

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS Y COMPUTACIÓN

“DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA PARA EL CONTROL Y
ADMINISTRACIÓN DE ASISTENCIA DE LOS AYUDANTES EN EL
LABORATORIO DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES
(LTIC) DE LA PUCE.”

LUIS FERNANDO GARCÍA CASTRO

DIRECTOR: ING. DAMIAN NICOLALDE

QUITO, 2017

DEDICATORIA

El presente trabajo está dedicado a todas las personas que me apoyaron y me guiaron tanto en mis estudios como en mi vida personal – profesional.

A mis padres, Margarita y Freddy, quienes supieron guiarme siempre por el buen camino y enseñarme a nunca rendirme ante cualquier situación.

A mis hermanos, con quienes compartí muchos momentos divertidos.

A mi esposa Pamelita, mi compañía invaluable con quien comparto esta hermosa profesión. El pilar en mi formación profesional.

A mi pequeño Martincito, eres mi todo y este proyecto va dedicado especialmente para ti. Todo lo hago y lo haré por ti desde el día de tu nacimiento hasta siempre. Nada me hace más feliz que verte sonreír.

A mis profesores, quienes me instruyeron desde pequeño, todo el conocimiento que poseo se los debo a ustedes.

Y a mis amigos, quienes proporcionaron una gran compañía en toda mi formación.

AGRADECIMIENTO

Me gustaría agradecer a todas las personas que me acompañaron en toda mi formación, desde pequeño. A mi familia y amigos, que siempre están pendientes de mí.

Un especial agradecimiento a mis Profesores y a mi Director de Proyecto de Titulación, supieron darme el conocimiento necesario y la oportunidad para que este proyecto sea desarrollado basado en las mejores prácticas.

A mi esposa Pamelita, mi gran motivación en nuestros proyectos se debe a usted. Y que este trabajo sea de los primeros de muchos que podamos desarrollar.

TABLA DE CONTENIDOS

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA Y METODOLOGÍAS ÁGILES	1
1.1 Fundamentos teóricos	1
1.1.1 Ingeniería de software	1
1.1.2 Especificación de requerimientos norma IEEE-830.....	1
1.1.2.1 Introducción	1
1.1.2.2 Descripción general.....	2
1.1.2.3 Requisitos específicos	3
1.1.3 Lenguaje unificado de modelado.....	4
1.1.3.1 Diagrama de casos de uso	4
1.1.3.2 Diagrama de clases.....	5
1.1.4 Sistema gestor de base de datos: PostgreSQL	6
1.1.4.1 Modelo conceptual de base de datos	6
1.1.4.2 Modelo físico de base de datos	7
1.1.5 Lenguajes de programación.....	7
1.1.5.1 JAVA 8.....	7
1.1.5.2 Framework: Primefaces.....	8
1.1.5.3 Framework: Javaserwer Faces (JSF).....	8
1.1.5.4 Javaserwer Pages (JSP)	8
1.2 Metodología Ágiles.....	9
1.2.1 Principios de los métodos ágiles.....	9
1.2.1.1 Participación del cliente	9
1.2.1.2 Entrega incremental.....	9
1.2.1.3 Personas, no procesos.....	9
1.2.1.4 Adoptar el cambio	10

1.2.1.5	Mantener simplicidad.....	10
1.2.2	Metodología extreme programming	10
CAPÍTULO II. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE		12
2.1	Introducción	12
2.1.1	Propósito.....	13
2.1.2	Ámbito del sistema	13
2.1.3	Definiciones, acrónimos y abreviaturas.....	13
2.1.4	Levantamiento de información.....	13
2.1.5	Visión general del documento	15
2.2	Descripción general	15
2.2.1	Perspectiva del producto.....	15
2.2.2	Funciones del producto.....	15
2.2.3	Características de los usuarios	15
2.2.4	Restricciones.....	16
2.2.5	Suposiciones y dependencias.....	16
2.2.6	Requisitos futuros	16
2.3	Requisitos específicos.....	17
2.3.1	Interfaces externas	17
2.3.1.1	Interfaz con el usuario	17
2.3.1.2	Interfaz con el hardware	17
2.3.1.3	Interfaz de comunicaciones	17
2.3.2	Funciones.....	17
2.3.2.1	Requerimiento funcional agregar tipos de catálogos	17
2.3.2.2	Requerimiento funcional editar tipo de catálogo	17
2.3.2.3	Requerimiento funcional cambiar estado de tipo de catálogo.....	18
2.3.2.4	Requerimiento funcional agregar catálogos	18

2.3.2.5	Requerimiento funcional editar catálogo	18
2.3.2.6	Requerimiento funcional cambiar estado de catálogo.....	18
2.3.2.7	Requerimiento funcional editar parámetro.....	19
2.3.2.8	Requerimiento funcional agregar personas	19
2.3.2.9	Requerimiento funcional editar persona	19
2.3.2.10	Requerimiento funcional cambiar estado de persona.....	19
2.3.2.11	Requerimiento funcional generar usuario	20
2.3.2.12	Requerimiento funcional cambiar estado de usuario	20
2.3.2.13	Requerimiento funcional restablecer contraseña.....	20
2.3.2.14	Requerimiento funcional cambiar estado de parámetro.....	20
2.3.2.15	Requerimiento funcional asociar personas al horario	21
2.3.2.16	Requerimiento funcional reasignar horas estudiantes.....	21
2.3.2.17	Requerimiento funcional registrar permisos	21
2.3.2.18	Requerimiento funcional descargar reporte de faltas.....	21
2.3.2.19	Requerimiento funcional descargar reporte de atrasos	22
2.3.2.20	Requerimiento funcional descargar reporte de total de faltas por atrasos	22
2.3.2.21	Requerimiento funcional descargar reporte de total de faltas justificadas	22
2.3.2.22	Requerimiento funcional descargar certificado de ayudantía	22
2.3.2.23	Requerimiento funcional cerrar período académico	23
2.3.2.24	Requerimiento funcional iniciar período académico	23
2.3.2.25	Requerimiento funcional registrar asistencia	23
2.3.3	Diagrama de casos de uso.....	24
CAPÍTULO III: DESARROLLO DEL SISTEMA		28
3.1	Diseño arquitectónico del sistema	28

3.1.1	Componentes	28
3.1.2	Diseño de base de datos.....	28
3.1.2.1	Diagrama conceptual de base de datos.....	29
3.1.2.2	Diagrama físico de base de datos	29
3.2	Diseño de la arquitectura de software.....	29
3.2.1	Capa de datos.....	29
3.2.1.1	Diagrama de clases en la capa de datos.....	29
3.2.1.2	Diseño de los métodos en la capa de datos	29
3.2.2	Capa de negocio.....	30
3.2.2.1	Diseño de los métodos en la capa de negocio	30
3.2.3	Capa de presentación.....	30
3.2.2.2	Diseño de los métodos en la capa de presentación.....	30
CAPÍTULO IV: IMPLEMENTACIÓN		31
4.1	Implementación en la nube	31
4.1.1	Google Cloud.....	31
4.2	Configuración servidor	31
4.3	Integración continua.....	32
4.4	Despliegue del sistema de control de asistencia – LTIC	33
4.5	Comprobación instalación del sistema.....	34
CAPÍTULO V: SEGURIDADES Y PRUEBAS		35
5.1	Reporte de ACUNETIX.....	35
5.1.1	Primefaces	35
5.1.2	Hibernate	36
5.2	Pruebas finales en el despliegue	38
CONCLUSIONES Y RECOMENDACIONES		40
Conclusiones.....		40

Recomendaciones	41
BIBLIOGRAFÍA	42
ANEXOS	43
Anexo 1: Diagrama conceptual de base de datos.	44
Anexo 2: Diagrama físico de base de datos.....	45
Anexo 3: Diagrama de clases.	46
Anexo 4: Diagrama de clases, diseño de los métodos de datos.....	47
Anexo 5: Diagrama de clases, diseño de los métodos de negocio.	48
Anexo 6: Diagrama de clases, diseño de los métodos de presentación.	49
Anexo 7: Glosario de términos.....	51
Anexo 8: Manual de usuario.....	52

TABLA DE ILUSTRACIONES

Ilustración 1. Ejemplo de diagrama de caso de uso. 16 de mayo de 2016.	4
Ilustración 2. Ejemplo de diagrama de clases. 18 de mayo de 2016.	6
Ilustración 3. Ejemplo de modelo conceptual de base de datos. 19 de mayo de 2016.	7
Ilustración 4. Ejemplo de modelo físico de base de datos. 19 de mayo de 2016.	7
Ilustración 5. Proceso del desarrollo ágil. 24 de mayo de 2016.	10
Ilustración 6. Ciclo de liberación en programación extrema. 25 de mayo de 2016.	10
Ilustración 7. Módulo de administración: tipos de catálogos. 13 de junio de 2016.	24
Ilustración 8. Módulo de administración: catálogos. 13 de junio de 2016.	24
Ilustración 9. Módulo de administración: parámetros. 13 de junio de 2016.	25
Ilustración 10. Módulo de administración: personas. 13 de junio de 2016.	25
Ilustración 11. Módulo de administración: usuarios. 13 de junio de 2016.	26
Ilustración 12. Módulo de asistencia. 13 de junio de 2016.	26
Ilustración 13. Módulo de reportes. 13 de junio de 2016.	27
Ilustración 14. Diseño arquitectónico del sistema. 18 de julio de 2016.	28
Ilustración 15. Diseño de la arquitectura de software. 20 de julio de 2016.	29
Ilustración 16. Dashboard de instancias de máquinas virtuales. 18 de febrero de 2017.	31
Ilustración 17. Versiones instaladas en la nube de Google. 18 de febrero de 2017.	32
Ilustración 18. Visualización del historial de despliegue por proyecto. 18 de febrero 2017.	33
Ilustración 19. Visualización del despliegue en el sistema de integración continua Jenkins. 18 de febrero de 2017.	33
Ilustración 20. Pantalla de ingreso al sistema. 18 de febrero de 2017.	34
Ilustración 21. Pantalla inicial del sistema. 18 de febrero de 2017.	34
Ilustración 22. Inicio del escaneo de vulnerabilidades. 18 de febrero de 2017.	35

Ilustración 23. Resultado final del escaneo en su primera ejecución. 18 de febrero de 2017.	36
Ilustración 24. Detalle del resultado de escaneo. 18 de febrero de 2017.	36
Ilustración 25. Visualización de los tres (3) escaneos ejecutados sobre el sistema. 18 de febrero de 2017.....	37
Ilustración 26. Detalle del último resultado del escaneo. 18 de febrero de 2017.....	37
Ilustración 27. Ingreso de credenciales. 18 de febrero de 2017.	38
Ilustración 28. Página de horarios para el sistema. 18 de febrero de 2017.	38
Ilustración 29. Ingreso de información de la persona. 18 de febrero de 2017.....	39
Ilustración 30. Ingreso incorrecto de la identificación PUCE. 18 de febrero de 2017.....	39
Ilustración 31. Ingreso correcto de la identificación PUCE. 18 de febrero de 2017.	39
Ilustración 32. Modelo conceptual de la base de datos. 22 de julio de 2016.	44
Ilustración 33. Modelo físico de la base de datos. 22 de julio de 2016.....	45
Ilustración 34. Diagrama de clases: entidades de la capa de datos. 23 de julio de 2016....	46
Ilustración 35. Diagrama de clases: métodos de la capa de datos. 24 de julio de 2016.	47
Ilustración 36. Diagrama de clases: métodos de la capa de negocio. 24 de julio de 2016 ..	48
Ilustración 37. Diagrama de clases: métodos de la capa de presentación administrativa. 25 de julio de 2016.	49
Ilustración 38. Diagrama de clases: métodos de la capa de presentación principal y de reportería. 26 de julio de 2016.....	50

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA Y METODOLOGÍAS ÁGILES

“Computer science is no more about computers than astronomy is about telescopes.” — Edsger W. Dijkstra.

1.1 Fundamentos teóricos

1.1.1 *Ingeniería de software*

Esta disciplina de ingeniería está muy inmersa a un aspecto global de producción del Software, desde el inicio correspondiente a las especificaciones del sistema hasta su etapa de mantenimiento una vez concluida la producción de la misma. Dicha disciplina consiste en desarrollar un sistema de software. Acorde a los principios de la Ingeniería de Software, los costos de software corresponden a un 60% para el desarrollo y un 40% para las pruebas, como nos indica Ian Sommerville en su libro “Ingeniería de Software”.

Aun cuando los proyectos de desarrollo de software son gestionados y desarrollados de forma muy profesional, cabe recalcar que existen muchas técnicas, las cuales se aplican dependiendo el tamaño del proyecto y su complejidad. Por ello, todos los proyectos de desarrollo software son totalmente distintos entre ellos.

1.1.2 *Especificación de requerimientos norma IEEE-830*

Las especificaciones de requerimientos de software (ERS) es un documento en el cual se detalla de forma clara los requerimientos, los factores que afectan en el desarrollo de un software que se va a desarrollar y las bases de acuerdo entre clientes y proveedores. La ERS debe cumplir con ciertos criterios, es decir, debe ser: correcta, ordenada, verificable, completa y sobre todo comprobable.

1.1.2.1 *Introducción*

Engloba ciertos elementos que nos indican el objetivo general del proyecto de desarrollo. En esta sección se puede indicar aspectos preliminares del proyecto.

- **Propósito.**

Permitió indicar el objetivo del documento de ERS, además a quién va dirigido.

- **Ámbito del sistema.**

En esta sección se pudo indicar ciertas características específicas del sistema, así como: el futuro nombre que tendrá; indicar que es lo que el sistema realizará y que no; beneficios, objetivos y metas a las cuales se desea alcanzar con el proyecto; en el caso de ser necesario, también se referencia aquellos documentos extras del sistema.

- **Definiciones, acrónimos y abreviaciones.**

En esta sección se definió todas las siglas, términos y abreviaciones que se utilizaron en el documento de ERS.

- **Levantamiento de información.**

Al no tener una sección de referencias para el documento de ERS, se aumentó esta sección indicando la información proporcionada por el cliente como levantamiento de información para el desarrollo del proyecto.

- **Visión general del documento.**

En esta sección se describió de forma sencilla los contenidos y la organización del resto del documento de ERS.

1.1.2.2 Descripción general

Engloba todos los factores que afectan directamente al sistema y a sus requerimientos. En esta sección no se describen los requisitos del sistema sino su contexto. Dentro de esta sección se especifica la siguiente información del futuro sistema.

- **Perspectiva del producto.**

Se indicó lo que se desea lograr con el sistema, y en el caso de aplicar, indicar si el sistema tendrá relación con otro u otros productos, es decir, si tendrá integración con otros sistemas.

- **Funciones del producto.**

Se especificó las funciones que el sistema realiza de acuerdo a lo establecido con el cliente, tomando en cuenta la organización del mismo para que pueda ser entendido.

- **Características de los usuarios.**

Se indicó a qué tipo de usuarios está orientado el sistema desarrollado, para ello es recomendable especificar la experiencia técnica de cada uno de ellos.

- **Restricciones.**

Como su nombre lo indica, aquí se describió todas las limitaciones o restricciones que tendrá el futuro sistema.

- **Suposiciones y dependencias.**

En esta sección indicó los parámetros que necesita el sistema para su correcto funcionamiento. Quiere decir que, si uno de ellos llegase a cambiar puede afectar directamente en el sistema.

- **Requisitos futuros.**

Aquí se pudo indicar posibles mejoras para las cuales se podría realizar un análisis completo para su futura implementación.

1.1.2.3 Requisitos específicos

En esta sección se especificó a gran detalle cada uno de los requisitos establecidos con el cliente a fin de satisfacer las necesidades del mismo y a su vez permita realizar las pruebas necesarias para la comprobación de estas. Los requisitos además deben cumplir con los siguientes principios: corrección, no ambigüedad, consistencia, clasificación, verificabilidad, modificables y de fácil mantenimiento.

- **Interfaces externas.**

Se indicó aquellos requisitos que afecten a los siguientes:

- Interfaz con el usuario.
- Interfaz con el hardware.
- Interfaz de comunicaciones.

- **Funciones.**

Esta subsección es posiblemente la más larga del documento, ya que se debe especificar todas las funciones las cuales deberá llevar a cabo el sistema. De ser necesario se podría utilizar notaciones gráficas o tablas.

1.1.3 Lenguaje unificado de modelado

UML es el lenguaje de modelado de sistemas más conocido, este permite detallar el plan del sistema por medio de métodos o procesos, tomando en cuenta que se lo realiza a través de gráficos, al ser uno de los lenguajes visuales más usados se clasifican en:

- **Estructurales**

- Diagrama de clases.
- Diagrama de componentes.
- Diagrama de despliegue.
- Diagrama de objetos.
- Diagrama de paquetes.
- Diagrama de perfiles.
- Diagrama de estructura compuesta.

- **Comportamiento**

- Diagrama de Actividades.
- Diagrama de Casos de Uso.
- Diagrama de máquina de estados.

- **Interacción**

- Diagrama global de interacciones.
- Diagrama de comunicación.
- Diagrama de secuencia.
- Diagrama de tiempo.

1.1.3.1 Diagrama de casos de uso

Tomando en cuenta que los casos de uso son más detallados y son diferentes a los diagramas de casos de uso se dice que estos muestran la forma y el orden en como el cliente maneja el sistema. Los principales elementos son: Actor, Casos de Uso y Relaciones, así como se muestra en la Ilustración 1.

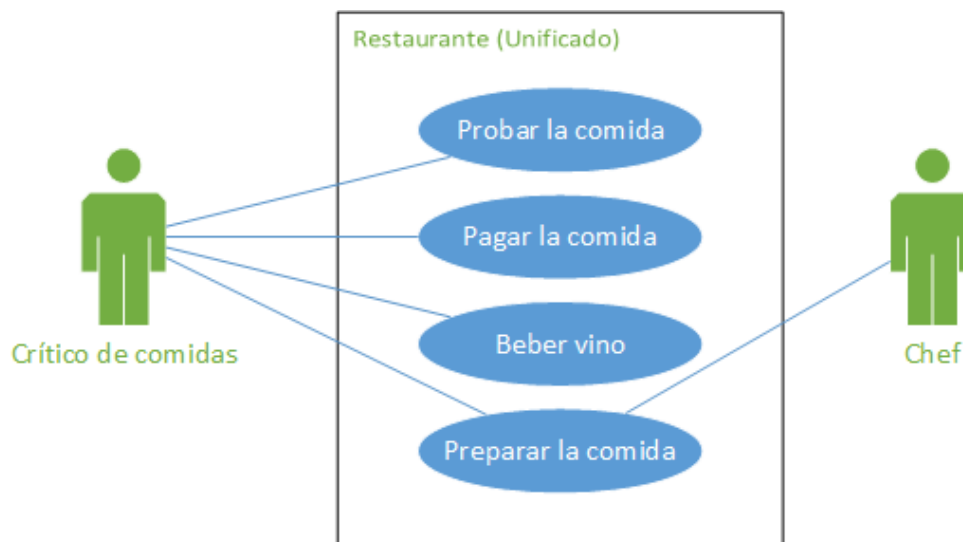


Ilustración 1. Ejemplo de diagrama de caso de uso. 16 de mayo de 2016.

- **Actor.**

Es la persona que interactúa directamente con el sistema, la persona que maneje y tenga conocimiento y buen manejo del mismo.

- **Caso de Uso.**

Es una operación que se realiza tras ser ejecutada una acción o petición ya sea del actor o respuesta a una invocación de otro caso de uso.

- **Relaciones.**

- Asociación: se denota con una flecha simple y su función principal es la de indicar la operatividad entre un actor o caso de uso y otro, puede ser semejante a una relación padre a hijo.
- Dependencia: se denota con una flecha entrecortada, indica la dependencia de una clase a otra, es decir, la segunda se crea a partir de la primera.
- Generalización: se denota con una flecha más grande, indica una doble función ya que puede ser utilizada como Herencia <<*extends*>> o Uso <<*uses*>>.

1.1.3.2 Diagrama de clases

Este diagrama describe la estructura del sistema a través de las clases definidas por el desarrollador y en las mismas los atributos, métodos y relaciones (generalización, asociación, agregación y composición).

- **Clase.**

La clase representa una entidad en la cual se definen atributos y métodos correspondientes.

- **Atributos y Métodos.**

Estos se muestran y definen en la clase con el tipo al que representan. Las características principales de cada uno de estos es que Los atributos pueden o no tener un valor definido o inicial, y en el caso de la definición de los métodos estos mostrarán el valor de retorno.

- **Relaciones.**

Representa la relación entre clases, que pueden ser de generalización, asociación, agregación y composición.

- Generalización: Indica herencia de una clase a otra.

- Asociación: Se representa con una línea simple, pueden ser unidireccionales (solo una clase conoce de la otra) o bidireccionales (ambas clases se conocen).

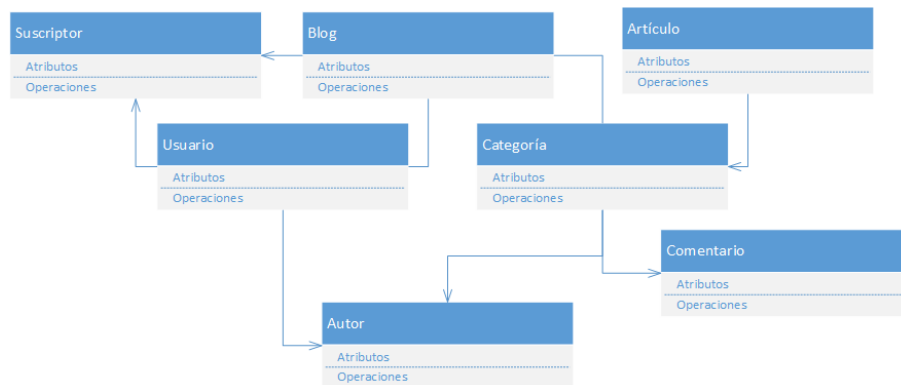


Ilustración 2. Ejemplo de diagrama de clases. 18 de mayo de 2016.

1.1.4 Sistema gestor de base de datos: PostgreSQL

Es un potente sistema gestor de base de datos orientada a objetos, dirigido por un grupo de desarrolladores que laboran de manera libre, actualmente es el sistema gestor de base de datos más considerado en todo proyecto, una de las características de PostgreSQL es que permite el control de concurrencias y de igual manera facilitar a usuarios tener acceso a la base de datos en modo de lectura, mientras esta se encuentre en espera, permitiendo así que los administradores de la misma puedan hacer algún tipo de mantenimiento mientras este se encuentre en ejecución.

Entre las opciones que se pueden realizar con la base de datos son: triggers, autorizaciones, afirmaciones, transacciones, respaldos y restricciones en el dominio.

