Universidad de las Ciencias Informáticas

Facultad 2

Sistema Informático para el Control de Recursos Humanos y Medios Tecnológicos de los Laboratorios del centro ISEC

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Ariam Figueredo Rojas

Tutores: Ing. Yadira Benavides Zaila

Ing. Víctor Ernesto Marín Martínez

Ciudad de la Habana

2013

DECLARACIÓN DE AUTORÍA

Declaro ser autor del presente trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste se firma el presente a los ____ días del mes ____ del año _____.

Firma Autor: *Firma Tutores*:

_____	_____	_____
Ariam Figueredo Rojas	Ing. Yadira Benavides Zaila	Ing. Víctor E. Marín Martínez

AGRADECIMIENTOS

A mi madre por el apoyo incondicional que siempre me ha brindado, por entenderme y ser la guía de mi vida.

A mi hermano y toda mi familia por siempre estar pendiente de mí.

A mis amigos por soportarme.

DEDICATORIA

Dedicar este logro:

A mi madre, mi hermano y toda mi familia por el apoyo que me han brindado.

A mis amigos.

RESUMEN

En el centro para la Informatización de la Seguridad Ciudadana (ISEC), no se sigue un proceso de negocio único para la gestión de la información vinculada al control y supervisión de los recursos humanos y medios tecnológicos. Esto dificulta la toma de decisiones por parte de los directivos de dicho centro, ya que no poseen la información que le permita conocer el comportamiento del personal en cuanto al cumplimiento de la jornada laboral se refiere.

Para transformar la situación planteada se implementó SICREL, un sistema informático basado en una aplicación de escritorio desarrollado con las tecnologías de .NET y como gestor de base de datos MySQL. Éste permite gestionar información vinculada a los recursos humanos y medios tecnológicos presentes en los diferentes laboratorios productivos del centro anteriormente mencionado, así como la realización de consultas sobre el estado de ocupación de estos locales y los datos del personal perteneciente a cada uno de los proyectos productivos por parte de los directivos de ISEC. Por tal motivo se llevó a cabo un estudio de los sistemas informáticos existentes en el mundo, dirigidos a la gestión de estos recursos. El propósito del presente trabajo de diploma es el de describir estudio llevado a cabo y la solución propuesta, empleándose en la misma para dirigir el proceso de desarrollo las metodologías ágiles Scrum y XP.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La industria del software ha desempeñado, un importante papel en el desarrollo económico mundial, debido a la creciente necesidad y evolución de las tecnologías de la información y las comunicaciones. Cada país, partiendo de sus posibilidades y limitaciones, realiza acciones en cuanto a la actualización, asimilación y perfeccionamiento de las nuevas tecnologías. Para lograrlo se hace necesario la administración de recursos humanos y medios tecnológicos. Estos elementos son factores claves para el desarrollo y progreso de todo tipo de empresa, sobre todo las que se desenvuelven en la industria del software, permiten con su buen manejo alcanzar un nivel de calidad y productividad elevadas, marcando así significativas pautas en el desarrollo presente y futuro.

El paso del tiempo, ha demostrado que la mayoría de las empresas, deben tomar medidas preventivas en la parte administrativa para conservar la salud del organismo, evitar prácticas ineficientes, mejorar los métodos y desempeño de esta, obteniéndose como principal resultado una disminución de los costos económicos y de tiempo.

Entre los nuevos retos del país, se encuentran el control y utilización racional de los recursos disponibles, tanto humanos como tecnológicos, que garantice un buen desempeño de la economía. El proceso de gestión de recursos humanos y medios tecnológicos, en instituciones del territorio nacional, requiere un gran volumen de documentación en formato físico, entre reportes, inventarios e informes. Las dificultades para el control de estos procesos, trae como resultado la ocurrencia de pérdidas y duplicación innecesaria de la información, por lo que el acceso a esta no es óptimo. Además, problemas en su manejo, imposibilitan la toma de decisiones por parte de los directivos de estas instituciones.

Una factible solución para erradicar estos percances, es la informatización de los procesos que intervienen en el control de la gestión de los recursos humanos y medios tecnológicos. En esta área, varias empresas pertenecientes a la industria del software nacional, tales como HavanaSoft, perteneciente a la provincia de la Habana, han empezado a dar los primeros pasos.

En la Universidad de las Ciencias Informáticas (UCI), no se cuenta con un único proceso de negocio para la gestión de la información relacionada con los recursos humanos y medios tecnológicos. Al no estar definidos estos procesos se le dificulta a directivos de los centros

productivos tener control sobre el capital humano existente, así como de las tecnologías y el estado de las mismas.

El centro para la Informatización de la Seguridad Ciudadana (ISEC), perteneciente a la Facultad 2, de dicha universidad, no se encuentra exento de esta situación. Los recursos humanos y medios tecnológicos de este centro se hallan distribuidos en diferentes áreas del campus universitario. La supervisión, por parte de los jefes de centro, del aprovechamiento de la jornada laboral y del estado de los medios tecnológicos, está obstaculizada por la dispersión geográfica presente.

Se desconoce también la cantidad de horas reales ocupadas en laboratorio, puesto que el mecanismo de entrada y salida implantado, se limita solo a libros de firmas para especialistas y adiestrados, no contemplándose el control de asistencia de los estudiantes que laboran en estos locales. El acceso de personal ajeno a los laboratorios, así como la entrada y salida de medios tecnológicos fuera de los asignados a la actividad productiva, como es el caso de discos duros externos y laptops personales, no se registran, posibilitando esto que se vea comprometida la integridad de la información que se maneja en los proyectos productivos. Además la información sobre el estado de los puestos de trabajo, depende de reportes diarios que realizan los técnicos de laboratorio y se llevan en formato físico.

Estos datos no se recopilan de forma accesible a la dirección del centro, ni por líderes de proyecto que son quienes establecen el tiempo de máquina de los estudiantes. La dirección del centro ISEC, necesita mantener una supervisión permanente sobre el estado de ocupación de los laboratorios, la disponibilidad de los puestos de trabajo, así como el control de la asistencia del personal que labora en ellos.

Por la situación anteriormente descrita, se identificó como **problema a resolver**: ¿Cómo facilitar la obtención de información, referente a la utilización de los medios tecnológicos y el cumplimiento de la jornada laboral de trabajadores y estudiantes, a los directivos del centro ISEC?

En consecuencia con lo anterior se determinó como **objeto de estudio** los procesos de gestión de la información vinculada a los recursos humanos y medios tecnológicos. Enmarcando el **campo de acción** sobre las herramientas informáticas para el control y supervisión de recursos humanos y medios tecnológicos. Teniéndose como **objetivo general**: desarrollar un sistema

informático para el control de recursos humanos y medios tecnológicos de los laboratorios del centro ISEC.

Los **objetivos específicos** serán:

- Caracterizar la información a gestionar por el sistema informático.
- Definir la arquitectura de la solución.
- Implementar un sistema informático que permita la gestión de información de recursos humanos y medios tecnológicos de los laboratorios del centro ISEC.

Para cumplir los objetivos trazados servirán de guía las siguientes **tareas de investigación**:

- Estudio del estado del arte de sistemas informáticos similares al que se quiere implementar, para identificar, de las características que estos poseen, cuáles se podrían adaptar al sistema que se presenta como solución.
- Estudio y selección de metodologías y herramientas a utilizar para el desarrollo del sistema informático.
- Entrevistas al cliente para realizar el levantamiento de requisitos.
- Análisis, diseño e implementación de una aplicación de escritorio para la gestión y supervisión de información referente a recursos humanos y medios tecnológicos, de los laboratorios del centro ISEC.
- Realización de pruebas a la aplicación funcional, para detectar posibles errores y corregirlos.

Se espera como resultado una aplicación informática que facilite el proceso de gestión de la información referente a los recursos humanos y medios tecnológicos de los laboratorios pertenecientes al centro ISEC.

Esta investigación se encuentra estructurada en diferentes capítulos, ellos son:

Capítulo 1 Fundamentación Teórica: presenta el estudio realizado sobre el estado del arte de sistemas relacionados con el control de recursos humanos y medios tecnológicos. También se describen herramientas y metodologías a utilizar para el desarrollo del sistema.

Capítulo 2 Análisis y Propuesta de Solución: se realiza un análisis de la gestión de información de los recursos humanos y medios tecnológicos en el centro ISEC, se describe el sistema informático a implementar, así como sus funcionalidades.

Capítulo 3 Diseño del Sistema: se detalla la estructura de la arquitectura empleada, describiéndose los principales patrones arquitectónicos y de diseño, así como el modelo entidades de datos.

Capítulo 4 Implementación y Prueba: se muestran las tareas de implementación, detallándose los estándares de programación empleados. También se describen las diferentes pruebas aplicadas al sistema informático desarrollado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo, se presenta el estudio del estado del arte realizado, analizándose algunas de las aplicaciones informáticas utilizadas, tanto a nivel nacional como internacional, enfocadas en la gestión de los recursos humanos y medios tecnológicos. También se describen las principales herramientas y metodologías a emplear en el desarrollo de la propuesta de solución.

1.1. Sistemas de gestión de recursos humanos y medios tecnológicos

A continuación se refleja el estudio realizado, respecto a sistemas informáticos existentes dirigidos a la automatización de procesos vinculados a la gestión de Recursos Humanos y Medios Tecnológicos.

El **Sistema de Gestión de Laboratorios** de la Escuela Politécnica Superior en la Universidad de Alicante en España, constituye una aplicación web cuya interfaz fue implementada en PHP y JSP, utilizando para la gestión de identidad y perfiles de usuario el directorio Netscape Directory Server de Sun y como gestor de base de datos MySQL.

Permite la gestión y asignación de turnos de práctica para las asignaturas de cada departamento en dependencia de su carga docente. Emite estadísticas sobre el control de la asistencia a los laboratorios, a las cuales el estudiantado también tiene acceso para revisar y controlar a qué prácticas ha asistido.

Brinda la posibilidad de acceder al inventario de cada laboratorio el cual ofrece información sobre hardware, software y medios de apoyo a la docencia. Permite ver el estado de las distintas instalaciones de los equipos así como el tiempo que tardó en realizarse. Se cuenta en el sistema con la gestión del inventario, en la cual se verifica la contabilización del hardware disponible, la generación estadística de incidencias y la gestión de las aplicaciones de los laboratorios, esto brinda la posibilidad de emitir informes en los cuales se plasma el estado de los laboratorios y estadísticas de su uso para su consulta por parte de los alumnos. [1]

Este sistema, a pesar de integrar elementos a tener en cuenta para el desarrollo de la propuesta de solución, no resuelve el problema planteado en su totalidad pues no permite a los directivos supervisar el estado de ocupación de estos locales, ni la gestión de tiempos de máquinas.

ASSETSNS es un Sistema de Gestión Integral estándar y parametrizado, siendo una aplicación informática cliente-servidor programada en Visual Basic 6.0 y Microsoft SQL Server 2000 que dispone de interfaces amigables y utiliza adicionalmente Crystal Reports 7.0 para la generación de reportes, que consta de varios módulos. Entre estos módulos se encuentran el de Inventario y Activos Fijos que permite controlar la existencia, disponibilidad y reserva de los diferentes recursos de almacenes, realizar conteos físicos de los productos y hacer ajustes en los precios de costo de los artículos almacenados. Por su parte el módulo Activos Fijos gestiona la ubicación física de los medios básicos, el control de alquiler, alta, baja, préstamo y traspasos hacia otras áreas dentro y fuera de la entidad.

También presenta un módulo de recursos humanos que posibilita el control íntegro de los recursos laborales: empleados y estructura organizativa de la entidad. Asimismo, proporciona opciones de seguridad que le permiten limitar el acceso a los diferentes procesos del sistema de acuerdo con el perfil de cada usuario. [2]

Pese a las características anteriormente mencionadas, ASSETSNS no presenta elementos que permitan el control de entrada y salida de personal y medios tecnológicos ajenos a las entidades donde se emplea.

RH Expert es un sistema experto para la gestión de los Recursos Humanos, desarrollado por la empresa HavanaSoft; de la Ciudad de la Habana. En dicho sistema, se desarrollaron un grupo de aplicaciones cliente-servidor que integran los programas que se utilizan diariamente para la gestión de los Recursos Humanos. Este sistema tiene un alcance general, es decir, puede ser usado en cualquier tipo de entidad sin importar el sector al que pertenezca. Permite generar reportes y estadísticas del personal que labora en la empresa [3]. Se divide en varios módulos entre ellos se tienen:

- ***Control de Personal***: Permite la adaptación a diferentes convenios y tipos de horarios, la gestión de la presencia del personal a través de un calendario visual, confeccionar un expediente exportable de la trayectoria laboral del trabajador, reportes con información de los datos de los trabajadores a nivel de áreas.
- ***Control de Asistencia***: Permite el control del archivo de los datos personales, a través de un Archivo de Entrada que gestiona la entrada inicial al sistema, un Archivo Activo para la gestión del personal activo y un Archivo Pasivo que gestiona el historial de trabajadores de la entidad. También gestiona el expediente personal del trabajador, administrando los datos

personales, la integración, la profesión y las publicaciones. Brinda la posibilidad de gestionar el cierre de período con el objetivo de consolidar la información de la asistencia hacia la entidad superior.

