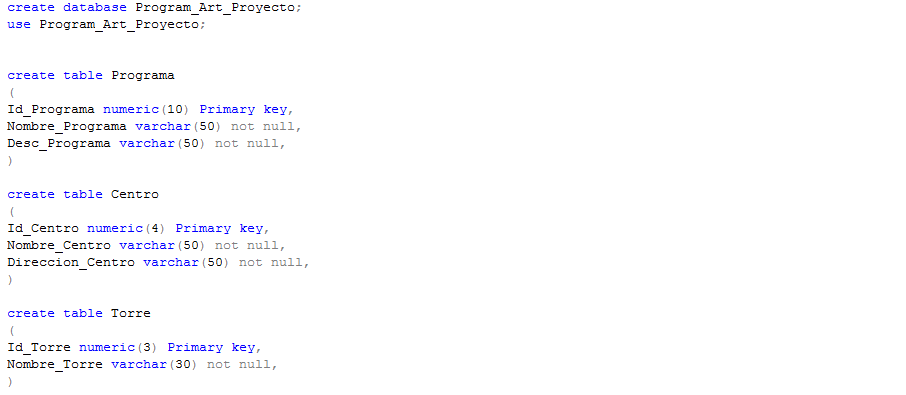
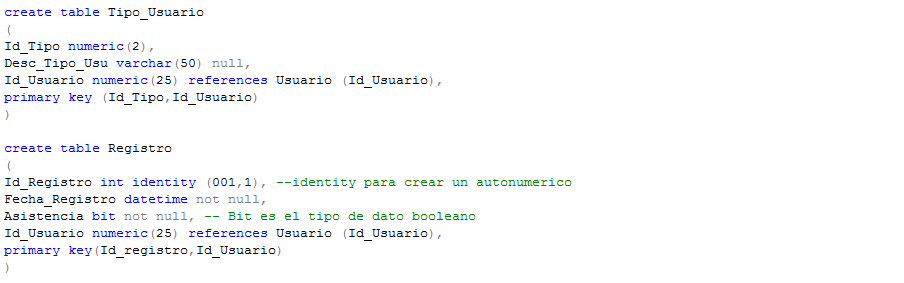
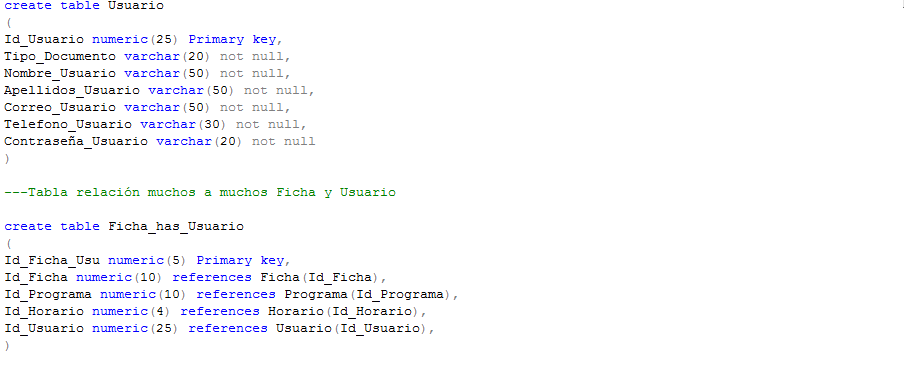
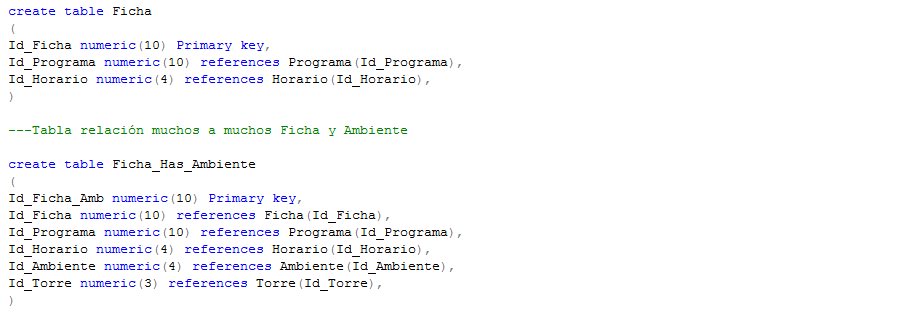
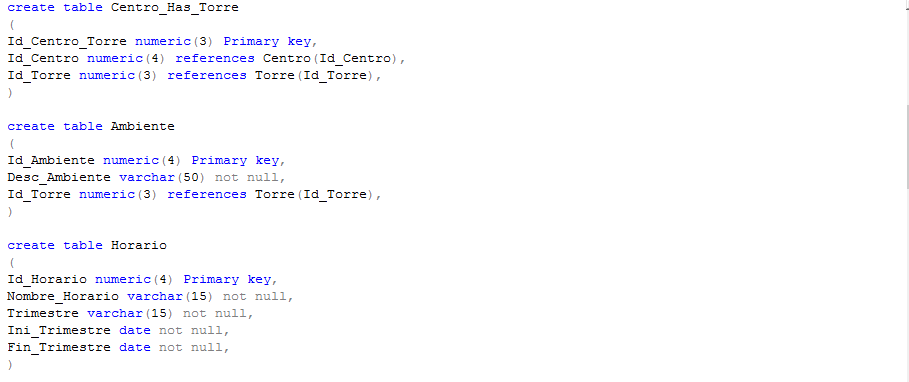