1.1.4.1 Modelo conceptual de base de datos

El diseño conceptual es considerado como la primera fase en el diseño de una base de datos en la cual se realiza un análisis previo de los requisitos del sistema y se genera un esquema conceptual. Para la representación del esquema conceptual se logra utilizando el Modelo Entidad/Relación Extendido.

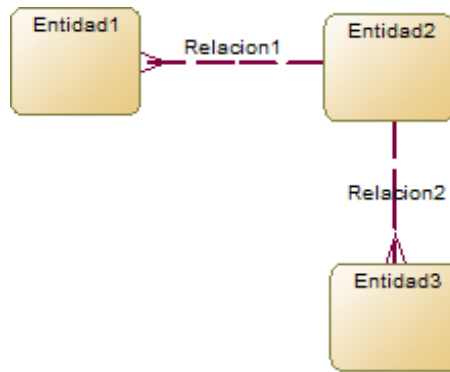


Ilustración 3. Ejemplo de modelo conceptual de base de datos. 19 de mayo de 2016.

1.1.4.2 Modelo físico de base de datos

Es un modelo específico de la base de datos desarrollada a partir del modelo conceptual y lógico, aunque no necesariamente en ciertos casos, tiene como objetivo satisfacer los requisitos del sistema y cumplir con las siguientes acciones: proporcionar seguridad en los datos, optimiza el consumo de los recursos y minimizar el espacio de almacenamiento.

Al crear el modelo físico de nuestra base de datos debemos tomar en cuenta las siguientes consideraciones:

- Redundancia de datos.
- Clustering.
- Uso de punteros.
- Locking.
- Hashing.

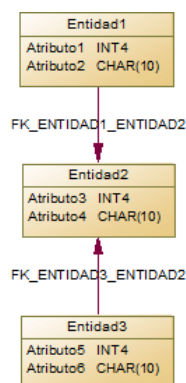


Ilustración 4. Ejemplo de modelo físico de base de datos. 19 de mayo de 2016.

1.1.5 Lenguajes de programación

1.1.5.1 JAVA 8

Es un lenguaje de programación el cual se encuentra actualizado a la versión 8, con nuevas mejoras y correcciones de bugs para mejorar el desarrollo y ejecución de programas en el mismo. Entre las características mejoradas para esta versión es la seguridad, la API de

fecha y la hora, nueva implementación del motor de Javascript Nashhorn integrada en JDK, a través de las APIS ya existentes, disponibilidad de un amplio conjunto de bibliotecas, robusto, portable y dinámico.

Es uno de los lenguajes más sencillos para manejar eliminando así la complejidad del lenguaje C e incorporando el uso la programación orientada a objetos. El código compilado en Java se ejecuta en una máquina virtual por lo que puede ser efectuado en cualquiera de estos entornos: Unix, Windows o Mac OS.

1.1.5.2 Framework: Primefaces

Primefaces posee un conjunto de componentes que permiten la creación de aplicativos WEB como lo es RichFaces para Android, Palm y iPhone, la creación de dichas aplicaciones es amigables y fáciles de usar para el usuario, ya que cuenta con diseños creativos y permite escoger entre los más de 25 temas prediseñados.

Entre las características de Primefaces podemos nombrar las siguientes:

- Posee más de 90 componentes OpenSource.
- Incorpora el concepto de “skins” por lo que nos favorece con los temas prediseñados.
- Utiliza JQuery y JQuery UI para los efectos visuales.
- Soporte de AJAX.

1.1.5.3 Framework: Javaser Faces (JSF)

La tecnología Javaser Faces basada en web, establece un estándar para crear interfaces de usuario. Las APIs de JSF están diseñadas para ser aprovechadas por herramientas que hacen más fácil su desarrollo, además usa Javaser Pages (JSP) que permite hacer el despliegue de las páginas. Las características de desarrollo de esta tecnología es permitir la navegación entre páginas, se ejecuta sobre la tecnología de Servlets, agrupación de componentes gráficos, basadas en el patrón de diseño MVC, maneja un archivo fácil de utilizar en XML para su configuración.

1.1.5.4 Javaser Pages (JSP)

Esta tecnología está basada en crear páginas WEB utilizando programación en lenguaje JAVA, y su principal objetivo es el de simplificar el desarrollo código creando una gran cantidad de contenido dinámico HTML con elementos especiales.¹

¹ (Burns, Schalk, & Griffin, 2010, pág. 5)

- **Arquitectura de aplicaciones JSF.**

La aplicación WEB debe tener un archivo de configuración llamado “*web.xml*”. Configuraciones adicionales u opcionales las cuales se encontrarán en el archivo “*faces-config.xml*”. Adicional se debe tomar en cuenta que, si la aplicación posee o hace referencias a librerías externas para su funcionamiento, se las deben incluir en el directorio “*WEB-INF/lib*”. Una vez que la aplicación WEB se encuentra configurada correctamente, se puede empezar a crear las vistas con páginas XHTML las cuales adapta HTML a XML.

1.2 Metodología Ágiles

Las metodologías ágiles de desarrollo software se enfocan en que el equipo de desarrollo se enfoque en el software en lugar del diseño y la documentación. Éstas metodologías poseen un enfoque incremental además de poseer un objetivo, la entrega pronta del software operativo o funcional al cliente, los cuales proporcionarán nuevos requerimientos o ajustes. Estos nuevos requerimientos serán incluidos en las próximas iteraciones.

1.2.1 Principios de los métodos ágiles

Los métodos ágiles tienen 5 principios que deben ser respetados:

1.2.1.1 Participación del cliente

Los clientes son la parte más importante, debido a que ellos nos indican cuáles son sus requerimientos, qué es lo que ellos desean. Los clientes deben intervenir durante todo el proceso de desarrollo y su función principal es la de ofrecer y priorizar nuevos requerimientos.

1.2.1.2 Entrega incremental

Los productos de software se desarrollan por incrementos, y el cliente es la persona aquella va a especificar los requerimientos e indicará aquellos que serán incluidos en cada iteración.

1.2.1.3 Personas, no procesos

El equipo de desarrollo es una parte fundamental para un proyecto de software y se debe permitir que estos miembros proporcionen y desarrollen sus propias formas de trabajar sin procesos establecidos previamente.

1.2.1.4 Adoptar el cambio

Diseñar el sistema para adaptar cambios futuros, pues los requerimientos del sistema pueden cambiar en el momento menos esperado.

1.2.1.5 Mantener simplicidad

Se debe mantener un trabajo activo para lograr eliminar o evitar el crear complejidad en el sistema, es decir, enfocarse en la simplicidad del mismo, tanto en el software como en los procesos de desarrollo.

Desarrollo ágil

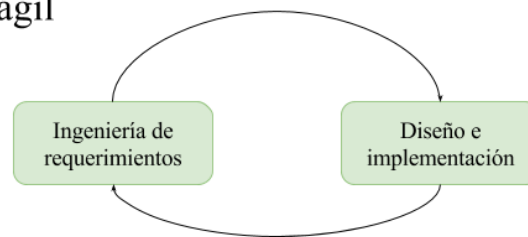


Ilustración 5. Proceso del desarrollo ágil. 24 de mayo de 2016.

1.2.2 Metodología extreme programming

Esta metodología también conocida como XP, es posiblemente la mayor conocido y usado en el desarrollo de sistemas software. Los requerimientos son tomados en historias de usuario, las cuales son consideradas directamente como tareas a desarrollar. Consiste en mantener un equipo de dos programadores y se suelen realizar pruebas previo a empezar a codificar.

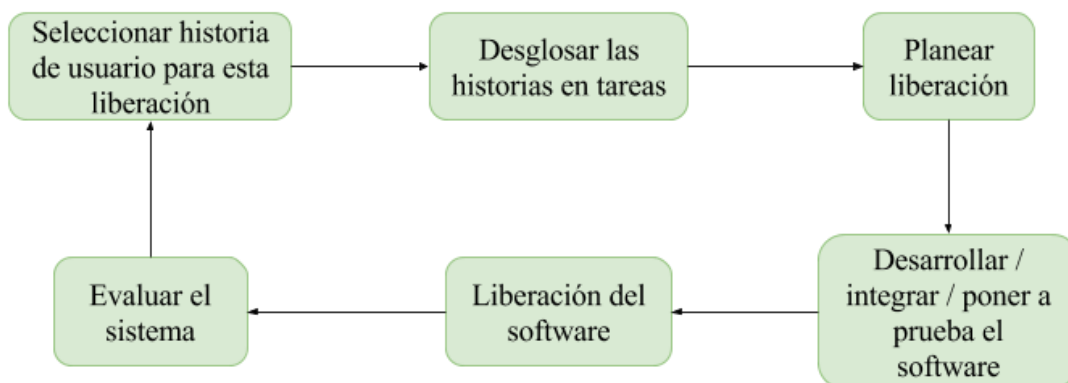


Ilustración 6. Ciclo de liberación en programación extrema. 25 de mayo de 2016.

Todas las pruebas creadas deben ejecutarse sin ningún problema una vez que el nuevo código forme parte del sistema (entregables). En la Ilustración 6 se puede visualizar el proceso que se realiza en cada iteración en un proyecto aplicando el método XP. El desarrollo incremental genera pequeñas y frecuentes liberaciones, de tal manera que se logra tener una parte funcional del sistema con las cuales el cliente puede ejecutar un plan de pruebas, y de ser el caso indicar los nuevos requerimientos.

CAPÍTULO II. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

2.1 Introducción

La unidad de servicio LTIC de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador; no dispone de un sistema que facilite el control y la administración de las asistencias de los alumnos ayudantes y becarios.

Según el reglamento de las normas internas del laboratorio de computación de la facultad de ingeniería de la Pontificia Universidad Católica del Ecuador, art. 1.- de los fines del laboratorio, literal c (22 de abril de 2004). Procurar la capacitación técnica en computación de docentes, administrativos, becarios, ayudantes y estudiantes de la facultad de ingeniería.

La unidad de servicio LTIC, posee un personal interno regido en una estructura de: director del laboratorio, asistente, becarios y ayudantes. Los becarios del laboratorio serán estudiantes designados conforme a la normativa permitida de la dirección general de estudiantes (DGE). Y los ayudantes serán designados por el consejo de escuela de sistemas bajo el régimen de acción social universitaria (servicio a la comunidad).

Debido a la necesidad de los estudiantes de la facultad de ingeniería, escuela de

Sistemas de la Pontificia Universidad Católica del Ecuador por completar los requisitos de graduación, correspondientes a realizar un porcentaje de horas de acción social, estas pueden ser desarrolladas ayudando a los docentes del instituto educativo. Además, cumplen dichas horas como estudiantes ayudantes de la unidad de servicio LTIC de la facultad de ingeniería.

Dentro de este marco, los señores y señoritas estudiantes de la facultad de ingeniería, acuden como ayudantes y becarios a la unidad LTIC para cumplir con sus horas de acción social en el LTIC. Actualmente el control de esta asistencia se realiza por medio de hojas impresas, las cuales los estudiantes proceden a firmarlas; este documento es único y respalda la presencia del ayudante con el puesto asignado, de igual manera para los becarios. No se mantiene un control previo y organizado, es decir; el docente encargado de esta área no dispone de una herramienta automatizada que le permita llevar un control de las asistencias de los estudiantes, aun cuando un número significativo de estudiantes acuden a esta área a realizar sus horas de acción social.

Por los motivos nombrados previamente, este sistema se implementará con el fin de mejorar la administración de la información con respecto a estudiantes ayudantes y becarios de los laboratorios de tecnologías de información y comunicaciones (LTIC) que cumplan sus determinadas horas en este lugar, dando paso al manejo adecuado y ordenado de dicha información mejorando y solucionando la problemática que este proceso conlleva.

Por medio del levantamiento de requerimientos según la norma IEEE 830, nos permitirá analizar todas las necesidades que el mismo va a cubrir para el uso del usuario final (encargado del área LTIC). En esta etapa de análisis se permitirá especificar los procesos que realizará el sistema definiendo diagrama de casos de uso, diagrama de secuencia. Mientras que en la etapa de diseño se definirá la estructura como tal del sistema, definiendo diagrama de clases, modelo conceptual de base de datos, modelo físico de base de datos; al manejarse una metodología de desarrollo ágil, no requiere llevar una documentación extensiva del sistema.

2.1.1 Propósito

El documento de ERS permitirá al docente encargado del LTIC conocer de forma macro las funcionalidades del sistema a desarrollarse, las cuales corresponden a los requerimientos completos del mismo.

2.1.2 Ámbito del sistema

Se desarrollará un sistema de gestión para el control y administración de asistencia de los alumnos ayudantes del LTIC, mismo que al ser puesto en ejecución permitirá al director del laboratorio administrar la asistencia de los alumnos ayudantes. Además de generar reportes que permitan verificar el resumen de las asistencias e imprimir el certificado correspondiente para el alumno que cumplió con las tareas.

El sistema tendrá como nombre Sistema de Control de Asistencia – LTIC.

2.1.3 Definiciones, acrónimos y abreviaturas

Ver Anexo 7: Diagrama de clases.

2.1.4 Levantamiento de información

- El sistema para el control del personal del LTIC, tiene como fin automatizar el control de ingresos, salidas, permisos, atrasos de todo el personal y reportería. El personal debe

registrar los ingresos y salidas a través del portal del sistema, el sistema almacena la información en la base de datos tomando como referencia la hora del servidor.

- Para los ingresos se permite registrar la entrada hasta 10 minutos tarde, entre 11 y 20 minutos tarde registra atraso y después de los 21 minutos el sistema ya no permite registrar.
- Para las salidas, el sistema permite registrar 5 minutos antes de la hora.
- Los ayudantes que se encuentran cumpliendo una jornada de 7:00 a 9:00 y de 18:00 a 20:00 se le multiplicará por 2 la cantidad de horas cumplidas.
- Los becarios deben cumplir 20 horas a la semana, siempre cumplen 340 horas para la impresión del certificado.
- Los módulos que tendrá el sistema son los siguientes:
 - i. Módulo de administración.
 - a. Creación de usuarios.
 - b. Administración de tablas base (catálogos).
 - c. Configuración de parámetros especiales (políticas de aceptación para el registro de entrada y atrasos).
 - ii. Módulo para la gestión del personal.
 - a. Creación de personal (becarios, ayudantes o voluntarios).
 - iii. Módulo de control de ingresos y egresos.
 - a. Registro de ingreso.
 - b. Registro de salida.
 - iv. Módulo de permisos.
 - a. Registro de permisos.
 - v. Módulo de reportes.
 - a. Cálculo de faltas.
 - b. Cálculo de los atrasos.
 - c. Cálculo de las faltas en base a la sumatoria de atrasos.
 - d. Cálculo de las faltas justificadas con permisos.
 - e. Cálculo de las horas de ayudantía que serán impresas en el certificado.
 - f. Impresión del certificado de ayudanta en el LTIC.

2.1.5 *Visión general del documento*

El presente documento contiene una descripción general y un detalle de las características del Sistema Control de Asistencia – LTIC, conteniendo las siguientes secciones:

- Perspectiva del producto.
- Funciones del producto.
- Características de los usuarios.
- Restricciones.
- Suposiciones y dependencias.
- Requisitos futuros.
- Interfaces externas.
- Funciones.
- Requisitos de rendimiento.
- Restricciones de diseño.
- Atributos del sistema.
- Otros requisitos.

2.2 **Descripción general**

2.2.1 *Perspectiva del producto*

El Sistema de Control de Asistencia – LTIC permitirá al docente encargado poder automatizar el proceso de registro de asistencia de los alumnos becarios, ayudantes y voluntarios, y de esta manera generar los reportes o certificados necesarios de forma rápida y sencilla. El sistema será totalmente independiente de cualquier producto el LTIC.

2.2.2 *Funciones del producto*

El sistema para el control de asistencia debe ser capaz de realizar las siguientes funciones:

- Crear, editar y cambiar el estado de tipos de catálogos.
- Crear, editar y cambiar el estado de catálogos.
- Editar parámetros.
- Crear, editar y cambiar el estado de personas.
- Crear, editar y cambiar el estado de usuarios.
- Crear, editar y cambiar el estado de horarios de asistencia.
- Registro de permisos.
- Visualización y descarga de reportes.

2.2.3 *Características de los usuarios*

El sistema mantiene el uso de cuatro tipos de usuarios o perfiles los cuales corresponden a: docente (administrador), becario, ayudante, voluntario.

El docente es la persona encargada del LTIC, en el sistema tendrá todos los permisos como tipo administrador.

El becario es la persona que brindara un soporte al docente encargado, en labores de registrar las personas ayudantes y voluntarios, además de registrar los horarios de asistencia.

La interacción del ayudante y voluntario con el sistema será de registro de asistencia.

2.2.4 Restricciones

- No tomará lista de los estudiantes.
- No llevará un control diario de las tareas.
- No asignará tiempos a las actividades de los estudiantes.
- No llevará un inventario de los artículos del LTIC.
- No se registrarán préstamos de los artículos del LTIC.
- No se realizará el registro de asistencia por medio de un sensor biométrico.

2.2.5 Suposiciones y dependencias

El sistema deberá operarse sobre un sistema operativo con GNU/Linux Ubuntu 16.04 server; la base de datos con la que se trabajará es PostgreSQL 9.5.5 y será orientado a la web.

2.2.6 Requisitos futuros

- Optimizar el registro de la asistencia por medio de un sensor.
- Manejar y administrar un inventario del LTIC para realizar los préstamos de los artículos.

2.3 Requisitos específicos

2.3.1 Interfaces externas

2.3.1.1 Interfaz con el usuario

El sistema debe ser fácil de comprender y fácil de usar, además de ser predictivo con el usuario ya que lo podrá utilizar cualquier persona que sepa operar un equipo de cómputo.

2.3.1.2 Interfaz con el hardware

Para el uso del sistema, usuario podrá acceder desde cualquier equipo informático que posea una conexión a internet.

2.3.1.3 Interfaz de comunicaciones

El sistema se encuentra expuesto en una red publico/privada, la cual utiliza el protocolo TCP/IP para sus conexiones entre los equipos de los usuarios finales y el servidor.

2.3.2 Funciones

2.3.2.1 Requerimiento funcional agregar tipos de catálogos

Id. Requerimiento	RF01-ADD-TCAT
Descripción	Permite agregar un nuevo tipo de catálogo en el sistema.
Parámetros de entrada	Datos del tipo de catálogo: código, nombre, descripción.
Resultado	Mensaje de información guardado exitosamente.
Proceso	1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de tipos de catálogos.
Postcondiciones	Se actualizará la tabla de tipos de catálogos.
Efectos secundarios*	Crea un nuevo tipo de catálogo.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.2 Requerimiento funcional editar tipo de catálogo

Id. Requerimiento	RF02-EDIT-TCAT
Descripción	Permite edita un tipo de catálogo en el sistema.
Parámetros de entrada	Datos del tipo de catálogo: nombre, descripción.
Resultado	Mensaje de información guardado exitosamente.
Proceso	1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de tipos de catálogos.
Postcondiciones	Se actualizará la tabla de tipos de catálogos.
Efectos secundarios*	Edita el tipo de catálogo seleccionado previamente.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.3 *Requerimiento funcional cambiar estado de tipo de catálogo*

Id. Requerimiento	RF03-STATUS-TCAT
Descripción	Permite cambiar el estado de un tipo de catálogo en el sistema.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de información guardado exitosamente.
Proceso	El usuario dará clic en el botón para activar/desactivar el registro.
Precondiciones	Acceder a la pantalla de tipos de catálogos.
Postcondiciones	Se actualizará la tabla de tipos de catálogos.
Efectos secundarios*	Edita el estado del tipo de catálogo.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.4 *Requerimiento funcional agregar catálogos*

Id. Requerimiento	RF04-ADD-CAT
Descripción	Permite agregar un nuevo catálogo en el sistema.
Parámetros de entrada	Datos del catálogo: tipo de catálogo, código, nombre, descripción.
Resultado	Mensaje de información guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de catálogos.
Postcondiciones	Se actualizará la tabla de catálogos.
Efectos secundarios*	Crea un nuevo catálogo.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.5 *Requerimiento funcional editar catálogo*

Id. Requerimiento	RF05-EDIT-CAT
Descripción	Permite edita un catálogo en el sistema.
Parámetros de entrada	Datos del catálogo: nombre, descripción.
Resultado	Mensaje de información guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de catálogos.
Postcondiciones	Se actualizará la tabla de catálogos.
Efectos secundarios*	Edita el catálogo seleccionado.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.6 *Requerimiento funcional cambiar estado de catálogo*

Id. Requerimiento	RF06-STATUS-CAT
Descripción	Permite cambiar el estado de un catálogo en el sistema.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de información guardado exitosamente.
Proceso	El usuario dará clic en el botón para activar/desactivar el registro.
Precondiciones	Acceder a la pantalla de catálogos.
Postcondiciones	Se actualizará la tabla de catálogos.
Efectos secundarios*	Edita el estado del catálogo.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.7 *Requerimiento funcional editar parámetro*

Id. Requerimiento	RF07-EDIT-PARAMETER
Descripción	Permite edita un parámetro en el sistema.
Parámetros de entrada	Datos del parámetro: nombre, descripción, valor (f, h, e, d, t).
Resultado	Mensaje de información guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de parámetros.
Postcondiciones	Se actualizará la tabla de parámetros.
Efectos secundarios*	Edita el parámetro seleccionado.
Prioridad	Medio.
Persona encargada	Docente.

2.3.2.8 *Requerimiento funcional agregar personas*

Id. Requerimiento	RF08-ADD-PEOPLE
Descripción	Permite agregar una nueva persona en el sistema.
Parámetros de entrada	Datos de la persona: tipo de identificación, número de identificación, nombre, apellido, género, fecha de nacimiento, identificación PUCE, correo personal.
Resultado	Mensaje de información guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de personas.
Postcondiciones	Se actualizará la tabla de personas.
Efectos secundarios*	Crea una nuevo persona.
Prioridad	Alta.
Persona encargada	Docente, becario.

2.3.2.9 *Requerimiento funcional editar persona*

Id. Requerimiento	RF09-EDIT-PEOPLE
Descripción	Permite edita una persona en el sistema.
Parámetros de entrada	Datos de la persona: número de identificación, nombre, apellido, género, fecha de nacimiento, identificación PUCE, correo personal, perfil.
Resultado	Mensaje de información guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de personas.
Postcondiciones	Se actualizará la tabla de personas.
Efectos secundarios*	Edita la persona seleccionada.
Prioridad	Alta.
Persona encargada	Docente, becario.

2.3.2.10 *Requerimiento funcional cambiar estado de persona*

Id. Requerimiento	RF10-STATUS-PEOPLE
Descripción	Permite cambiar el estado de una persona en el sistema.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de información guardado exitosamente.
Proceso	El usuario dará clic en el botón para activar/desactivar el registro.
Precondiciones	Acceder a la pantalla de personas.
Postcondiciones	Se actualizará la tabla de personas.
Efectos secundarios*	Edita el estado de la persona.
Prioridad	Alta.
Persona encargada	Docente, becario.