A partir del estudio realizado se determinó que RH Expert presenta importantes características para el control de Recursos Humanos, tales como: asistencia, control de entrada y salida, reportes y estadísticas; pudiéndose emplear como una herramienta, por los directivos, para apoyar la toma de decisiones. Sin embargo, este sistema no cumple con todos los requerimientos demandados, pues no brinda la posibilidad de gestionar tiempos de máquina ni la entrada y salida de personal ajeno.

1.2. Metodología de desarrollo del software

El desarrollo de software no es una tarea fácil, por lo que la selección de una adecuada metodología para dirigir este proceso requiere gran atención, pues de ella puede depender el éxito del producto final, así como el cumplimiento de las expectativas que se planteen por parte del cliente [4]. Existen numerosas metodologías para llevar a cabo el desarrollo de software, dividiéndose estas en dos grupos: las pesadas y las ágiles.

Las metodologías pesadas o tradicionales (como también se le conocen), invierten una significativa cantidad de tiempo y esfuerzo en preparación y planificación para conseguir que el desarrollo del software se ejecute en el menor tiempo posible y con los menores incidentes, estableciendo rigurosamente actividades involucradas, artefactos que se deben producir, y herramientas a utilizar. Por ello estas metodologías no facilitan el desarrollo rápido de aplicaciones, pues requieren de una excesiva burocracia documental y no se adaptan a los cambios por lo que no son métodos adecuados cuando se trabaja en un entorno donde los requisitos pueden variar fácilmente. [5]

Por otro lado las metodologías ágiles, no se centran tanto en la preparación y planificación, invirtiendo la mayor parte del tiempo en el desarrollo, la realización de pruebas y la rectificación. Estas metodologías centran su objetivo en el valor del resultado, reducción del tiempo de desarrollo, agilidad y flexibilidad teniendo como efecto la gestión ágil de proyectos, que no se formula sobre el concepto de anticipación (requisitos, diseño, planificación y seguimiento) sino sobre el de adaptación (visión, exploración y adaptación), esto ofrece garantías a las demandas principales de la industria. Brindan mayor valor al individuo, a la colaboración entre el cliente y el equipo de desarrollo, definiendo un proceso incremental de desarrollo con iteraciones cortas. Esto demuestra su efectividad en proyectos con requisitos cambiantes. [6]

Para dar respuesta a la problemática planteada se hace necesario desarrollar y construir un producto a la par de la investigación y del descubrimiento de los requisitos, de tal manera que se logre una mejor adaptabilidad a los posibles cambios que pudiesen presentarse. Para suplir esta necesidad, luego de realizado el análisis de los tipos de metodologías de desarrollo de software existentes, se decide el empleo de metodologías de desarrollo ágil, sobre las cuales se realiza un estudio, el cual se presenta a continuación, para seleccionar la más factible a la hora de desarrollar la solución en pos de dar respuesta al problema.

1.2.1. Scrum

Scrum es una metodología que define un marco para gestión de proyectos de software basándose en la adaptación continua a las circunstancias de la evolución del proyecto. [4] A partir del concepto o visión de la necesidad del cliente, el empleo de esta metodología permite la construcción de un sistema informático de forma incremental a través de iteraciones breves, las cuales producen una parte completamente terminada y operativa. Con el empleo de esta metodología se puede controlar de forma empírica y adaptable la evolución del proyecto pues para ello utiliza prácticas de la gestión ágil tales como revisión de las iteraciones, desarrollo incremental, desarrollo evolutivo y colaboración. Estos elementos sirven de ayuda para organizar a las personas y el flujo de trabajo. Permite una mejor adaptabilidad a posibles cambios que puedan ocurrir en los requerimientos del sistema a lo largo de su desarrollo, en intervalos cortos y regulares, adecuando el producto que se esté construyendo, a las necesidades del cliente. [7]

Scrum presenta tres artefactos:

- **Pila del producto:** es una lista de los requisitos del cliente, que puede ir creciendo y evolucionando, desde la visión inicial del producto y a lo largo de su desarrollo.
- **Pila de tareas de las iteraciones:** es la lista de tareas que se realizan durante el desarrollo de cada iteración, permitiendo generar el incremento previsto.
- **Incremento:** son los resultados obtenidos al finalizar cada una de las iteraciones.

Dentro de las principales ventajas del empleo de Scrum se tiene que el valor que aporta cada requisito del proyecto y cuándo se espera que esté completado, es indicado y establecido a través de las expectativas del cliente; permite utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo. También al término de cada iteración, el cliente puede aprovechar la parte del producto, completada hasta ese momento, para hacer pruebas de concepto con usuarios o consumidores y tomar decisiones en función del resultado

obtenido. Vale destacar que el riesgo al que se enfrenta el equipo, está limitado a los requisitos que se puede desarrollar en una iteración. Todo ello trae consigo que de manera regular se vaya mejorando y simplificando la forma de trabajar. [7]

Otra de las características de Scrum es que permite la integración de elementos de otras metodologías de desarrollo ágil, como es el caso de eXtreme Programming o XP como es comúnmente conocida. A continuación se detallan las particularidades de dicha integración.

1.2.2. Integración de Scrum y XP

XP se basa en la simplicidad, la comunicación y la reutilización continua de código, uno de sus principales objetivos es el de potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software [8]. Es una metodología que sirve de guía para el proceso ingenieril, además que facilita una documentación discreta y dinamismo en el desarrollo del sistema informático propuesto. Presenta un desarrollo dirigido por pruebas, que es una técnica utilizada en el diseño e implementación de software. Con su empleo minimiza el número de defectos que pudiesen surgir durante la construcción del software, lográndose que este sea altamente reutilizable y adaptable ante futuros cambios.

Permite, mediante la inclusión de prácticas de refactorización, en las tareas de diseño y codificación, reestructurar el sistema sin cambiar su comportamiento para remover duplicación, mejorar la comunicación y simplificar o añadir flexibilidad [9]. Ello implica, en la aplicación de Scrum para el desarrollo de la solución, que se eviten problemas de degradación del sistema o de la arquitectura por la evolución continua del producto.

La integración de estas características de XP a Scrum ofrece una estrategia tecnológica. Con la introducción de los procedimientos ágiles presentes en ambas metodologías se puede mejorar el desarrollo de la actividad productiva.

Son metodologías especialmente indicadas para su uso en proyectos de pequeños equipos de trabajo, rápido cambio de requisitos imprecisos y volubles, orientándose a una entrega rápida de resultados y una alta flexibilidad. La utilización en conjunto de estas ayuda a fomentar el trabajo, encaminado en una dirección común con un objetivo claro, permitiendo seguir el avance de las tareas a realizar, viéndose así el progreso del trabajo. Ofrece al cliente la posibilidad de comprobar sistemáticamente si se van cumpliendo sus expectativas, logrando así estimular y comprometer al mismo con el proyecto, pues irá notando el crecimiento del

producto. El hecho de completar los requisitos, en función del valor que aportan al cliente, minimiza la probabilidad de ocurrencia grandes cambios en la evolución del sistema. También brinda al equipo de desarrollo una soberanía en la planificación del trabajo en cada iteración. [10]

La integración entre estas dos metodologías proporciona un desempeño efectivo en el desarrollo del sistema, dadas las características que ambas poseen: Scrum se enfoca en las prácticas de organización y gestión, mientras que XP se centra más en las prácticas de programación. Esa es la razón por la que funcionen tan bien juntas, pues tratan de áreas diferentes y se complementan entre ellas [10]. Esto permite obtener mejores resultados en la organización y planificación del trabajo, contribuyendo a la entrega de un producto que satisfaga tanto al cliente como al equipo de desarrollo, haciendo por ende viable el empleo de dicha integración, para dirigir así el proceso de desarrollo de la solución al problema planteado.

1.3. Lenguaje de programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

1.3.1. C#

C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo haciéndolo sencillo y moderno, permitiendo que cualquiera que esté familiarizado con estos lenguajes pueda adaptarse fácilmente a su uso, permitiéndoles ser productivos. Proporciona seguridad de tipos y está orientado a objetos. Fue diseñado para superar algunos problemas de Java. En particular, la diferencia sustancial entre valores y objetos y la carencia de delegados que facilitaran la implementación. C# ha evolucionado mucho desde su aparición, pero mantiene una coherencia en el diseño que lo hace fácil de aprender. Aunque es un lenguaje con herencia simple, implementación de interfaces y memoria con recolección automática, como Java, se diferencia de éste en numerosos aspectos importantes. C# integra eficazmente varios conceptos de la programación funcional, como las funciones anónimas y las clausuras.

Evita la utilización del patrón de consulta y asignación de valor a campos (*get* y *set*) propia de Java mediante las denominadas propiedades. Finalmente cabe advertir que la implementación

de tipos genéricos en C# es sólida, pues conserva información de tipos y distingue entre valores y objetos en el parámetro de tipo.

C# está estandarizado y su definición se encuentra en *Standard ECMA*[1] *334 C# Language Specification*[2]. El código creado mediante C# se compila como código administrado, lo cual significa que se beneficia de los servicios de *CLR*[3] y en conjunto con .NET Framework, proporcionan interoperabilidad entre lenguajes, recolección de elementos no utilizados, mejora de la seguridad en el tratamiento de tipos y mayor compatibilidad entre versiones. Entre los principales elementos que presenta se encuentra que posee un ambiente unificado y elegante, el manejo de errores está basado en excepciones e impide la utilización de variables no inicializadas. [11]

Fundamentación de la selección del lenguaje de programación

Para el desarrollo de la solución se decide utilizar el lenguaje de programación C# en su versión 4.0, puesto que permite llevar a cabo tareas comunes que implican tipos genéricos, interoperabilidad heredada y trabajar con modelos de objetos dinámicos sencillos. Además ofrece soporte a toda la tecnología Windows Foundation con las cuales se trabaja. La jerarquía de llamadas que presenta, permite navegar por el código, propiciando un mejor entendimiento del flujo de este, evaluando los cambios que se realizan en él. Las nuevas funcionalidades de esta versión permiten a los programadores alcanzar un nivel avanzado debido a facilidad de su código. Partiendo de las características anteriormente mencionadas y por ser parte de la línea de desarrollo del centro ISEC, en aplicaciones de escritorio, se emplea este lenguaje de programación para la implementación de la solución.

1.4. Plataforma de desarrollo Microsoft .NET

Se utiliza como plataforma de desarrollo Microsoft .NET, esta permite la ejecución de aplicaciones, brindando herramientas y servicios necesarios para desarrollar modernas aplicaciones, proporcionando mecanismos robustos y seguros, haciendo óptima la ejecución de las mismas.

[1] *European Computer Manufacturers Association*: organización internacional basada en membrecías de estándares para la comunicación y la información

[2] Este estándar internacional especifica la forma y establece la interpretación de programas escritos en el lenguaje de programación C#.

[3] *Common Language Runtime*: Entorno en tiempo de ejecución que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo.

Ofrece un entorno de ejecución con máquina virtual para un lenguaje intermedio de máquina propio. Diferentes lenguajes se traducen a ese lenguaje de máquina y un compilador genera el código nativo, que es lo que realmente se ejecuta.

.NET sigue el estándar *ECMA 335 Common Language Infrastructure*[4] (CLI). La implementación de Microsoft del CLI se conoce por CLR. Hay una implementación libre de CLI desarrollada por Novell[5] conocida como Mono[6]. Acompaña al entorno un conjunto de librerías gigantesco, aspecto en el que .NET va significativamente por delante de Mono.

Los componentes principales de esta plataforma son:

- Un entorno de ejecución de aplicaciones, que es un componente de software cuya función es la de ejecutar las aplicaciones .NET e interactuar con el sistema operativo ofreciendo sus servicios y recursos.
- Un conjunto de bibliotecas de funcionalidades y controles reutilizables, con una enorme cantidad de componentes ya programados listos para ser consumidos por otras aplicaciones.
- Un conjunto de lenguajes de programación de alto nivel, con sus respectivos compiladores, que permiten el desarrollo de aplicaciones sobre dicha plataforma.
- Un conjunto de utilitarios y herramientas de desarrollo para simplificar las tareas más comunes del proceso de desarrollo de aplicaciones.

La plataforma Microsoft .NET está completamente basada en el paradigma de Orientación a Objetos además que es multilenguaje: esto permite codificar aplicaciones sobre esta plataforma sin necesidad de tener que aprender un único lenguaje en específico de programación de alto nivel, de los cuales ofrece una amplia lista de opciones. Microsoft .NET permite el desarrollo de aplicaciones empresariales de misión crítica, entendiéndose por esto que permite la creación y ejecución de aplicaciones de porte corporativo que sean críticas para la operación de tipos variados de organizaciones. Si bien también es sugestivo para desarrolladores no profesionales, estudiantes y entusiastas, su ventaja más significativa radica en su capacidad para soportar aplicaciones grandes y complejas.

[4] Este estándar internacional define el lenguaje común de infraestructura (CLI) en el cual las aplicaciones escritas en múltiples lenguajes de alto nivel pueden ser ejecutados en diferentes ambientes del sistema sin la necesidad de volver a escribir esas aplicaciones para tener en cuenta las características únicas de esos ambientes.

[5] Compañía de origen estadounidense dedicada al software.

[6] Entorno de desarrollo de código abierto que permite a los desarrolladores ejecutar aplicaciones .NET en Linux, Solaris, Unix y Mac platforms.