Ilustración 5: Diccionario de datos

## 7.3 Construcción

### 7.3.1 Base de datos

****

****

## 7.4 Implementación

### 7.4.1 Diagrama de despliegue

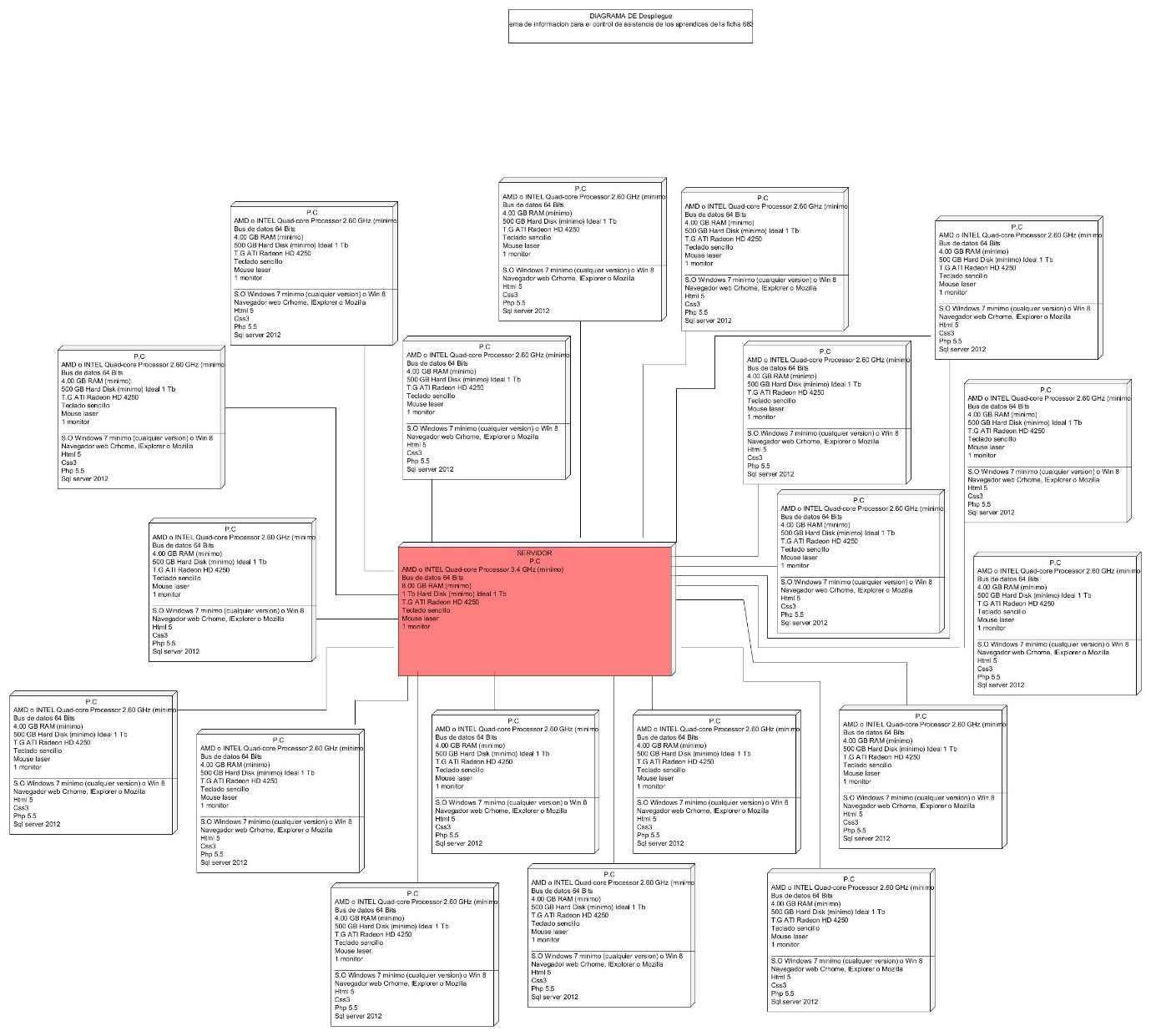


Ilustración 6: Diagrama de despliegue

# 8. INGENIERÍA DEL PRODUCTO

## 8.1 Diagramas de distribución

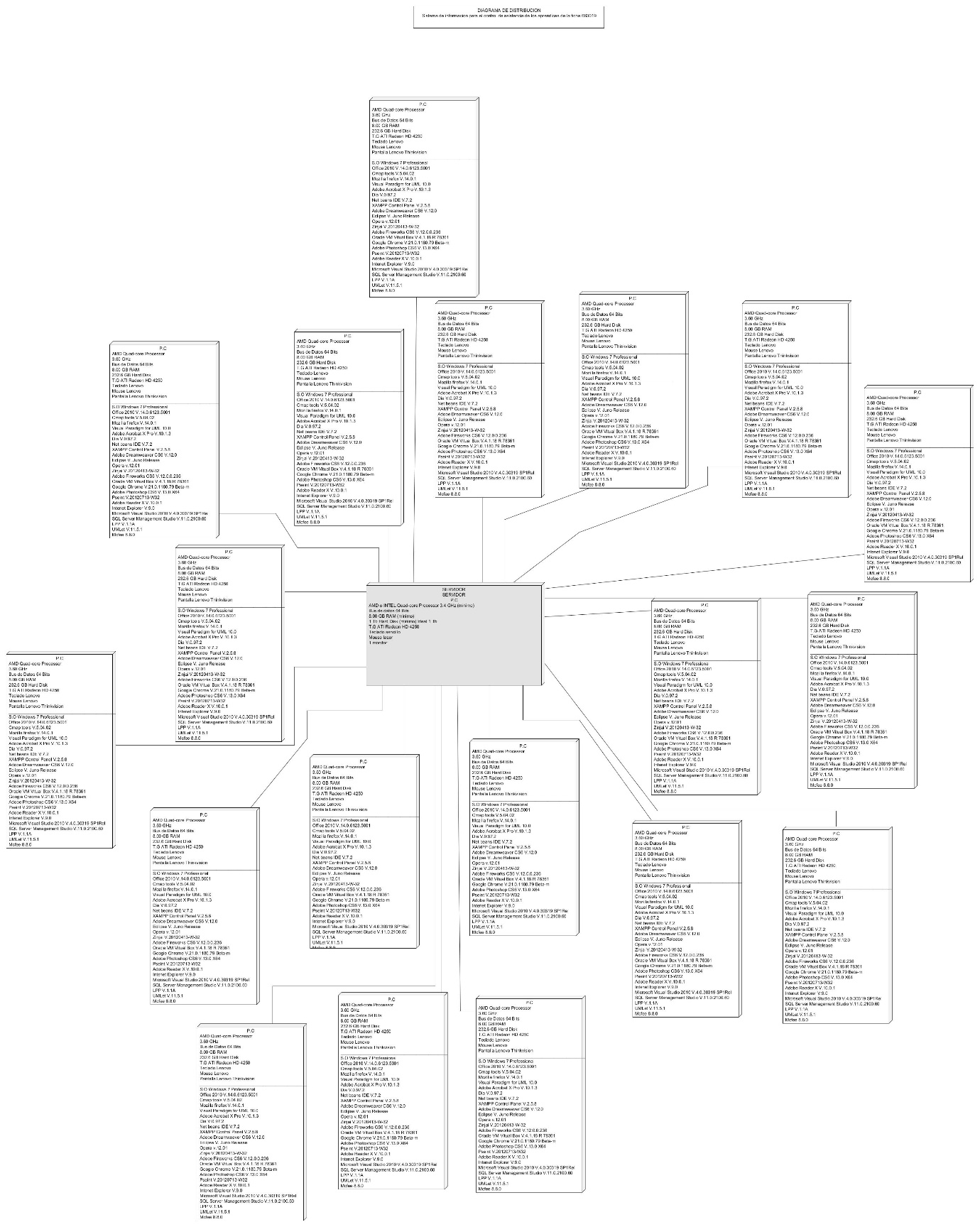


Ilustración 7: Diagrama de distribución

## 8.2 Servicios web

### 8.2.1 Diagrama de clases

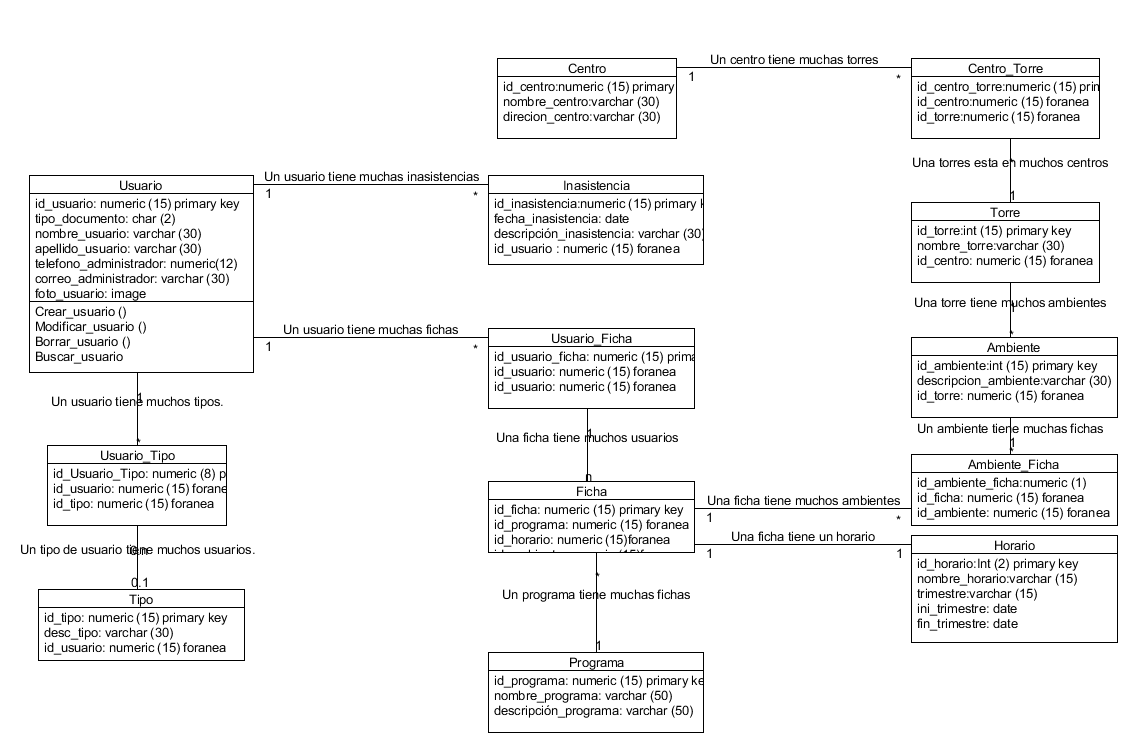


Ilustración 8: Diagrama de clases

### 8.2.2 Diagramas casos de uso

**Administrador:**

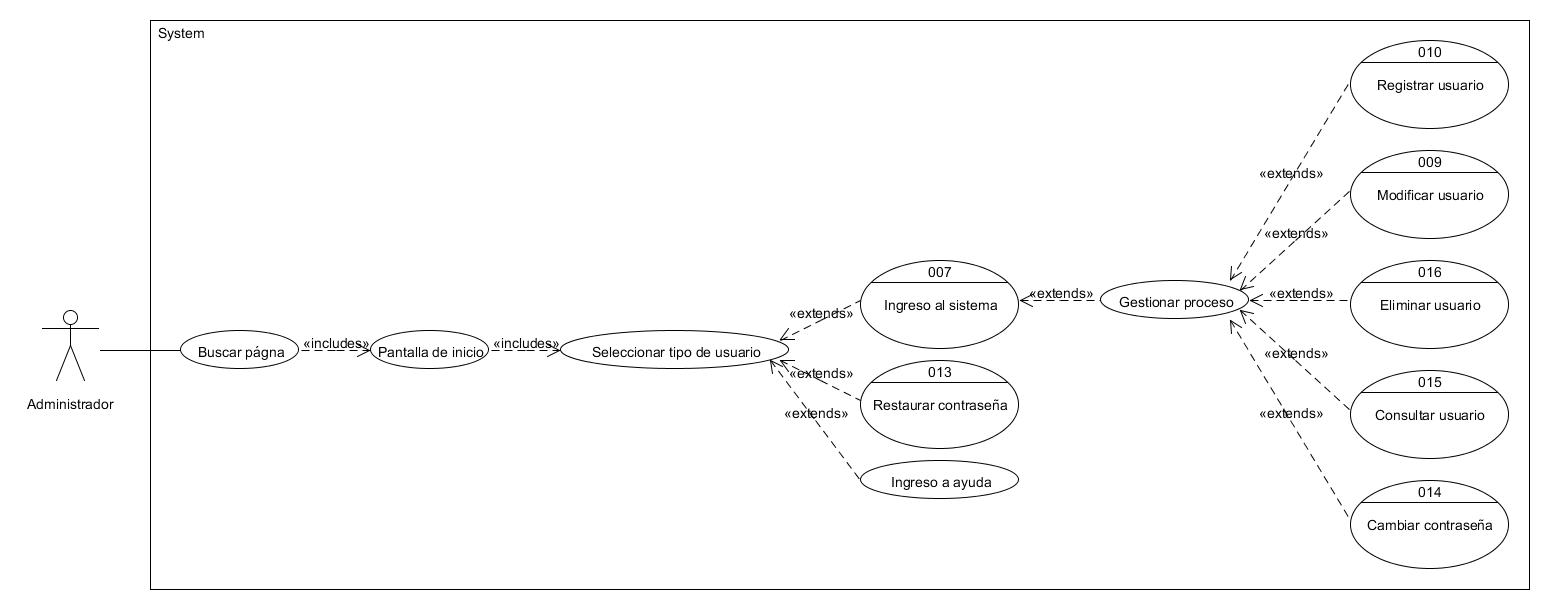


Ilustración 9: Caso de uso general administrador

**Instructor:**

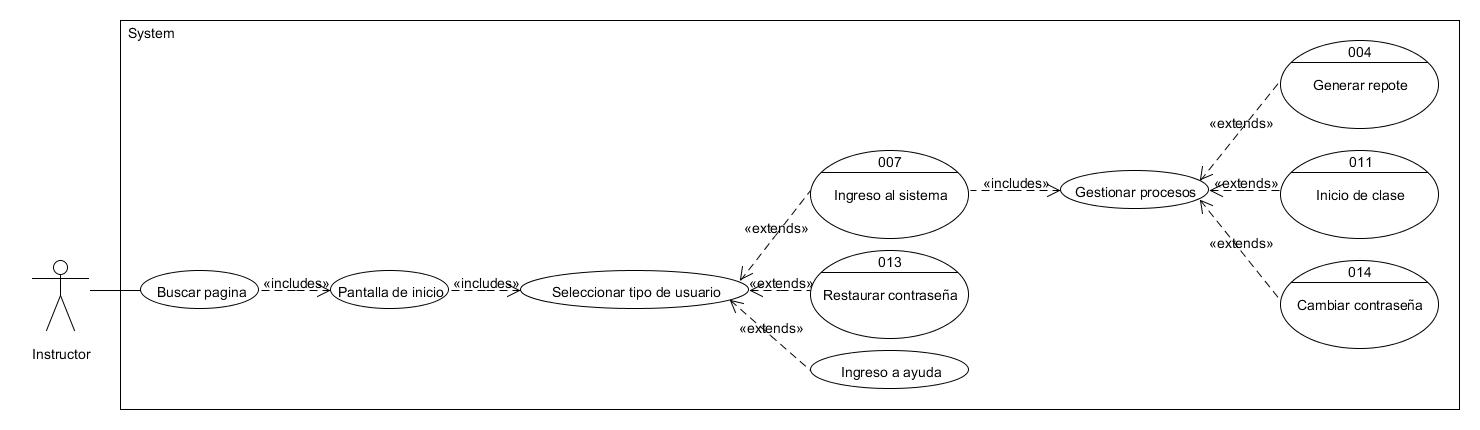


Ilustración 10: Caso de uso general Instructor

**Aprendiz:**

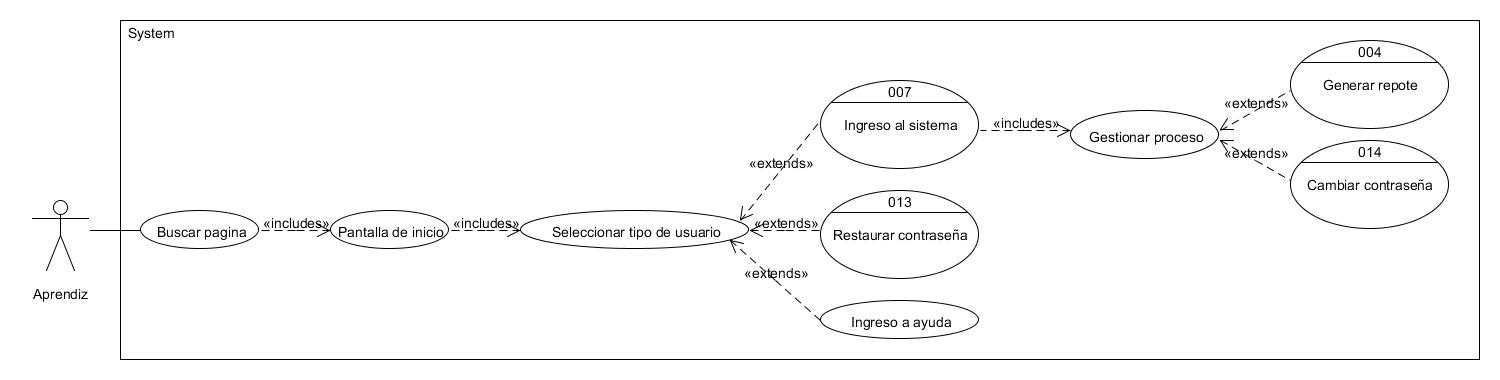
****

Ilustración 11: Caso de uso general Aprendiz

C.U 001 Registro de ingreso, C.U 002 Registrar asistencia, C.U 003 Registrar Asistencia

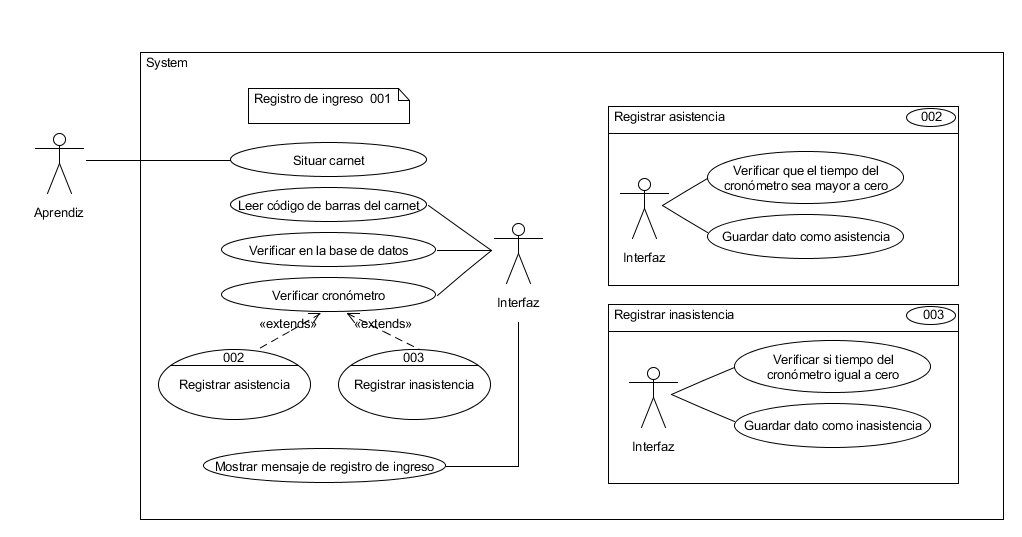


Ilustración 12: Caso de uso 001, 002, 003

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 001 | | | | **Nombre** | | Registro de ingreso de asistencia. |
| Propósito u objetivo | | | Registrar la asistencia o inasistencia de los aprendices. | | | | | |
| Actores | | | Aprendiz, Sistema. | | | | | |
| Entradas | | | * Código. | | | | | |
| Salidas | | | * Mensaje de registro de ingreso. | | | | | |
| Pre-Condiciones | | | * El aprendiz tiene que estar registrado. * El aprendiz tiene que situar correctamente el carnet. | | | | | |
| Post-Condición | | | Condición final de éxito:   * Se registró la asistencia correctamente. | | | | | |
| Condición final de fallo:   * No se pudo leer el código. * No se registró la asistencia por retardo del aprendiz. | | | | | |
| Flujo | | | | | | | | |
| Actor: Aprendiz | | | | | **Sistema: Program Art** | | | |
| 1 | Situar carnet. | | | |  | |  | |
|  |  | | | | **2** | | Leer código de barras del carnet. | |
|  |  | | | | **3** | | Verificar en la base de datos. | |
|  |  | | | | **4** | | Verificar el tiempo del cronómetro. | |
|  |  | | | | **5** | | Registrar asistencia o inasistencia. | |
|  |  | | | | **6** | | Mostrar mensaje de registro de ingreso. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | Asistencia: el tiempo es mayor a cero.  Inasistencia: el tiempo es menor a cero. | | | | |
| Variaciones (Excepciones): | | | | -Lectura de código errónea: El aprendiz no situó adecuadamente el carnet o el lector presento fallas y no se pudo realizar la lectura del código de barras.  -Retardo justificado: el aprendiz justificó el retardo y se le agrego el debido comentario al registro de asistencia. | | | | |

C.U 004 Generar reporte, C.U 005 Archivo pdf, C.U 006 Archivo Excel

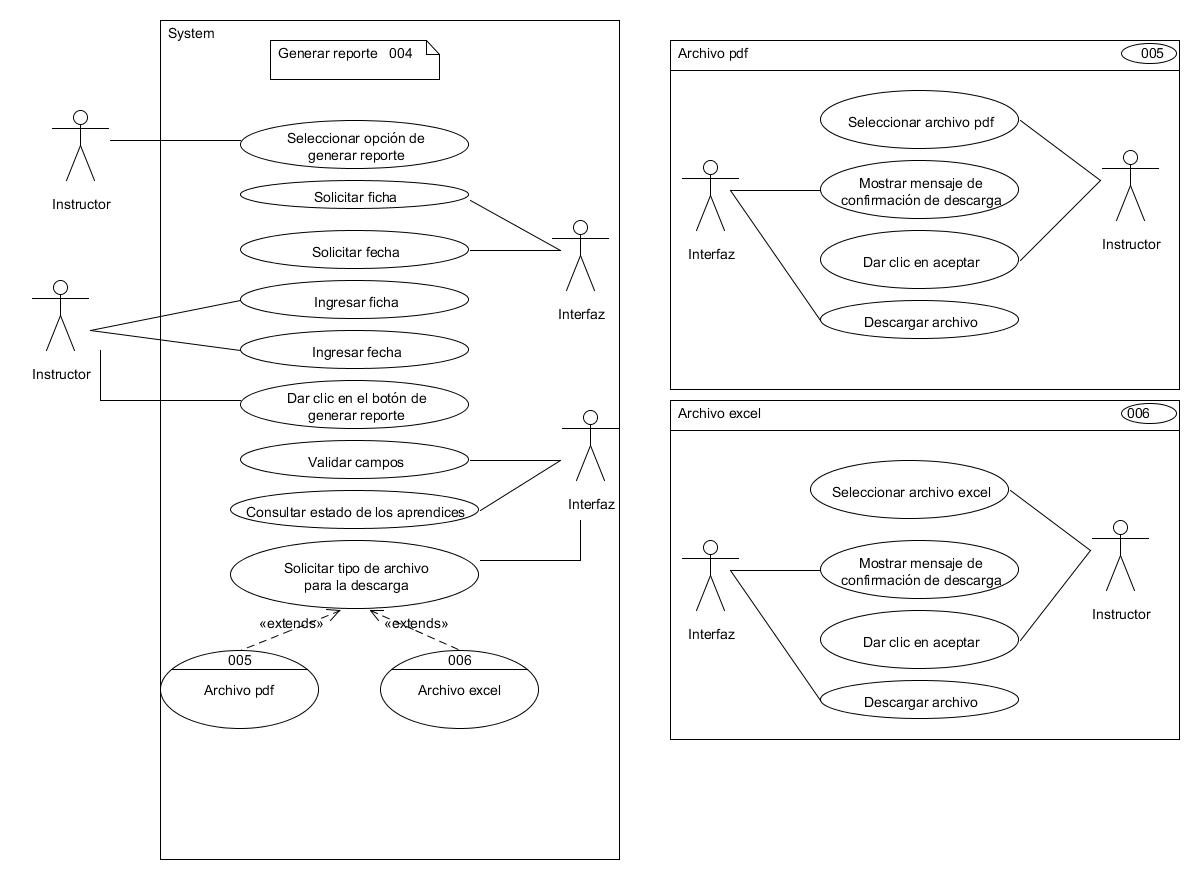


Ilustración 13: Caso de uso 004, 005, 006

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 004 | | | | **Nombre** | | Generar reporte |
| Propósito u objetivo | | | Generar reporte de una ficha. | | | | | |
| Actores | | | Instructor, Sistema. | | | | | |
| Entradas | | | * Ficha * Fecha | | | | | |
| Salidas | | | * Solicitud de datos requeridos. * Mensaje de confirmación. | | | | | |
| Pre-Condiciones | | | * La fecha tiene que ser menor al día de consulta. * El instructor tiene que estar vinculado con la ficha a consultar. * La ficha tiene que estar registrada en el sistema. | | | | | |
| Post-Condición | | | Condición final de éxito:   * Se generó correctamente el reporte. | | | | | |
| Condición final de fallo:   * No se pudo generar el reporte. * Datos mal ingresados. | | | | | |
| Flujo | | | | | | | | |
| Actor: Instructor | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar opción de generar reporte. | | | |  | |  | |
|  |  | | | | **2** | | Solicitar ficha a consultar. | |
|  |  | | | | **3** | | Solicitar fecha. | |
| 4 | Ingresar ficha. | | | |  | |  | |
| 5 | Ingresar fecha. | | | |  | |  | |
| 6 | Dar clic en el botón de generar reporte. | | | |  | |  | |
|  |  | | | | **7** | | Validar campos | |
|  |  | | | | **8** | | Consultar estado de los aprendices. | |
|  |  | | | | **9** | | Mostrar mensaje de confirmación de descarga. | |
| 10 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **11** | | Descargar archivo. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | -Información errónea: la información ingresada no coincidió con ninguna registrada. | | | | |