2.3.2.11 *Requerimiento funcional generar usuario*

Id. Requerimiento	RF11-GEN-USER
Descripción	Permite generar el usuario de una persona para usar el sistema.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de generación correcta de usuario.
Proceso	El usuario dará clic en el botón para generar usuario.
Precondiciones	Acceder a la pantalla de personas.
Postcondiciones	Se actualizará la tabla de personas.
Efectos secundarios*	Crea un nuevo usuario.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.12 *Requerimiento funcional cambiar estado de usuario*

Id. Requerimiento	RF12-STATUS-USER
Descripción	Permite cambiar el estado de un usuario en el sistema.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de información guardado exitosamente.
Proceso	El usuario dará clic en el botón para activar/desactivar el registro.
Precondiciones	Acceder a la pantalla de usuarios.
Postcondiciones	Se actualizará la tabla de usuarios.
Efectos secundarios*	Edita el estado del usuario.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.13 *Requerimiento funcional restablecer contraseña*

Id. Requerimiento	RF13-PASS-USER
Descripción	Permite recuperar la contraseña del usuario.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de reinicio de contraseña de usuario.
Proceso	El usuario dará clic en el botón para restablecer usuario.
Precondiciones	Acceder a la pantalla de usuarios.
Postcondiciones	Se actualizará la tabla de usuarios.
Efectos secundarios*	Edita la contraseña del usuario.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.14 *Requerimiento funcional cambiar estado de parámetro*

Id. Requerimiento	RF15-RECOVER-PASSWORD
Descripción	Permite recuperar la clave de inicio de sesión.
Parámetros de entrada	Usuario generado para el sistema.
Resultado	Mensaje de información enviada exitosamente.
Proceso	El usuario dará clic en el link de recuperar clave.
Precondiciones	Acceder a la pantalla de inicio de sesión.
Postcondiciones	Ninguna.
Efectos secundarios*	Edita la clave del usuario y la envía por correo electrónico.
Prioridad	Bajo.
Persona encargada	Docente, becario.

2.3.2.15 *Requerimiento funcional asociar personas al horario*

Id. Requerimiento	RF16-ASSOCIATE-PEOPLE
Descripción	Permite asociar los alumnos al horario para el control de asistencia.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de asociación guardada exitosamente.
Proceso	El usuario dará clic en uno de los botones presentes en la tabla del horario para asociar a uno o varios estudiantes en dicho horario.
Precondiciones	Acceder a la pantalla de horarios.
Postcondiciones	Se actualizará la tabla de asociación de personas al horario.
Efectos secundarios*	Crea una nueva asociación de alumnos al horario seleccionado.
Prioridad	Alta.
Persona encargada	Docente, becario.

2.3.2.16 *Requerimiento funcional reasignar horas estudiantes*

Id. Requerimiento	RF14-REASSIGN-HOURS
Descripción	Permite reasignar horas de asistencia de un estudiante a otro.
Parámetros de entrada	Datos de reasignación: alumno, día reasignado hora entrada, hora salida.
Resultado	Mensaje de información guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Ingresar los parámetros de entrada obligatorios (marcados con *). 3. Esperar el resultado.
Precondiciones	Acceder a la pantalla de asociación de personas al horario.
Postcondiciones	Ninguna.
Efectos secundarios*	Crea un registro para las horas reasignadas.
Prioridad	Bajo.
Persona encargada	Docente.

2.3.2.17 *Requerimiento funcional registrar permisos*

Id. Requerimiento	RF17-ASSOCIATE-PERMISSION
Descripción	Permite registrar un permiso con justificación o no para el módulo de reportería.
Parámetros de entrada	Ninguna.
Resultado	Mensaje de permiso guardado exitosamente.
Proceso	<ol style="list-style-type: none"> 1. Ingresar a la pantalla correspondiente. 2. Dar clic en el horario en el cual desea registrar el permiso. 3. Dar clic en el botón de permisos y seleccionar la persona a la cual se está registrando el permiso. 4. Esperar el resultado.
Precondiciones	Acceder a la pantalla de asociar alumnos al horario.
Postcondiciones	Ninguna.
Efectos secundarios*	Registra un nuevo permiso a la persona seleccionada.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.18 *Requerimiento funcional descargar reporte de faltas*

Id. Requerimiento	RF18-DOWN-ABSENT
Descripción	Permite descargar el cálculo del total de faltas por un período específico.
Parámetros de entrada	Período.
Resultado	Archivo PDF con el reporte en el período específico.
Proceso	El usuario dará clic en el botón descargar reporte.
Precondiciones	Acceder a la pantalla de cálculo de faltas.
Postcondiciones	Se descargará el reporte.
Efectos secundarios	Ninguno.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.19 *Requerimiento funcional descargar reporte de atrasos*

Id. Requerimiento	RF19-DOWN-DELAY
Descripción	Permite descargar el cálculo del total de atrasos por un período específico.
Parámetros de entrada	Período.
Resultado	Archivo PDF con el reporte en el período específico.
Proceso	El usuario dará clic en el botón descargar reporte.
Precondiciones	Acceder a la pantalla de cálculo de atrasos.
Postcondiciones	Se descargará el reporte.
Efectos secundarios	Ninguno.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.20 *Requerimiento funcional descargar reporte de total de faltas por atrasos*

Id. Requerimiento	RF20-DOWN-ABSENT-DELAY
Descripción	Permite descargar el cálculo del total de faltas en base a la sumatoria de atrasos por un período específico.
Parámetros de entrada	Período.
Resultado	Archivo PDF con el reporte en el período específico.
Proceso	El usuario dará clic en el botón descargar reporte.
Precondiciones	Acceder a la pantalla de cálculo de las faltas en base a la sumatoria de atrasos.
Postcondiciones	Se descargará el reporte.
Efectos secundarios	Ninguno.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.21 *Requerimiento funcional descargar reporte de total de faltas justificadas*

Id. Requerimiento	RF21-DOWN-ABSENT-JUSTIFIED
Descripción	Permite descargar el cálculo del total de faltas justificadas con permisos por un período específico.
Parámetros de entrada	Período.
Resultado	Archivo PDF con el reporte en el período específico.
Proceso	El usuario dará clic en el botón descargar reporte.
Precondiciones	Acceder a la pantalla de cálculo de las faltas justificadas con permisos.
Postcondiciones	Se descargará el reporte.
Efectos secundarios	Ninguno.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.22 *Requerimiento funcional descargar certificado de ayudantía*

Id. Requerimiento	RF22-DOWN-ASSISTANTSHIP
Descripción	Permite descargar el cálculo del total de las horas de ayudantía que serán impresas en el certificado.
Parámetros de entrada	Período.
Resultado	Archivo PDF con el certificado en el período específico.
Proceso	El usuario dará clic en el botón descargar certificados.
Precondiciones	Acceder a la pantalla de cálculo de las horas de ayudantía.
Postcondiciones	Se descargarán los certificados.
Efectos secundarios	Ninguno.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.23 *Requerimiento funcional cerrar período académico*

Id. Requerimiento	RF23-CLOSE-PERIOD
Descripción	Permite cerrar el período académico actual.
Parámetros de entrada	Ninguna.
Resultado	Mensaje del cierre exitoso del período actual.
Proceso	El usuario dará clic en el botón descargar certificados.
Precondiciones	Acceder a la pantalla de horarios.
Postcondiciones	Ninguna.
Efectos secundarios*	<ul style="list-style-type: none"> • Se eliminan todas las asociaciones de alumnos a los horarios. • Se elimina el valor de la parametrización del período académico actual para poder configurar el nuevo. • Se desactivan todas las personas con perfil de ayudante.
Prioridad	Alta.
Persona encargada	Docente.

2.3.2.24 *Requerimiento funcional iniciar período académico*

Id. Requerimiento	RF24-OPEN-PERIOD
Descripción	Permite inicial el nuevo período académico.
Parámetros de entrada	Datos del parámetro: descripción, valor texto.
Resultado	Mensaje del cierre exitoso del período actual.
Proceso	El usuario accederá a la pantalla de inicio del nuevo período académico donde ingresará y seleccionará los parámetros descritos anteriormente. El usuario deberá llenar todos los campos remarcados con un asterisco (*), lo que indica que es obligatorio.
Precondiciones	Acceder a la pantalla de horarios.
Postcondiciones	Ninguna.
Efectos secundarios*	Se configura el valor de la parametrización del período académico actual.
Prioridad	Alta.
Persona encargada	Docente, becario

2.3.2.25 *Requerimiento funcional registrar asistencia*

Id. Requerimiento	RF25-REG-ATTENDANCE
Descripción	Permite registrar la asistencia en el ingreso y salida.
Parámetros de entrada	Datos de asistencia: identificación PUCE.
Resultado	Mensaje de ingreso o salida indicando la fecha y hora del registro.
Proceso	El usuario accederá a la pantalla de inicio del sistema donde ingresará el parámetro descrito anteriormente. El usuario deberá llenar todos los campos remarcados con un asterisco (*), lo que indica que es obligatorio.
Precondiciones	Acceder a la pantalla de inicio del sistema.
Postcondiciones	Ninguna.
Efectos secundarios*	Se agrega un registro con la asistencia de la persona.
Prioridad	Alta.
Persona encargada	Becario, ayudante, voluntario

* Acciones que se realizan sobre la base de datos.

2.3.3 Diagrama de casos de uso

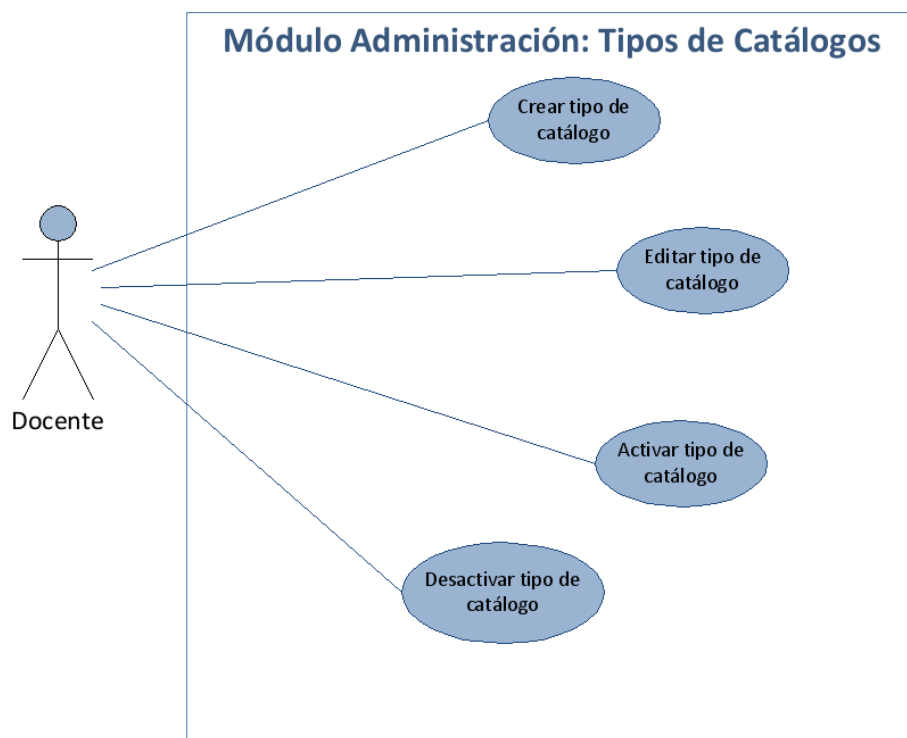


Ilustración 7. Módulo de administración: tipos de catálogos. 13 de junio de 2016.

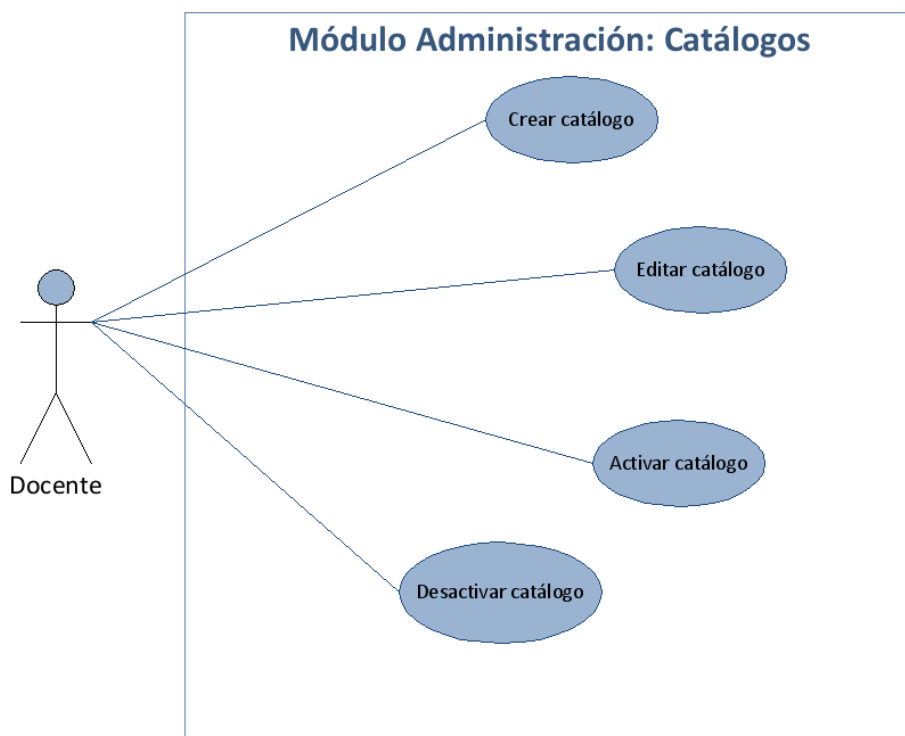


Ilustración 8. Módulo de administración: catálogos. 13 de junio de 2016.

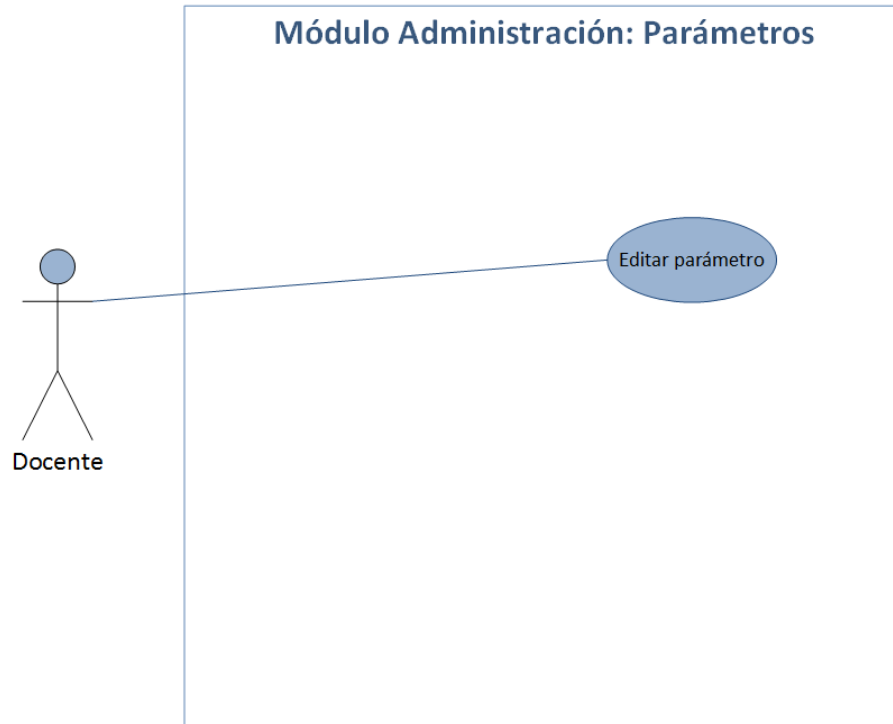


Ilustración 9. Módulo de administración: parámetros. 13 de junio de 2016.

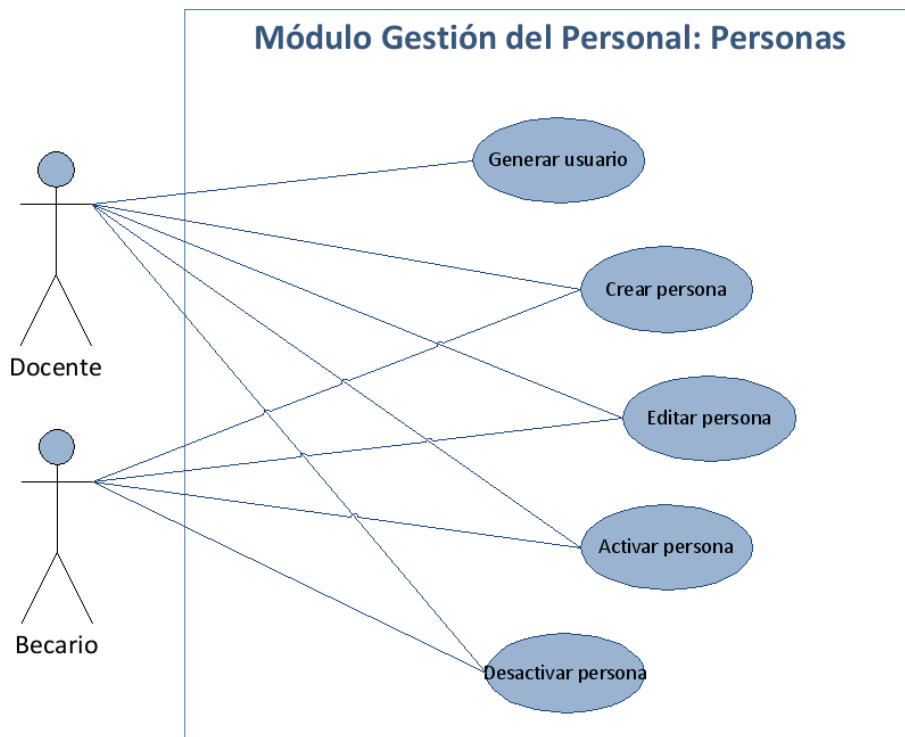


Ilustración 10. Módulo de administración: personas. 13 de junio de 2016.

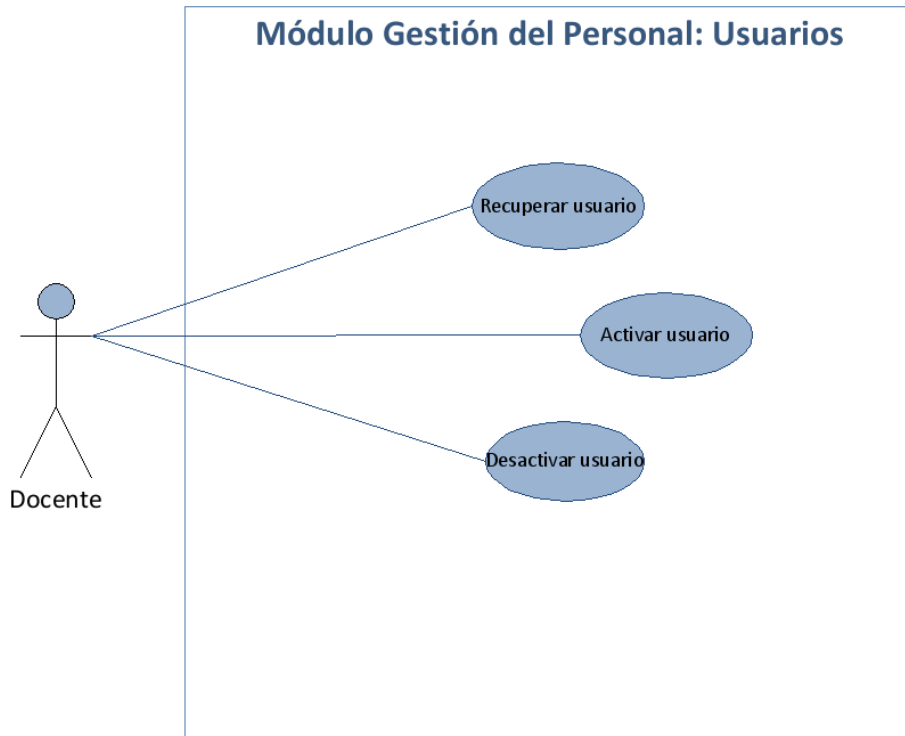


Ilustración 11. Módulo de administración: usuarios. 13 de junio de 2016.

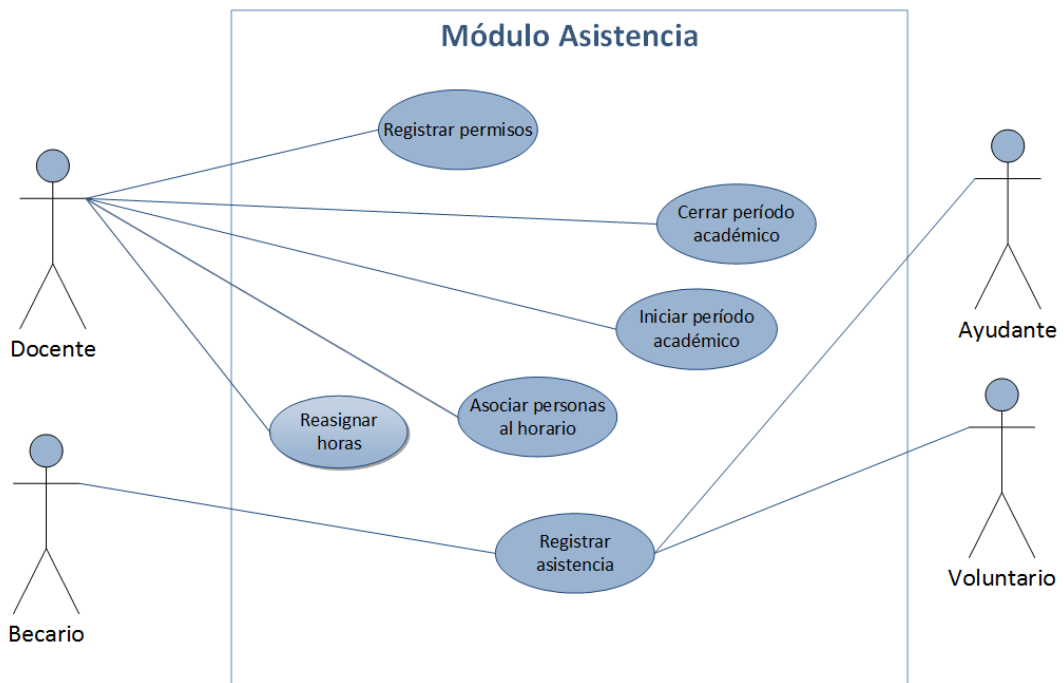


Ilustración 12. Módulo de asistencia. 13 de junio de 2016.