.NET fue diseñado de manera tal que permite proveer un único modelo de programación, uniforme y consistente, para todo tipo de aplicaciones (ya sean de formularios Windows, de consola, aplicaciones Web o aplicaciones móviles, por solo citar algunas) y para cualquier dispositivo de hardware. Esto representa un gran cambio con respecto a las plataformas anteriores a .NET, las cuales tenían modelos de programación, bibliotecas, lenguajes y herramientas distintas según el tipo de aplicación y el dispositivo de hardware.

No sólo se integra fácilmente con aplicaciones desarrolladas en otras plataformas Microsoft, sino también con aquellas desarrolladas en otras plataformas de software, sistemas operativos o lenguajes de programación. Para esto hace un vasto uso de estándares globales que son de uso extensivo en la industria, algunos ejemplos de estos estándares son *XML*[7], *HTTP*[8], *SOAP*[9], *WSDL* [10]y *UDDI*[11]. [12]

1.5. Microsoft .NET Framework 4.0

Microsoft .NET Framework es el marco de trabajo y ejecución de runtime que administra aplicaciones para toda la tecnología .NET, por lo que representa un elemento indispensable dentro esta. Incorpora CLR, que proporciona una administración de la memoria y otros servicios del sistema, y una biblioteca de clases completa, que permite a los programadores aprovechar el código sólido y confiable para todas las áreas importantes del desarrollo de aplicaciones.

Posee mejoras en las características de Windows Communication Foundation (WCF), para la presentación de servicios de forma segura. Presenta un conjunto de librerías para implementar aplicaciones interactivas, denominada Windows Presentation Foundation (WPF). [12]

.NET Framework 4.0 permite la realización del mapeo Objeto/Relacional, gracias a ADO.NET Entity Framework (EF), a través del cual se pueden crear aplicaciones de acceso a datos programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional.

---

[7] *Extensible Markup Language*: Lenguaje de marcado extensible que proporciona un formato para describir datos.

[8] *Hypertext Transfer Protocol*: Protocolo de transferencia de hipertexto, permite transferir información en ficheros hipertexto a través de Internet.

[9] *Simple Object Access Protocol*: Protocolo simple de acceso a objetos que define el formato XML para los mensajes de intercambio en el uso de un servicio web.

[10] *Web Services Description Language*: Lenguaje de descripción de servicios web, define un esquema para los documentos XML que, a su vez, definen los servicios web.

[11] *Universal Description, Discovery, and Integration:* Descripción, descubrimiento e integración universales, definen un medio estándar para publicar y descubrir información acerca de los servicios Web XML

Para la realización de consultas estilo SQL (Lenguaje de Consulta Estructurada) a colecciones de objetos, el Framework 4.0 de .NET presenta la tecnología LINQ, esta permite un desarrollo mucho más ágil, pues no es necesaria la realización de algoritmos complejos y propensos a errores para el tratamiento de colecciones de objetos. A continuación se describen cada una de estas tecnologías. [12]

1.5.1. WCF

WCF es un marco de trabajo para la creación de aplicaciones orientadas a servicios. Posibilita enviar datos como mensajes asincrónicos de un extremo de servicio a otro. Estos pueden formar parte de un servicio disponible continuamente hospedado por *Internet Information Services*, o un servicio hospedado en una aplicación, también pueden ser un cliente de un servicio que solicita datos a otro extremo de servicio. Los mensajes pueden ser tan simples como un carácter o una palabra que se envía como XML, o tan complejos como una secuencia de datos binarios. [13]

Las principales características que presenta WCF son:

- ***Orientado a servicios***: Debido al empleo de estándares de servicios web, permite crear aplicaciones orientadas a servicios, por lo que se empleará una arquitectura orientada a servicios (SOA), la cual se basa en el manejo de servicios web para enviar y recibir datos, en la propuesta de solución que se plantea. El trabajo con servicios tendrá como ventaja desacoplar estos entre una aplicación y otra en lugar de tenerlos incluidos en el código. Una relación de acoplamiento débil implica que cualquier cliente creado en cualquier plataforma puede conectar con cualquier servicio siempre y cuando se cumplan los contratos esenciales.
- ***Interoperabilidad***: Implementa estándares modernos para la interoperabilidad de servicios web. Ofreciendo compatibilidad con los protocolos de la infraestructura de estos servicios a través de las características de los contratos de canales, y los protocolos de aplicación, estos últimos se consiguen mediante XML y WSDL.
- ***Metadatos de servicios***: Admite la publicación de metadatos de servicios utilizando los formatos especificados en los estándares de la industria, como WSDL y esquemas XML. Estos metadatos pueden utilizarse a la hora de generar y configurar automáticamente clientes para el acceso a los servicios de WCF. Los metadatos se pueden publicar sobre HTTP y HTTPS.
- ***Contratos de datos***: Al basarse en .NET Framework, incluye métodos con código sencillo para proporcionar los contratos de datos que se desea emplear. Básicamente, mientras se

escribe el código del servicio usando Visual C# u otro lenguaje, WCF permite controlar los datos creando clases que representan una entidad de datos con propiedades que pertenecen a la misma. Cuando se han creado estas clases, el servicio genera automáticamente los metadatos que permiten a los clientes ajustarse a los tipos de datos que se han diseñado.

- ***Seguridad***: Posibilita cifrar los mensajes para proteger la privacidad de estos, así como obligar a los usuarios a autenticarse antes de permitirles la recepción de mensajes. [14]

1.5.2. WPF

WPF es un marco de trabajo de interfaz de usuario que permite la construcción de una aplicación cliente enriquecida visualmente e interactivas. Su plataforma de desarrollo admite un amplio conjunto de características de desarrollo, incluyendo un modelo de aplicaciones, recursos, controles, gráficos, diseño, enlace de datos, documentos y seguridad. Esta tecnología utiliza el lenguaje XAML[12] para proporcionar un modelo declarativo para la programación de aplicaciones. [15]

Entre los elementos que presenta destacan:

- Separación de apariencia y lógica.
- Soporte del patrón orden (command).
- Fácil conexión a fuentes de datos vía ligaduras (bindings),
- Simplificación de trabajo con objetos observables mediante propiedades de dependencia.
- Herencia de valores para propiedades por relación jerárquica entre componentes
- Personalización completa de componentes mediante plantillas.

1.5.3. EF

EF es un conjunto de tecnologías de ADO.NET para el desarrollo de aplicaciones de software orientadas a datos. Con el empleo de esta tecnología, modelar las entidades y sus relaciones es sencillo, pues permite crear una aplicación de acceso a datos programando con un modelo conceptual en lugar de programar directamente con un esquema de almacenamiento relacional, incluyendo tipos con herencia, miembros complejos y relaciones.

Brinda la posibilidad de trabajar con datos en forma de objetos y propiedades específicas del dominio, ahorrando preocupaciones por las tablas y columnas de la base de datos subyacente

[12] *Extensible Aplication Markup Language*: Lenguaje de marcado de aplicaciones extensible basado en XML desarrollado por Microsoft.

donde se almacenan estos. Así se puede trabajar con un nivel mayor de abstracción, acarreando consigo la reducción de código y el mantenimiento del mismo. Esta tecnología es compatible con LINQ proporcionando la validación de sintaxis en el momento de compilar consultas sobre un modelo conceptual. [16]

1.5.4. LINQ

LINQ es un conjunto de características que agrega eficaces capacidades de consulta a la sintaxis de C#, pudiéndose extender para utilizar potencialmente cualquier tipo de almacén de datos. Simplifica el trabajo objetos que proporcione una enumeración de elementos, además convierte una consulta en una construcción de lenguaje de primera clase en C#. Las consultas se escriben para colecciones de objetos fuertemente tipadas, utilizando palabras claves del lenguaje y operadores con los que se está familiarizado. Es un modelo de programación que simplifica y unifica la implementación de acceso a cualquier tipo de dato, no impone el uso de una arquitectura específica facilitando la implementación de varias arquitecturas existentes para acceso a datos. [17]

1.6. Microsoft Visual Studio 2010 Ultimate

Microsoft Visual Studio 2010 Ultimate es un IDE que soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, proporcionando un entorno integrado de herramientas e infraestructura de servidor, que simplifica todo el proceso de desarrollo de aplicaciones. Permite crear servicios web en cualquier entorno que soporte la plataforma .NET. Ofrece resultados usando procesos productivos, predecibles y personalizables, aumentando la transparencia y el seguimiento durante el ciclo de vida a partir de análisis detallados. Permite crear soluciones nuevas y mejorar otras existentes, con herramientas de elaboración de prototipos, arquitectura y desarrollo. Aumenta la productividad debido al empleo de herramientas de prueba y depuración integradas para detectar y corregir errores de un modo rápido y sencillo, permitiendo reducir el coste de desarrollo de las soluciones.

Microsoft Visual Studio en su versión 2010 es un exhaustivo paquete de herramientas de administración del ciclo de vida de las aplicaciones que incorpora nuevas características mejoradas que permite la sencillez de todo el proceso de desarrollo y que garantiza la calidad de los resultados, desde el diseño a la implementación a través del uso de herramientas de pruebas que presenta. Se selecciona este IDE para crear una solución que permita darle respuesta al problema planteado dada las facilidades que presenta en cuanto a productividad

que permiten rapidez en el desarrollo de aplicaciones, porque utiliza en su marco de trabajo el Framework 4.0 de .NET e integra el lenguaje de programación C#. [18]

1.7. Sistema Gestor de Bases de Datos (SGBD)

Los SGBD son software que permiten la utilización y/o la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios. Se compone de tres lenguajes: uno para la definición de datos, otro para la manipulación de estos y por ultimo un lenguaje de consulta.

Los objetivos fundamentales de un SGBD consisten en:

- Suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos, ni el método de acceso empleado.
- Servir de interfaz entre la base de datos, el usuario y las aplicaciones.
- Definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.

El propósito general de los SGBD es el de manejar de forma clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. [19]

1.7.1. MySQL

MySQL es uno de los gestores de base de datos que, desde su surgimiento, por características tales como la fiabilidad, velocidad, rendimiento, facilidad de administración y conexión con otros productos, ha ganado prestigio durante su evolución.

Es un sistema de gestión de bases de datos relacional de implementación multihilo, aprovechando así la potencia de sistemas multiprocesadores. Es rápido y de fácil empleo, lo cual le permite establecer relaciones para soportar y manejar una significativa carga de datos de forma segura. Se puede utilizar en entornos cliente/servidor o incrustados. Agrupa las múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo, haciendo uso del lenguaje SQL estandarizado para el almacenamiento, actualización y acceso a información. Vale destacar que emplea la Licencia Pública General de la GNU[13] para cualquier uso compatible.[20]

[13] Acrónimo recursivo para "*Gnu No es Unix*".

La versión 5.0.67 de MySQL incluye nuevas características que le permite construir aplicaciones seguras, mejorando las reglas del negocio y facilitando el acceso a los metadatos. También Soporta transacciones entre múltiples ambientes de bases de datos. Es un producto ligero que ofrece rendimiento mejorado, permitiendo definir las tablas que se utilizarán en la futura implantación. Además como las versiones que le antecedieron, es no propietario.

Fundamentación de la elección del SGBD MySQL 5.0.67

Existe una gran cantidad de softwares que emplean MySQL, lo que demuestra su aceptación y da garantía de ser confiable. Para el trabajo con la base de datos, en la solución propuesta se decidió emplear el SGBD MySQL en su versión 5.0.67. Es un gestor que facilita el trabajo a los desarrolladores, dada la existencia de un cúmulo de librerías y herramientas que permiten su integración con numerosos lenguajes de programación, como es el caso de C#. Además de su fácil instalación y configuración, también constituye un gestor multiplataforma, confiable, robusto, estable y que dada su gran escalabilidad, control de concurrencia y funcionalidades se ubica entre los SGBD más utilizados. Es software libre y utiliza la licencia GPL para definir qué se puede o no hacer.

Conclusiones del capítulo

- Se realizó un estudio de los diferentes sistemas existentes dirigidos a automatizar procesos de control de recursos humanos y medios tecnológicos, de este el autor del presente trabajo concluye que los mismos presentan características relevantes a tener en cuenta a la hora de desarrollar la aplicación que se propone como solución, pero de forma general no dan respuesta a la problemática planteada en su totalidad.
- Se realizó un análisis de diferentes tecnologías existentes para el desarrollo de la propuesta de solución, definiéndose como metodología el empleo de Scrum, a la cual se le integrarán las características que ofrece la metodología XP en el área de pruebas, contribuyendo esto a la obtención de un producto que satisfaga tanto al cliente como al equipo de desarrollo.
- También se definieron, para las tareas de implementación, las herramientas a utilizar en el desarrollo del sistema informático, así como el SGBD a emplear para el almacenamiento de datos.

CAPÍTULO 2: ANÁLISIS Y PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se realiza el análisis de los procesos existentes en el centro ISEC en cuanto a la gestión de los recursos humanos y medios tecnológicos. Puntualizando los principales elementos de la propuesta de solución y especificando sus funcionalidades a través de la pila de tareas del producto de cada una de las iteraciones.

2.1. Análisis de los procesos existentes

Los procesos llevados a cabo por el centro ISEC, encaminados a la gestión de recursos humanos y medios tecnológicos no se encuentran bien definidos, esto dificulta el control de la información referente a los recursos anteriormente mencionados, según datos proporcionados por directivos del centro. Dada esta situación, una significativa cantidad de recursos materiales son requeridos para registrar dichos datos.