C.U 007 Ingreso al sistema

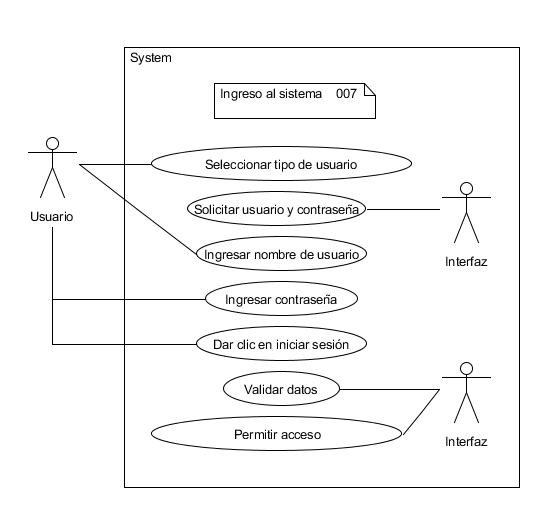


Ilustración 14: Caso de uso 007

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 007 | | | | **Nombre** | | Ingresar al sistema |
| Propósito u objetivo | | | El usuario (Administrador, Instructor o Aprendiz) pueda ingresar al sistema. | | | | | |
| Actores | | | Administrador, Instructor o Aprendiz y Sistema. | | | | | |
| Entradas | | | * Nombre de usuario * Contraseña | | | | | |
| Salidas | | | * Tipos de usuarios * Pantalla de ingreso | | | | | |
| Pre-Condiciones | | | * El usuario debe estar registrado. * El usuario debe escoger el tipo de usuario correspondiente al que está registrado. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario ingreso con éxito. | | | | | |
| Condición final de fallo:   * El usuario no pudo ingresar. * Usuario o contraseña incorrectos. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador, Instructor o aprendiz | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar tipo de usuario. | | | |  | |  | |
|  |  | | | | **2** | | Solicitar usuario y contraseña. | |
| 3 | Digitar nombre de usuario. | | | |  | |  | |
| 4 | Digitar contraseña. | | | |  | |  | |
| 5 | Oprimir botón iniciar sesión | | | |  | |  | |
|  |  | | | | **6** | | Validar campos. | |
|  |  | | | | **7** | | Permitir acceso. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

C.U 008 Generar reporte aprendiz

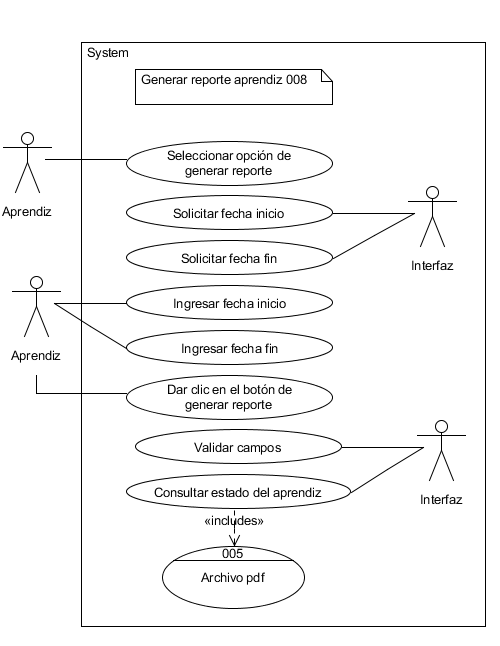


Ilustración 15: Caso de uso 008

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 008 | | | | **Nombre** | | Generar reporte aprendiz |
| Propósito u objetivo | | | Que el usuario (Aprendiz) pueda generar un reporte. | | | | | |
| Actores | | | Aprendiz, Sistema. | | | | | |
| Entradas | | | * Fecha inicio * Fecha fin | | | | | |
| Salidas | | | * Solicitud de fechas. * Mensaje de confirmación de descarga. | | | | | |
| Pre-Condiciones | | | * El aprendiz tiene que haber iniciado sesión. * Las fechas no deben ser mayores a la del día de consulta. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El reporte se generó correctamente. | | | | | |
| Condición final de fallo:   * El reporte no se pudo generar. | | | | | |
| Flujo | | | | | | | | |
| Actor: Aprendiz | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar opción de generar reporte. | | | |  | |  | |
|  |  | | | | **2** | | Solicitar fecha inicio. | |
|  |  | | | | **3** | | Solicitar fecha fin. | |
| 4 | Ingresar fecha inicio. | | | |  | |  | |
| 5 | Ingresar fecha fin. | | | |  | |  | |
| 6 | Dar clic en el botón de generar reporte. | | | |  | |  | |
|  |  | | | | **7** | | Validar campos. | |
|  |  | | | | **8** | | Consultar estado del aprendiz. | |
|  |  | | | | **9** | | Mostrar mensaje de confirmación de descarga. | |
| 10 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **11** | | Descargar archivo. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

C.U 009 Modificar usuario

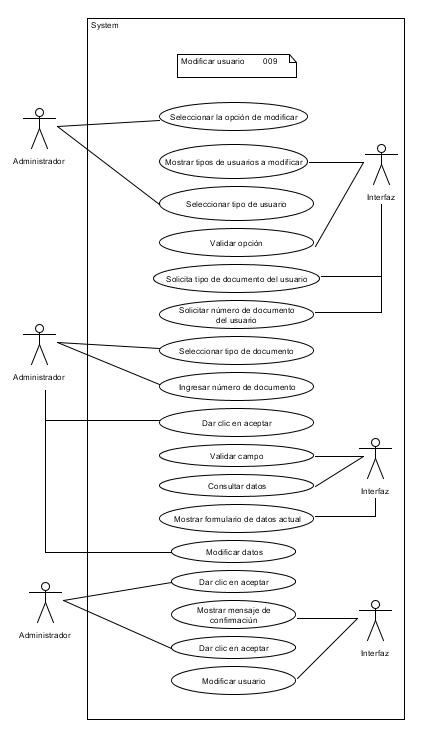


Ilustración 16: Caso de uso 009

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 009 | | | | **Nombre** | | Modificar usuario |
| Propósito u objetivo | | | Modificar los datos de un usuario registrado. | | | | | |
| Actores | | | Administrador, Sistema. | | | | | |
| Entradas | | | * Documento * Nombre * Apellido * Correo * Teléfono * Cargo * Titulo * Nombre de usuario * Contraseña * Confirmación de contraseña | | | | | |
| Salidas | | | * Tipos de usuarios * Solicitud de nombre de usuario * Formulario de datos actual * Mensaje de confirmación | | | | | |
| Pre-Condiciones | | | * El usuario tiene que estar registrado. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario se modificó con éxito. | | | | | |
| Condición final de fallo:   * El usuario no se pudo modificar. * Datos mal ingresados. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar la opción de modificar. | | | |  | |  | |
|  |  | | | | **2** | | Mostrar tipo de usuario a modificar. | |
| 3 | Seleccionar tipo de usuario a modificar | | | |  | |  | |
|  |  | | | | **4** | | Validar opción. | |
|  |  | | | | **5** | | Solicitar nombre de usuario a modificar. | |
| 6 | Ingresar nombre de usuario. | | | |  | |  | |
| 7 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **8** | | Validar campo. | |
|  |  | | | | **9** | | Consultar usuario. | |
|  |  | | | | **10** | | Mostrar formulario de datos actual de usuario. | |
| 11 | Modificar datos. | | | |  | |  | |
| 12 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **13** | | Mostrar mensaje de  Confirmación. | |
| 14 | Dar clic en aceptar | | | |  | |  | |
|  |  | | | | **15** | | Modificar usuario | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | Tipo de usuario: si el usuario es aprendiz, instructor o administrador el formulario adicionara o quitara algunos datos. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

C.U 010 Registrar usuario

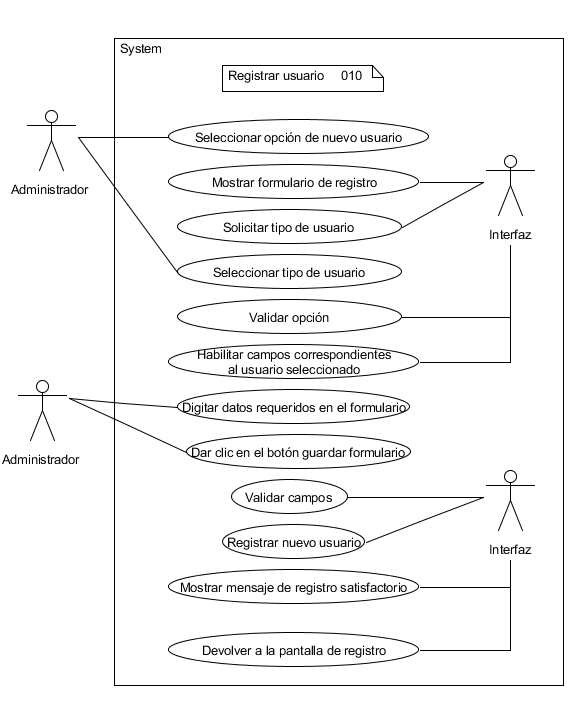


Ilustración 17: Caso de uso 010

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 010 | | | | **Nombre** | | Registrar usuario |
| Propósito u objetivo | | | Registrar un usuario en el sistema. | | | | | |
| Actores | | | Administrador, Sistema. | | | | | |
| Entradas | | | * Documento * Nombre * Apellido * Correo * Teléfono * Cargo * Titulo * Nombre de usuario * Contraseña * Confirmación de contraseña | | | | | |
| Salidas | | | * Formulario de registro * Tipos de usuarios * Mensaje de registro exitoso * Pantalla de registro | | | | | |
| Pre-Condiciones | | | * El usuario no tiene que estar registrado. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario se registró con éxito. | | | | | |
| Condición final de fallo:   * El usuario no se pudo registrar. * Datos mal ingresados.   El usuario ya está registrado. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar opción de nuevo usuario. | | | |  | |  | |
|  |  | | | | **2** | | Mostrar formulario de registro. | |
|  |  | | | | **3** | | Solicitar tipo de usuario | |
| 4 | Seleccionar tipo de usuario. | | | |  | |  | |
|  |  | | | | **5** | | Validar opción. | |
|  |  | | | | **6** | | Habilitar campos correspondientes al usuario seleccionado. | |
| 7 | Digitar datos requeridos en el formulario. | | | |  | |  | |
| 8 | Dar clic en el botón guardar formulario | | | |  | |  | |
|  |  | | | | **9** | | Validar campos | |
|  |  | | | | **10** | | Registrar nuevo usuario | |
|  |  | | | | **11** | | Mostrar mensaje de registro satisfactorio | |
|  |  | | | | **12** | | Devolver a la pantalla de registro | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | Tipo de usuario: si el usuario es Administrador, Instructor o Aprendiz el sistema bloquea o desbloquea algunos campos. | | | | |
| Variaciones (Excepciones): | | | | -Información incompleta: falta llenar información en el registro de usuario. Se solicita al usuario que complete la información.  - Registro ya existente: si ya existe un registro, se solicitará al usuario que lo cambie o termine el caso de uso. | | | | |

C.U 011 Inicio de clase, C.U 012 Configuración de cronómetro

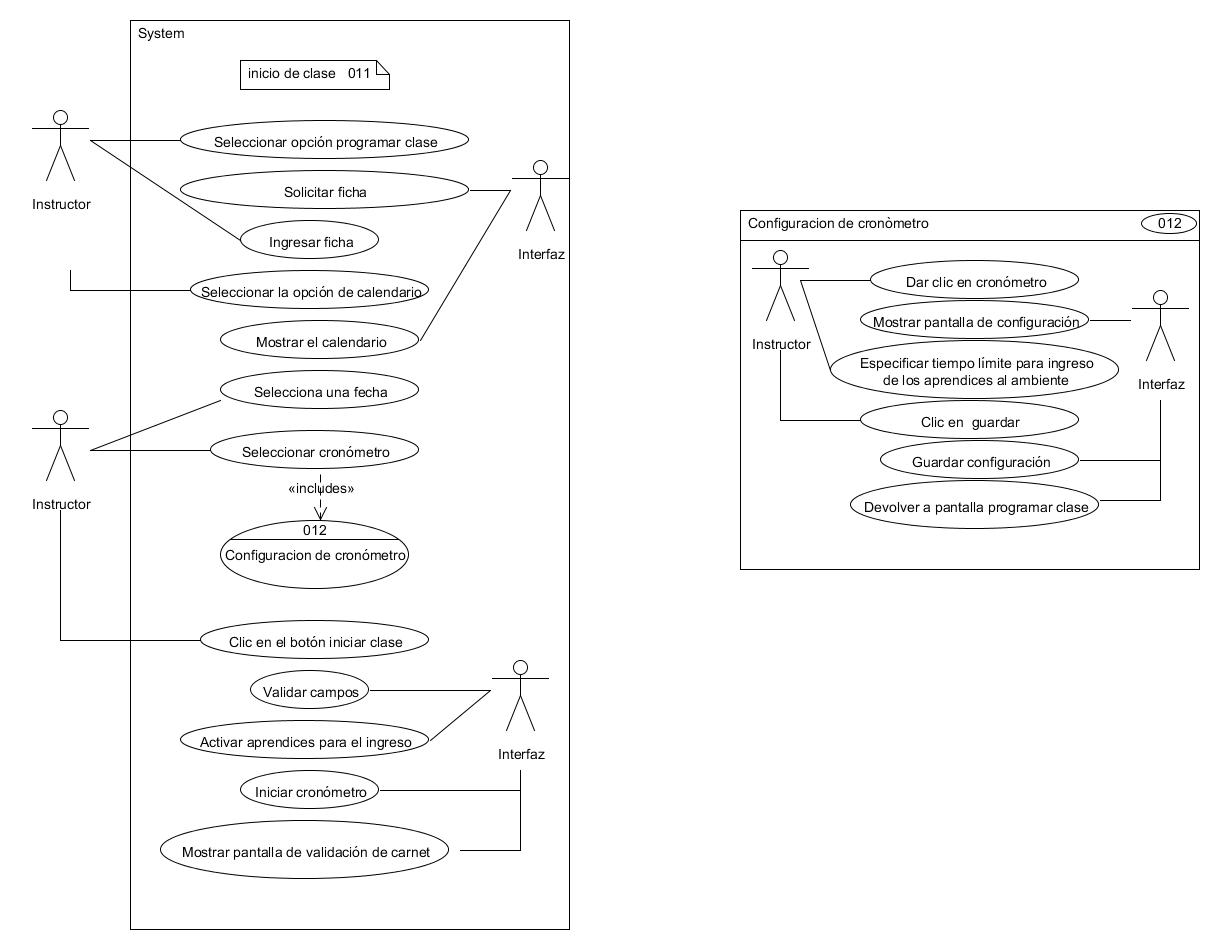


Ilustración 18: Caso de uso 011, 012

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 011 | | | | **Nombre** | | Inicio de clase |
| Propósito u objetivo | | | Registrar un usuario en el sistema. | | | | | |
| Actores | | | Instructor, Sistema. | | | | | |
| Entradas | | | * Ficha * Fecha * Tiempo en el cronómetro | | | | | |
| Salidas | | | * Solicitud de datos * Calendario * Pantalla de configuración de cronómetro * Pantalla de validación de carnet | | | | | |
| Pre-Condiciones | | | * El instructor tiene que estar vinculado con la ficha a programar. * El tiempo del cronómetro tiene que ser mayor a cero. * La fecha tiene que ser igual a la del día a programar. | | | | | |
| Post-Condición | | | Condición final de éxito:   * La clase se programó con éxito. | | | | | |
| Condición final de fallo:   * La clase no se pudo programar. * Datos mal ingresados. | | | | | |
| Flujo | | | | | | | | |
| Actor: Instructor | | | | | **Sistema: Program Art** | | | |
| 1 | Ingresar al módulo programar clase. | | | |  | |  | |
|  |  | | | | **2** | | Solicitar ficha. | |
| 3 | Ingresar ficha. | | | |  | |  | |
| 4 | Oprimir la opción de calendario. | | | |  | |  | |
|  |  | | | | **5** | | Mostrar el calendario. | |
| 6 | Selecciona una fecha | | | |  | |  | |
| 7 | Dar clic en cronómetro. | | | |  | |  | |
|  |  | | | | **8** | | Mostrar pantalla de configuración. | |
| 9 | Especificar tiempo límite para ingreso de los aprendices al ambiente. | | | |  | |  | |
| 10 | Oprimir el botón de guardar. | | | |  | |  | |
|  |  | | | | **11** | | Guardar configuración. | |
|  |  | | | | **12** | | Devolver a pantalla programar clase. | |
| 13 | Oprime el botón iniciar clase. | | | |  | |  | |
|  |  | | | | **14** | | Validar campos. | |
|  |  | | | | **15** | | Activar aprendices para el ingreso. | |
|  |  | | | | **16** | | Iniciar cronómetro. | |
|  |  | | | | **17** | | Mostrar pantalla de validación de carnet por código de barra. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | Configuración de cronómetro: el instructor puede configurar el cronómetro o dejar el tiempo que esta por defecto. | | | | |
| Variaciones (Excepciones): | | | | -Información incompleta: falta llenar información para inicio de clase. | | | | |