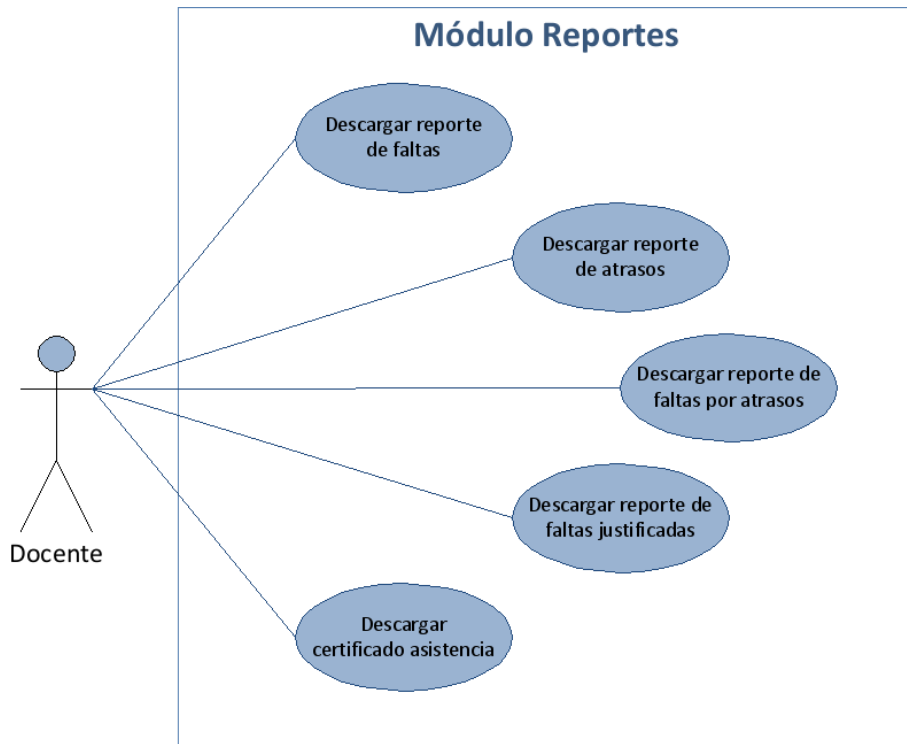


Ilustración 13. Módulo de reportes. 13 de junio de 2016.

CAPÍTULO III: DESARROLLO DEL SISTEMA

3.1 Diseño arquitectónico del sistema

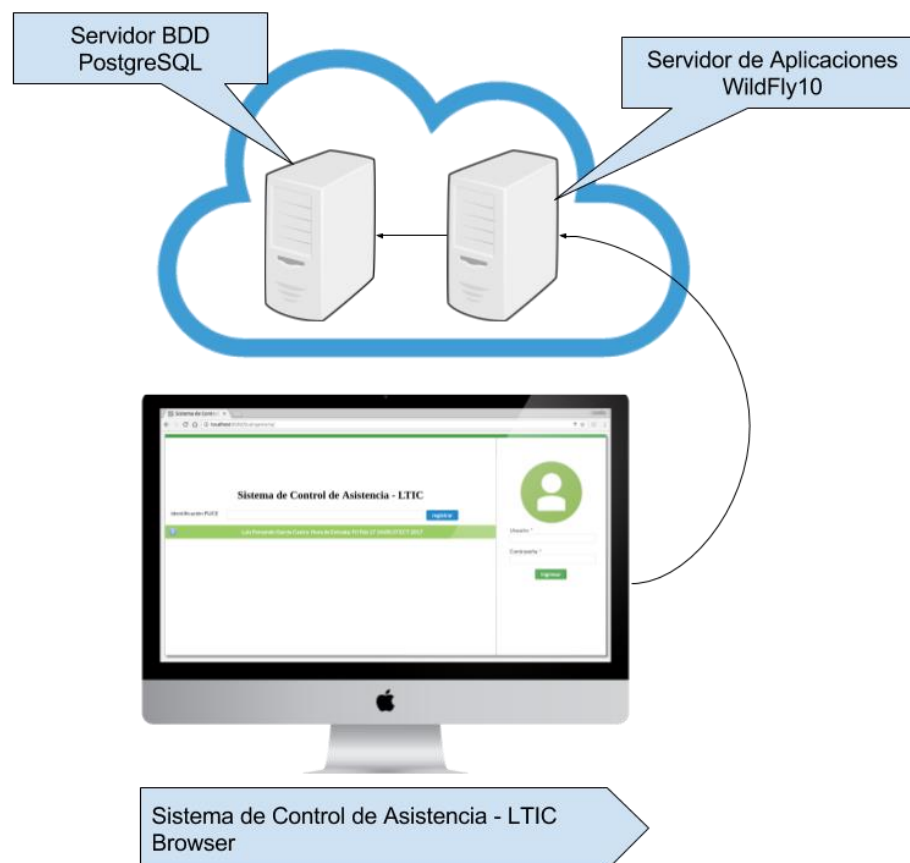


Ilustración 14. Diseño arquitectónico del sistema. 18 de julio de 2016.

3.1.1 Componentes

- **Servidor de base de datos PostgreSQL:** Permite centralizar la información registrada para el sistema.
- **Servidor de aplicaciones WildFly:** Permitirá montar el aplicativo WEB para su uso.
- **Browser:** Herramienta mediante la cual se podrá acceder al sistema.

3.1.2 Diseño de base de datos

Como se indicó en la especificación de requerimientos, se utilizará una base de datos PostgreSQL 9.5.5 instalado en un sistema operativo con GNU/Linux. A continuación, se definen los nombres que se utilizarán en el diseño de la base de datos:

- Nombre de la base de datos: sca_ingenieria.
- Usuario de inicio de sesión: Itic.

- Nombre del esquema: sca.

3.1.2.1 Diagrama conceptual de base de datos

Ver Anexo 1: Diagrama conceptual de base de datos.

3.1.2.2 Diagrama físico de base de datos

Ver Anexo 2: Diagrama físico de base de datos.

3.2 Diseño de la arquitectura de software

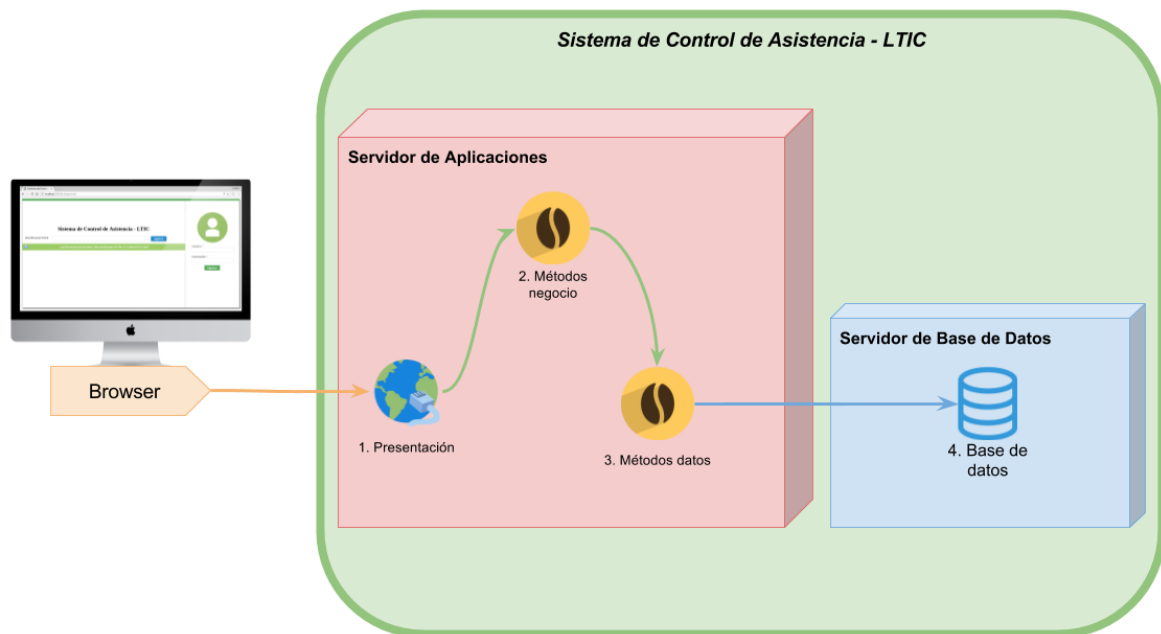


Ilustración 15. Diseño de la arquitectura de software. 20 de julio de 2016.

El sistema será desarrollado en tres capas la cual consiste en las siguientes: capa de datos, capa de negocio y capa de presentación.

3.2.1 Capa de datos

La capa de datos consiste en establecer la conexión con la base de datos por medio de la librería Hibernate. Esta capa estará conformada por las clases POJOs y las clases de acceso de datos (DAO).

3.2.1.1 Diagrama de clases en la capa de datos

Ver Anexo 3: Diagrama de clases.

3.2.1.2 Diseño de los métodos en la capa de datos

Ver Anexo 4: Diagrama de clases, diseño de los métodos de datos.

3.2.2 *Capa de negocio*

En la capa de negocio se definirá la comunicación entre la capa de presentación con la capa de datos. Las clases creadas en esta capa se las conoce como objetos de transferencia de datos (Data Transaction Objects – DTO).

3.2.2.1 *Diseño de los métodos en la capa de negocio*

Ver Anexo 5: Diagrama de clases, diseño de los métodos de negocio.

3.2.3 *Capa de presentación*

La última capa en el proyecto es la capa de presentación, es aquella visual para el usuario final aquí se definirá la interfaz de usuario del sistema y se definirán los métodos utilizados por las JSP.

3.2.2.2 *Diseño de los métodos en la capa de presentación*

Ver Anexo 6: Diagrama de clases, diseño de los métodos de presentación.

CAPÍTULO IV: IMPLEMENTACIÓN

4.1 Implementación en la nube

4.1.1 Google Cloud

La implementación del Sistema de Control de Asistencia – LTIC, se la realizará en dos fases. La primera fase consiste en instalarlo sobre la nube de Google para realizar las pruebas funcionales de todo el sistema y correr las pruebas de seguridad necesarias para proseguir con la segunda fase.

La segunda fase consiste en realizar la implementación en un servidor proporcionado por el área de LTIC, configurado con los implementos necesarios para realizar la instalación del sistema.

Como se puede visualizar en la Ilustración 16, se posee un proyecto llamado “SCA Ingeniería” y una máquina virtual identificada como “sca-ltic”. Posee una configuración inicial de CPU, memoria RAM e IP estática pública (104.197.167.164)

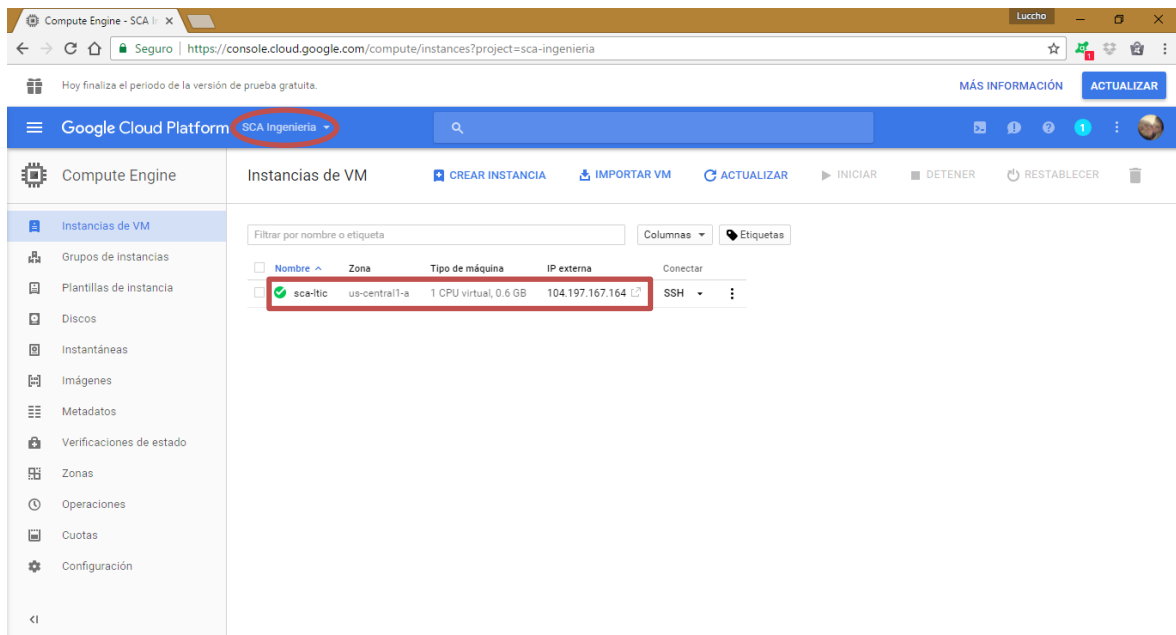


Ilustración 16. Dashboard de instancias de máquinas virtuales. 18 de febrero de 2017.

4.2 Configuración servidor

La configuración inicial del servidor nace en la nube de Google, con la configuración de firewall el cual va a permitir las conexiones al sistema, indicando rangos de IP y puertos para sus conexiones. Para el proyecto se debe permitir en el rango de IP de conexiones la

dirección o direcciones IP pública de la Pontificia Universidad Católica del Ecuador al puerto 80 y 5432, correspondientes al sistema WEB y a la base de datos respectivamente.

El servidor necesita tener los siguientes componentes:

- Java8 (requerimiento del sistema y servidor de aplicaciones).
- WildFly 10.
- PostgreSQL 9.5.5.
- Repositorio Maven.

```
luis_fernando_garcia_castro@sca-ltic:~$ java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
luis_fernando_garcia_castro@sca-ltic:~$ psql --version
psql (PostgreSQL) 9.5.5
luis_fernando_garcia_castro@sca-ltic:~$ mvn --version
Apache Maven 3.3.9
Maven home: /usr/share/maven
Java version: 1.8.0_121, vendor: Oracle Corporation
Java home: /usr/lib/jvm/java-8-oracle/jre
Default locale: es_EC, platform encoding: UTF-8
OS name: "linux", version: "4.4.0-62-generic", arch: "amd64", family: "unix"
luis_fernando_garcia_castro@sca-ltic:~$
```

Ilustración 17. Versiones instaladas en la nube de Google. 18 de febrero de 2017.

4.3 Integración continua

Para realizar la instalación de nuevas versiones en el servidor como ajustes a las pruebas finales o implementaciones futuras, se instala un sistema de integración continua “Jenkins”. Este se conectará al repositorio GIT en el cual se encuentra el código fuente del proyecto y ejecutará una serie de rutinas para realizar el despliegue del sistema en el servidor de aplicaciones nombrado anteriormente. Será entregado las credenciales del usuario administrador Jenkins al igual que las credenciales para acceder al repositorio GIT.

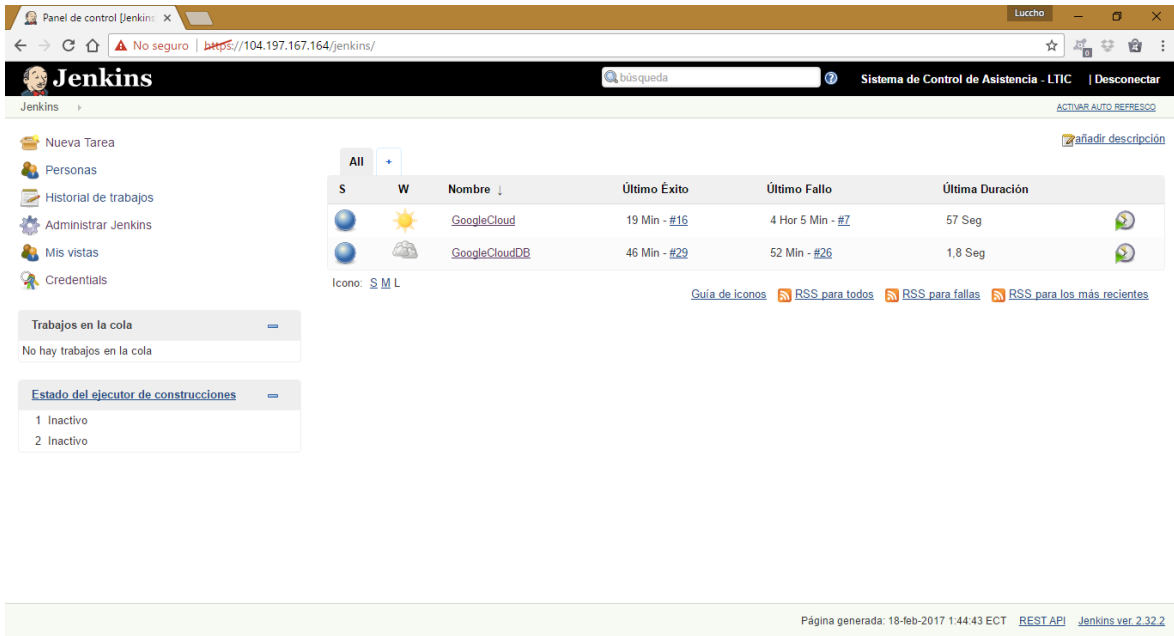


Ilustración 18. Visualización del historial de despliegue por proyecto. 18 de febrero de 2017.

4.4 Despliegue del sistema de control de asistencia – LTIC

Para realizar el despliegue del Sistema de Control de Asistencia – LTIC, se realizará de forma automática por medio del sistema de integración continua. El despliegue se realizará cada vez que se realice una actualización en el código fuente en la rama “gcloud”. Para ejecutar el despliegue en el servidor proporcionado por el LTIC, de igual manera se realizará automáticamente con una actualización en el código fuente en la rama “master”.

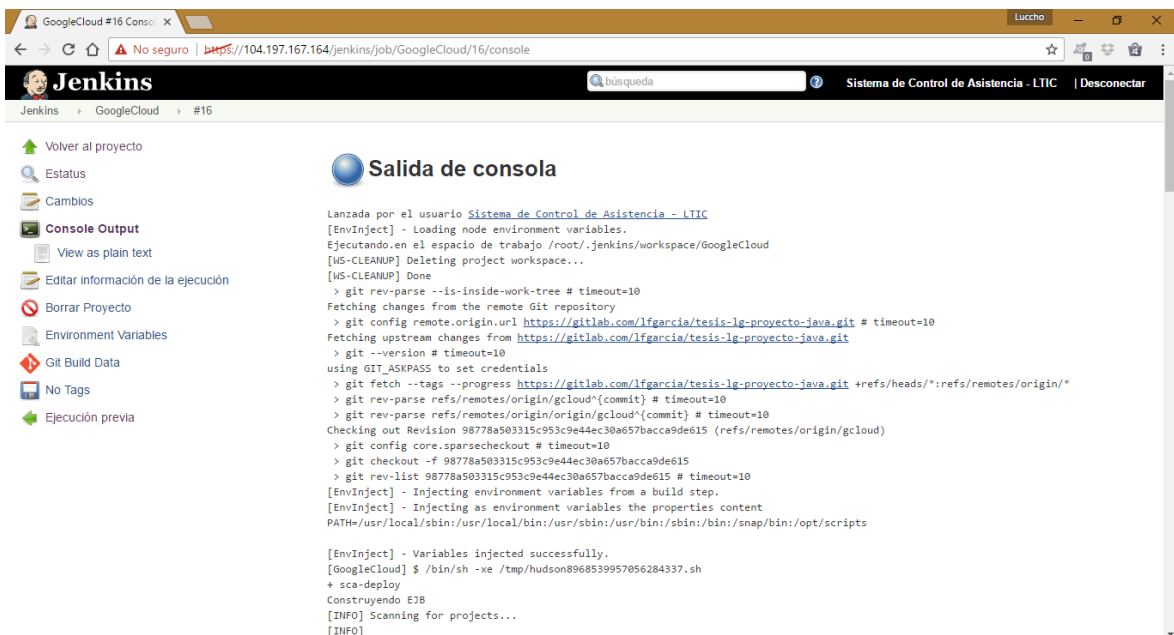


Ilustración 19. Visualización del despliegue en el sistema de integración continua Jenkins. 18 de febrero de 2017.

4.5 Comprobación instalación del sistema

El último proceso en la implementación del sistema, será comprobar que el despliegue se realizó correctamente dirigiéndose a la dirección WEB <http://sca-ltic.von-development-studio.com/ScaIngenieria/>

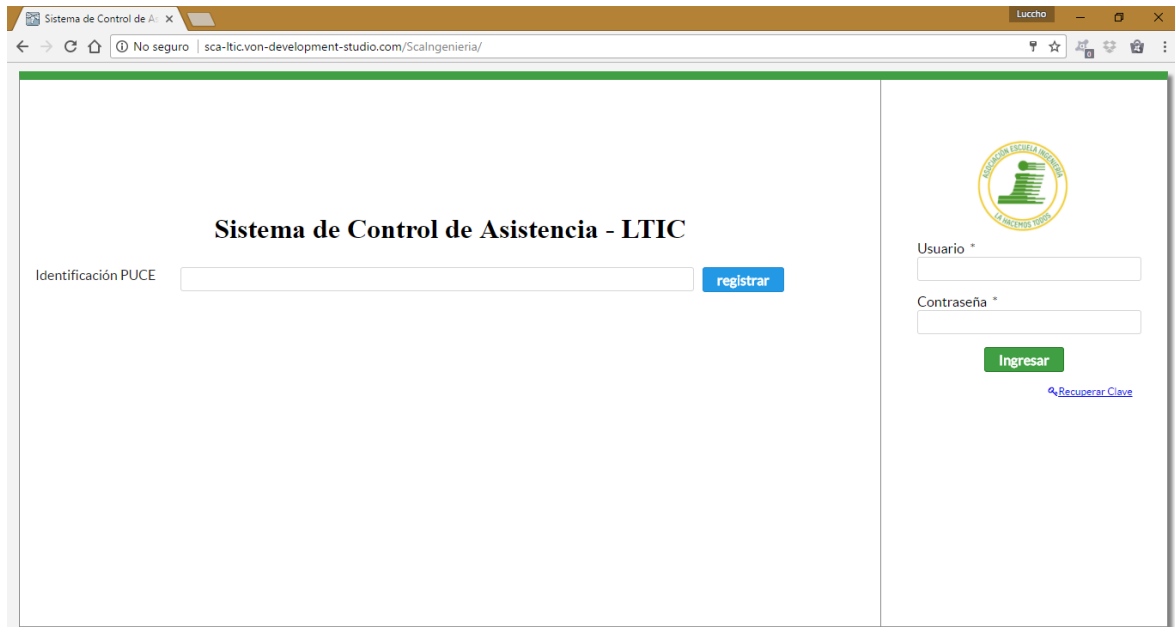


Ilustración 20. Pantalla de ingreso al sistema. 18 de febrero de 2017.