Procesar ese volumen de información se convierte en una tarea engorrosa a la hora de realizar consultas para la posterior presentación de reportes, necesarios en el análisis de los recursos con los que se cuenta y efectuar así la toma decisiones. Muchas veces esto se encuentra condicionado por la existencia de datos duplicados y/o pérdida de estos, causado por el deficiente control que se tiene o por el deterioro que presenta esta documentación.

La asistencia de los miembros de los proyectos productivos se registra a través de libros de firmas para adiestrados y especialistas, haciendo complejo este proceso necesario para tener conocimiento del cumplimiento de la jornada laboral. Vale destacar que en estos libros de firma no se contemplan a los estudiantes vinculados a estos proyectos, por lo que no se puede llevar un seguimiento de los registros de asistencia de los mismos así como de entrada y salida en el horario laboral. También se realiza de forma manual la asignación de puestos de trabajos a los integrantes del proyecto, estableciéndose tiempos de máquina según las afectaciones que presente cada uno. El control del cumplimiento de la jornada laboral, a partir de la asignación de tiempos de máquina, se registran en papel o en documentos digitales Excel, información que no está asequible a los directivos del centro ISEC. A causa de lo descrito, no es posible la realización de reportes que les permitan tomar decisiones sobre los recursos humanos con los que se cuenta en cuanto a la asistencia y el cumplimiento de la jornada laboral.

La información relativa a los medios tecnológicos presentes en los laboratorios pertenecientes al centro, solo se llevan a través de reportes de Control Diario de Laboratorios en Áreas

Productivas (ver Figura 1). Esto trae consigo el desconocimiento del estado real de estos medios y de las características específicas que presentan, para ubicar el personal y los proyectos según los requerimientos de estos, y poder lograr un mejor desempeño en el desarrollo de las actividades productivas.

CONTROL DIARIO DE LAB. ÁREAS PRODUCTIVAS	Turno		Laboratorio	FECHA			No Folio CD	Área				No Folio JT
	Ent	Rec		Dia	Mes	Año						
Pertlericos	Cantidades (U)			Afectaciones Generales:				Afectación al Servicio:				
	Entrega	Recibo	Falta	Falta	Aspectos	N/E	N/EN	G	Por	No de PCs		T
PC L/P					Aire 1				DVD/CD			
Monitores L/P					Aire 2				HDD			
Teclados L/P					Aire 3				SO			
Mouse L/P					Aire 4				Red/PC			
Sillas L/P					Medios no Tecnológicos	B	R	M	Red/Tor-Par			
Mesas L/P									Mouse			
Switch					Est. puertas				Teclado			
Cables de Red					Est. Ventanas				Video			
C. Aliment.					Est. Luces				Se Reinicia			
UPS					Est. de la RED				S_Rotos			
Sellos/Odatos					Limpieza				Fuera Domin			
Scanner Barra												
Cesto												
Teléfonos					Técnicos Involucrados en la Entrega							
AP					ENTREGA	Dia	Mes	Año	Hora	RECIBE	Dia	Mes	Año	Hora
Regulador														
TV					Nombre:					Nombre:				
Scanner					Solapin		Firma			Solapin		Firma		
Lightpen														
Impresora					Observaciones:									
Neveras														
Gaveteros														
Armarios														
L/P(Lab/Proy), SO(Sist.Oper.), N/E(No Enc.), N/EN(No Enfna), G(Gotea),B(Bien), R(Reg.), M(Mal), AP(Access P),HDD(Disco Duro)														

Figura 1. Reporte del control diario de laboratorios

Otro de los elementos, que no son manejados efectivamente, es el de control de acceso de las personas ajenas a los locales dirigidos a la producción. Ello puede traer consigo que se pueda ver comprometida la integridad de la información que se esté manejando en los proyectos.

Es por lo anteriormente planteado, que se precisa de un sistema informático que permita automatizar la gestión de los elementos necesarios para llevar el control y supervisión de los recursos humanos y medios tecnológicos de los laboratorios productivos pertenecientes al centro ISEC.

2.2. Propuesta de solución

Con el objetivo de dar solución a la problemática planteada, se decidió implementar un sistema informático para el control recursos humanos y medios tecnológicos presentes en los laboratorios del centro ISEC, al cual se le decidió llamar SICREL por mutuo acuerdo entre el cliente y el equipo de desarrollo. Dicho sistema se divide en dos aplicaciones, una será el cliente y la otra el servidor.

El cliente es una aplicación de escritorio que permite a los usuarios, según el rol que posean, interactuar con las diferentes funcionalidades del sistema. Esto se realiza a través de una

interfaz de usuario desarrollada con la tecnología de .NET WPF. A continuación se describen los roles, a través de los cuales se pueden establecer las relación de las personas con SICREL.

Tabla 1. Roles identificados.

Rol	Descripción
Técnico de Laboratorio	Trabajador responsable del laboratorio. Encargado de registrar la entrada y salida de personas al laboratorio.
Administrador del Sistema	Persona que configura y controla todos los datos del sistema.
Integrante del Proyecto	Personal que forma parte del proyecto y que trabaja en los laboratorios. Podrá acceder a los datos referentes al recorrido en el trascurso del desarrollo del proyecto productivo al que pertenece, como es el caso de la asistencia.
Jefe de Proyecto	Persona que se encuentra al frente de los proyectos. Planifica el tiempo de máquina del personal del proyecto del cual es líder, además tiene acceso a toda la información referente a los recursos humanos del mismo, así como de los medios tecnológicos asignados a la actividad productiva.
Directivo del Centro	Personal que integra la dirección del centro ISEC. Puede acceder a toda la información referente a los recursos humanos pertenecientes a cada uno de los proyectos productivos del centro, así como a la de los medios tecnológicos presentes en los laboratorios asignados al mismo.

Por el lado del servidor se encuentra el Proveedor de Servicios SICREL, este es el encargado de publicar los servicios implementados gracias a las facilidades que brinda WCF. Estos servicios son necesarios para el correcto trabajo de cada una de las funcionalidades en el lado del cliente, los mismos se publican y describen con el empleo de WSDL, siendo establecida la comunicación a través del protocolo SOAP y haciéndose la transferencia de datos con XML. Este proveedor concentra toda la lógica de negocio de los servicios, comunicándose directamente con el servidor de base de datos y los servicios externos que se están empleando (ver Figura 2), como son Assets, Akademos e Identificación para obtener los principales datos de trabajadores y estudiantes.

Con el apoyo en los servicios externos se facilita el trabajo a la hora de almacenar los datos de personas, puesto que no se tendrán que introducir uno a uno, a menos que se presente algún inconveniente con las prestaciones de los servicios mencionados, lo cual no supone ningún problema ya que esta acción se podría llevar a cabo manualmente.

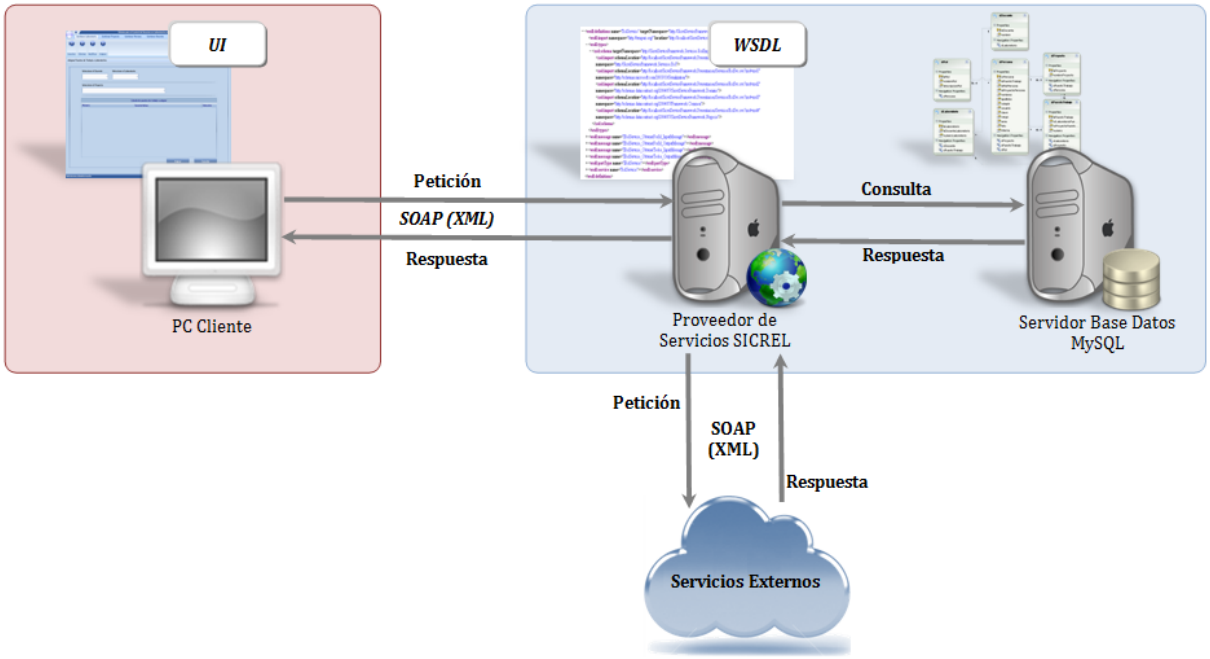

Figura 2. Estructura del sistema.

El sistema permitirá gestionar los laboratorios asignados al centro ISEC y los docentes en los cuales estos se encuentran, haciendo posible la gestión de los puestos de trabajos asignados para la actividad productiva junto a sus características. Se podrá gestionar los tiempos de máquina por parte de los jefes de proyecto a cada uno de sus integrantes, facilitando el control de la asistencia y del cumplimiento de la jornada laboral. Los directivos del centro podrán tener conocimiento de la ocupación real de los laboratorios en cualquier momento, así como los datos referentes al personal de los proyectos. A continuación se describen las funcionalidades a través de la pila de tareas del producto de cada una de las iteraciones.

2.4. Pila de Tareas del Producto

La Pila de Tareas del Producto (*Product Backlog*) no es más que una lista de funcionalidades que el sistema debe presentar. La pila del producto se sitúa en el área de necesidades de negocio desde el punto de vista del cliente. [10] Entre los atributos que componen este artefacto se tienen:

ID: Identificador único de la funcionalidad, simplemente un número autoincremental. Esto permite no perder de vista la funcionalidad, en caso de variar su nombre.

Funcionalidad: Nombre de la funcionalidad solicitada por el cliente, suficientemente clara como para distinguirla de las otras.

Descripción: Descripción corta de la funcionalidad, suficientemente clara como para que el Dueño de Producto comprenda aproximadamente de qué se habla.

Estimación: Valoración inicial acerca de cuanto trabajo es necesario para implementar la funcionalidad. Este valor se estima según la proporción ideal días/hombre.

Prioridad: Importancia que el dueño del producto da a la funcionalidad.

2.4.1. Pila de Tareas del Producto Iteración 1

El principal objetivo de esta Iteración es el de diseñar e implementar las funcionalidades relativos al módulo Administrador, junto a la funcionalidad Autenticación de usuario. La misma tiene una duración de 25 días en los cuales se llevará a cabo la implementación y pruebas de las funcionalidades expuestas en la pila de tareas de la primera iteración.

Tabla 2. Pila de tareas del producto Iteración 1

Pila de Tareas del Producto Iteración 1		***Duración***: 25 días		
ID	**Funcionalidad**	**Descripción**	**Estimación**	**Prioridad**
1	***Insertar docente***	El sistema debe permitir insertar los datos de docentes.	2	Alta
2	***Modificar docente***	El sistema debe permitir modificar los datos de docentes.	2	Alta
3	***Eliminar docente***	El sistema debe permitir eliminar docentes.	2	Alta
4	***Insertar proyecto***	El sistema debe permitir insertar los datos de proyectos.	1	Alta
5	***Modificar proyecto***	El sistema debe permitir modificar los datos de proyectos.	1	Alta
6	***Eliminar proyecto***	El sistema debe permitir eliminar	1	Alta

		proyectos.		
7	***Insertar laboratorio***	El sistema debe permitir insertar los datos de laboratorios.	2	Alta
8	***Modificar laboratorio***	El sistema debe permitir modificar los datos de laboratorios.	1	Alta
9	***Eliminar laboratorio***	El sistema debe permitir eliminar laboratorios.	1	Alta
10	***Asignar proyectos a puestos de trabajo***	El sistema debe permitir asignar a los puestos de trabajo el proyecto que estará empleando dicho puesto.	2	Alta
11	***Insertar persona***	El sistema debe permitir insertar los datos de personas a partir de los servicios brindados por la UCI.	2	Alta
12	***Insertar persona manualmente***	El sistema debe permitir insertar los datos de personas manualmente en caso de fallo de los servicios UCI.	2	Alta
13	***Modificar persona***	El sistema debe permitir modificar los datos de personas.	2	Alta
14	***Eliminar persona***	El sistema debe permitir eliminar personas.	1	Alta
15	***Autenticar usuario***	El sistema debe permitir la autenticación de los usuarios para poder acceder a las funcionalidades que le corresponde.	2	Media
16	***Insertar puesto de trabajo***	El sistema debe permitir insertar puestos de trabajo.	2	Media
17	***Asignar puesto de trabajo a personas***	El sistema debe permitir asignar puestos de trabajo a las personas que trabajan en un proyecto.	2	Media
18	***Modificar puesto de trabajo***	El sistema debe permitir modificar puestos de trabajo.	2	Media

19	***Eliminar puesto de trabajo***	El sistema debe permitir eliminar puestos de trabajo.	1	Media

2.4.2. Pila de Tareas del Producto Iteración 2

El objetivo de esta Iteración es el de diseñar e implementar las funcionalidades relativas a los módulos Jefe de Proyecto, Técnico de Laboratorio, Integrante del Proyecto y Directivo del Centro. Esta tiene una duración de 20 días en los cuales se llevará a cabo la implementación y pruebas de las funcionalidades expuestas en la pila de tareas de la primera iteración.