C.U 013 Restaurar contraseña

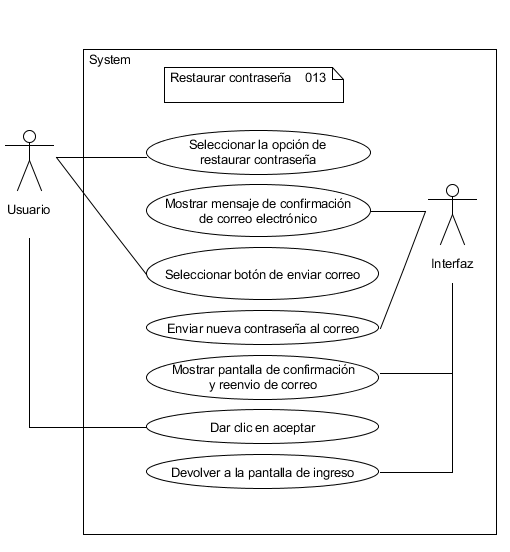


Ilustración 19: Caso de uso 013

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 013 | | | | **Nombre** | | Restaurar contraseña |
| Propósito u objetivo | | | Registrar un usuario en el sistema. | | | | | |
| Actores | | | Administrador, Instructor o Aprendiz y Sistema. | | | | | |
| Entradas | | | * Nueva contraseña * Confirmación de nueva contraseña | | | | | |
| Salidas | | | * Mensaje de confirmación de correo electrónico * Solicitud de nueva contraseña y confirmación * Pantalla de ingreso al sistema | | | | | |
| Pre-Condiciones | | | * El usuario tiene que estar registrado. * El usuario debe haber registrado un correo valido. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario pudo restaurar la contraseña con éxito. | | | | | |
| Condición final de fallo:   * El usuario no pudo restaurar la contraseña. * La contraseña y la confirmación no coinciden. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador, Instructor o Aprendiz | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar la opción de restaurar contraseña. | | | |  | |  | |
|  |  | | | | **2** | | Mostrar mensaje de confirmación de correo electrónico. | |
| 3 | Seleccionar botón de enviar correo. | | | |  | |  | |
|  |  | | | | **4** | | Enviar link al correo. | |
| 5 | Abrir link enviado al correo. | | | |  | |  | |
|  |  | | | | **6** | | Solicitar nueva contraseña. | |
| 7 | Ingresar nueva contraseña. | | | |  | |  | |
| 8 | Confirmar nuevamente contraseña. | | | |  | |  | |
| 9 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **10** | | Validar campos. | |
|  |  | | | | **11** | | Cambiar contraseña. | |
|  |  | | | | **12** | | Mostrar pantalla de ingreso al sistema. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

C.U 014 Cambiar contraseña

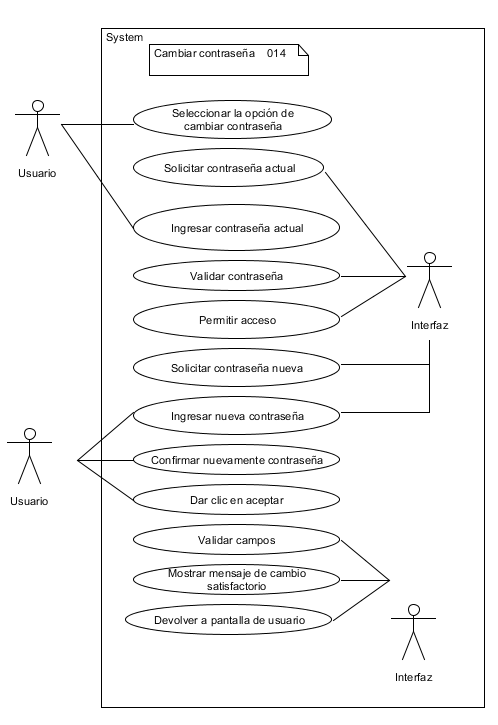


Ilustración 20: Caso de uso 014

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 014 | | | | **Nombre** | | Cambiar contraseña |
| Propósito u objetivo | | | Que un usuario pueda cambiar su contraseña. | | | | | |
| Actores | | | Administrador, Instructor o Aprendiz y Sistema. | | | | | |
| Entradas | | | * Contraseña actual * Nueva contraseña * Confirmación de nueva contraseña | | | | | |
| Salidas | | | * Solicitud de datos * Mensaje de cambio exitoso * Pantalla de usuario | | | | | |
| Pre-Condiciones | | | * El usuario tiene que haber iniciado sesión. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario cambio su contraseña con éxito. | | | | | |
| Condición final de fallo:   * El usuario no pudo cambiar su contraseña. * La contraseña actual no es correcta. * La nueva contraseña y la confirmación no coinciden. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar la opción de cambiar contraseña. | | | |  | |  | |
|  |  | | | | **2** | | Solicitar contraseña actual. | |
| 3 | Ingresar contraseña actual. | | | |  | |  | |
|  |  | | | | **4** | | Validar contraseña. | |
|  |  | | | | **5** | | Permitir acceso. | |
|  |  | | | | **6** | | Solicitar contraseña nueva. | |
|  |  | | | | **7** | | Solicitar confirmación de nueva contraseña. | |
| 8 | Ingresar nueva contraseña. | | | |  | |  | |
| 9 | Confirmar nuevamente contraseña. | | | |  | |  | |
| 10 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **11** | | Validar campos. | |
|  |  | | | | **12** | | Mostrar mensaje de cambio satisfactorio. | |
|  |  | | | | **13** | | Mostrar pantalla de usuario. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

C.U 015 Consultar usuario

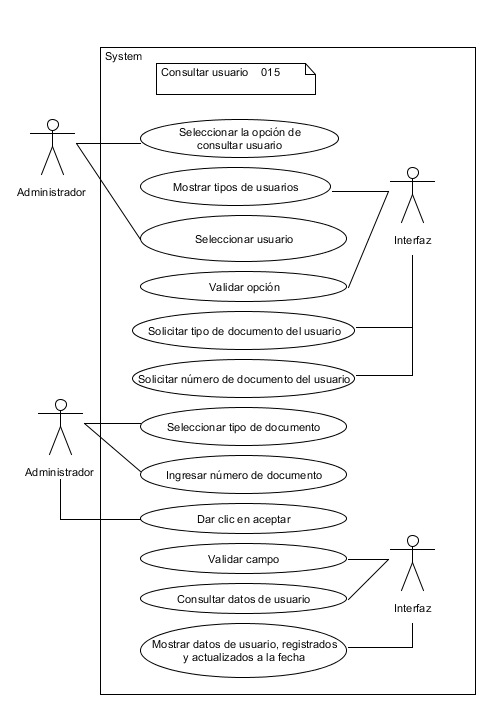


Ilustración 21: Caso de uso

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 015 | | | | **Nombre** | | Consultar usuario |
| Propósito u objetivo | | | Consultar un usuario registrado en el sistema. | | | | | |
| Actores | | | Administrador, Sistema. | | | | | |
| Entradas | | | * Nombre de usuario (documento). | | | | | |
| Salidas | | | * Tipos de usuarios. * Solicitud de nombre de usuario (documento). * Datos registrados y actualizados a la fecha. | | | | | |
| Pre-Condiciones | | | * El usuario tiene que estar registrado. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario se buscó con éxito. | | | | | |
| Condición final de fallo:   * El usuario no se pudo buscar. * El nombre de usuario está mal ingresado. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar la opción de consultar usuario. | | | |  | |  | |
|  |  | | | | **2** | | Mostrar tipos de usuarios. | |
| 3 | Seleccionar usuario. | | | |  | |  | |
|  |  | | | | **4** | | Validar opción. | |
|  |  | | | | **5** | | Solicita nombre de usuario a consultar. | |
| 6 | Ingresar nombre de usuario. | | | |  | |  | |
| 7 | Digitar datos requeridos en el formulario. | | | |  | |  | |
| 8 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **9** | | Validar campo. | |
|  |  | | | | **10** | | Consultar datos de usuario. | |
|  |  | | | | **11** | | Mostrar datos de usuario, registrados y actualizados a la fecha. | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

C.U 016 Eliminar usuario

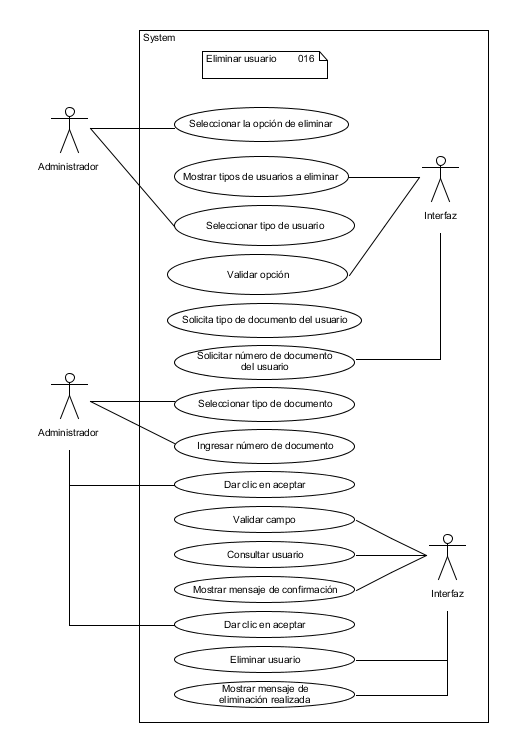


Ilustración 22: Caso de uso 016

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: | | Program Art | | | | Fecha: | | 30 de Julio de 2015 |
| Autor: | | Esneider Rojas  Cindy Saldaña | | | | **Versión:** | | 4.0 |
| Id Caso de Uso | | CU – 016 | | | | **Nombre** | | Eliminar usuario |
| Propósito u objetivo | | | Eliminar un usuario registrado en el sistema. | | | | | |
| Actores | | | Administrador, Sistema. | | | | | |
| Entradas | | | * Nombre de usuario (documento). | | | | | |
| Salidas | | | * Tipos de usuarios. * Solicitud de nombre de usuario (documento). * Mensaje de confirmación. * Mensaje de eliminación exitosa. | | | | | |
| Pre-Condiciones | | | * El usuario tiene que estar registrado. * El usuario no tiene que estar activo. | | | | | |
| Post-Condición | | | Condición final de éxito:   * El usuario se eliminó con éxito. | | | | | |
| Condición final de fallo:   * El usuario no se pudo eliminar. * El nombre de usuario está mal ingresado. | | | | | |
| Flujo | | | | | | | | |
| Actor: Administrador | | | | | **Sistema: Program Art** | | | |
| 1 | Seleccionar la opción de eliminar. | | | |  | |  | |
|  |  | | | | **2** | | Mostrar tipos de usuario a eliminar. | |
| 3 | Seleccionar tipo de usuario. | | | |  | |  | |
|  |  | | | | **4** | | Validar opción. | |
|  |  | | | | **5** | | Solicitar nombre usuario. | |
| 6 | Ingresar usuario. | | | |  | |  | |
| 7 | Dar clic en aceptar. | | | |  | |  | |
|  |  | | | | **8** | | Validar campo. | |
|  |  | | | | **9** | | Consultar usuario. | |
|  |  | | | | **10** | | Mostrar mensaje de confirmación. | |
| 11 | Dar clic en aceptar | | | |  | |  | |
|  |  | | | | **12** | | Eliminar usuario | |
|  |  | | | | **13** | | Mostrar mensaje de eliminación realizada | |
|  |  | | | |  | |  | |
| Variaciones (Alternativas): | | | | No aplica. | | | | |
| Variaciones (Excepciones): | | | | No aplica. | | | | |