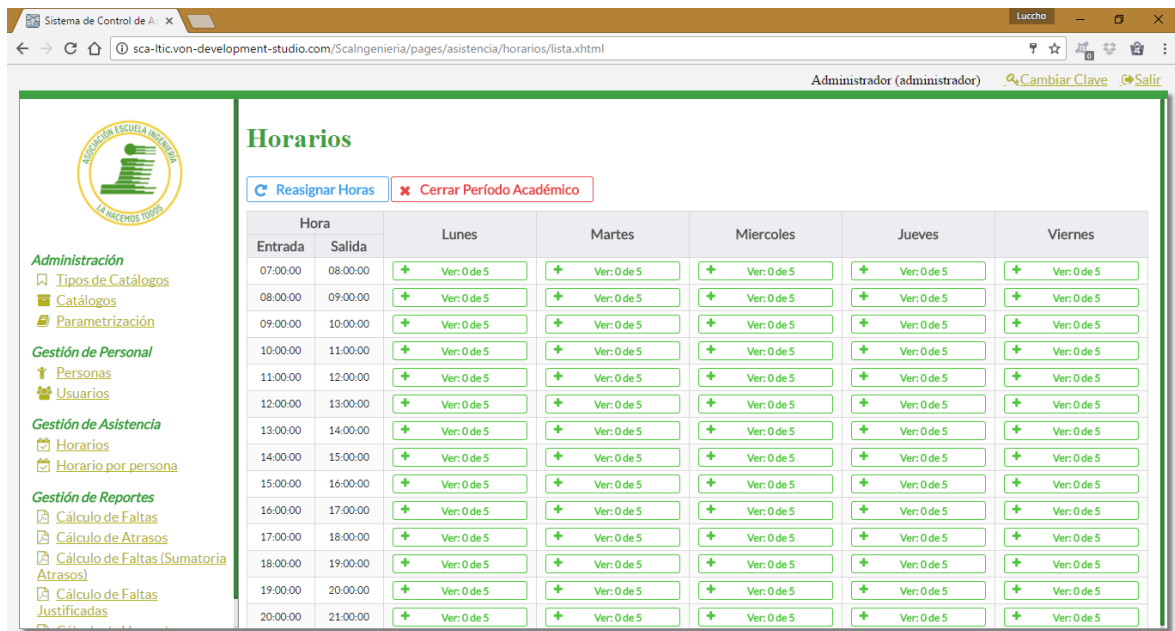


Ilustración 21. Pantalla inicial del sistema. 18 de febrero de 2017.

CAPÍTULO V: SEGURIDADES Y PRUEBAS

5.1 Reporte de ACUNETIX

Para realizar un escaneo de seguridad sobre el aplicativo, se utilizó la herramienta de Escaneo de Vulnerabilidades en Línea de ACUNETIX en su período de prueba. Este permite realizar un escaneo completo de la seguridad. La versión de prueba utilizada nos indica únicamente los fallos o vulnerabilidades que posee el sistema, más no el lugar en donde se encuentran como se puede visualizar en la Ilustración 24.

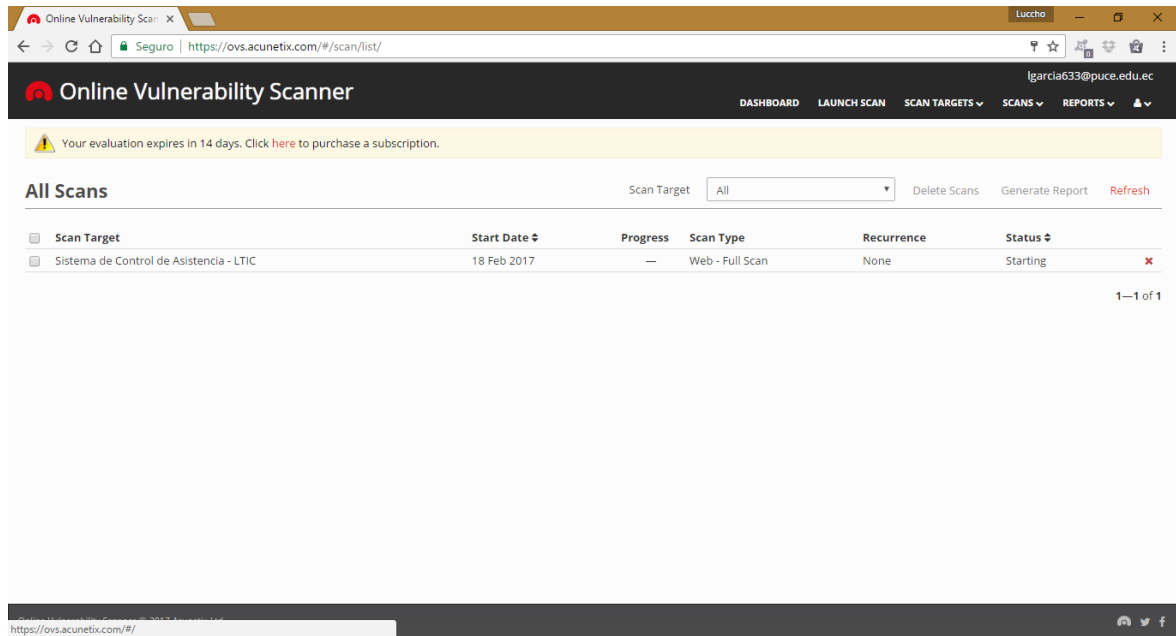


Ilustración 22. Inicio del escaneo de vulnerabilidades. 18 de febrero de 2017.

Dependiendo el tamaño del proyecto el proceso del escaneo se toma un tiempo respectivo. El aplicativo realiza ciertas actividades para detectar las vulnerabilidades, como Orígenes permitido (CROSS ORIGIN), inyección de código (SQL o XSS), entre otras.

Las nombradas anteriormente se logran controlar en el sistema por medio de configuraciones específicas y por el uso de frameworks que permite bloquear estas vulnerabilidades de forma más simple y sencilla.

5.1.1 Primefaces

En las configuraciones por defecto de Primefaces evita una gran cantidad de vulnerabilidades, pero no al 100%; para lograrlo se definen ciertos atributos o configuraciones específicas para aumentar la seguridad en el sistema.

5.1.2 *Hibernate*

Este framework permite la conexión con la base de datos y aumenta la seguridad en cuanto a la inyección de código SQL por medio de los parámetros enviados entre la capa de presentación a la capa de datos.

En las siguientes ilustraciones se muestran los reportes obtenidos per medio de la herramienta:

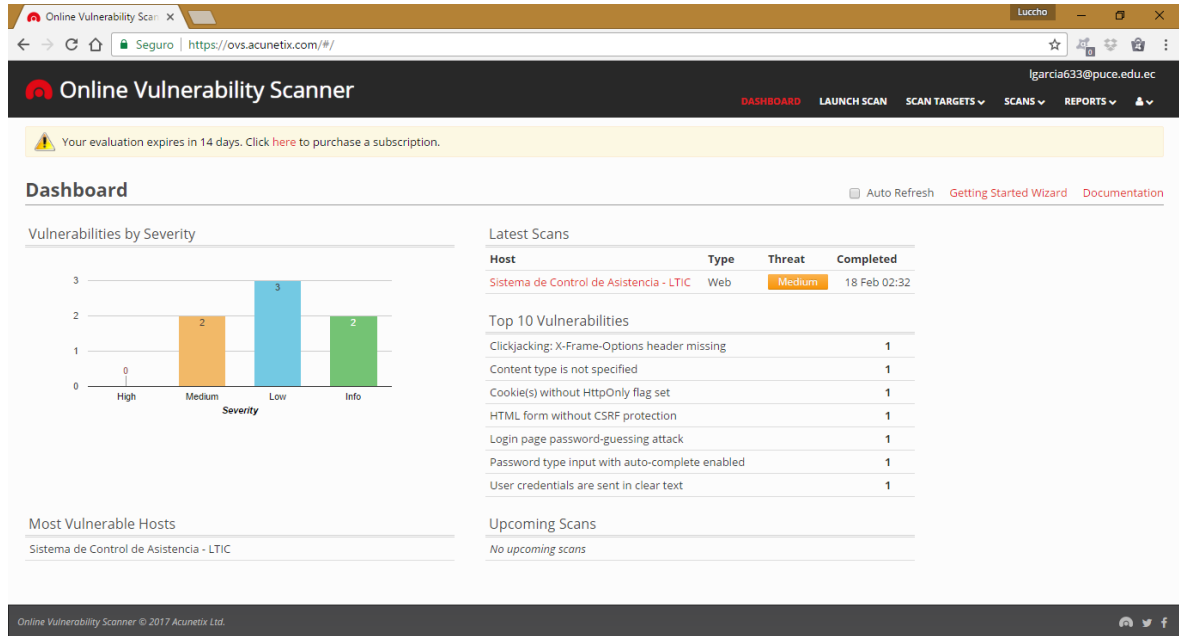


Ilustración 23. Resultado final del escaneo en su primera ejecución. 18 de febrero de 2017.

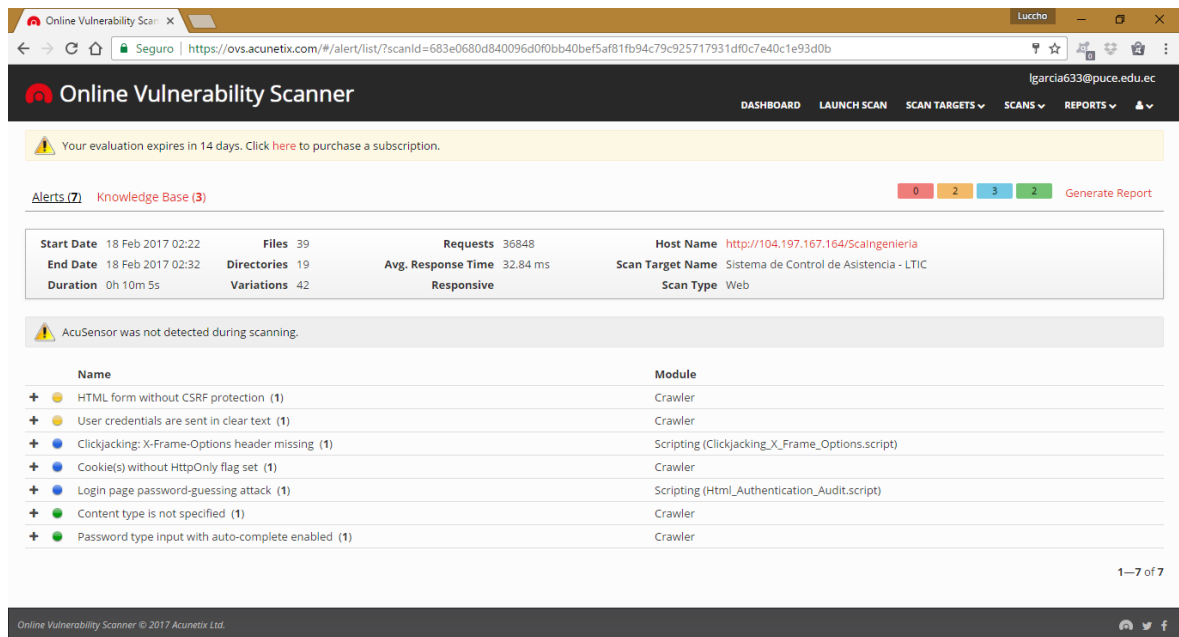


Ilustración 24. Detalle del resultado de escaneo. 18 de febrero de 2017.

Después de este reporte se procedió a realizar los ajustes válidos para corregir las vulnerabilidades indicadas en el reporte de la Ilustración 24.

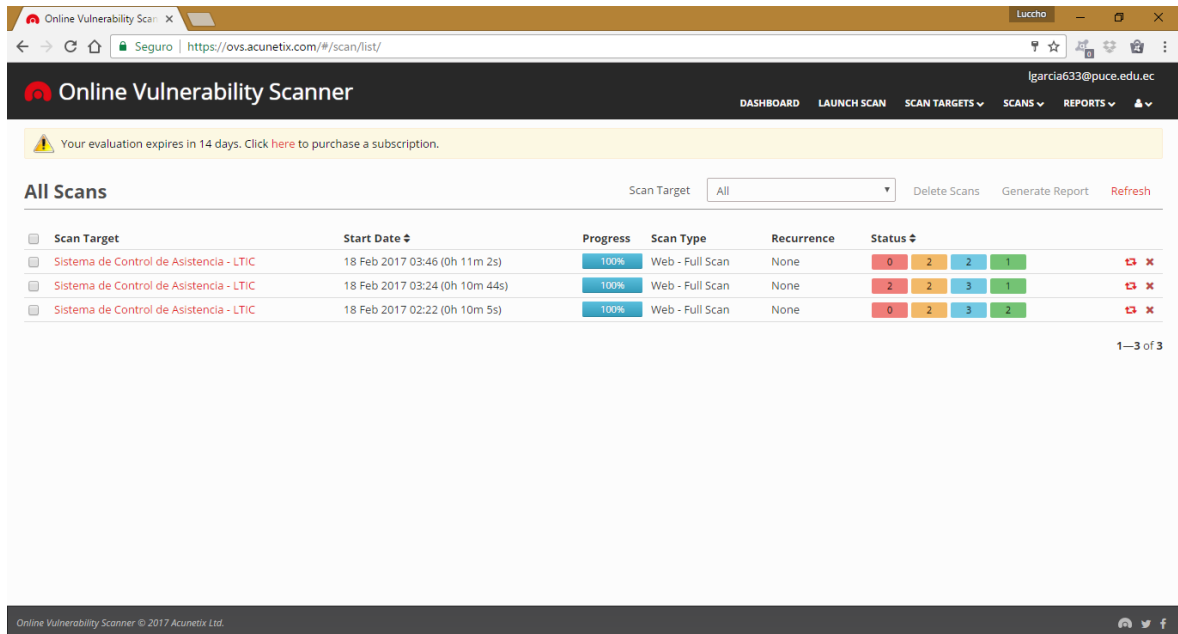


Ilustración 25. Visualización de los tres (3) escaneos ejecutados sobre el sistema. 18 de febrero de 2017.

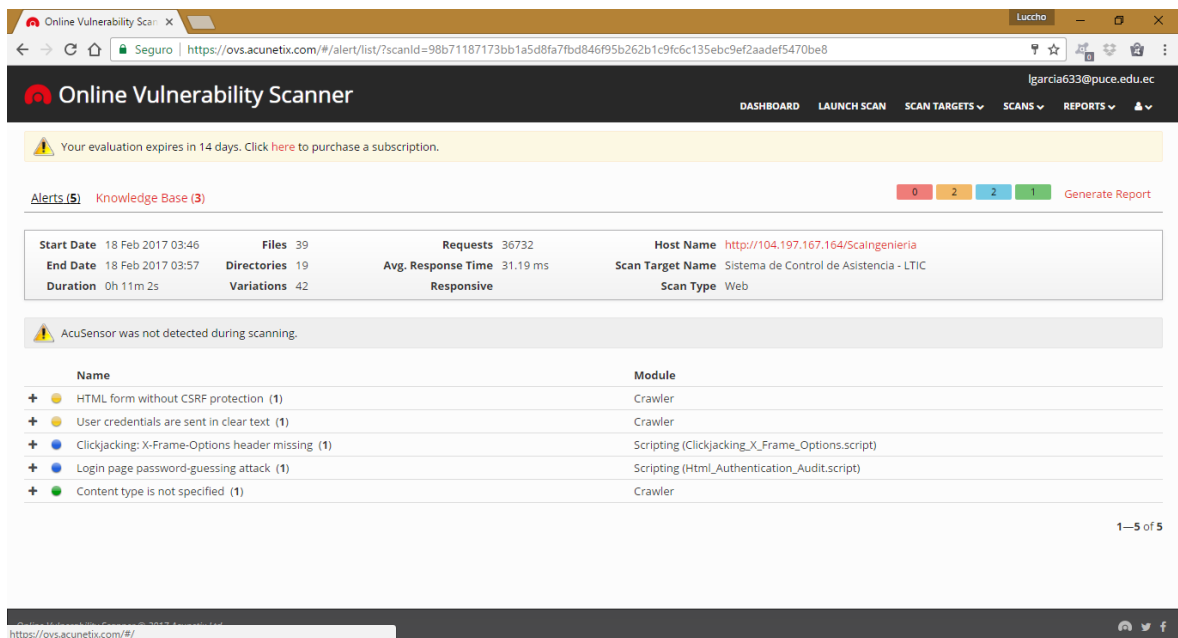


Ilustración 26. Detalle del último resultado del escaneo. 18 de febrero de 2017.

5.2 Pruebas finales en el despliegue

Para realizar la primera prueba integral del sistema se procederá a realizar un ingreso de sesión con las siguientes credenciales, se visualiza en la Ilustración 27.

Usuario: administrador

Contraseña: @dministr@dor



Usuario *

administrador

Contraseña *

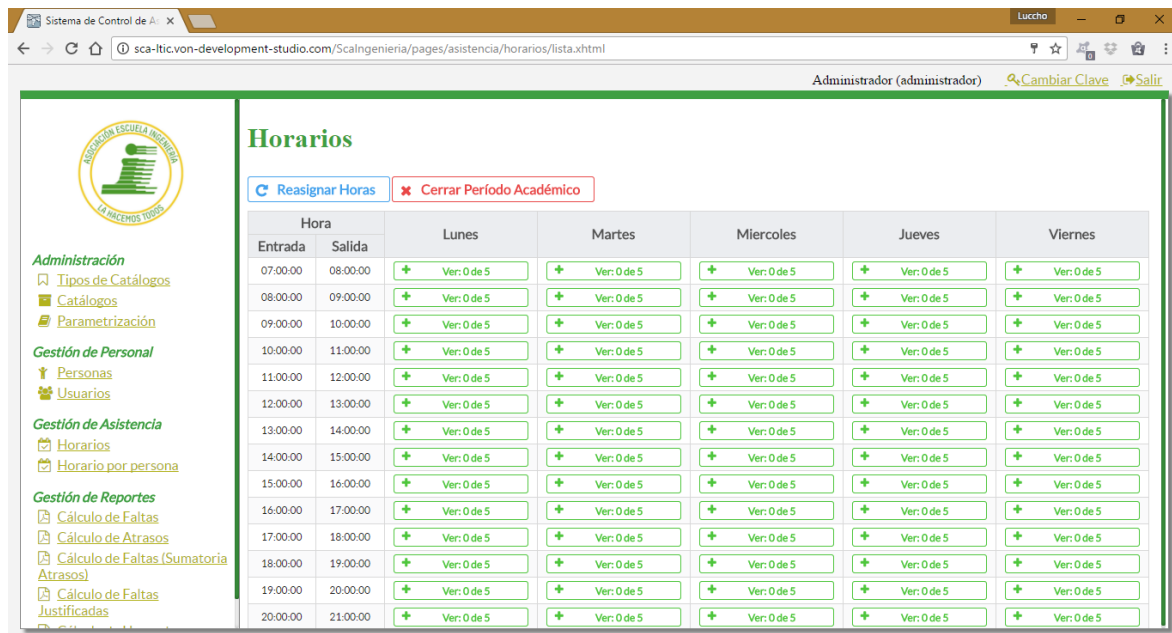
●●●●●●●●●●

Ingresar

[Recuperar Clave](#)

Ilustración 27. Ingreso de credenciales. 18 de febrero de 2017.

La pantalla de bienvenida será la de asociación de los estudiantes al horario, este proceso de asociación se realizará una vez por cada inicio del período académico y las personas encargadas de realizarlo es el docente encargado del área o los estudiantes becarios.



Sistema de Control de A. x Luccho

Administrador (administrador) [Cambiar Clave](#) [Salir](#)

Horarios

[Reasignar Horas](#) [Cerrar Período Académico](#)

Hora		Lunes	Martes	Miércoles	Jueves	Viernes
Entrada	Salida					
07:00:00	08:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
08:00:00	09:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
09:00:00	10:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
10:00:00	11:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
11:00:00	12:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
12:00:00	13:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
13:00:00	14:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
14:00:00	15:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
15:00:00	16:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
16:00:00	17:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
17:00:00	18:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
18:00:00	19:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
19:00:00	20:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5
20:00:00	21:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5	+ Ver: 0 de 5

Ilustración 28. Página de horarios para el sistema. 18 de febrero de 2017.

Se procederá a registrar un usuario de tipo ayudante para realizar el registro de asistencia para lo cual se dirige a la sección de Gestión de Personal: Personas. Dar clic en el botón nuevo y llenar los datos como se muestra en la Ilustración 29.

Tipo Identificación: *	Cédula
Número de Identificación: *	1310176175
Nombre: *	Luis Fernando
Apellido: *	García Castro
Genero:	Masculino
Fecha de Nacimiento:	22-07-1990
Usuario PUCE: *	lgarcia633
Correo Institucional: *	lgarcia633@puce.edu.ec
Correo Personal:	luis.fernando.garcia.cast
Perfil: *	Ayudante

Guardar

Ilustración 29. Ingreso de información de la persona. 18 de febrero de 2017.

Una vez creada la persona, su identificación de la PUCE servirá como campo único para registrar su asistencia en la página inicial del sistema, esta ejecutará ciertas validaciones como, por ejemplo, la existencia de la identificación como se puede observar en la Ilustración 30. En caso de ser el ingreso correcto nos indicará el día y la hora en la cual se registró la asistencia como se indica en la Ilustración 31.

Sistema de Control de Asistencia - LTIC

Identificación PUCE

No se encontró la persona con Identificación: lgarcia63

Ilustración 30. Ingreso incorrecto de la identificación PUCE. 18 de febrero de 2017.

Luis Fernando García Castro. Hora de Entrada: Sat Feb 18 11:33:04 ECT 2017

Ilustración 31. Ingreso correcto de la identificación PUCE. 18 de febrero de 2017.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- La metodología ágil aplicada al proyecto XP (Extreme programming) facilitó el desarrollo del proyecto con entregables funcionales del sistema en sus distintos módulos como el módulo administrativo, gestión de personal y reportería.
- El diseño inicial del sistema, previo a su codificación, puede permitir que el mismo sea robusto así sea pequeño o grande. Debido a que se tiene un concepto claro de lo que el cliente desea y cuáles son sus requerimientos.
- La ingeniería de requerimientos es un área lo suficientemente completa que permite realizar un análisis entre el o los clientes y el proveedor. Existen varias normas y estándares que se pueden aplicar como lo es la norma IEEE 830 aplicada en el presente proyecto e identificada en el Capítulo 2 de Especificación de Requerimientos de Software, la cual nos da una visión macro del sistema a ser desarrollado.
- Disponer de varios ambientes para un sistema permite tener un mayor ámbito de pruebas con el cliente y el equipo de desarrollo. La implementación de estos ambientes puede llevar algo de tiempo, pero una vez que se encuentre listo puede ahorrar mucho tiempo que habitualmente se utiliza para realizar despliegues de los sistemas informáticos, como se puede evidenciar en el Capítulo 4 de Implementación en el cual se realiza la instalación de un ambiente de pruebas en la nube con componentes de integración continua; de tal forma teniendo disponible sus servicios para realizar pruebas de forma más rápida.