Tabla 3 Pila de tareas del producto Iteración 2

Pila de Tareas del Producto Iteración 2		***Duración***: 20 días		
ID	**Funcionalidad**	**Descripción**	**Estimación**	**Prioridad**
20	***Insertar tiempo de máquina***	El sistema debe permitir insertar los datos de tiempos de máquina.	2	Alta
21	***Modificar tiempo de máquina***	El sistema debe permitir modificar los datos de tiempos de máquina.	2	Alta
22	***Eliminar tiempo de máquina***	El sistema debe permitir eliminar tiempos de máquina.	1	Alta
23	***Registrar entrada de persona interna***	El sistema debe permitir registrar entrada de personas internas	3	Alta
24	***Registrar entrada de persona externa***	El sistema debe permitir registrar entrada de personas externas.	2	Alta
25	***Registrar salida de persona***	El sistema debe permitir registrar salida de personas	2	Alta
26	***Consultar datos de personas proyectos***	El sistema debe permitir mostrar a los directivos del centro los datos referentes a al personal que laboran en cada proyecto.	3	Media
27	***Consultar datos ocupación actual de laboratorios***	El sistema debe permitir mostrar a los directivos del centro los datos referentes a la ocupación actual de los laboratorios.	3	Media

28	***Consultar datos planificación de tiempo de máquina***	El sistema debe permitir mostrar los datos del tiempo de máquina asignado a los integrantes del proyecto.	3	Baja
29	***Consultar datos trayectoria de asistencia***	El sistema debe permitir mostrar los datos de la trayectoria en cuanto a asistencia de los integrantes del proyecto.	3	Baja

2.5. Requerimientos adicionales

A continuación se expondrán los requerimientos no funcionales que de forma general deberá presentar SICREL.

2.5.1. Seguridad

- Se deben emplear certificados para autenticar el cliente con el servidor estableciendo la seguridad en los mensajes que son intercambiados en el canal de comunicación.
- La interacción con el sistema estará controlada; pues la información será consultada y/o modificada solo por el personal autorizado; para lo cual se establecieron roles con funciones específicas.

2.5.2. Disponibilidad

- El sistema debe ser capaz de resolver los problemas de conectividad, en caso que cambie su dirección el servidor donde están hosteados los servicios, posibilitar la opción de actualizar la dirección.

2.5.3. Rendimiento

- El sistema debe tener un tiempo de respuesta rápido y eficiente, 7 segundos como máximo.
- Los servicios deben ser capaces de soportar una cantidad escalable de 100 peticiones concurrentes.

2.5.4. Interfaz de usuario

- Las dimensiones de la interfaz de usuario serán de 1024 X 768 píxeles; presentará por defecto un color azul claro y los textos con un tipo de letra Segoe UI de tamaño 11pt.
- Tanto la interfaz, como los mensajes, ya sean de Error, Información o Advertencia deberán estar en idioma español. El aspecto de la apariencia debe ser estándar. Los mensajes deben ser claros.

2.5.5. Software

- El servidor de base de datos funcionará sobre el sistema operativo GNU/Linux, teniendo instalado el SGBD MySQL 5.0.67.
- El proveedor de servicios deberá utilizar el sistema operativo Windows 7, ya que será necesario utilizar Microsoft .NET Framework 4.0 e Internet Information Services 7.0.
- Las estaciones de trabajo clientes deberán utilizar cualquier sistema operativo de la familia Windows superior a XP SP2, también tendrán que presentar la instalación del Microsoft .NET Framework 4.0.

2.5.6. Hardware

Para el correcto funcionamiento de los componentes en el lado del servidor se hace necesario contar como mínimo con los siguientes elementos:

- 1 PC Intel Celeron a 2.6 GHz, 2 GB de memoria RAM y 150 GB de disco duro para el servidor de base de datos
- 1 PC Intel Celeron a 2.6 GHz, 2 GB de memoria RAM y 80 GB de disco duro para el proveedor de servicios.

Para las PC en puestos de trabajo situados en el lado del cliente se deberá contar con al menos 512 MB de RAM. También presentar 2 GB de espacio libre, para almacenar la información en caso del fallo de la red.

Conclusiones del Capítulo

- Se identificaron los diferentes roles que estarán interactuando con el sistema, y se establecieron las acciones que pueden realizar estos sobre el mismo.
- Se describieron las pilas de tareas del producto de cada una de las iteraciones, obteniéndose así las funcionalidades que el sistema debe presentar según las necesidades del cliente.
- Se obtuvo la descripción de los requerimientos no funcionales necesarios para el correcto funcionamiento del sistema.

CAPÍTULO 3: DISEÑO DEL SISTEMA

Introducción

En el presente capítulo se detallan los elementos referentes al diseño de la aplicación como es el caso de la arquitectura, los patrones de diseño a emplear y el modelo entidades de datos utilizado.

3.1. Arquitectura de software

La arquitectura de software representa un esquema lógico de organización estructural de alto nivel para sistema de software; que consiste en un conjunto de patrones y abstracciones coherentes. Donde se provee una serie de subsistemas predefinidos, especificando sus responsabilidades, e incluye reglas y guías para organizar las relaciones entre ellos. Especifica las propiedades generales a la estructura del sistema, y repercuten en la arquitectura de sus subsistemas. La selección de una arquitectura es una decisión fundamental al desarrollar un sistema de software.

3.1.1. Estilo Arquitectural Cliente-Servidor

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa como depósito de datos y funciona tal cual un SGBD. Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor es decir realiza peticiones. Ambas partes deben estar conectadas entre sí mediante una red. [21]

Dentro de sus principales características se encuentran:

- Es un estilo para sistemas distribuidos.
- Divide el sistema en una aplicación cliente, una aplicación servidora y una red que las conecta.
- Describe una relación entre el cliente y el servidor en la cual el primero realiza peticiones y el segundo envía respuestas
- Puede usar un amplio rango de protocolos y formato de datos para comunicar la información.

Los principios básicos que presenta este estilo son:

- El cliente realiza una o más peticiones, espera por las respuestas y las procesa a su llegada.
- El cliente normalmente se conecta a uno o a un número reducido de servidores al mismo tiempo.
- El cliente interactúa directamente con el usuario, a través de una interfaz gráfica.

- El servidor no realiza ninguna petición al cliente
- El servidor envía los datos en respuesta a las peticiones realizadas por los clientes conectados.

Una representación gráfica de este tipo de estilo arquitectónico sería la siguiente:

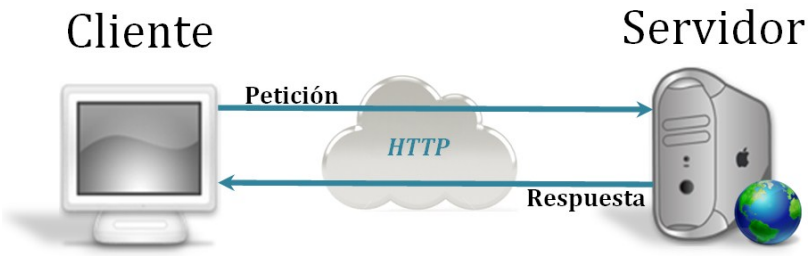

Figura 3. Arquitectura Cliente-Servidor

3.1.2. Arquitectura Orientada a Servicios (SOA)

SOA presenta un diseño del tipo Cliente-Servidor donde una aplicación se conforma por servicios de software y consumidores de esos servicios conocidos como clientes. Esta arquitectura establece un marco de diseño para la integración de aplicaciones independientes, permitiendo el acceso a funcionalidades desde la red, ofreciéndose estas como servicios [22]. A continuación se muestran los elementos que podrían observarse dentro de esta arquitectura.

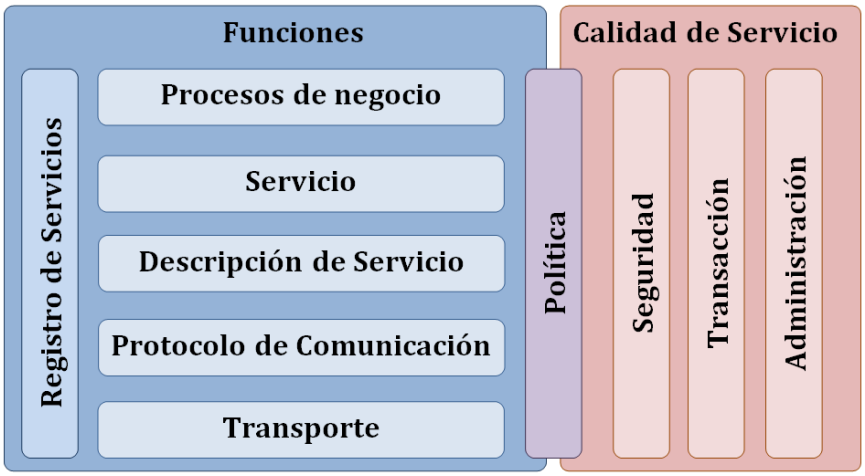

Figura 4. Elementos de la arquitectura SOA

Elementos de las Funciones:

- ***Transporte***: es el mecanismo utilizado para llevar las peticiones de servicio desde un consumidor hacia un proveedor de servicio, y las respuestas desde el proveedor hacia el consumidor.

- ***Protocolo de comunicación de servicios***: es un mecanismo a través del cual un proveedor y un consumidor de servicios comunican qué se solicita y qué está siendo respondido.
- ***Descripción de servicio***: es un esquema que describe al servicio, cómo este debe invocarse, y qué datos requiere para ser invocado con éxito.
- ***Servicio***: describe el servicio que se encuentra disponible para utilizar.
- ***Procesos de Negocio***: es la colección de servicios, invocados en una secuencia particular con un conjunto específico de reglas, para satisfacer un requisito de negocio.
- ***Registro de Servicios***: es un repositorio de descripciones de servicios y datos que pueden utilizar los proveedores de servicios para publicar sus servicios, así como los consumidores de servicios para descubrir o hallar servicios disponibles.

Elementos Calidad de Servicio:

- ***Política***: conjunto de condiciones o reglas bajo las cuales un proveedor hace el servicio disponible para consumidores.
- ***Seguridad***: conjunto de reglas que pueden aplicarse para identificar, autorizar y controlar el acceso de consumidores a los servicios.
- ***Transacciones***: conjunto de atributos aplicados a un grupo de servicios para entregar un resultado consistente.
- ***Administración***: conjunto de atributos aplicados para manejar servicios proporcionados o consumidos.

Las colaboraciones en SOA siguen el paradigma descubrir, ligar e invocar, donde un consumidor de servicios realiza la localización dinámica de un servicio, es decir descubre el servicio, consultando el registro de servicios para hallar uno que cumpla con un criterio determinado, que es donde el proveedor publica dicho servicio. De existir el servicio, el registro proporciona al consumidor la interfaz de contrato y la dirección del servicio proveedor, pudiéndose invocar los servicios para su utilización por parte del consumidor. [23]

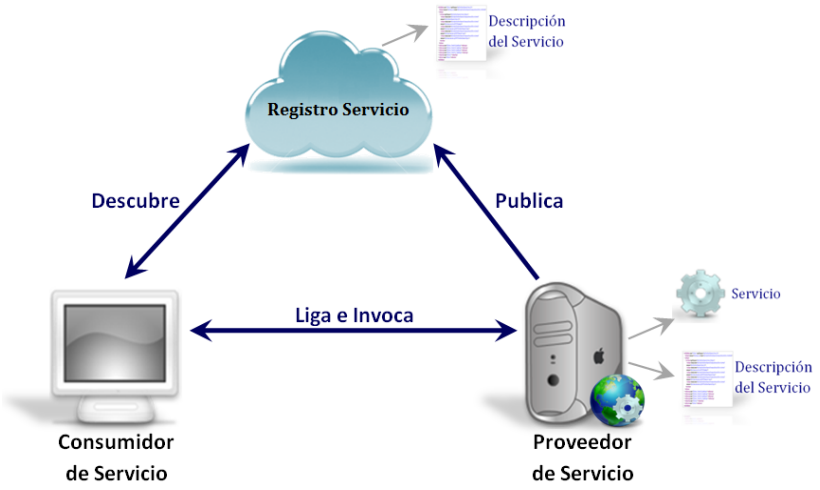

Figura 5. Colaboración en SOA

Cada entidad puede tomar uno de posibles roles correspondientes, siendo estos consumidor, proveedor y/o registro:

- ***Consumidor de servicios***: pudiese ser una aplicación, un módulo de software u otro servicio que demanda la funcionalidad proporcionada por un servicio, ejecutándola a partir de un contrato de interfaz.
- ***Proveedor de servicios***: entidad accesible a través de la red que acepta y ejecuta consultas de consumidores, publica servicios y sus respectivos contrato de interfaces en el registro de servicios para que el consumidor pueda descubrir y acceder al servicio.
- ***Registro de servicios***: hace posible el descubrimiento de servicios, los cuales están contenidos en un repositorio de servicios disponibles, permitiendo exponer las interfaces de los proveedores de servicios a los consumidores interesados.