### 8.2.3 Diagramas de colaboración

-Registro de ingreso

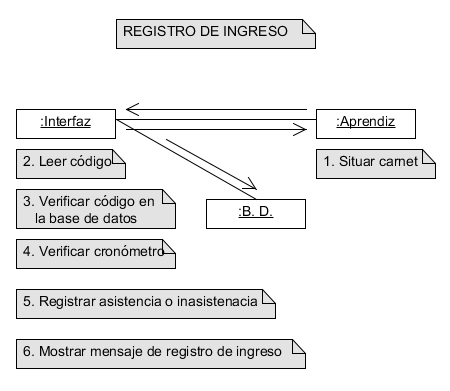


Ilustración 23: Diagrama de colaboración Registro de ingreso

-Generar reporte

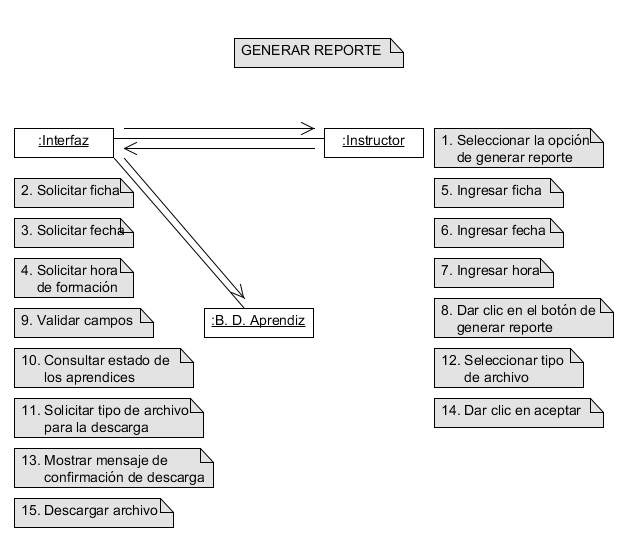


Ilustración 24: Diagrama de colaboración Generar reporte

-Ingresar al sistema

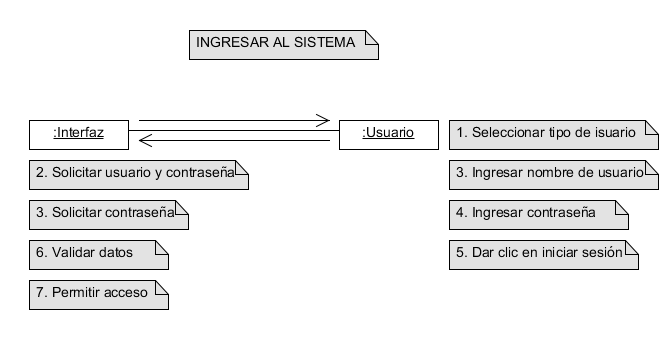


Ilustración 25: Diagrama de colaboración Ingresar al sistema

-Generar reporte aprendiz

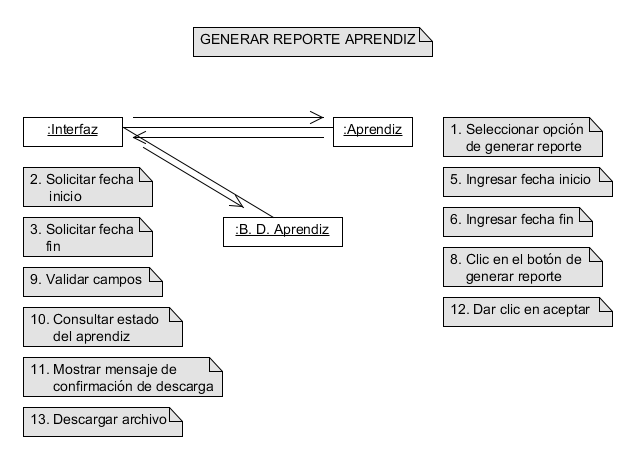


Ilustración 26: Diagrama de colaboración Generar reporte aprendiz

-Modificar usuario

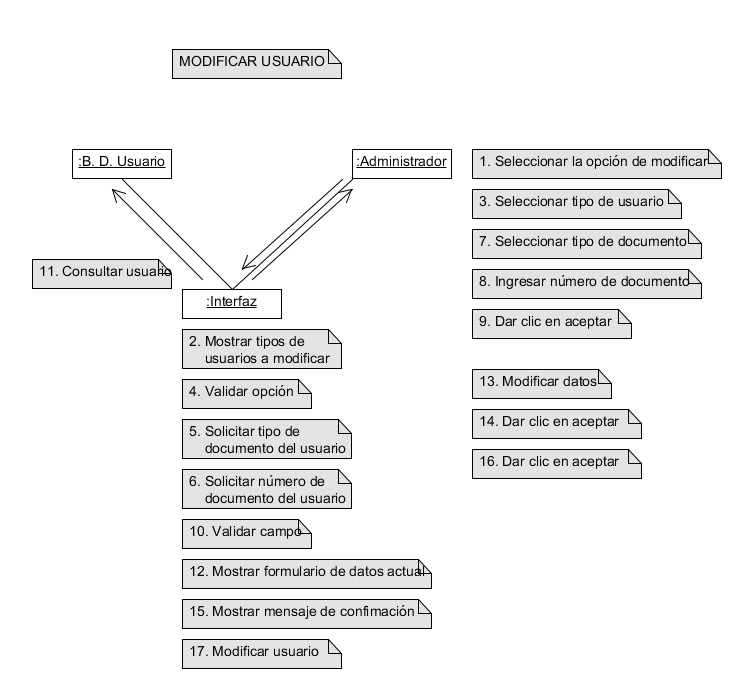


Ilustración 27: Diagrama de colaboración Modificar usuario

-Registrar usuario

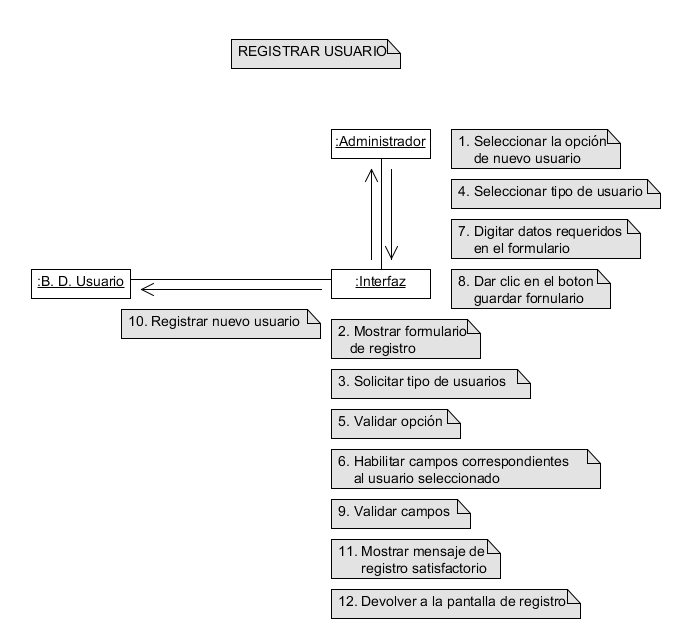


Ilustración 28: Diagrama de colaboración Registrar usuario

-Inicio de clase

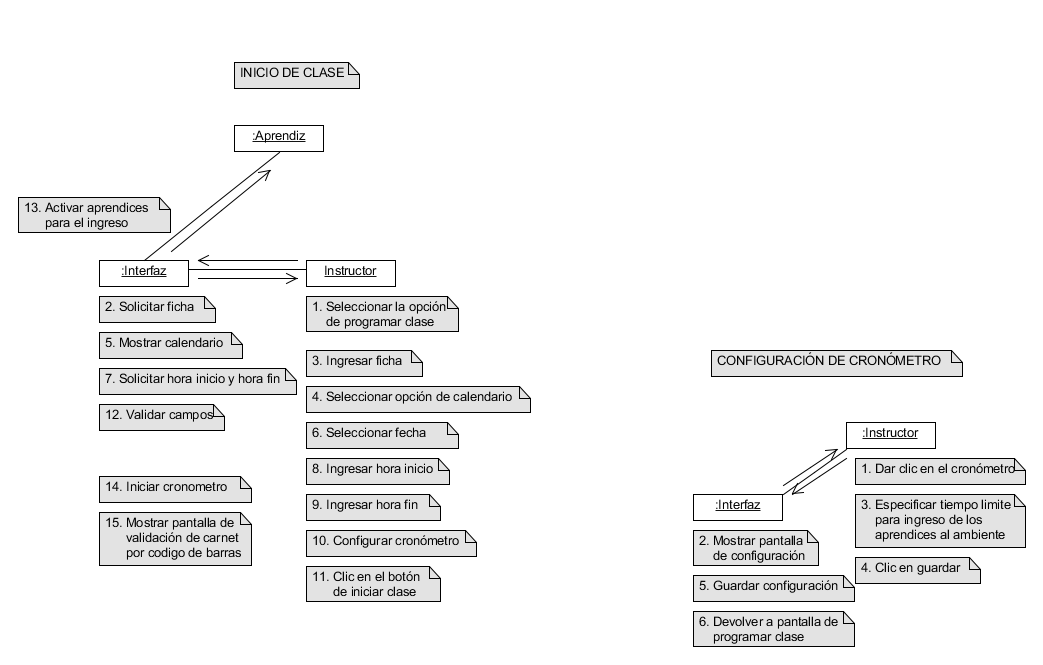


Ilustración 29: Diagrama de colaboración Inicio de clase

-Restaurar contraseña

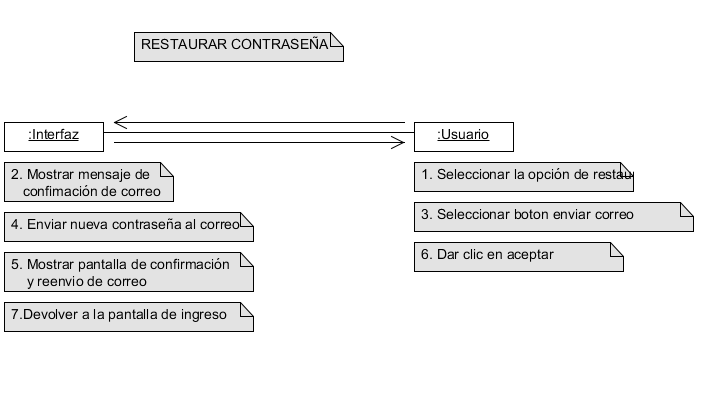


Ilustración 30: Diagrama de colaboración Restaurar contraseña

-Cambiar contraseña

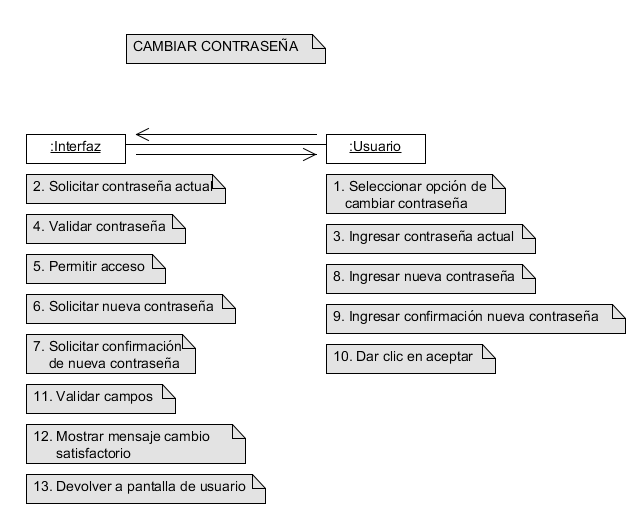


Ilustración 31: Diagrama de colaboración Cambiar contraseña

-Consultar usuario

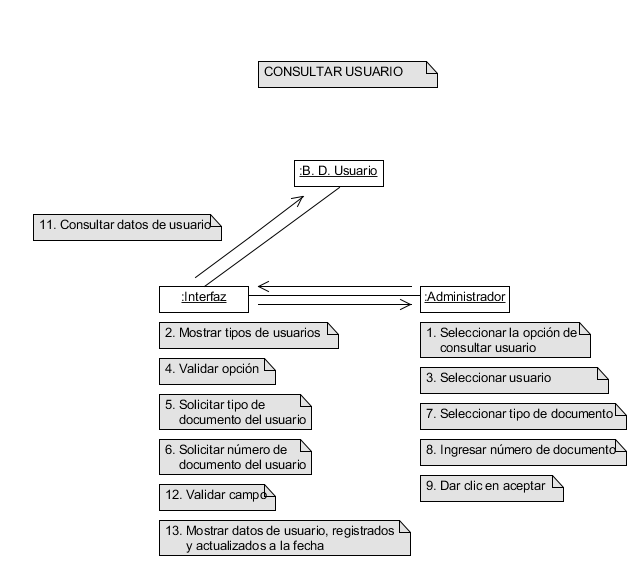


Ilustración 32: Diagrama de colaboración Consultar usuario

-Eliminar usuario

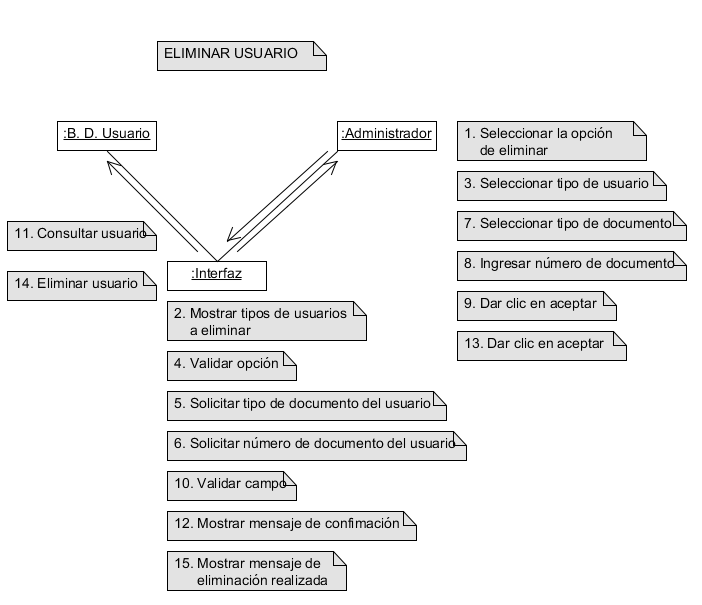


Ilustración 33: Diagrama de colaboración Eliminar usuario

## 8.4 Aplicación web - base de datos

## 8.5 Aplicación cliente

### 8.5.1 Diagramas de actividades

-Registro de ingreso

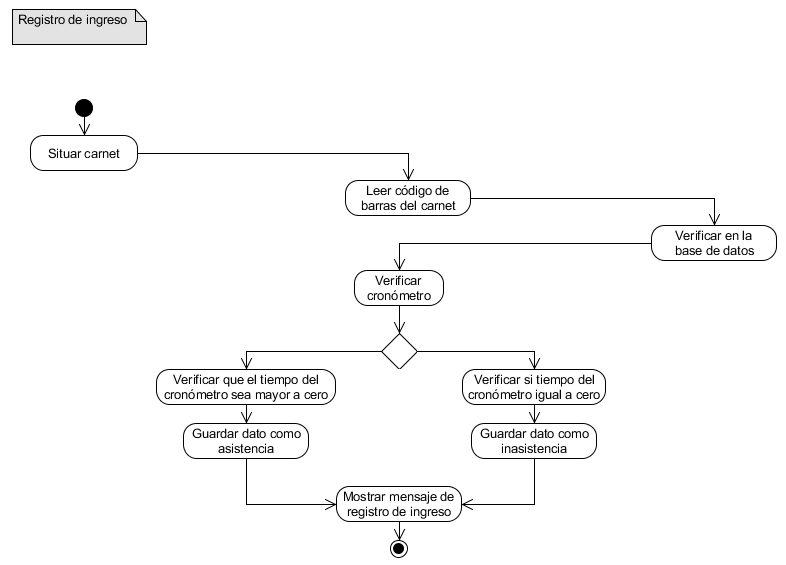


Ilustración : Diagrama de actividades Registro de ingreso

-Generar reporte

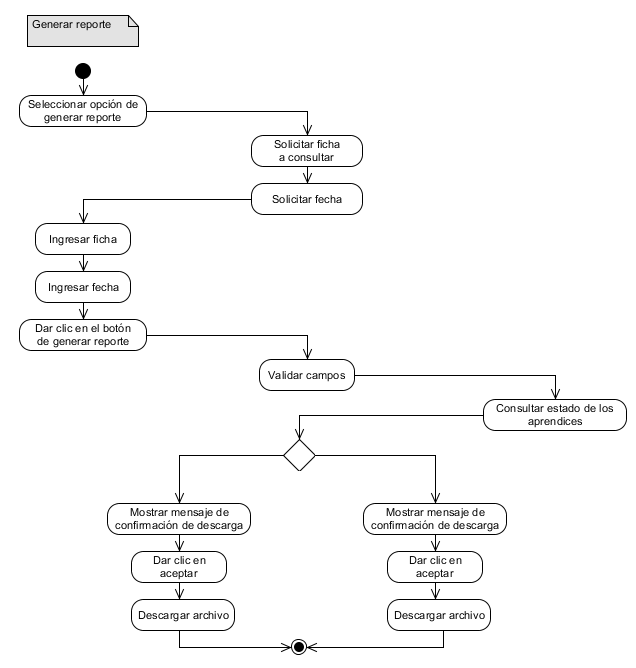


Ilustración : Diagrama de actividades Generar reporte

-Ingreso al sistema

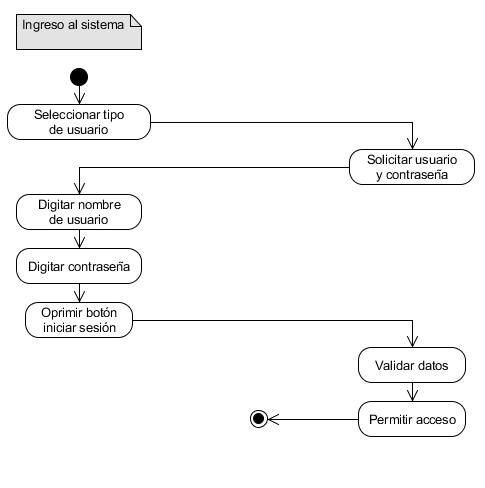


Ilustración : Diagrama de actividades Ingreso al sistema

-Generar reporte aprendiz

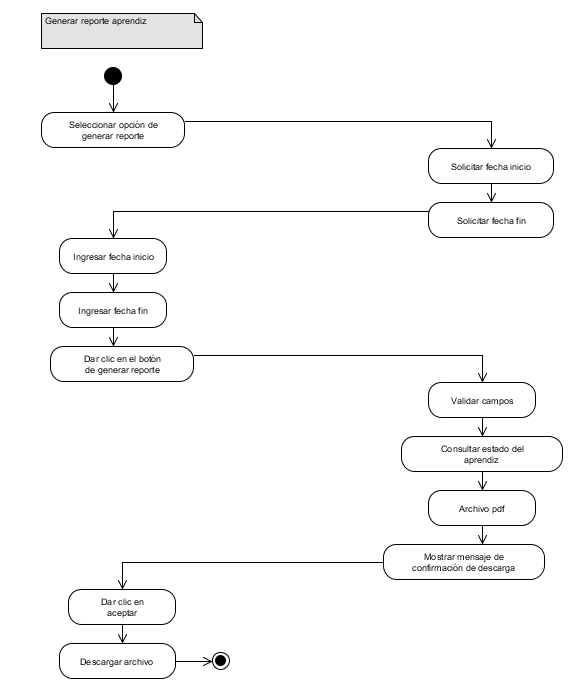
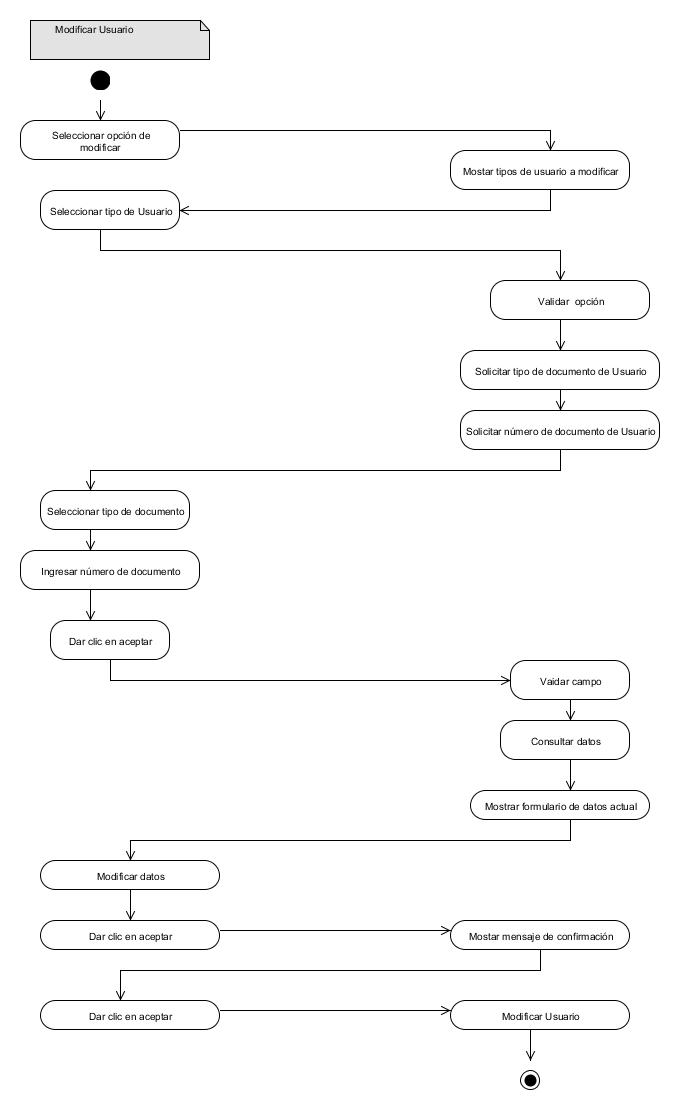