Recomendaciones

- Investigar a detalle las distintas metodologías ágiles para saber que metodología se puede aplicar en el proyecto que se realice. El conocer y tener claro estos conceptos nos lleva al éxito en el desarrollo de un sistema, puesto que el uso de metodologías ágiles de desarrollo permitirá cumplir con los tiempos establecidos y de igual manera cumplir con las iteraciones en un alto porcentaje.
- Realizar un correcto ejercicio en la aplicación de las normas y estándares en el desarrollo, como lo es la ingeniería de requerimientos con la norma IEEE 830; permitirá identificar de forma macro lo que el cliente desea que haga el sistema y lo que el equipo de desarrollo realizará.
- No confiar 100% en los frameworks que se utilicen en el desarrollo de un sistema, puesto que puede generar conflictos, o puede llegar a ocasionar problemas en las funcionalidades del sistema ya que no se tienen un completo control sobre los frameworks. Como se puede evidenciar en el Capítulo 5 de Seguridades y Pruebas, el reporte del escaneo de vulnerabilidades del sistema nos dio resultados de vulnerabilidades en el framework de Primefaces y de Hibernate. Es recomendable mantener el correcto uso entre las áreas como es el desarrollo de software y el diseño de base de datos. Por ejemplo, la estructura de base de datos se encuentra creada por medio de SQL; por este motivo la librería de Hibernate no crea objetos nuevos inutilizados sobre la base de datos.
- Se recomienda tener al menos un ambiente de pruebas que pueda ser accedido desde cualquier parte, esto permitirá ejecutar procesos de terceros, como lo fue el análisis de vulnerabilidades y poder conocer los problemas que puede llegar a tener el sistema desarrollado. Como se puede evidenciar en el Capítulo 4 de Implementación, se configuró el ambiente de pruebas en la nube de Google lo cual permite poner a disposición los servicios del sistema desarrollado.

BIBLIOGRAFÍA

- Anton, C. (9 de Julio de 2015). *Qué es PostgreSQL y cuáles son sus ventajas*. Recuperado el 19 de Agosto de 2016, de Platzi: <https://platzi.com/blog/que-es-postgresql/>
- Burns, E., Schalk, C., & Griffin, N. (2010). *JavaServer Faces 2.0: The Complete Reference*. Altamonte Springs: McGraw-Hill Companies.
- Calero, C., Genero, M., & Ruiz, F. (31 de Julio de 2012). *Base de Datos (diseño conceptual, lógico y físico)*. Recuperado el 18 de Agosto de 2016, de SlideShare: <http://es.slideshare.net/claudiachiri/base-de-datos-diseo-conceptuallogico-y-fisico>
- Groussard, T. (2014). *JAVA 8. Los fundamentos del lenguaje Java*. Barcelona: Ediciones ENI.
- IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. New York: The Institute of Electrical and Electronics Engineers, Inc.
- Matus, A., Guerra, A., Gaona, C., Benalcázar, S., & García, L. (11 de Junio de 2013). *Especificación de Requerimientos de Software*. Obtenido de DropBox: https://www.dropbox.com/s/h2ux5y4hv4jg6dv/04%20-%20GAFIMEV_ERS_1.3.docx?dl=0
- Momjian, B. (2001). *PostgreSQL: Introduction and Concepts*. California: Addison-Wesley.
- Ottinger, J., Linwood, J., & Minter, D. (2016). *Beginning Hibernate for Hibernate 5*. New York: APRESS.
- Programación en Java*. (29 de Abril de 2015). Recuperado el 30 de Agosto de 2016, de WIKILIBROS. Libros libres para un mundo libre: https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Java
- Salinas, P. (30 de Octubre de 1996). *Casos de Uso (Use Case)*. Recuperado el 17 de Agosto de 2016, de Departamento de Ciencias de la Computación. Universidad de Chile: <https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>
- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). México: PEARSON EDUCACIÓN.
- Viñé, E. (30 de Junio de 2010). *Introducción a Primefaces*. Recuperado el 23 de Agosto de 2016, de AdictosAlTrabajo: <https://www.adictosaltrabajo.com/tutoriales/introduccion-primefaces/>

ANEXOS

Anexo 1: Diagrama conceptual de base de datos.

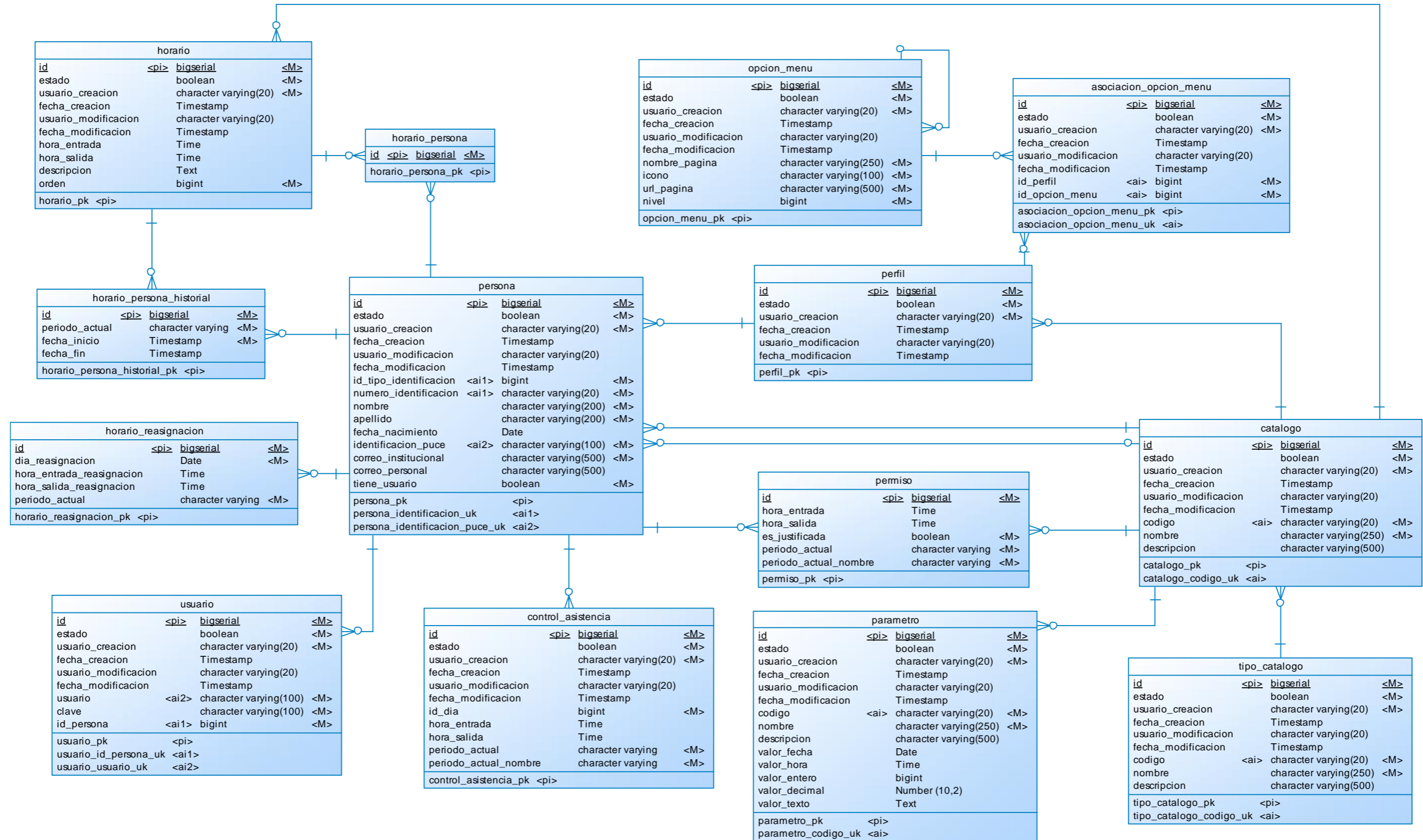


Ilustración 32. Modelo conceptual de la base de datos. 22 de julio de 2016.

Anexo 2: Diagrama físico de base de datos.

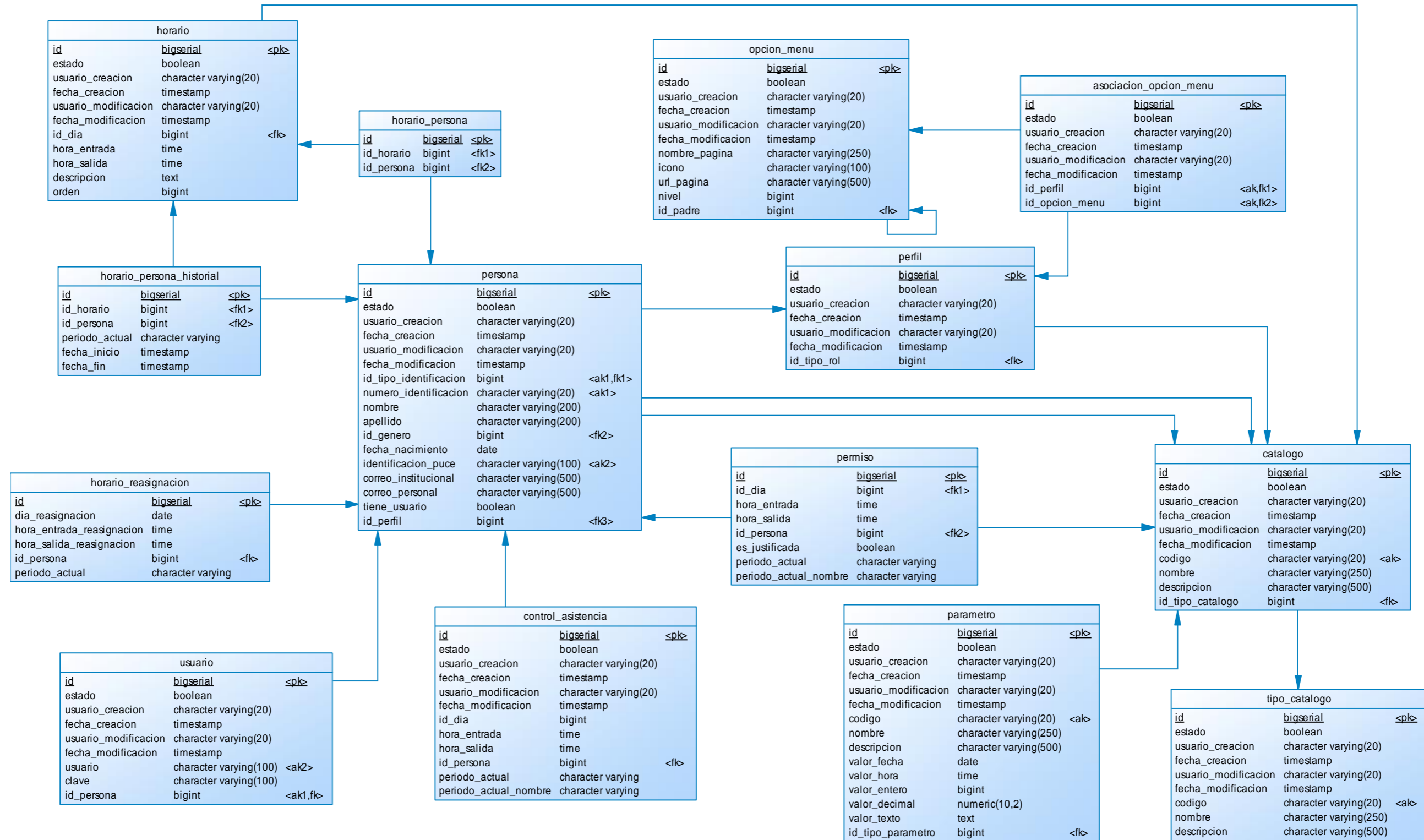


Ilustración 33. Modelo físico de la base de datos. 22 de julio de 2016.

Anexo 3: Diagrama de clases.

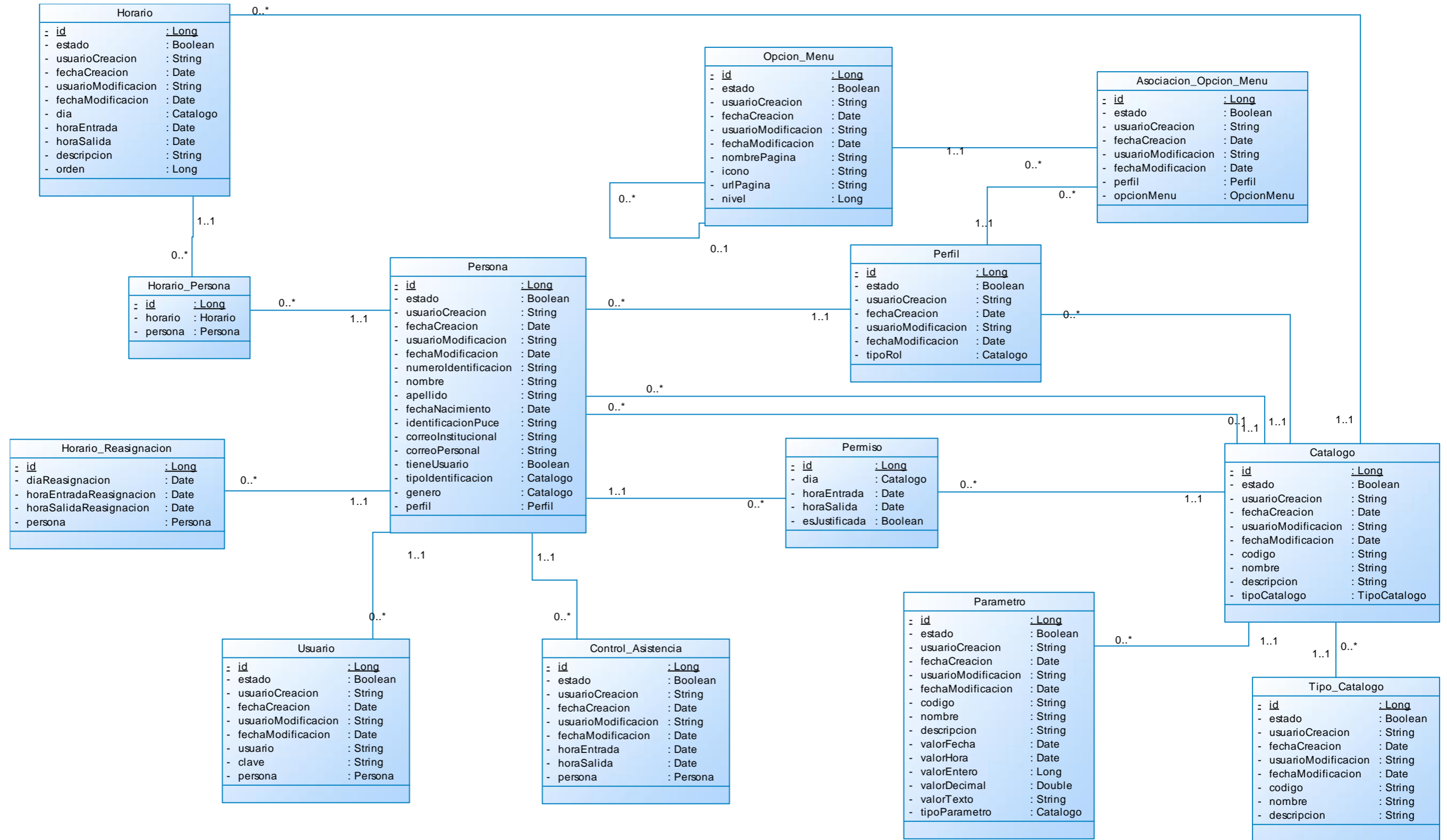


Ilustración 34. Diagrama de clases: entidades de la capa de datos. 23 de julio de 2016.

Anexo 4: Diagrama de clases, diseño de los métodos de datos.

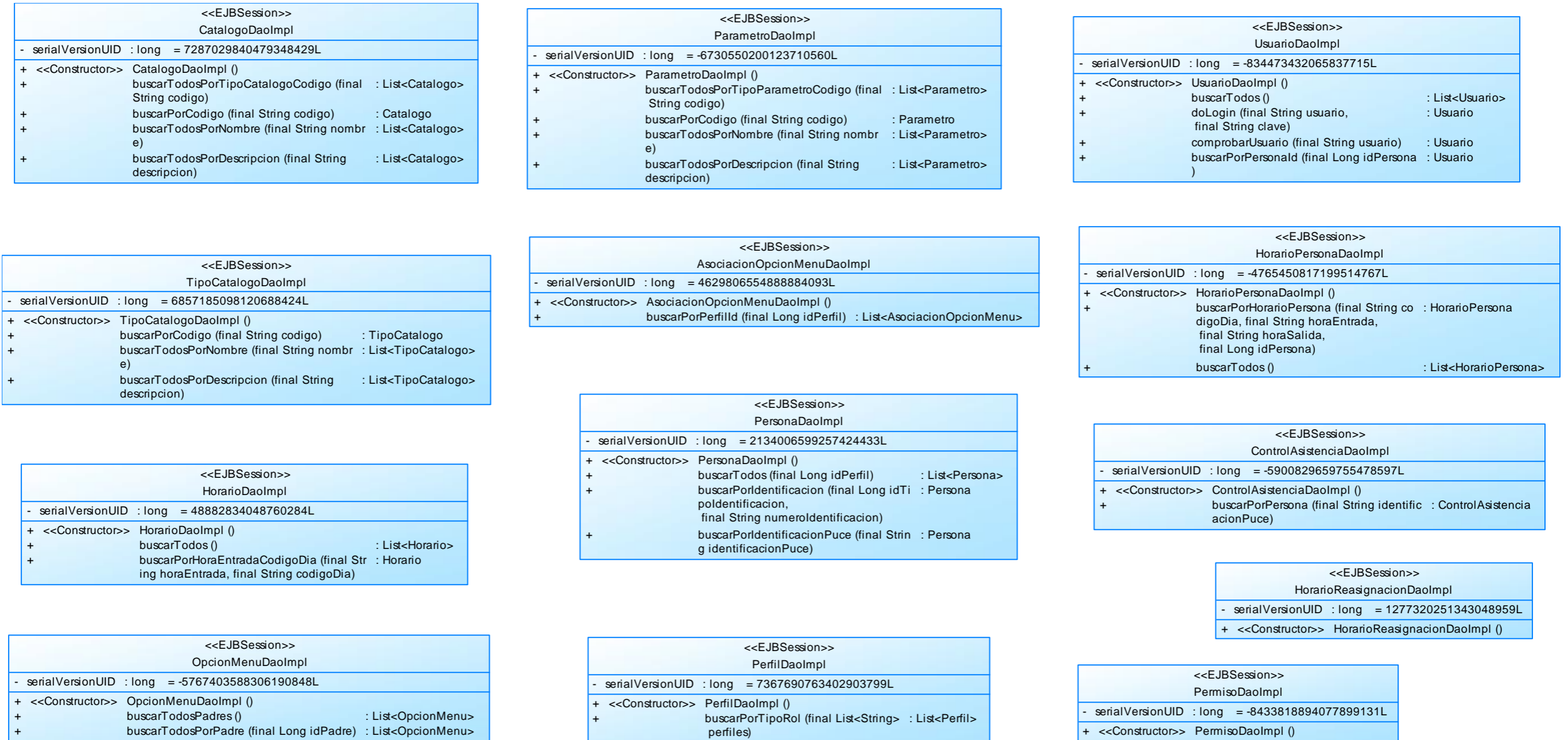


Ilustración 35. Diagrama de clases: métodos de la capa de datos. 24 de julio de 2016.

Anexo 5: Diagrama de clases, diseño de los métodos de negocio.

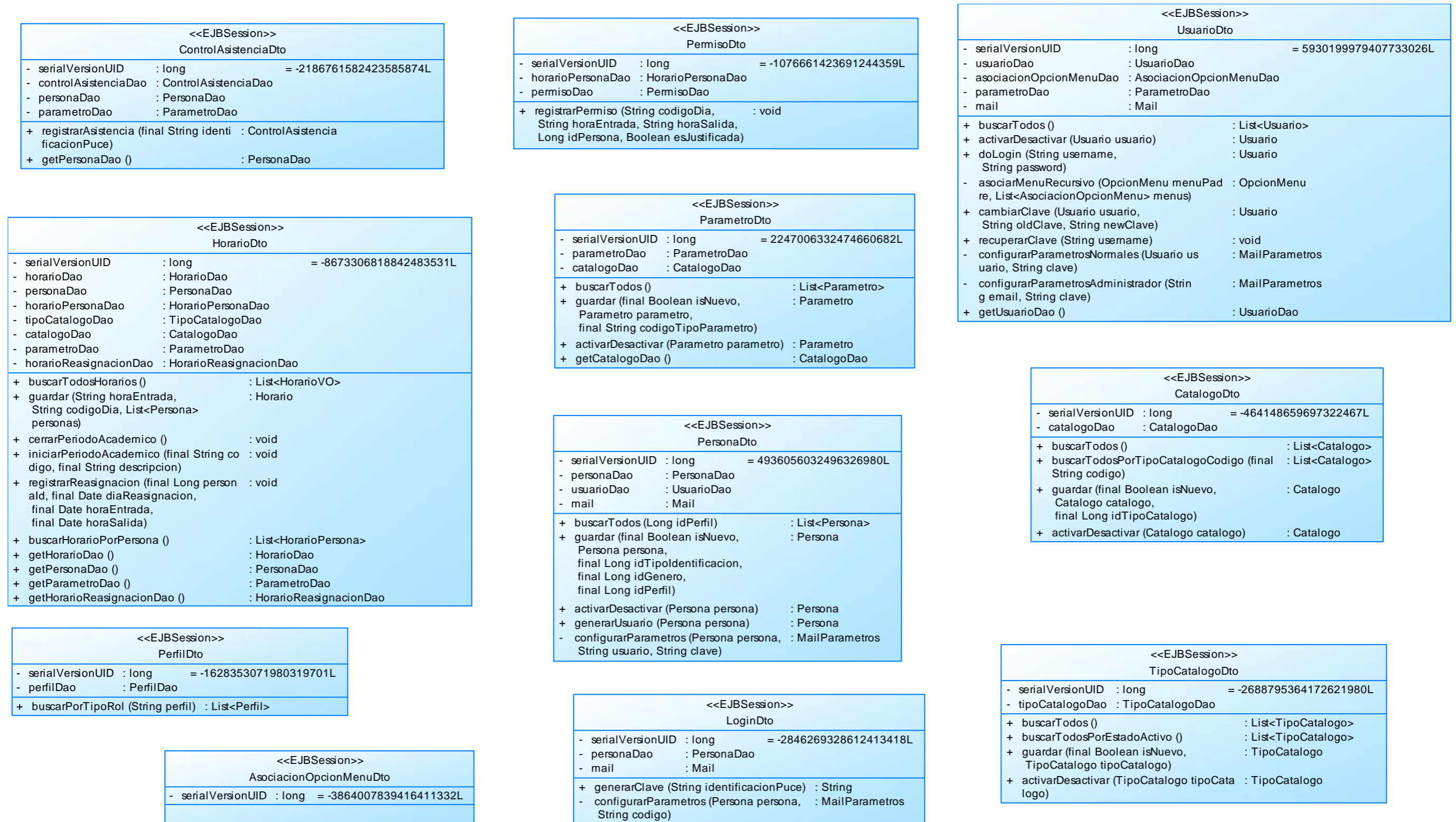


Ilustración 36. Diagrama de clases: métodos de la capa de negocio. 24 de julio de 2016

Anexo 6: Diagrama de clases, diseño de los métodos de presentación.