Las operaciones llevadas a cabo entre entidades son:

- ***Publicar***: para el acceso al servicio se debe publicar la descripción de este, permitiendo al consumidor poder descubrirlo e invocarlo.
- ***Descubrir***: un consumidor de servicios localiza un servicio que cumpla con un cierto criterio consultando al registro de servicios.
- ***Ligar e Invocar***: al obtener la descripción del servicio, el consumidor podrá invocarlo haciendo uso de la información proporcionada por dicha descripción.

El empleo de SOA proporciona como ventajas que favorece la reutilización, al desacoplar las capas de una aplicación, permitiendo la utilización de los servicios proveídos por terceros. También aumenta la flexibilidad, pues admite la interoperabilidad entre sistemas gracias a la prestación de servicios que ofrece. Además mejora el proceso de construcción de software, dado el desacople presente en el desarrollo de servicios y procesos permitiendo agilizar la realización de mantenimiento, mejorando a su vez la usabilidad de las aplicaciones ya que presenta al usuario la información dispersa en distintos sistemas de forma integrada.

3.1.3. Patrón Modelo-Vista-Vista del Modelo (MVVM)

Un patrón de diseño ayuda a definir la arquitectura a la hora de diseñar una aplicación, identificando problemas y aportando soluciones. MVVM es un patrón genérico de arquitectura para las capas de presentación, especialmente diseñado para su empleo en aplicaciones desarrolladas con WPF [24]. Está definido a partir del patrón Modelo Vista Controlador (MVC) y Modelo Vista Presentador (MVP), de ellos se pueden identificar elementos que presentan en común:

- ***Modelo***: identifica a la lógica de negocio, acceso a datos y operaciones sobre los mismos
- ***Vista***: define a la Interfaz a través de la cual el usuario interactúa con la aplicación.
- ***Controlador/Presentador***: actúa como capa intermedia para la gestión de las operaciones que efectúa el usuario sobre la vista, accediendo al modelo si la operación lo requiere.

En MVVM resaltan algunas diferencias en cuanto a estos elementos:

- ***Vista***: Al igual que en MVC y MVP, la vista siguen siendo la interfaz de usuario, está definida en XAML. Es la encargada de mostrar datos y generar eventos como respuesta a las acciones del usuario, enlazando los controles al modelo mediante *Data Binding*, esta característica que ofrece WPF, es la encargada de enlazar la interfaz con el modelo o contexto de datos, de tal manera que un cambio en el modelo de datos se replique automáticamente en la interfaz y a su vez un cambio en la interfaz se transfiera de forma automática sobre el modelo.
- ***Modelo de la Vista***: Es el encargado de manejar los eventos que genera la vista, ya sea recogiendo o mostrando datos. Es la capa intermedia del patrón que comunica el modelo con la vista, preparando los datos del modelo para que puedan ser consumidos por la vista.
- ***Modelo***: Es el encargado de mantener la persistencia de los datos. El modelo de datos, se obtendrá al generar la referencia sobre los servicios web.

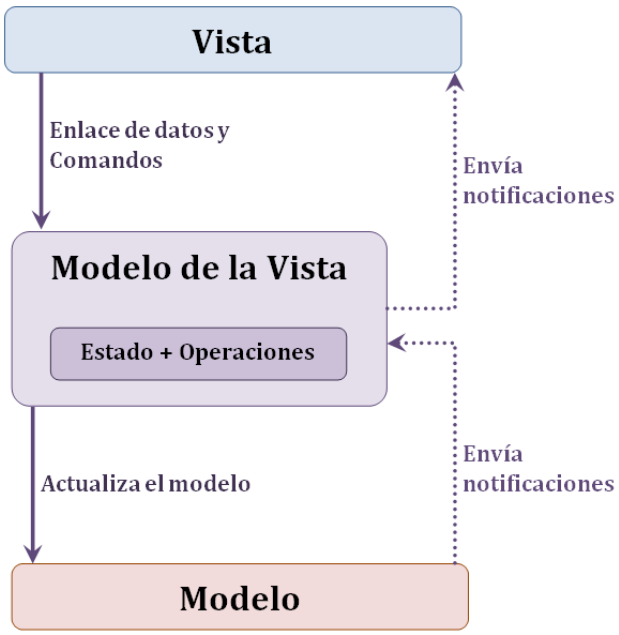

Figura 6. Estructura del patrón de diseño MVVM

El objetivo de la aplicación del MVVM es permitir presentar y gestionar de manera sencilla los datos que van a ser trasladados. Este patrón fue diseñado para hacer uso de funciones específicas de WPF, facilitar la separación de desarrollo/diseño de la sub-capa vista del resto del patrón, eliminando virtualmente todo el code behind que presenta [24]. Entre los beneficios que este patrón de diseño aporta al desarrollo del sistema se tienen:

- Un modelo de la vista proporciona un único almacén de estados y políticas de presentación, lo que mejora la reutilización del modelo (desacoplándolo de las vistas) y facilita el reemplazo de estas
- Aporta facilidades a la hora de realizar pruebas unitarias a la aplicación, pues permite separar la lógica de las vistas de los controles visuales.
- Las pruebas se encargarán exclusivamente del modelo y del modelo de la vista.

3.1.4. Descripción de la arquitectura

SICREL presenta una arquitectura Cliente-Servidor Orientada a Servicios, la puesta en práctica de ésta permite la centralización de la gestión de información y la separación de responsabilidades, facilitando y clarificando el diseño del sistema. Esto trae como ventaja que el control de acceso a los recursos e integridad de los datos sean realizados por el servidor, evitando el daño por parte de programas clientes no autorizados que intenten acceder a este. Además, el realizar mantenimiento, al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, posibilita reemplazar, reparar, actualizar, o incluso trasladar

un servidor, mientras que los clientes no se verán afectados por ese cambio o dichas afectaciones serán mínimas. La estructura modular facilita la integración de nuevas tecnologías y el crecimiento de la infraestructura, favoreciendo así la escalabilidad de la solución.

Una muestra lógica de la arquitectura utilizada se representa en la Figura 14.

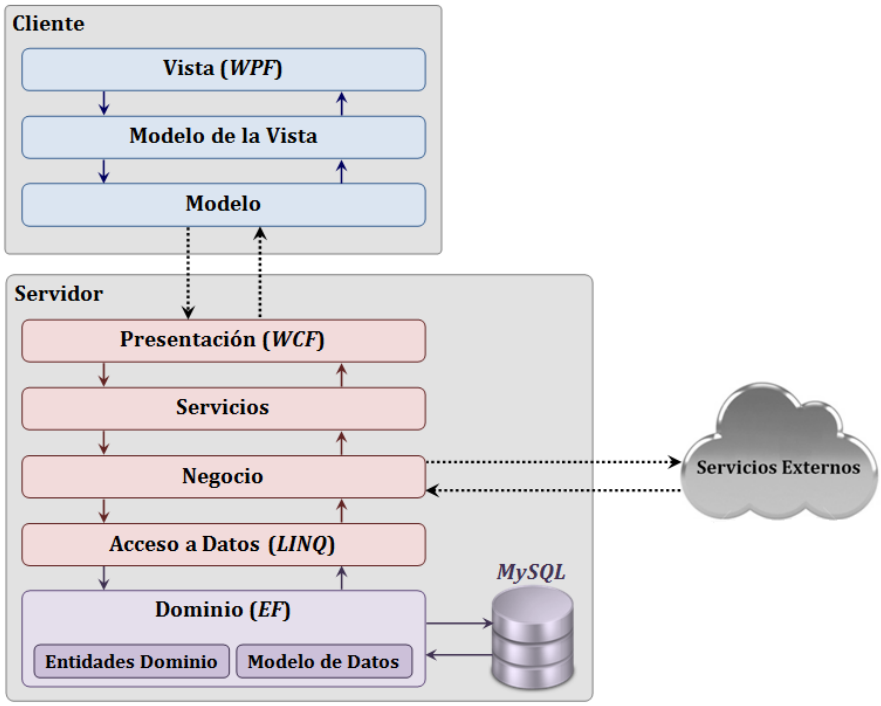

Figura 7. Vista lógica de la arquitectura empleada

El cliente sigue el patrón MVVM, el mismo está diseñado para su uso en aplicaciones en la cual la presentación de la interfaz de usuario esté realizada con la tecnología WPF. Esto permite un mayor desacople debido a la separación lógica que presenta de sus capas, lo cual trae como ventaja que se pueda hacer una mayor reutilización de los componentes y flexibilidad para el cambio de las interfaces de usuario sin tener que alterar la lógica.

El servidor presenta una arquitectura SOA, siguiendo una estructura en capas bien definida para lograr descomponer cada uno de los servicios, de esta forma se minimiza las dependencias del negocio y se agiliza el desarrollo. Es importante destacar que ésta arquitectura reduce al máximo el intercambio de información, el cual se realiza sólo entre capas inmediatas. Los servicios han sido diseñados teniendo en cuenta los requerimientos del cliente. Cada servicio es una entidad de software que encapsula funcionalidad de negocios y proporciona dicha funcionalidad a otras entidades a través de interfaces públicas bien definidas.

A continuación se describen cada una de las capas de la arquitectura propuesta:

3.1.5. Componentes Cliente

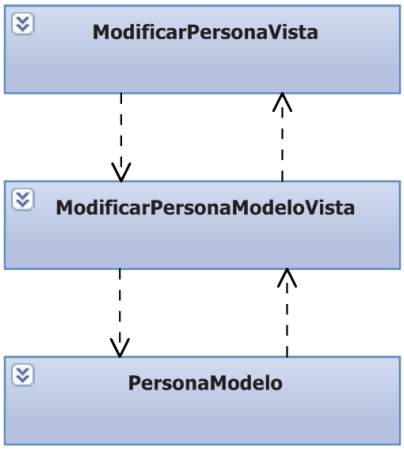

Figura 8. Interacción entre los componentes de la funcionalidad *Modificar Persona* aplicado el patrón MVVM.

- ***Vista*:** es la encargada del control de la interacción con el usuario y de definir la interfaz gráfica del sistema diseñada con la tecnología WPF (ver Figura 8). Su principal característica es la de proporcionar los mecanismos básicos para que el usuario utilice la aplicación (aplicado el patrón MVVM se encuentra en la clase *ModificarPersonaVista*, Figura 9).

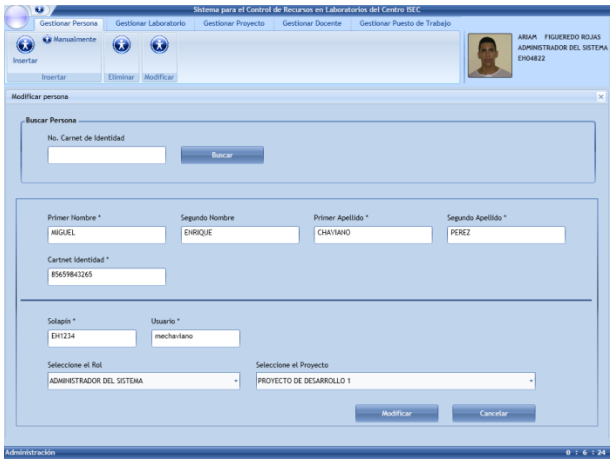

Figura 9. Vista del sistema modificar persona

- ***Modelo de la Vista*:** maneja la lógica referente a la presentación del sistema, ayudando a sincronizar y orquestar las interacciones del usuario. Permite separar el comportamiento de las interfaces de usuario de sus componentes gráficos, pudiéndose reutilizar dicha lógica y patrones desde otras vistas. Sirve como intermediario entre la vista y el modelo (ej. *ModificarPersonaModeloVista* ver Figura 9).
- ***Modelo*:** se encarga de persistir los datos en el sistema, a través de la comunicación con el proveedor de servicios (ej. *PersonaModelo* ver Figura 9).

3.1.6. Componente Servidor

A continuación se muestra la interacción existente entre las clases en cada una de las capas del servidor para la publicación de servicios.