Ilustración : Diagrama de actividades Generar reporte aprendiz

-Modificar usuario



-Registrar usuario

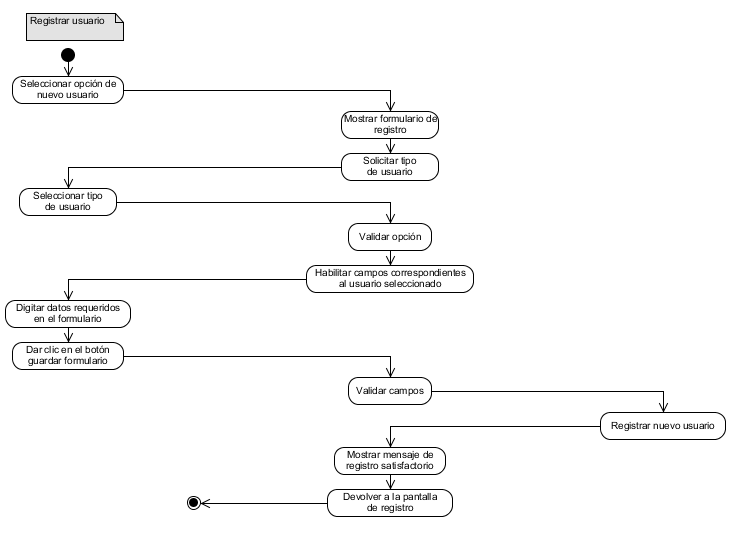


Ilustración : Diagrama de actividades registrar usuario

-Inicio de clase

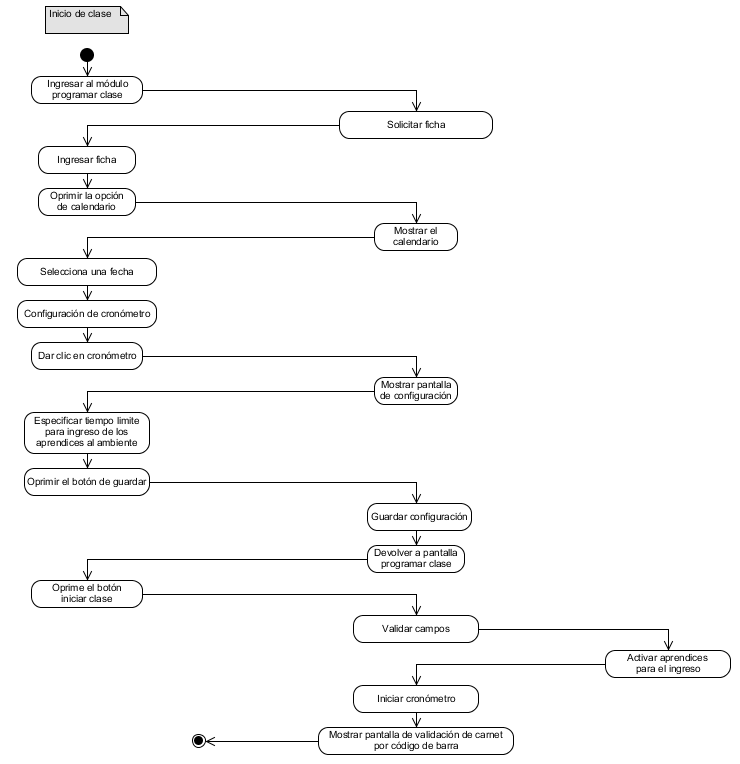


Ilustración : Diagrama de actividades Inicio de clase

-Restaurar contraseña

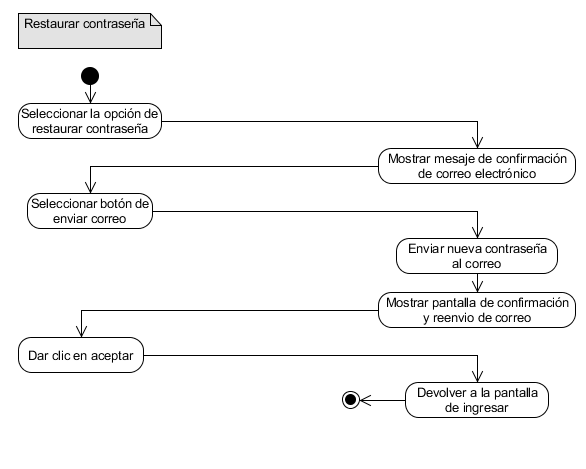


Ilustración 40: Diagrama de actividades Restaurar contraseña

-Cambiar contraseña

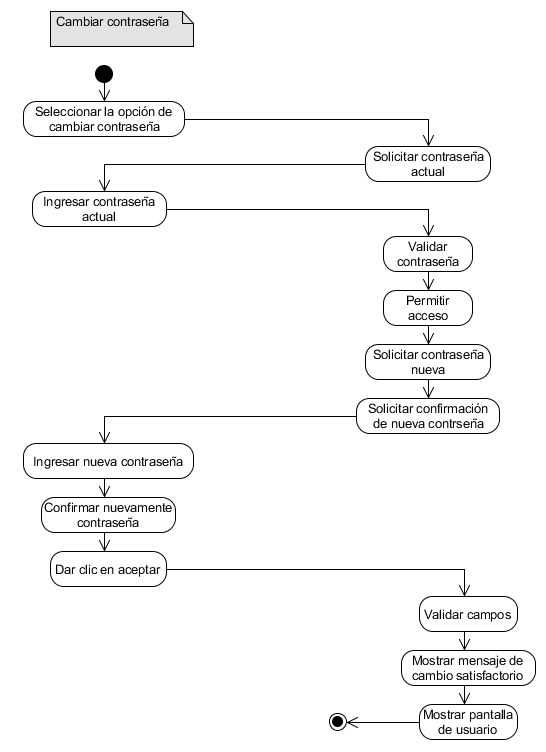


Ilustración : Diagrama de actividades Cambiar contraseña

-Consultar usuario

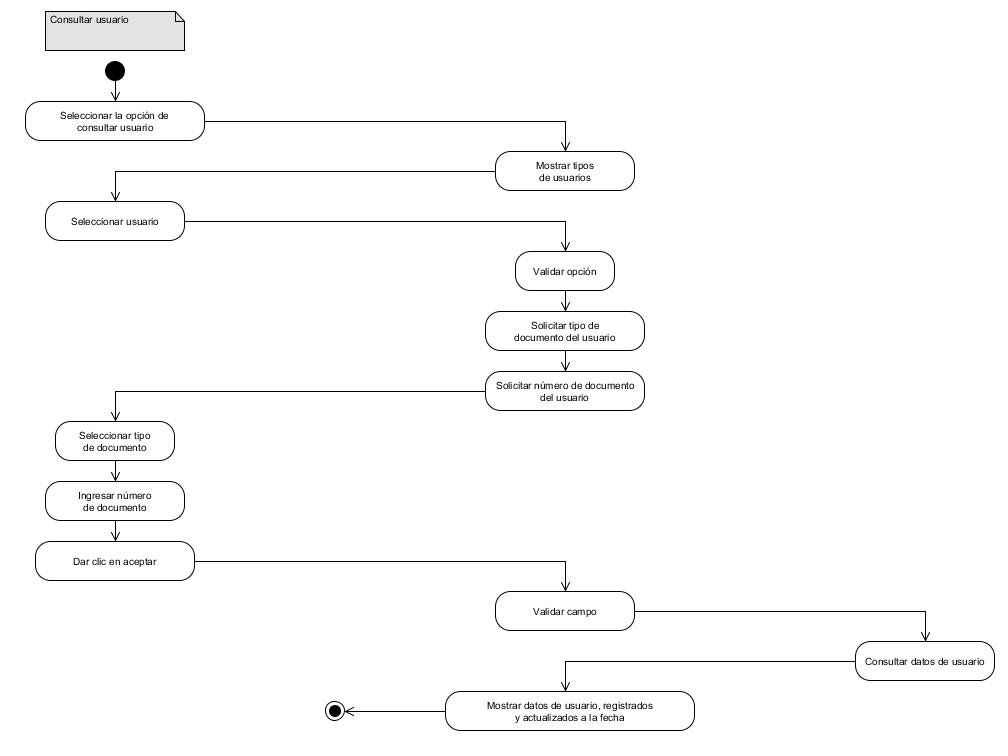


Ilustración : Diagrama de actividades Consultar usuario

-Eliminar usuario

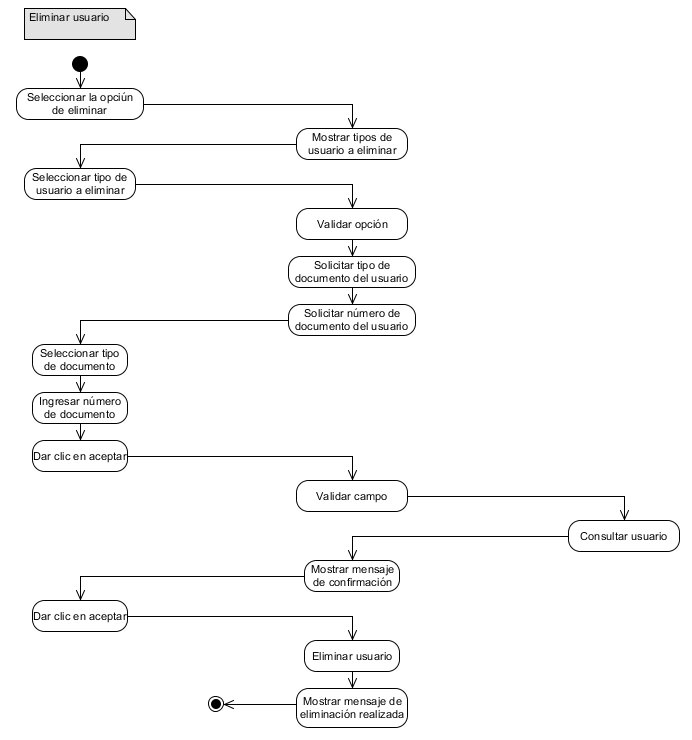


Ilustración : Eliminar usuario

### 8.5.2 Diagramas de secuencia

-Registro de ingreso

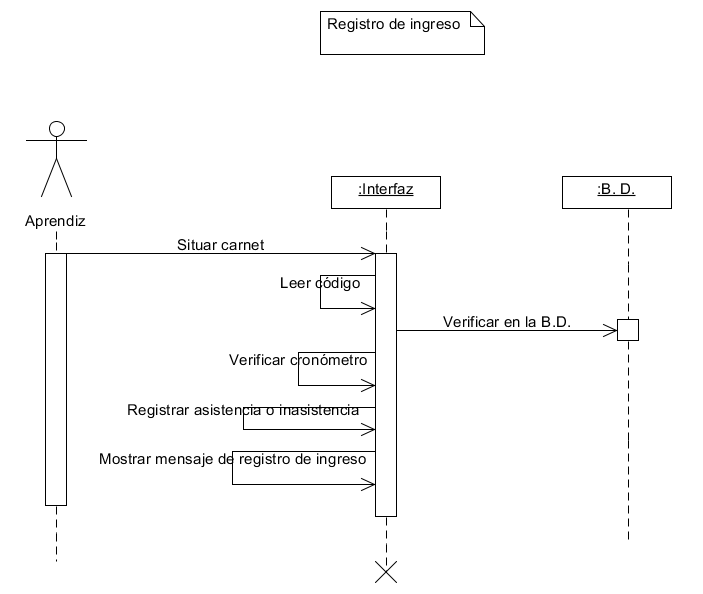


Ilustración : Diagrama de secuencia Registro de ingreso

-Generar reporte

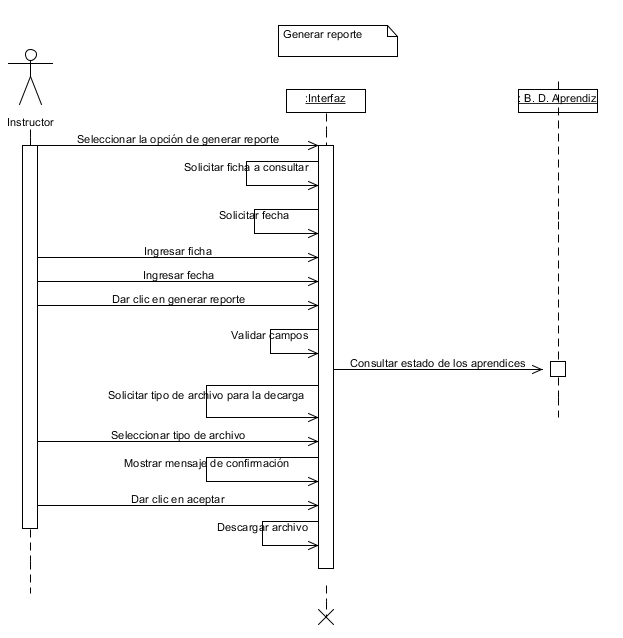


Ilustración : Diagrama de secuencia Generar reporte

-Ingreso al sistema

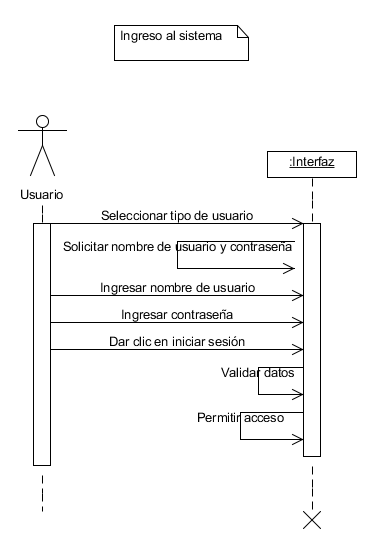


Ilustración : Diagrama de secuencia Ingreso al sistema

- Generar reporte aprendiz

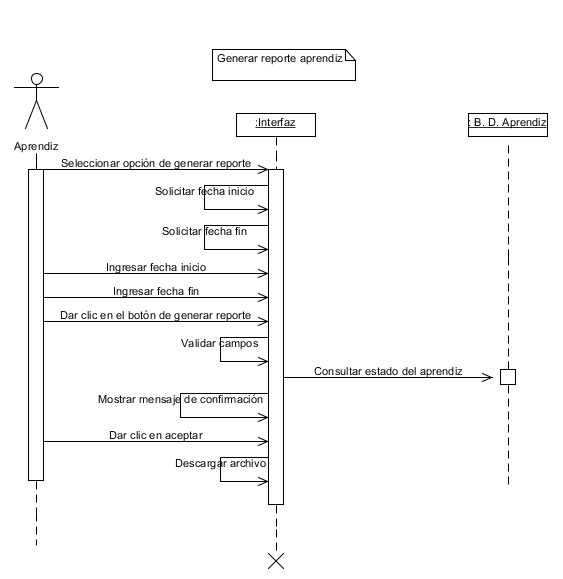


Ilustración : Diagrama de secuencia Generar reporte aprendiz

-Modificar usuario

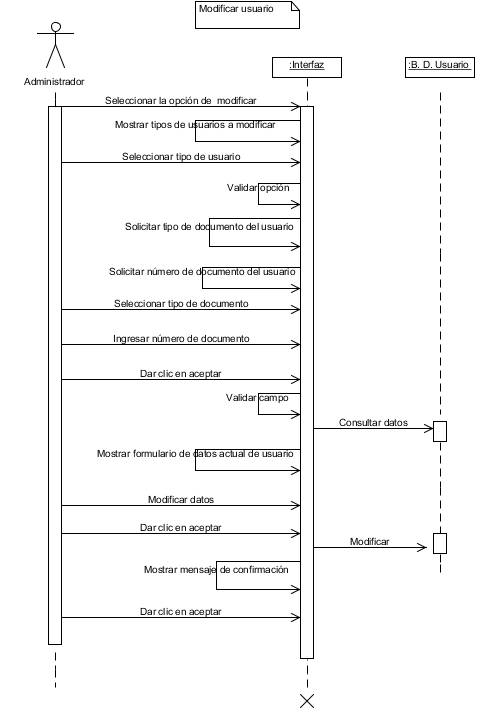