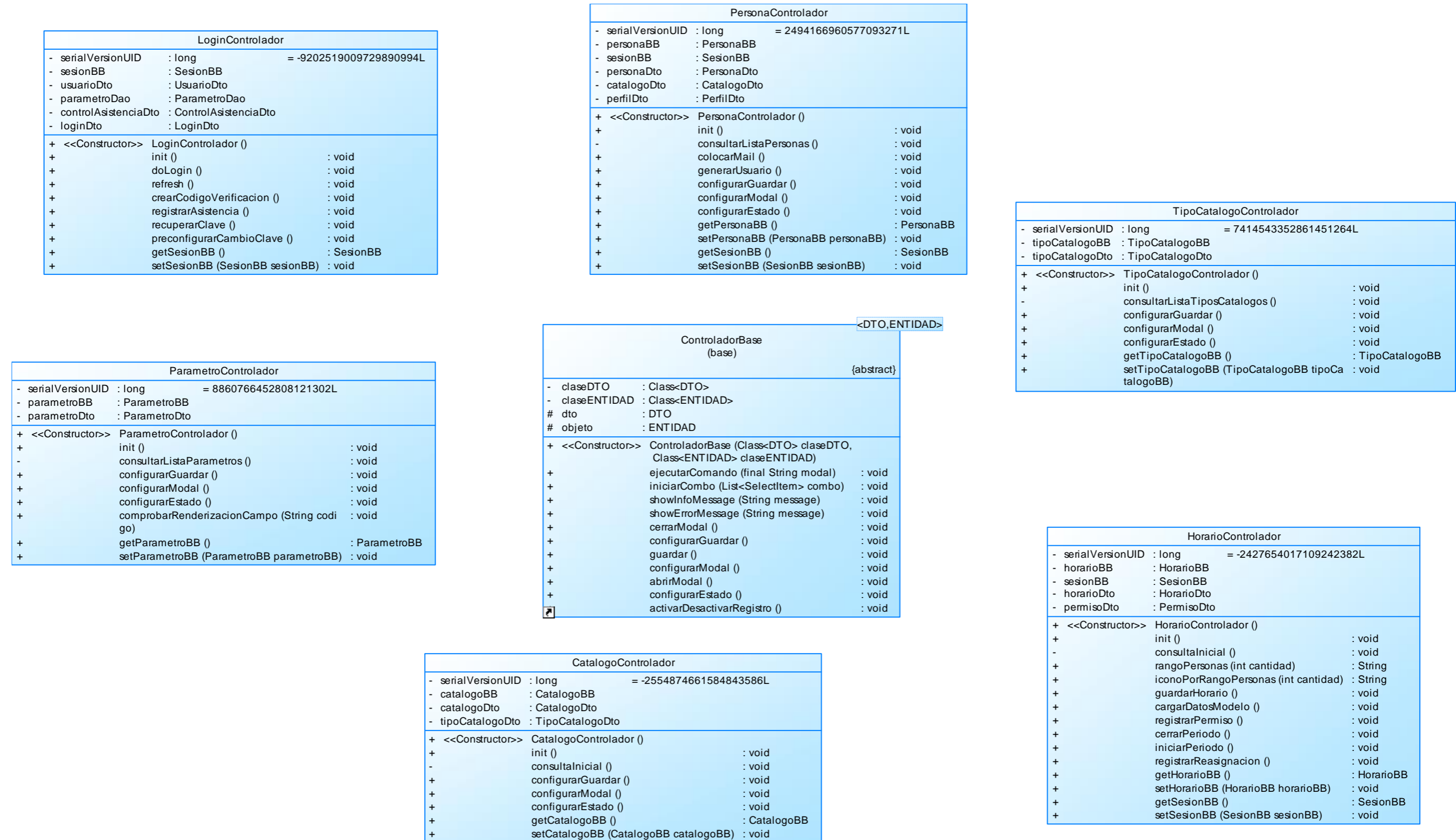


Ilustración 37. Diagrama de clases: métodos de la capa de presentación administrativa. 25 de julio de 2016.

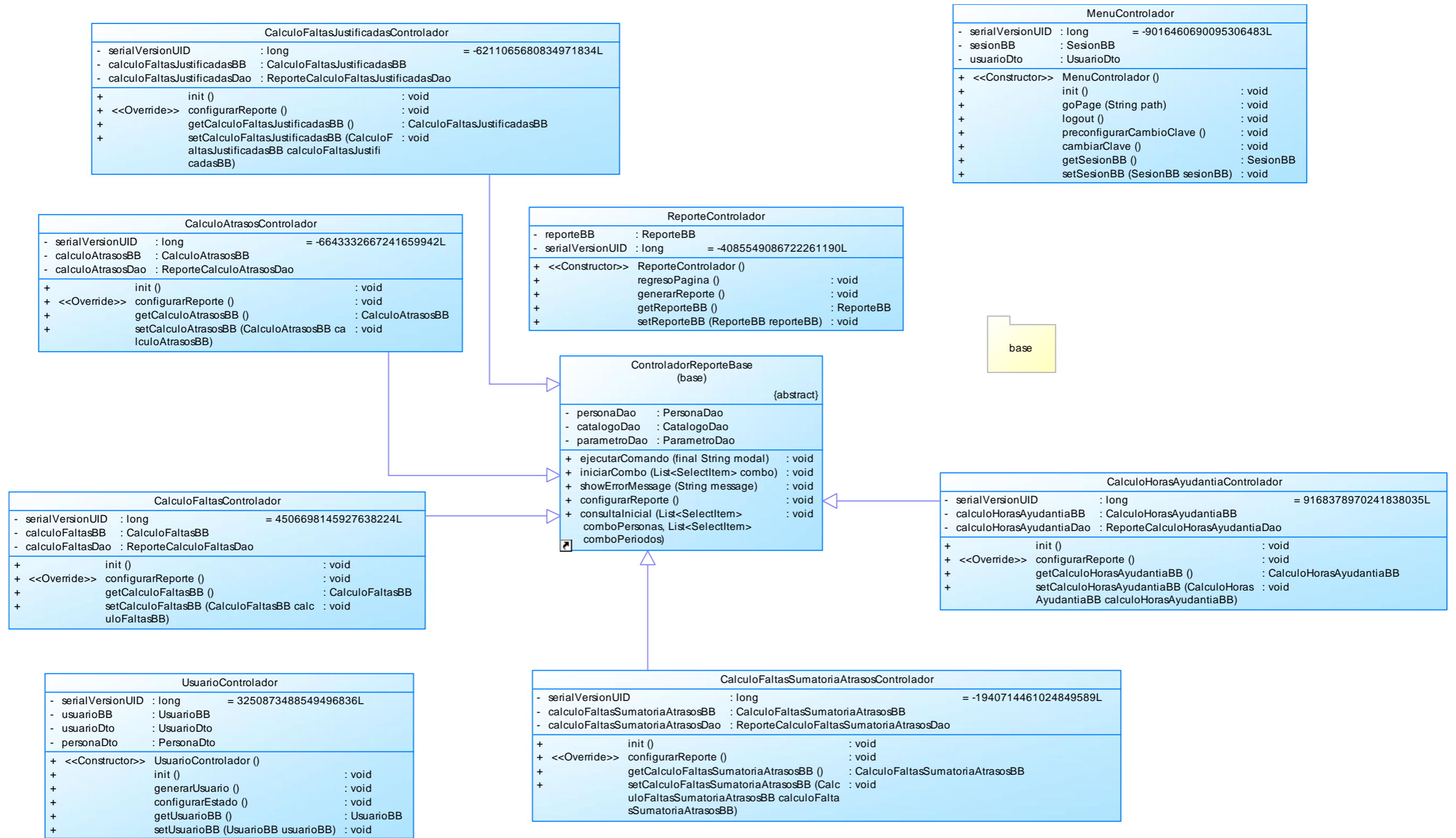


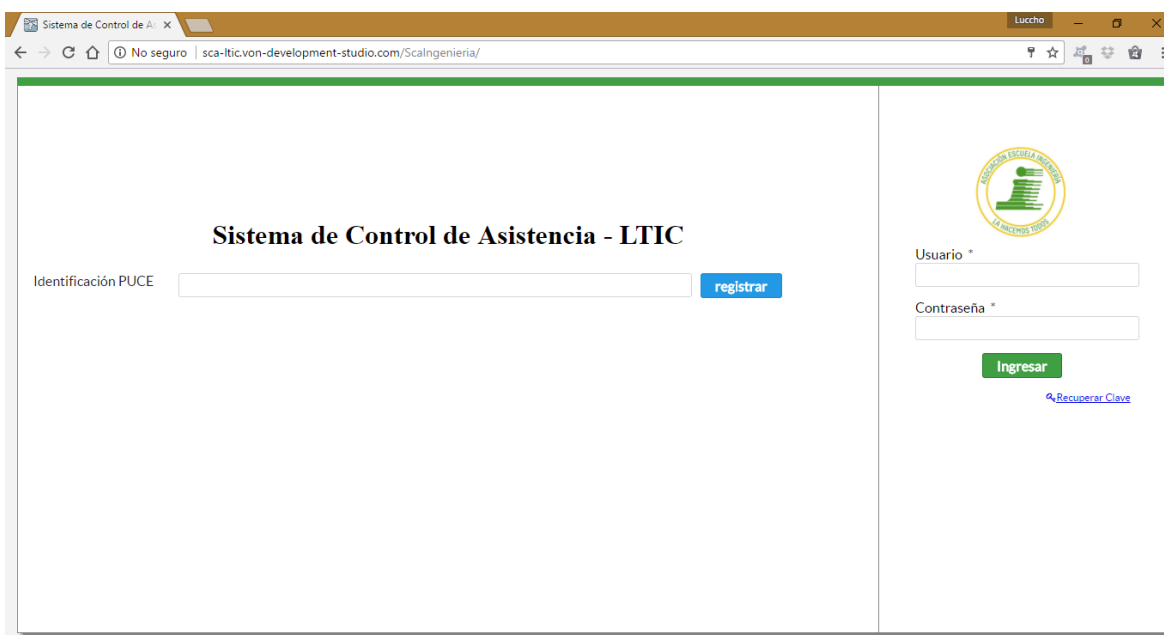
Ilustración 38. Diagrama de clases: métodos de la capa de presentación principal y de reportería. 26 de julio de 2016.

Anexo 7: Glosario de términos.

- SCA: Sistema de Control de Asistencia.
- Hardware: hace referencia las partes tangibles de un computador.
- Software: a diferencia del hardware, este corresponde a todo aquello que no se puede tocar en un computador tales como programas informáticos.
- Backup: respaldo de la información.
- SO: sistema operativo.
- Software libre: software libre de licencias.
- LTIC: Laboratorio de Tecnologías de Información y Comunicaciones.
- Framework: conceptos, prácticas y criterios estandarizados.
- ERS: acrónimo de Especificación de Requerimiento de Software.
- Trigger: también conocido como disparador y se encuentra relacionados con las tablas de una base de datos.
- Hashing: conocido como hash y su principal función es generar llaves que representa un archivo, documento, texto, registro, etc.
- Clustering: representa la técnica que combina sistemas y trabajan como un recurso consolidado.
- API: Application Programming Interface y sirve como un conjunto de funcionalidades las cuales pueden ser utilizados en distintos sistemas o aplicaciones.
- JDK: Java Development Kit, es el conjunto de herramienta para el desarrollo de aplicaciones y programas en JAVA.
- Skins: elementos gráficos que se pueden aplicar en un aplicativo
- Servlets: representan clases escritos en JAVA cuya función es aumentar las capacidades de un servidor.
- Browser: navegador web.
- Maven: repositorio de librerías para el desarrollo de software.
- AJAX: funciones que permiten realizar llamadas asíncronas a un servidor.
- GIT: repositorio de código fuente en distintos lenguajes de programación o archivos en general.

Anexo 8: Manual de usuario.

El Sistema de Control de Asistencia – LTIC tienen una interfaz muy amigable para el usuario final los cuales son segmentados en tres grupos. El docente encargado del área como administrador del sistema; los estudiantes becarios, los cuales tendrán acceso limitado al sistema y facilitaran una ayuda al docente encargado en temas administrativos como lo es registrar nuevas personas en el sistema y asignarlas en un horario específico; y los estudiantes ayudantes y voluntarios, su única interacción con el sistema es en la pantalla principal para registrar su asistencia como se muestra a continuación:



Si el usuario ingresó una credencial no registrada en el sistema, se le mostrará un mensaje erróneo:



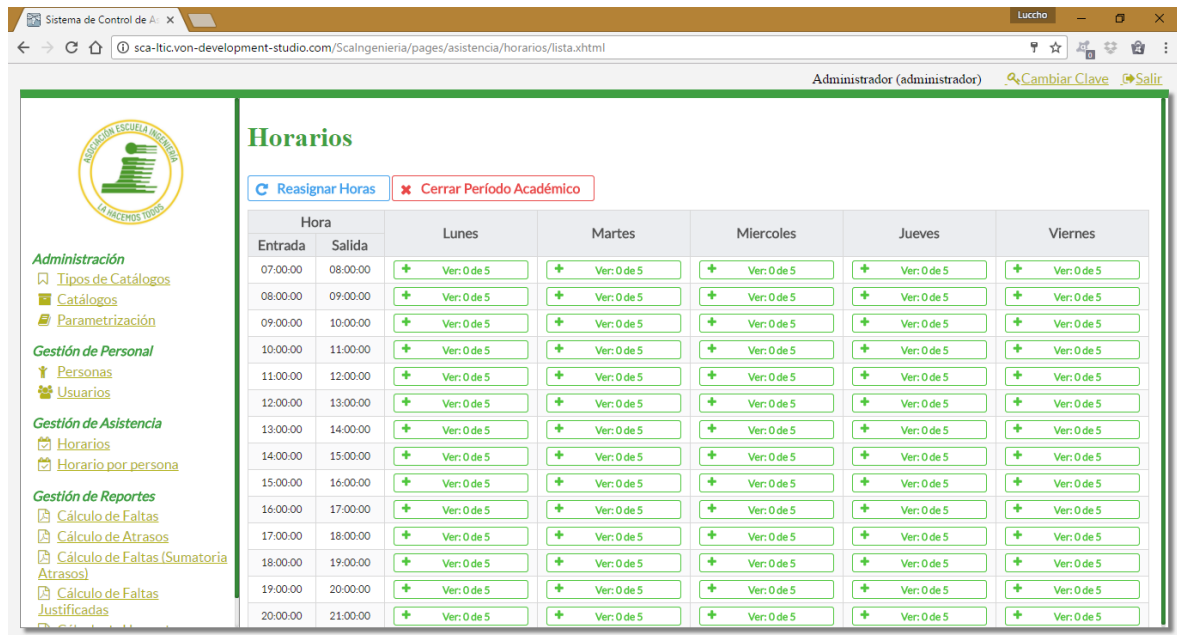
Si las credenciales son correctas, se indica la fecha y hora en la que se registró y de igual manera en la salida:

Luis García Castro. Hora de Entrada: Mon Mar 27 22:59:33 ECT 2017

Luis García Castro. Hora de Salida: Mon Mar 27 23:00:51 ECT 2017

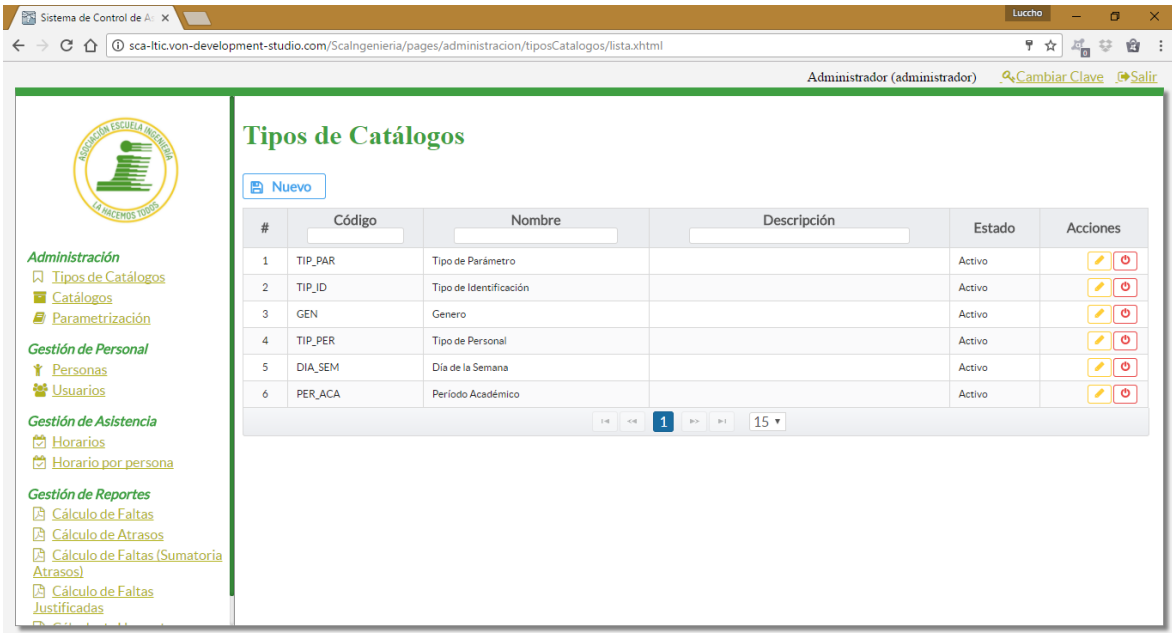
Si se realizó un ingreso correcto en el sistema con uno de los perfiles permitidos, el sistema mostrará como pantalla inicial el horario que se encuentra registrado hasta ese momento. Y además de mostrar en la sección de la izquierda una lista de menú de opciones.

En la sección superior se indicara el nombre y las credenciales de la persona que ingreso, además mostrará dos links que le permitirán al usuario cambiar su contraseña o salir del sistema.

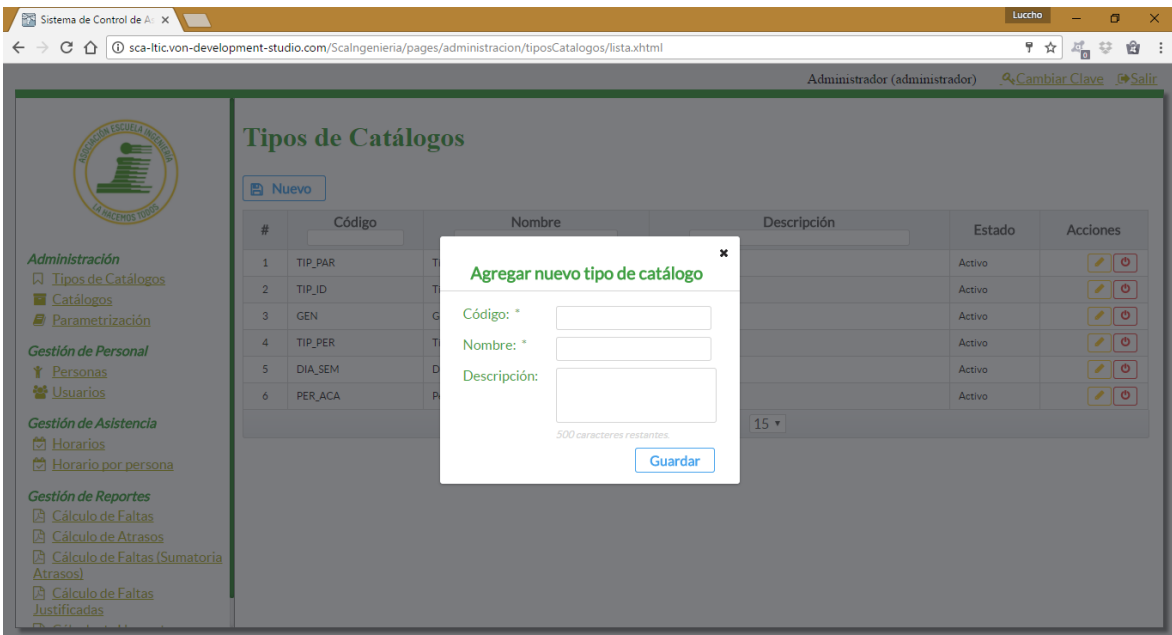


El sistema engloba 3 pantallas generales para la administración del sistema que corresponden a: Tipos de Catálogos, Catálogos y Parametrización:

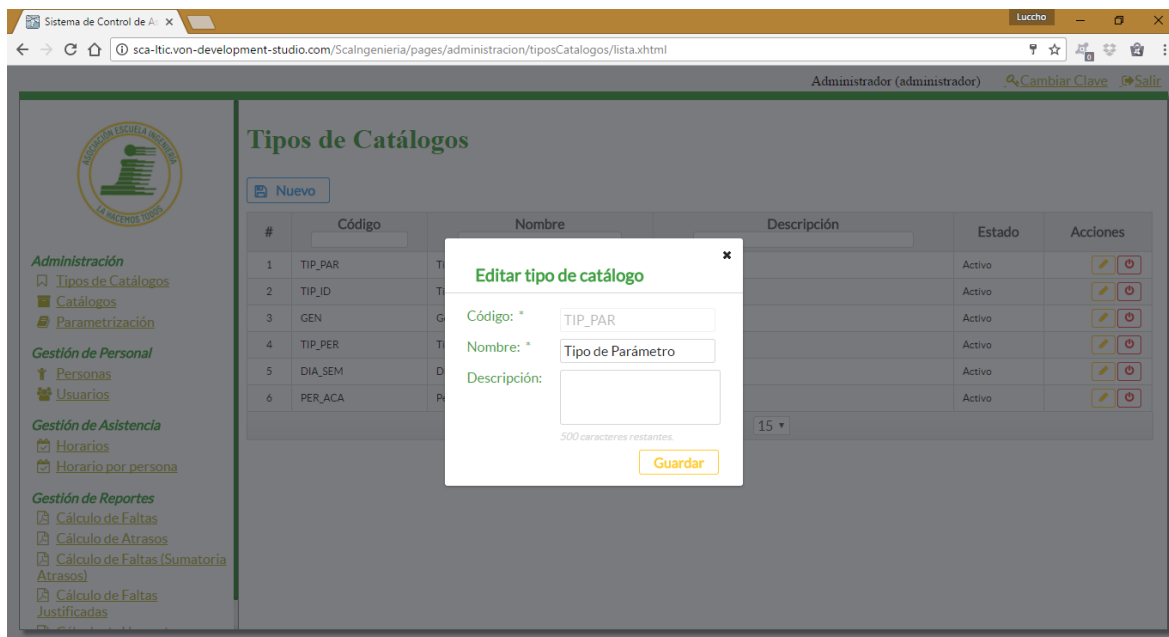
Como su palabra lo indica los tipos de catálogos engloban de forma general todos los catálogos que utilizara el sistema. La pantalla tiene un listado general de lo que se encuentra registrado en la base de datos. Tiene además tres botones que permiten agregar nuevos registros, editar registros existentes y activar/desactivar los registros.