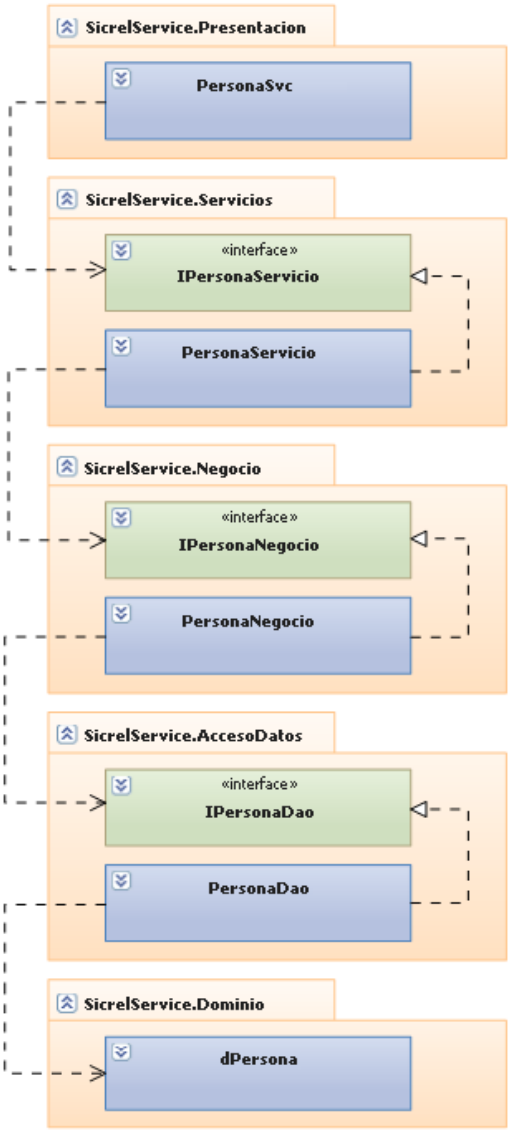

Figura 10. . Interacción entre clases de la arquitectura del proveedor de servicios (Servicios Persona).

- **Presentación:** presenta los servicios que son publicados y posteriormente consumidos por la aplicación cliente, haciendo uso de la tecnología WCF, esta permite publicar en los mismos las funcionalidades que presenta a través del empleo de *WSDL* (ver Figura 11).

```
- <wsdl:definitions name="PersonaServicio" targetNamespace="http://SicrelServiceFramework.Servicios.Persona">
    <wsdl:import namespace="http://tempuri.org/" location="http://localhost/SicrelServiceFramework.Presentacion/Servicios/PersonaSvc.svc?wsdl=wsdl0"/>
  - <wsdl:types>
    + <xsd:schema targetNamespace="http://SicrelServiceFramework.Servicios.Persona/Imports"></xsd:schema>
    </wsdl:types>
  + <wsdl:message name="IPersonaServicio_Modificar_InputMessage"></wsdl:message>
  + <wsdl:message name="IPersonaServicio_Modificar_OutputMessage"></wsdl:message>

  + <wsdl:portType name="IPersonaServicio"></wsdl:portType>
  + <wsdl:service name="PersonaServicio"></wsdl:service>
  </wsdl:definitions>
```

Figura 11. WSDL referente a PersonaSvc, para la funcionalidad *Modificar Persona*.

- **Servicios:** presenta las operaciones de cada servicio, y a través de una interfaz permite la interacción con cada una de las capas con las que se comunica (ver Figura 12). Emplea contratos que básicamente informan al cliente la acción del servicio (ver Figura 13).

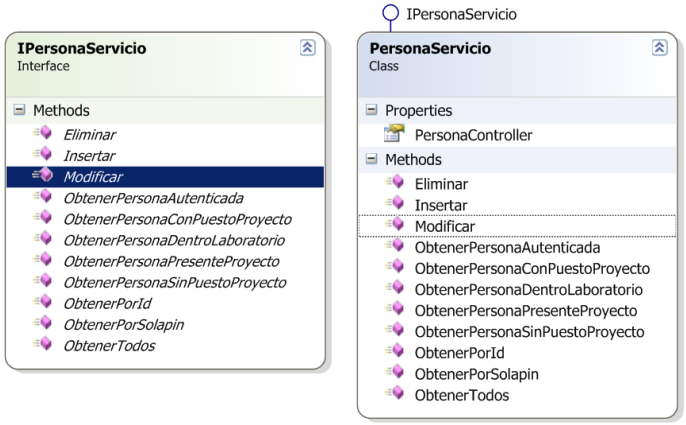

Figura 12. Interfaz IPersonaServicio y clase PersonaServicio que la implementa.

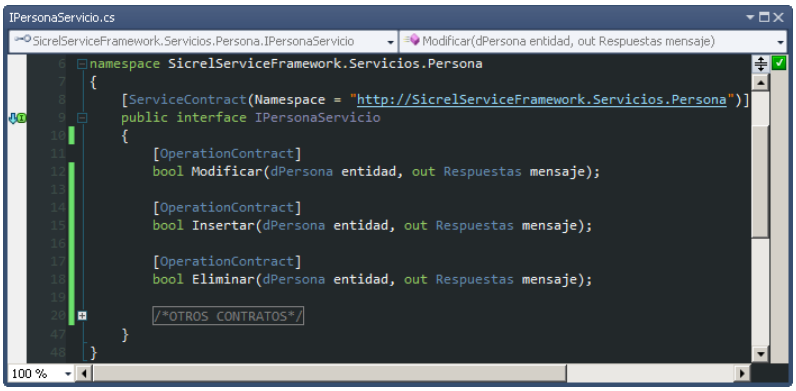

Figura 13. Código de la interfaz del servicio IPersonaServicio.

- **Negocio:** recibe las peticiones de la capa superior (*SicrelService.Servicios*) y tiene la responsabilidad de manejar todas las operaciones sobre una entidad en específico, así como todas las entidades que por conceptos de composición se encuentran relacionadas con esta. Permite gestionar la comunicación con servicios provistos por sistemas externos a la aplicación. También interactúa con la capa de acceso a datos, manejando los datos que son necesarios para el negocio del servicio. (ver Figura 14 y 15)

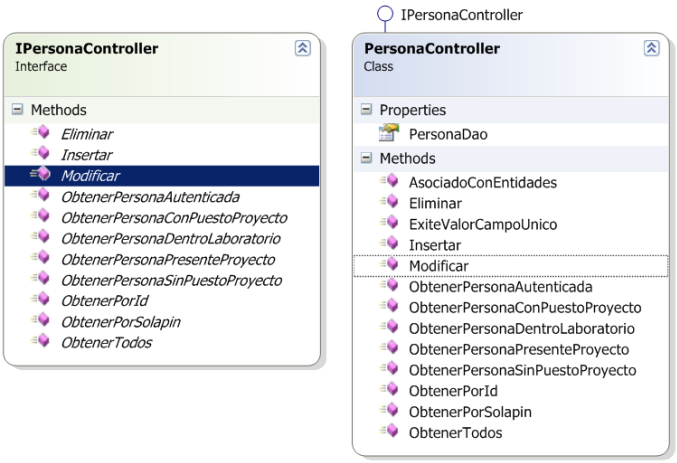

Figura 14. . Interfaz IPersonaController y clase PersonaController que la implementa.

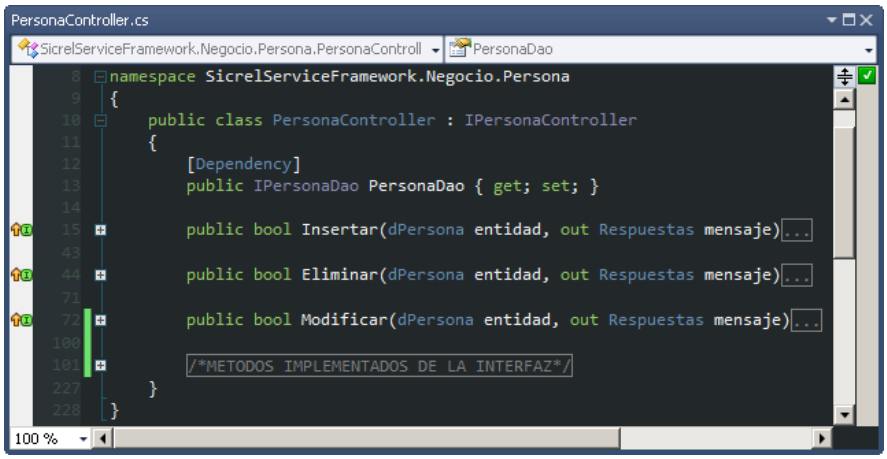

Figura 15. Código de la clase PersonaControler.

- **Acceso Datos:** encargada de obtener y persistir información en el servidor de bases de datos. Se relaciona directamente con los servicios definidos en el negocio (ver en Figura 10). Utiliza la tecnología LINQ, que permite el desarrollo de funcionalidades y la utilización de operadores genéricos, que facilitan el trabajo con las entidades mapeadas. Presenta un conector que es el encargado de establecer comunicación entre las fuentes de datos y el sistema.

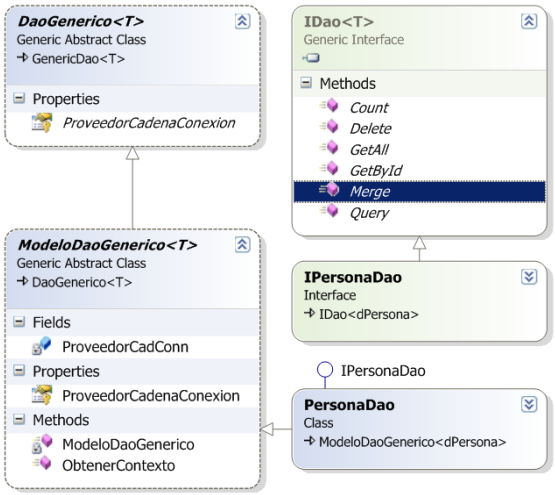

Figura 16. Interfaz IPersonaDao, clase PersonaDao que la implementa y clases auxiliares para la persistencia de datos.

- **Dominio:** presenta el conjunto de clases mapeadas, realizado por la tecnología EF, que representan, a través de un modelo entidades de datos (ver Figura 20), cada una de las entidades existentes en la base de datos (Figura 17).

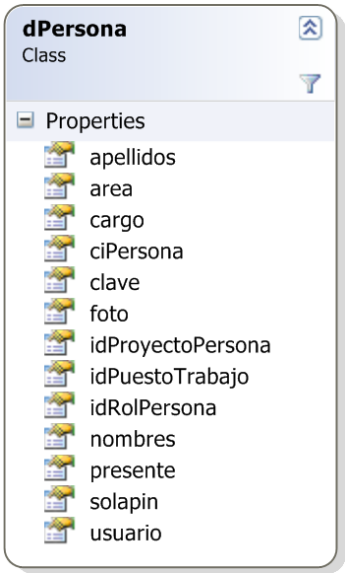

Figura 17. Entidad dPersona.

3.2. Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software, brindando una solución ya probada y documentada a estos [25].

Durante el desarrollo del sistema se identificaron problemas que se repiten o son análogos y para su solución se han empleado un conjunto de patrones que dan flexibilidad al código. Estos

patrones, permiten la reutilización del esfuerzo en el desarrollo de la solución con el fin de materializar buenas prácticas de programación. Los patrones de diseño empleados en la solución son:

Fachada: Permite proporcionar una interfaz unificada de alto nivel, ejemplo **ITiempoDeMaquinaDao**, para un conjunto de clases del sistema, como es el caso de **NegocioTiempoDeMaquina** que instancia un objeto de la interfaz tiempo de máquina, simplificando el acceso a dicho conjunto de clases (ej. **TiempoDeMaquinaDao** que implementa los métodos de dicha interfaz), pues las capas sólo se comunican entre sí a través de esta única interfaz.

Figura 18. Patrón Fachada

Fábrica: Proveerá una interfaz para crear familias de objetos relacionados o dependientes sin especificar los tipos concretos de clases (ej. **IDao<T>**). Su uso se encuentra centrado a la creación de los conectores correspondientes al acceso a datos que se esté utilizando (en la clase **ITiempoDeMaquinaDao**, quien hereda de **IDao** y es implementada a su vez por **TiempoDeMaquinaDao**), así como en la obtención de los servicios a utilizar por las clases (métodos ***Count***, ***Delete***, ***GetAll***, ***GetById***, ***Merge*** y ***Query***). Un ejemplo de cómo se aplica este patrón en la solución propuesta se muestra a continuación.

Figura 19. Patrón Fábrica

DAO[14]**:** Permite implementar el mecanismo de acceso requerido para trabajar con la fuente de datos. Los componentes de negocio que se basan en DAO utilizan la interfaz más simple expuesta por DAO para sus clientes (**ITiempoDeMaquinaDao**). DAO oculta completamente los detalles del origen de datos de la aplicación hacia sus clientes. Debido a que la interfaz expuesta por el DAO a los clientes no cambia cuando los datos subyacentes cambian su implementación de código (en la clase **ITiempoDeMaquinaDao**), este modelo permite al DAO adaptarse a los sistemas de almacenamiento sin que ello afecte al cliente o componentes de negocio. Esencialmente, DAO actúa como un adaptador entre el componente (**NegocioTiempoDeMaquina**) y la fuente de datos (**EntidadTiempoDeMaquina**). Emplea también el patrón Fábrica. A continuación se detalla este patrón (Figura 11).