Ilustración : Diagrama de secuencia Modificar usuario

-Registrar usuario

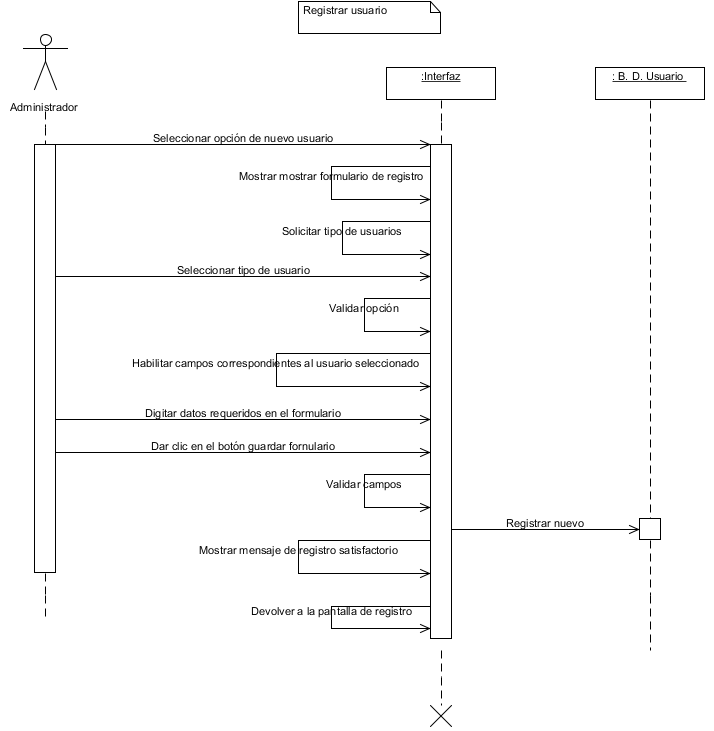


Ilustración : Diagrama de secuencia Registrar usuario

-Inicio de clase

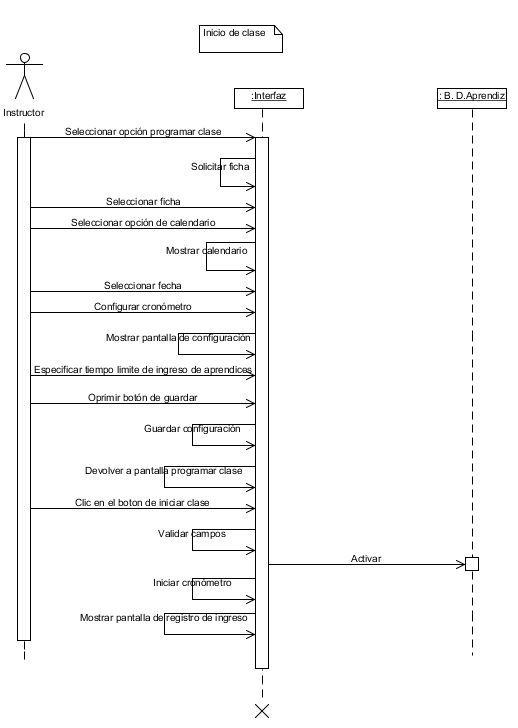


Ilustración : Diagrama de secuencia Inicio de clase

-Restaurar contraseña

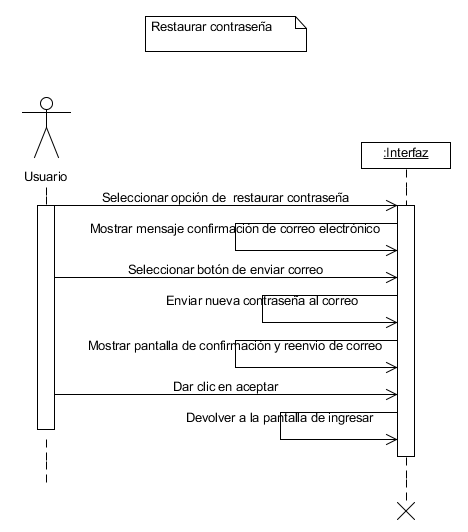


Ilustración : Diagrama de secuencia Restaurar contraseña

-Cambiar contraseña

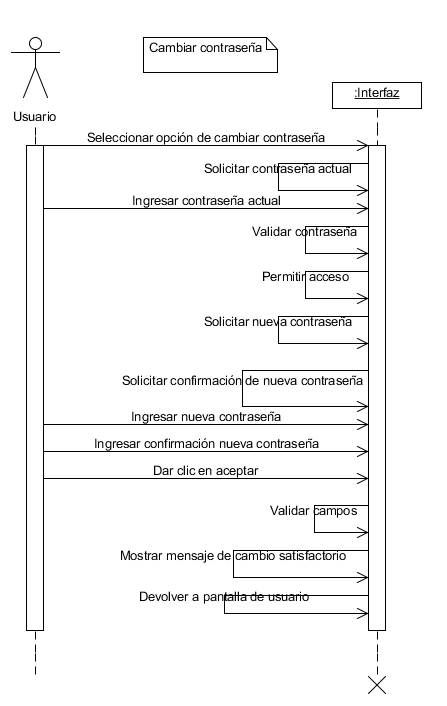


Ilustración : Diagrama de secuencia Cambiar contraseña

-Consultar usuario

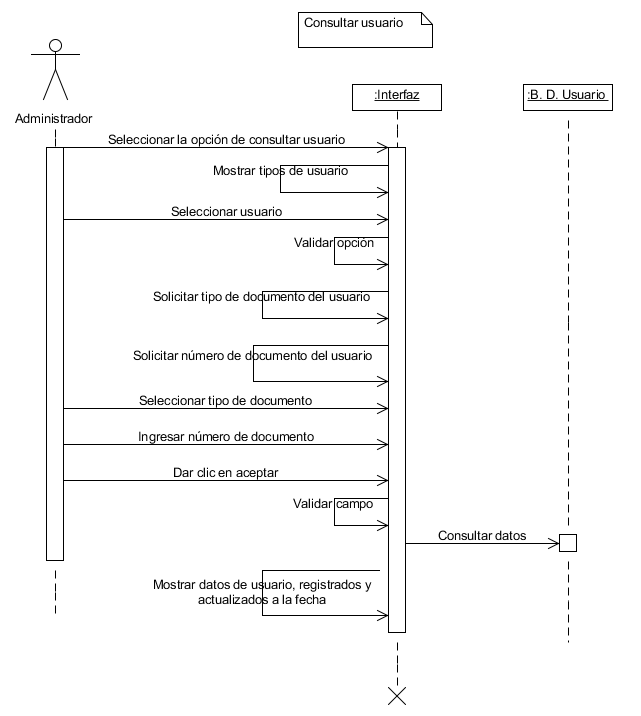


Ilustración Consultar usuario

-Eliminar usuario

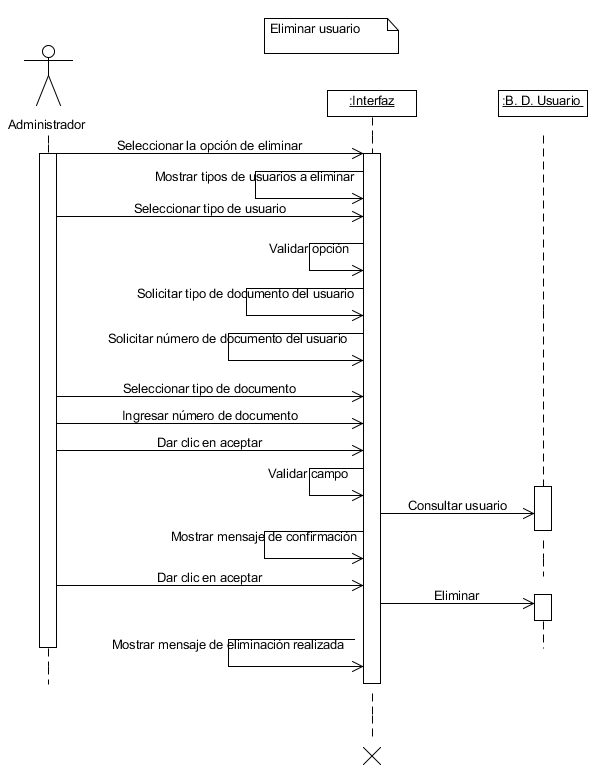


Ilustración Eliminar usuario

### 8.5.3 Diagramas de estado

Administrador

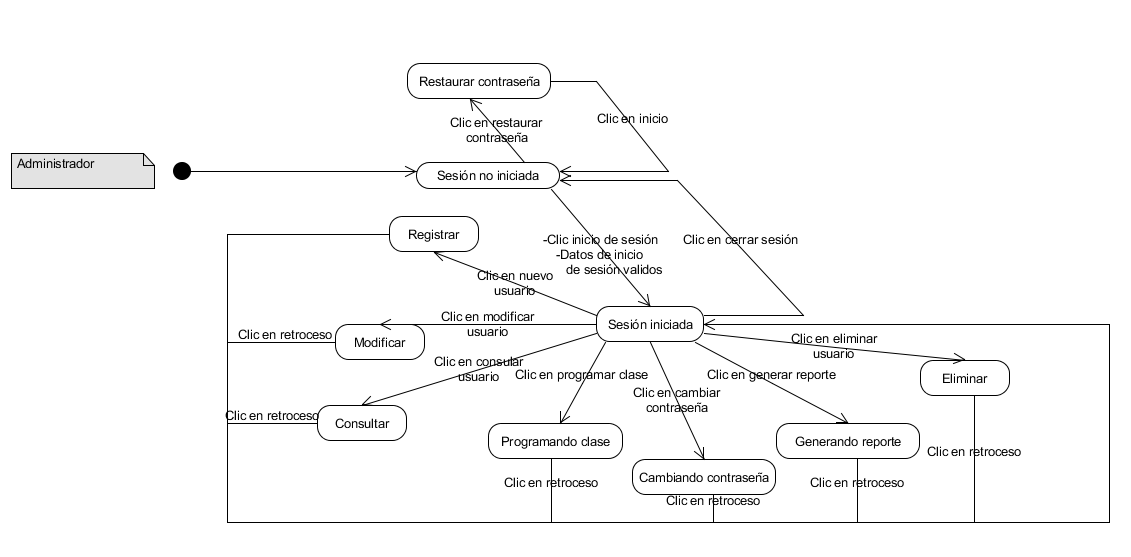


Ilustración : Diagrama de estado Administrador

Instructor

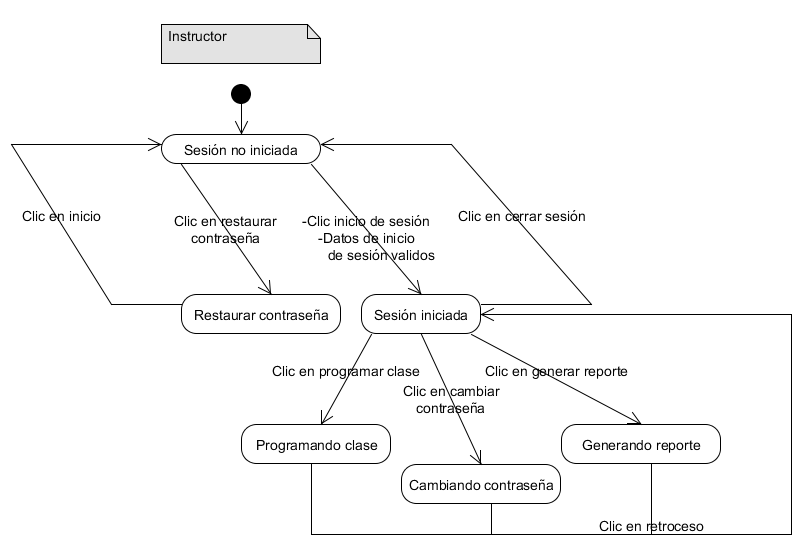


Ilustración : Diagrama de estado Instructor

Aprendiz

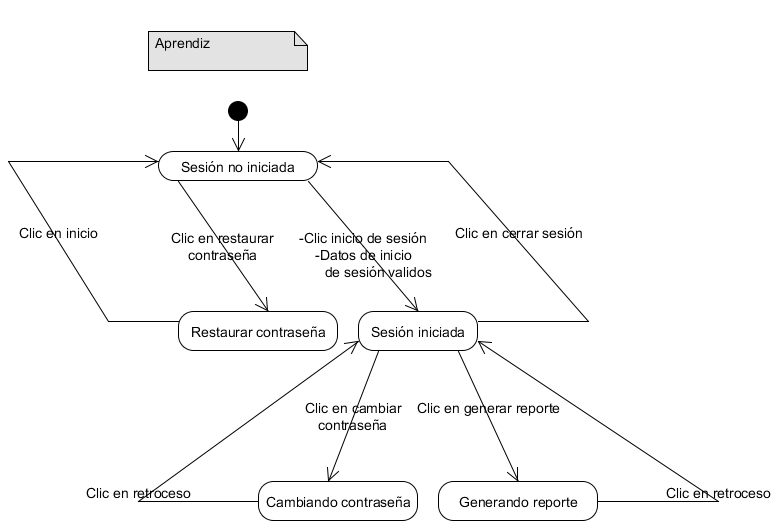


Ilustración : Diagrama de estado Aprendiz

### 

# 9. Recomendaciones

* Para la correcta ejecución del proyecto es necesario que el usuario tenga un navegador Mozilla Firefox, Google Chrome y/o Internet Explorer (versión 8 en adelante).
* Capacidad de memoria RAM de mínimo 2Gb para el fácil acceso al sistema y mejor rendimiento.
* Conexión a internet con un ancho de banda de mínimo 50Mb.