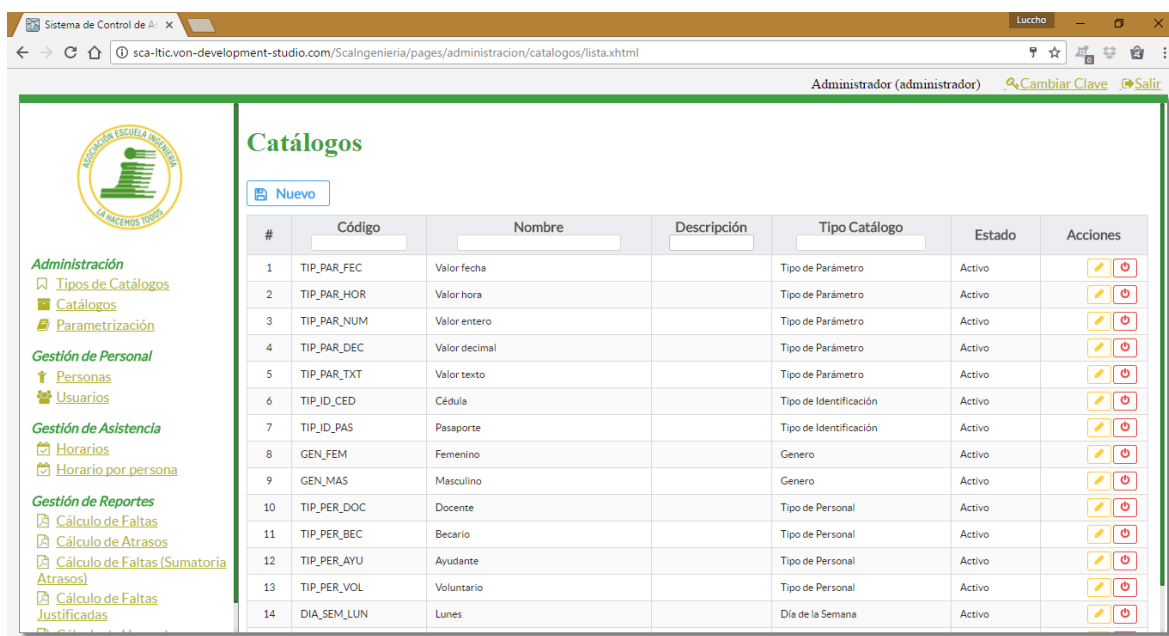
Los campos principales son un código y un nombre, y como campo opcional se encuentra una breve descripción.



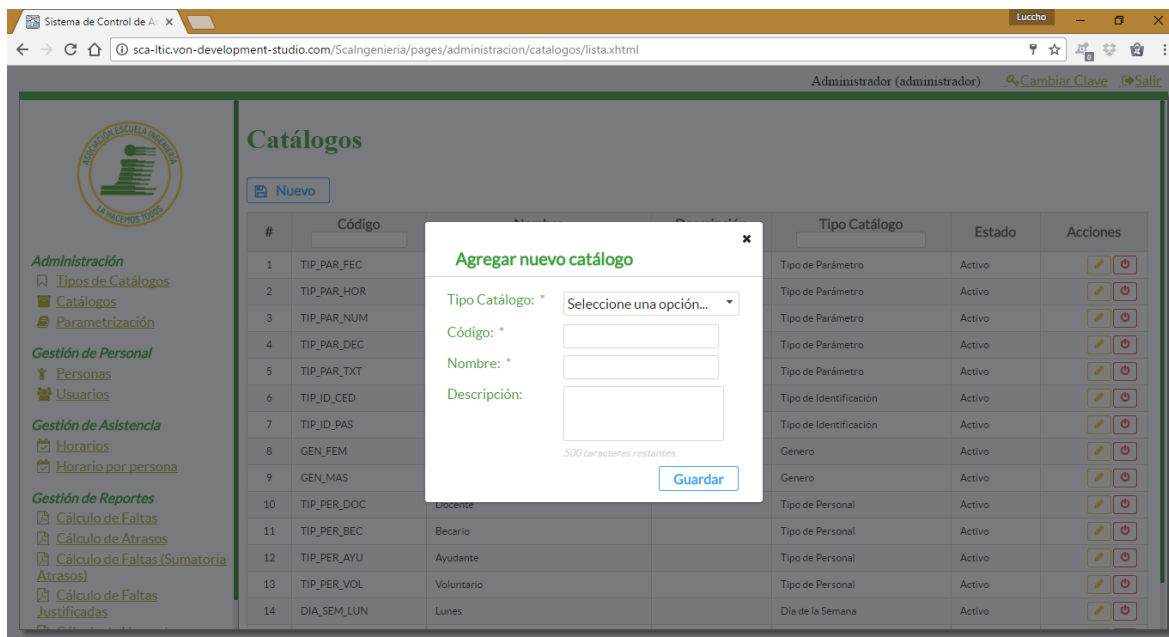
Cuando se seleccionó la opción de editar el registro, el sistema solo permite cambiar el nombre y la descripción.



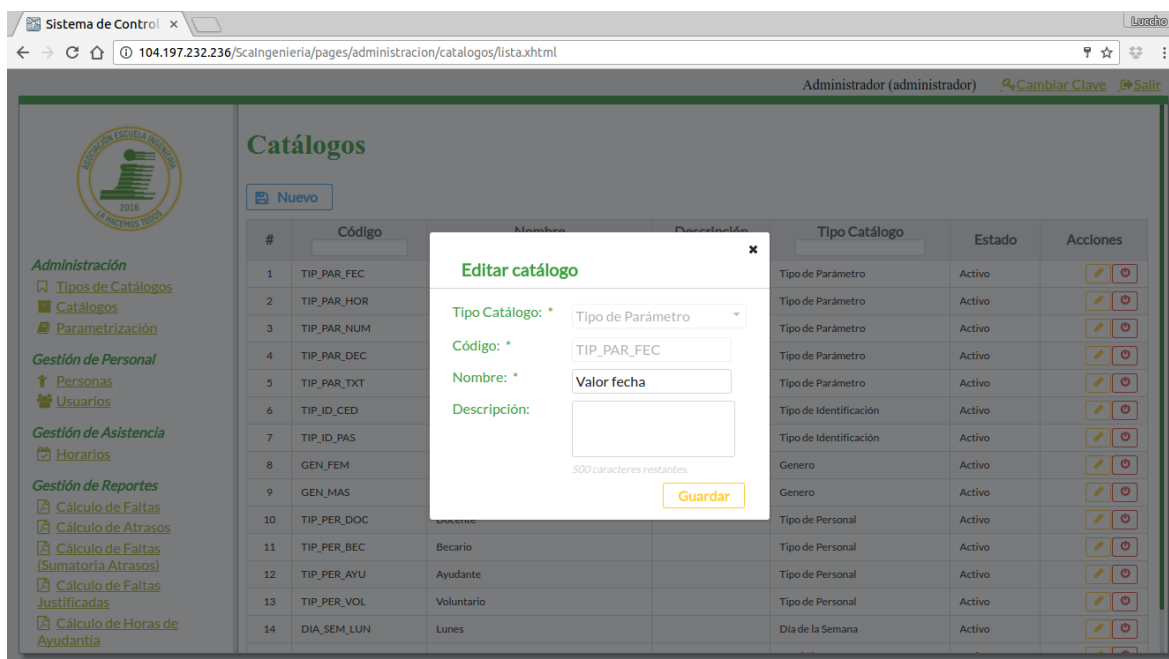
En los catálogos se tienen las características generales de las cuales se puede seleccionar en el sistema como por ejemplo tipos de identificación, género o el tipo de persona. La pantalla tiene un listado general de lo que se encuentra registrado en la base de datos. Tiene además tres botones que permiten agregar nuevos registros, editar registros existentes y activar/desactivar los registros.



Los campos principales son el tipo de catálogo, un código y un nombre, y como campo opcional se encuentra una breve descripción.

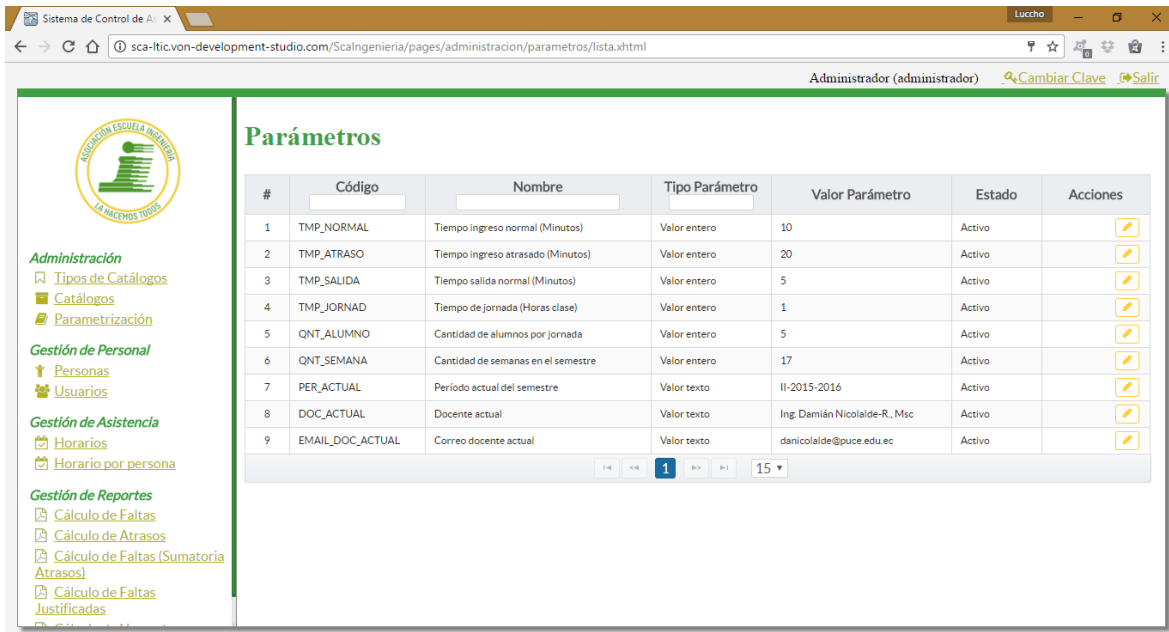


Cuando se seleccionó la opción de editar el registro, el sistema solo permite cambiar el nombre y la descripción.

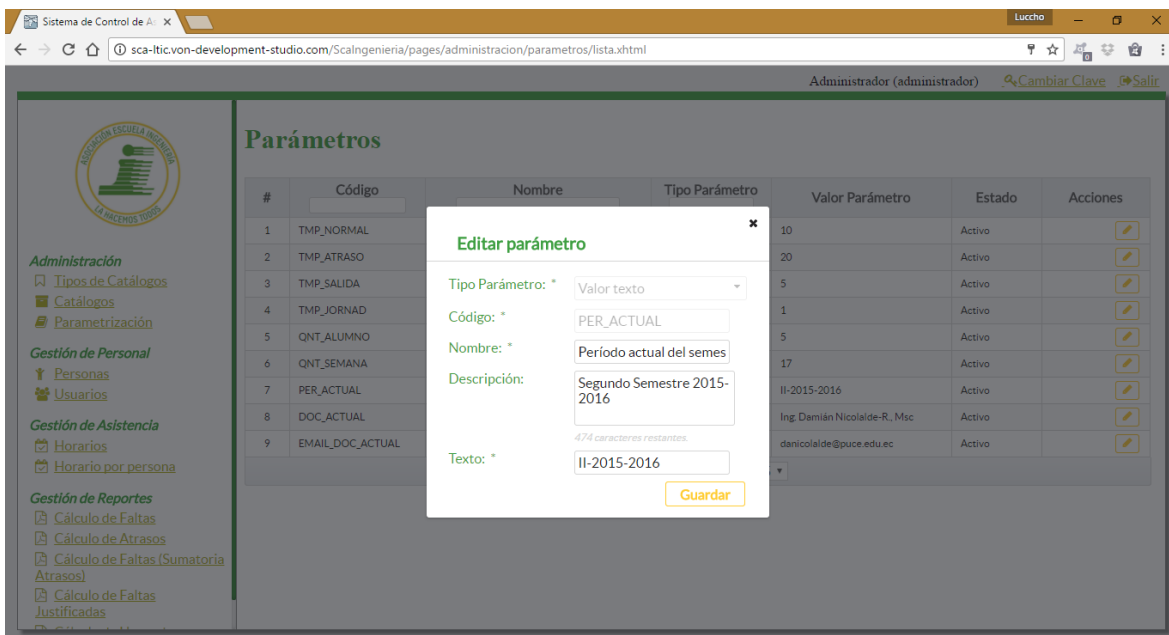


Para la parametrización, es recomendable realizarla previo al inicio del periodo académico además se recomienda que sus valores eviten ser cambiados en el transcurso del mismo porque puede interferir en la funcionalidad completa del sistema para ese periodo académico. Los parámetros más importantes que se debe dar mantenimiento de forma constante en el inicio del período académico correspondientes a la cantidad de semanas en el semestre, período actual del sistema y el docente actual encargado. Su correcto

mantenimiento permite poder generar los certificados y reportes necesarios de forma correcta.



Para su correcto mantenimiento se deben editar los parámetros creados.




El módulo de gestión de personal permite registrar a los alumnos en el sistema en los tres distintos perfiles que corresponden a: Becario, Ayudante y Voluntario. Es importante registrar de forma correcta la identificación de la PUCE, ya que al correo institucional se realizara envío de información relevante, en el caso de crear un usuario del sistema se enviara a este correo las credenciales de acceso.

Sistema de Control de A... X Luccho

sca-Itic.von-development-studio.com/Scalngenieria/pages/personal/personas/lista.xhtml

Administrador (administrador) Cambiar Clave Salir



Personas

Nuevo

#	Número de Identificación	Nombres	Apellidos	Usuario PUCE	Estado	Acciones
No records found.						

15

Administración

- Tipos de Catálogos
- Catálogos
- Parametrización

Gestión de Personal

- Personas
- Usuarios

Gestión de Asistencia

- Horarios
- Horario por persona


Gestión de Reportes

- Cálculo de Faltas
- Cálculo de Atrasos
- Cálculo de Faltas (Sumatoria Atrasos)
- Cálculo de Faltas Justificadas

Sistema de Control de A... X Luccho

sca-Itic.von-development-studio.com/Scalngenieria/pages/personal/personas/lista.xhtml

Administrador (administrador) Cambiar Clave Salir



Personas

Nuevo

Agregar nuevo tipo de catálogo

Tipo Identificación: * Seleccione una opción...

Número de Identificación: *

Nombre: *

Apellido: *

Genero: Seleccione una opción...

Fecha de Nacimiento: 01-01-1970

Usuario PUCE: *

Correo Institucional: *

Correo Personal:

Perfil: * Seleccione una opción...

Guardar

#	Número de Identificación	Usuario PUCE	Estado	Acciones
No records found.				

Administración

- Tipos de Catálogos
- Catálogos
- Parametrización

Gestión de Personal

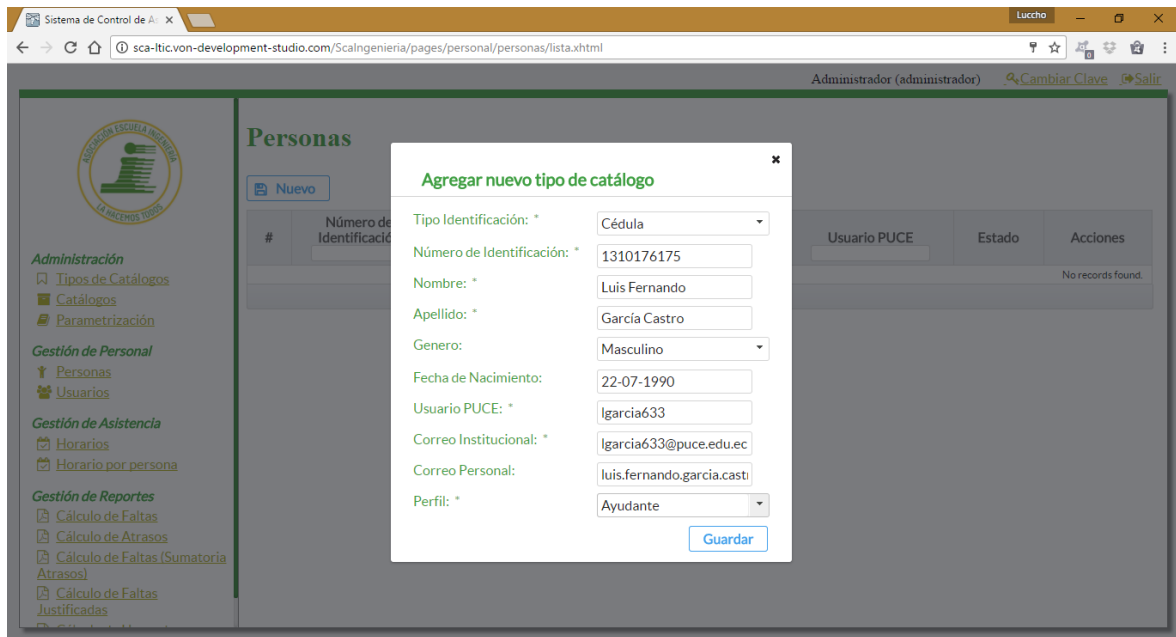
- Personas
- Usuarios

Gestión de Asistencia

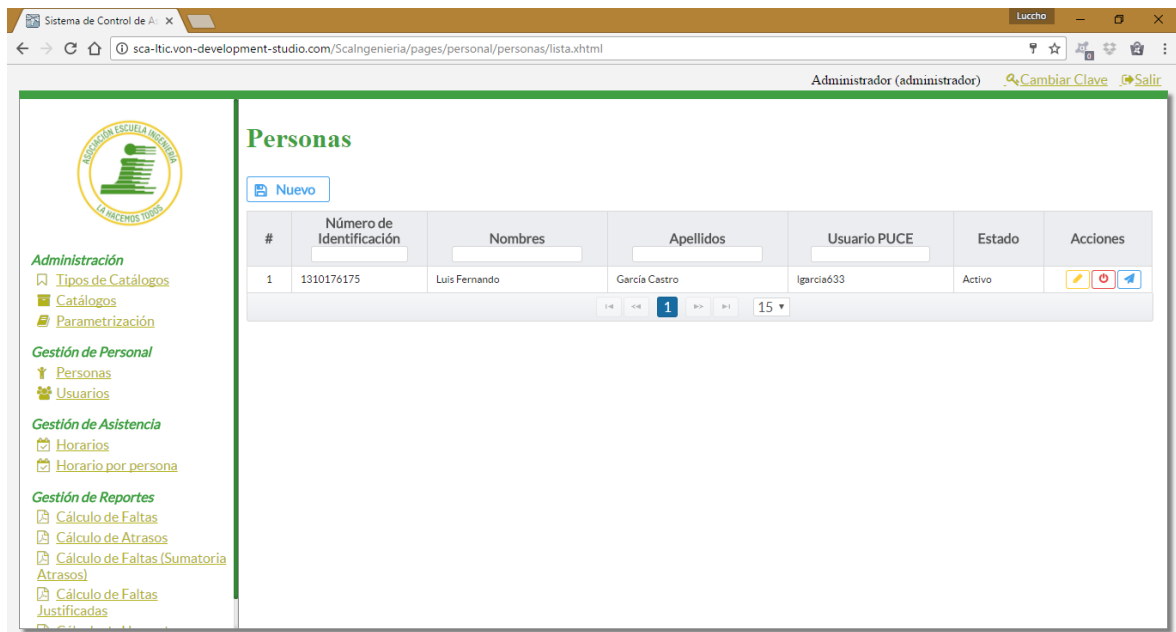
- Horarios
- Horario por persona

Gestión de Reportes

- Cálculo de Faltas
- Cálculo de Atrasos
- Cálculo de Faltas (Sumatoria Atrasos)
- Cálculo de Faltas Justificadas



La pantalla en general tiene un listado de las personas registradas, un botón para crear una nueva persona, de editar, actualizar estado y un botón para crear el usuario para el ingreso del sistema.



En la pantalla principal del sistema se permite asociar los alumnos a un horario específico. De tal manera, los alumnos tienen la obligación de registrar su asistencia en el sistema.

Sistema de Control de A... X Luccho

sca-lic.von-development-studio.com/Scalngenieria/pages/asistencia/horarios/lista.xhtml

Administrador (administrador) Cambiar Clave Salir

Horarios

Reasignar Horas

Hora		Jueves	Viernes
Entrada	Salida		
07:00:00	08:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
08:00:00	09:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
09:00:00	10:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
10:00:00	11:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
11:00:00	12:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
12:00:00	13:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
13:00:00	14:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
14:00:00	15:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
15:00:00	16:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
16:00:00	17:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
17:00:00	18:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
18:00:00	19:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
19:00:00	20:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5
20:00:00	21:00:00	+ Ver: 0 de 5	+ Ver: 0 de 5

Asociar alumnos al horario

Permisos

Disponible	Actual
	* Usuario 1 Prueba 1
	* Luis Fernando García Castro
	* Usuario 2 Prueba 2

Guardar

Administración

- Tipos de Catálogos
- Catálogos
- Parametrización

Gestión de Personal

- Personas
- Usuarios

Gestión de Asistencia

- Horarios
- Horario por persona

Gestión de Reportes

- Cálculo de Faltas
- Cálculo de Atrasos
- Cálculo de Faltas (Sumatoria Atrasos)
- Cálculo de Faltas Justificadas