# 10. Conclusiones

1) Para el desarrollo del proyecto, es importante hacer una recolección de datos, porque estos son fundamentales para solucionar las necesidades del cliente y recopilar los requerimientos del cliente.

2) El proyecto no necesita de una infraestructura compleja para su desarrollo.

3) El proyecto está diseñado con un modelo escalable, para que sea compatible con cualquier tipo de sistema.

# 11. Referencias

Monsó, J. (1994). Sistemas de identificación y control automáticos (II)

Barcelona: Editorial Marcombo S.A.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Gamba Roa, C (2010). Control de acceso con verificación de identidad por medio de código de barras. Bogotá, Colombia.

Alonso Bernate, J (2012). Análisis y diseño del software para el control de asistencia de estudiantes. Bogotá, Colombia.

Gutiérrez Ricardo, J (2007). Estudio de factibilidad para el control de acceso biométrico, en una empresa empleando lectores de huella digital. Bogotá, Colombia.

Sanabria Fernández, J (2011). CATEBus: Control de asistencia en transporte escolar. Madrid, España.

Guzmán Martínez, R (2008). Implantación de código de barras en un almacén de electrodomésticos. México.

Rivero Cornelio, Guardia Carlos, Reig José. Bases de datos relacionales: Diseño físico (Orientado a BD2 para z/OS de IBM).

Richard Barker. El modelo entidad-relación. Ediciones Díaz de Santos.

Sánchez Jorge. Diseño conceptual de Base de Datos.

# 12. Webgrafía

Definición UMLet. Recuperado de <http://blog.desdelinux.net/umlet-modelado-uml-en-linux/>

Definición Balsamiq mockups. Recuperado de <http://www.applesfera.com/aplicaciones-os-x-1/balsamiq-mockup-una-muy-buen-herramienta-para-esbozar-tus-futuras-apps>

Definición MySQL Workbench. Recuperado de <http://blog.lawebdelprofe.com/2013/05/10-herramientas-visualesgui-para-mysql.html>

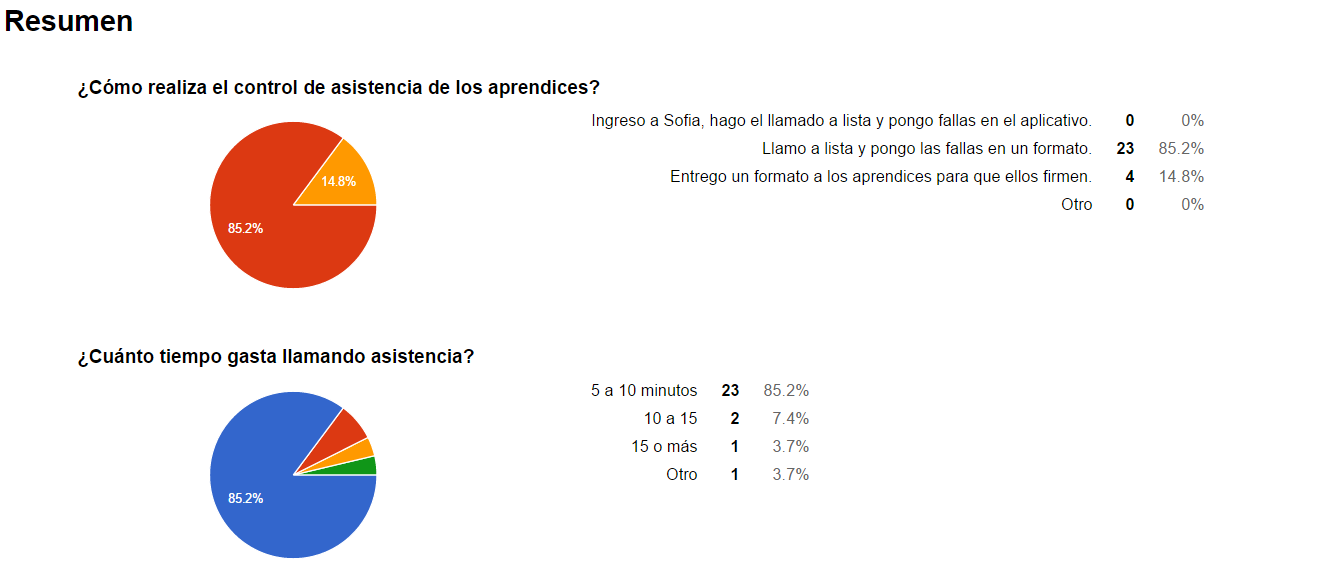
Definición Microsoft SQL Server. Recuperado de <https://es.wikipedia.org/wiki/Microsoft_SQL_Server>

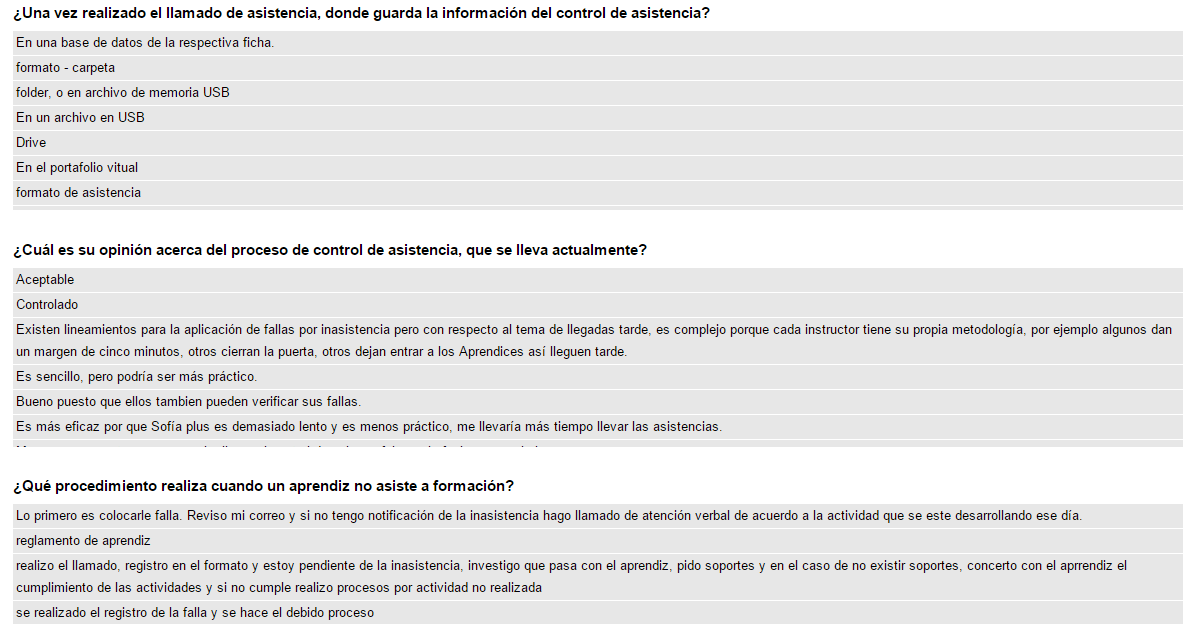
Definición ASP.NET. Recuperado de <https://es.wikipedia.org/wiki/ASP.NET>

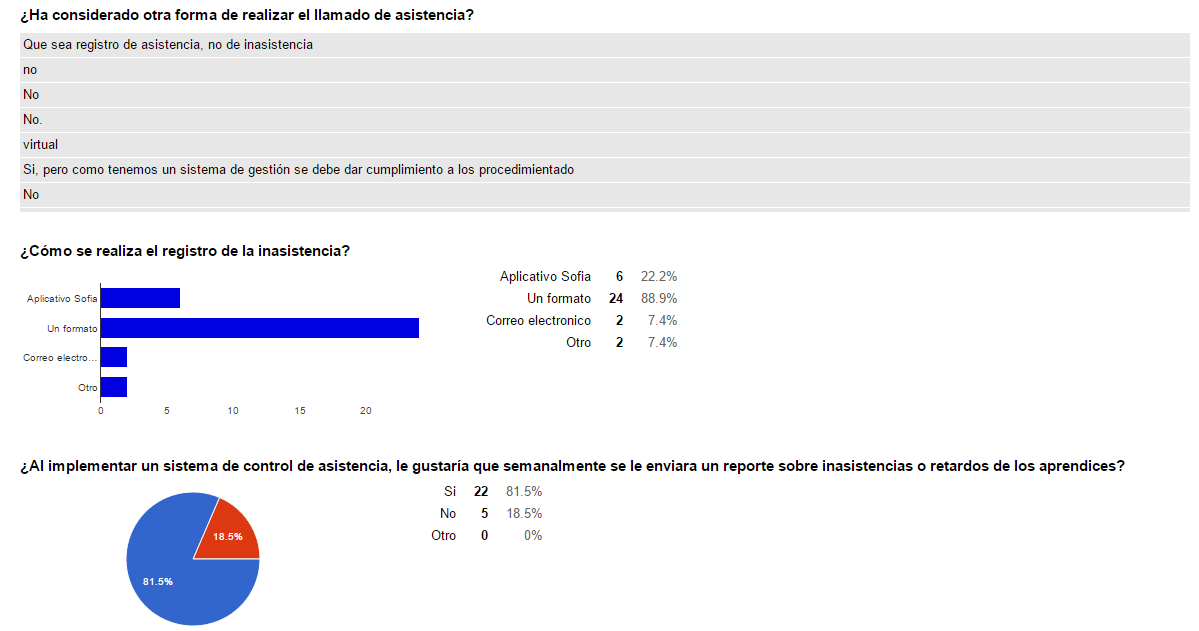
Definición MySQL Workbench. Recuperado de <https://msdn.microsoft.com/es-es/library/dd537667(v=vs.100).aspx>

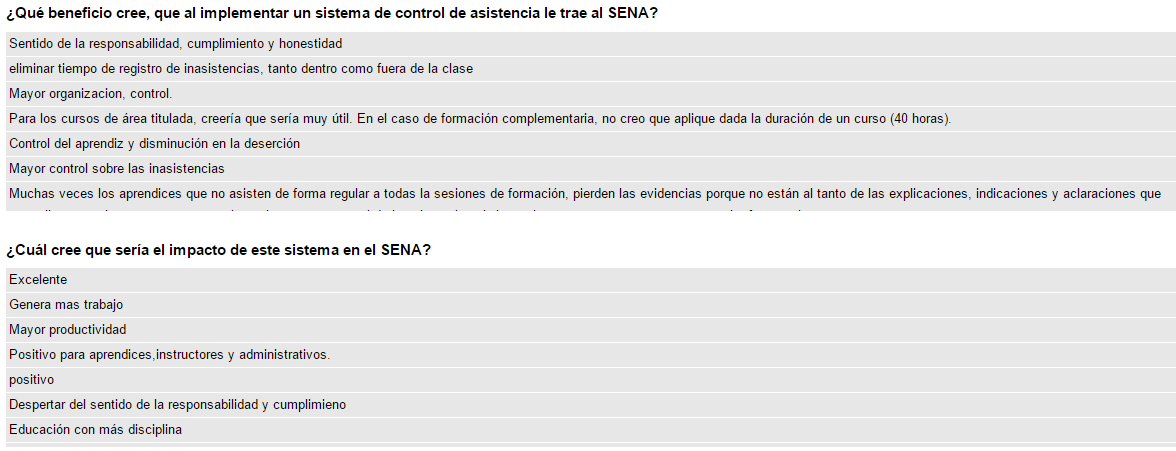
# 13. Anexos

**Anexo 1** Resultados encuesta a instructores SENA.









**Anexo 2** Entrevista Ing. Marina Casallas Silva (líder programas ADSI).

**ENTREVISTA**

Realizada a la Ingeniera Marina Casallas

1. **¿Qué beneficio cree usted que el sistema de control de asistencia le traería al SENA?**

**RTA:** Pues nos ayuda a la deserción, a mirar por que los muchachos dejan de venir al Sena, ya que es uno de nuestros mayores problemas, es uno de los índices que más alto tiene el Sena. El control de asistencia ayudaría mucho.

1. **¿Cuál cree que sería el impacto de este sistema en el SENA?**

**RTA:** Seria de una gran ayuda, porque sería un control de asistencia diferente a los cotidianos.

1. **¿Cree que con la implementación de este sistema se reducirían los costos de papelería para el SENA y contribuirá con el medio ambiente?**

**RTA:** Si, ayuda ya que no se tendría que utilizar tanta papelería.

1. **¿Cree usted que el llamado de asistencia interfiere con el inicio de su clase?**

**RTA:** Si y no porque en mi caso yo cuento a los estudiantes y hago el llamado de asistencia, así hago un control. Si interfiere porque cuando voy a ponerles a los aprendices la falla en Sofía, es porque ya llevo 2 o 3 sesiones y hay interrupción.

1. **¿Cuánto tiempo gasta usted llamando asistencia?**

**RTA:** Lo ideal es que la actividad de llamar asistencia se haga en clase y no que se llegue a la casa a buscar las listas porque se pierde tiempo y estoy ocupando horas que son mías para la formación.

1. **¿Cuál es su opinión acerca del proceso del control de asistencia?**

**RTA: (**sin respuesta)

1. **¿Cómo realiza usted el control de asistencia?**

**RTA:** Desde el aplicativo Sofía, se llama lista y se cuenta a los estudiantes. Paso una hoja de asistencia para que los aprendices la firmen.

1. **¿Una vez finalizado el llamado de asistencia donde guarda la información de la asistencia a clase?**

**RTA:** Lo guardo en mi portafolio.

1. **¿Qué procedimiento hace usted cuando un aprendiz llega después de la hora establecida al ambiente de formación?**

**RTA:** Se evidencia en la hoja de control y se escribe “pendiente”.

1. **¿Una vez realizado el llamado de atención que se hace con este?**

**RTA:** Se da en un escrito, el cual es llevado al coordinador académico.

1. **¿Qué procedimiento hace usted cuando un aprendiz no asiste al ambiente de formación?**

**RTA:** Se evidencia en la hoja de control. Además le envió un correo preguntándole el por qué no ha asistido a clase y tampoco ha enviado las actividades.

1. **¿Cómo se realiza el registro de la inasistencia por medio de un formato, correo?**

**RTA:** Nosotros tenemos nuestros formatos que es donde se debe evidenciare cada registro, además de la planilla escrita y también Sofía.

1. **¿Registrada la inasistencia que sea hace con esta?**

**RTA:** Se evidencia en Sofía.

1. **¿Cuál es su opinión acerca del proceso del control de asistencia?**

RTA: Se utiliza mucho tiempo. Es un desgaste.

1. **¿Cuáles son algunos de los errores más comunes durante proceso?**

RTA: (sin respuesta)

1. **¿Se utiliza algún software donde registra la inasistencia o las llegadas tarde a clase?**

**RTA:** El aplicativo Sofía.

1. **¿Cómo hacen la actualización de la información que contiene el software?**

**RTA:** Lo ideal es que se realice por semana.

1. **¿Implementando este sistema le gustaría que semanalmente se le enviara un reporte sobre inasistencias o retardos de los aprendices?**

**RTA:** Sería muy bueno.

1. **¿Ha considerado otra forma de realizar el llamado de asistencia?**

**RTA:** No, la actual es la que más nos da resultado.

1. **¿Le gustaría que se implementara este sistema?**

**RTA:** Si.

**Anexo 3** Pruebas realizadas a la aplicación

**Anexo 4** Manual de usuario

**Anexo 5** Manual técnico

**\*Nota:** los anexos 3, 4 y 5 se encuentran en la carpeta de la documentación del proyecto.