[14] ***D**ata **A**ccess **O**bjects*

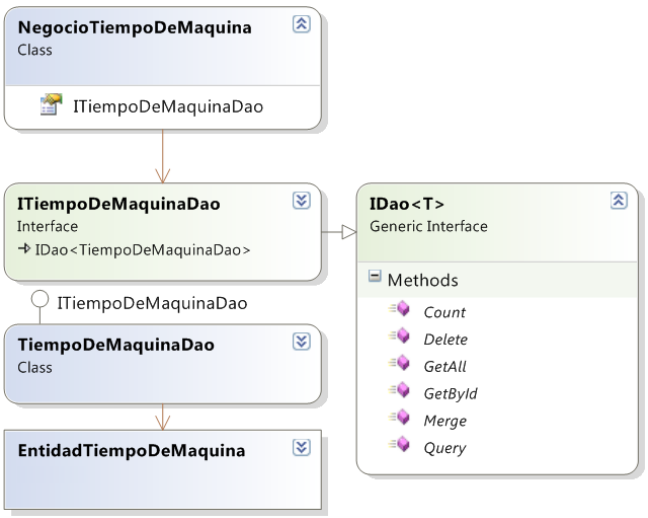

Figura 20. Patrón de diseño DAO

Bajo acoplamiento: Dada la necesidad de lograr una alta reutilización se decide aplicar este patrón. Esto permite crear clases más independientes y reutilizables implicando una mayor productividad. Como consecuencia los cambios efectuados sobre una capa no repercuten sobre las demás, o los efectos son mínimos. El empleo de este patrón puede verse en la Figura 15 donde cada capa del proveedor de servicios sólo se comunica con la inmediata superior e inferior (ej. En la capa de servicios ***PersonaServicio*** interactúa con la capa de negocio a través de la interfaz ***IPersonaNegocio***, siendo implementados sus métodos por la clase ***PersonaNegocio***, y esta última a su vez se comunica con la capa de acceso a datos a través de la interfaz ***IPersonaDao***).

Alta cohesión: Este patrón permite mantener la complejidad dentro de límites manejables del diseño, mejorándose la claridad y facilidad a la hora de entenderse este. También simplifica el mantenimiento y la mejora de las funcionalidades soportando mayor capacidad de reutilización y generando un bajo acoplamiento. En otras palabras la alta cohesión trae consigo que una clase presente responsabilidades sujetas a una funcionalidad única. Por ejemplo ***PersonaNegocio*** se encarga única y exclusivamente de la lógica de negocio (Figura 15), colaborando con las capas de servicios (donde se establecen solamente los contratos de los servicios a publicar) y acceso a datos (encargada de la persistencia de la información relativa a las diferentes entidades).

3.3. Modelo Entidades de Datos (EDM[15])

A través de un modelo de datos se obtiene una descripción abstracta sobre la representación de los datos en un SGBD. Los componentes que lo conforman son entidades existentes en el sistema a modelar y que se pueden manipular. Los modelos contienen a su vez características propias de estos objetos y la forma en que se relacionan entre sí. EDM describe la estructura de los objetos del negocio sirviendo como puente entre la aplicación y los datos almacenados [16], a continuación se muestra el EDM del sistema desarrollado.

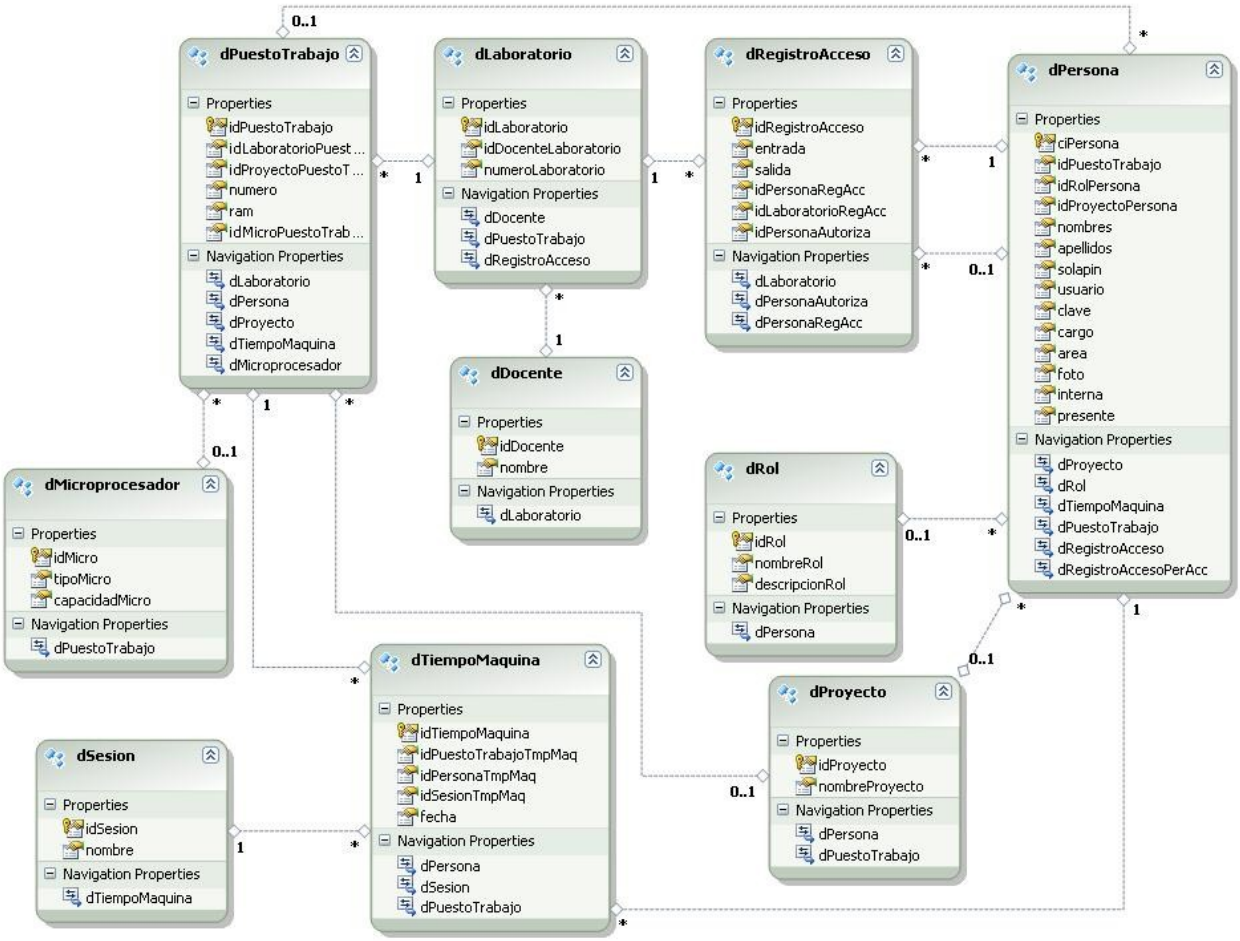

Figura 21. EDM del sistema desarrollado.

Conclusiones del Capítulo

- Se determinaron los patrones arquitectónicos y de diseño que estarán presentes en la solución, permitiendo alcanzar mayor organización en la misma.

[15] Por sus siglas en inglés.

- Se visualiza el Modelo Entidades de Datos obtenido, que será utilizado para el trabajo con las diferentes entidades de la base de datos.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Introducción

En el presente capítulo se abordan las tareas de implementación, contempladas a través de la pila de tareas de las iteraciones. Se describen los estándares de programación empleados en el desarrollo del sistema. También se describen, siguiendo la metodología XP, los tipos de pruebas a realizar sobre la solución propuesta, con el objetivo de comprobar el correcto funcionamiento del sistema.

4.1. Pila de Tareas de las Iteraciones

La Pila de Tareas de la Iteración (*Sprint Backlog*) es la lista de tareas que se realizan en una iteración. Descompone las funcionalidades de la pila del producto en las tareas necesarias para construir un incremento, es decir, lograr una parte completa y operativa del producto. Cada una de las tareas tiene asignada una persona, además presenta también el tiempo que aún falta para terminarla. Es útil porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance diario, e identificar riesgos y problemas sin necesidad de procesos complejos de gestión. [7]

Los componentes principales de la pila de tareas de la iteración son:

- ***ID:*** Identificador de la funcionalidad a la que pertenece la tarea.
- ***Lista de Tareas:*** lista de las tareas asociadas a cada una de las funcionalidades.
- ***Estado:*** estado de terminación en la que se encuentra la tarea.
- ***Estima:*** estimación realizada sobre la duración de la tarea.
- ***Real:*** duración real de la tarea.
- ***Responsable:*** persona responsable de llevar a cabo la tarea.

4.1.1. Pila de Tareas de la Iteración 1

Tabla 4. Pila de tareas de la Iteración 1

Pila de Tareas de la Iteración 1		**Responsable:** *Ariam Figueredo Rojas* **Duración:** *25 días*		
ID	**Lista de Tareas**	**Estado**	**Estima**	**Real**
0	Crear Modelo de Datos	100%	2	2
	Establecer conexión con la Base de Datos	100%	3	2
	Generar Script de la base de datos para MySQL	100%	1	1

1	Diseñar e implementar formulario Insertar docente Implementar funcionalidad Insertar docente	100%	1	1
2	Diseñar e implementar formulario Modificar docente Implementar funcionalidad Modificar docente	100%	1	1
3	Diseñar e implementar formulario Eliminar docente Implementar funcionalidad Eliminar docente	100%	1	1
4	Diseñar e implementar formulario Insertar proyecto Implementar funcionalidad Insertar proyecto	100%	1	1
5	Diseñar e implementar formulario Modificar proyecto Implementar funcionalidad Modificar proyecto	100%	1	1
6	Diseñar e implementar formulario Eliminar proyecto Implementar funcionalidad Eliminar proyecto	100%	1	1
7	Diseñar e implementar formulario Insertar laboratorio Implementar funcionalidad Insertar laboratorio	100%	1	1
8	Diseñar e implementar formulario Asignar proyectos a puestos de trabajo Implementar funcionalidad Asignar proyectos a puestos de trabajo	100%	2	1
9	Diseñar e implementar formulario Modificar laboratorio Implementar funcionalidad Modificar laboratorio	100%	1	1
10	Diseñar e implementar formulario Eliminar laboratorio Implementar funcionalidad Eliminar laboratorio	100%	1	1
11	Diseñar e implementar formulario Insertar persona Implementar funcionalidad Insertar persona	100%	1	1
12	Diseñar e implementar formulario Insertar persona manualmente Implementar funcionalidad Insertar persona manualmente	100%	1	1
13	Diseñar e implementar formulario Modificar persona Implementar funcionalidad Modificar persona	100%	1	1
14	Diseñar e implementar formulario Eliminar persona Implementar funcionalidad Eliminar persona	100%	1	1
15	Diseñar e implementar formulario Autenticar Usuario	100%	3	2

	Implementar funcionalidad Autenticar usuario			
16	Diseñar e implementar formulario Insertar puesto de trabajo Implementar funcionalidad Insertar puesto de trabajo	100%	1	1
17	Diseñar e implementar formulario Asignar puesto de trabajo a personas Implementar funcionalidad Asignar puesto de trabajo a personas	100%	1	1
18	Diseñar e implementar formulario Modificar puesto de trabajo Implementar funcionalidad Modificar puesto de trabajo	100%	1	1
19	Diseñar e implementar formulario Eliminar puesto de trabajo Implementar funcionalidad Eliminar puesto de trabajo	100%	1	1

4.1.2. Pila de Tareas de la Iteración 2

Tabla 5. Pila de tareas de la Iteración 2

Pila de Tareas de la Iteración 2	**Responsable:** ***Ariam Figueredo Rojas*** **Duración:** *20 días*			
ID	**Lista de Tareas**	**Estado**	**Estima**	**Real**
20	Diseñar e implementar formulario Insertar tiempo de máquina Implementar funcionalidad Insertar tiempo de máquina	100%	1	1
21	Diseñar e implementar formulario Modificar tiempo de máquina Implementar funcionalidad Modificar tiempo de máquina	100%	2	1
22	Diseñar e implementar formulario Eliminar tiempo de máquina Implementar funcionalidad Eliminar tiempo de máquina	100%	2	1
23	Diseñar e implementar formulario Registrar entrada a persona interna Implementar funcionalidad Registrar entrada a persona interna	100%	5	4
24	Diseñar e implementar formulario Registrar entrada de persona externa Implementar funcionalidad Registrar entrada de persona	100%	2	1

	externa			
25	Diseñar e implementar formulario Registrar salida de persona	100%	2	1
	Implementar funcionalidad Registrar salida de persona			
26	Diseñar e implementar formulario Consultar datos del personal de un proyecto	100%	3	2
	Implementar funcionalidad Consultar datos del personal de un proyecto	100%	2	1
27	Diseñar e implementar formulario Consultar ocupación de los laboratorios	100%	1	1
	Implementar funcionalidad Consultar ocupación de los laboratorios.			
28	Diseñar e implementar formulario Consultar datos planificación del tiempo de máquina	100%	4	3
	Implementar funcionalidad Consultar datos planificación del tiempo de máquina			
29	Diseñar e implementar formulario Consultar datos de trayectoria de asistencia de personas	100%	4	3
	Implementar funcionalidad Consultar datos de trayectoria de asistencia de personas			

4.2. Estándares de programación

Como XP enfatiza más la parte relativa al código, es indispensable que se sigan ciertos estándares de programación. Estos mantienen el código legible para los miembros del equipo, facilitando los cambios y su mantenimiento. A continuación se describen estándares utilizados en la implementación de la propuesta de solución.

4.2.1. Ficheros

- **Organización de ficheros:** Un fichero consta de secciones separadas por líneas en blanco y líneas de comentario, ningún fichero debe tener más de 2000 líneas de código.
- **Ficheros de código fuente:** Cada fichero contiene una única clase o interfaz.
- **Comentarios:** Los comentarios deben añadir claridad al código, ser concisos.
- **Número de declaraciones por línea:** Se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

- **Sentencias simples:** Cada línea debe contener una sola sentencia.
- **Líneas en blanco:** Se debe usar una línea en blanco entre métodos, variables locales de un método y la primera sentencia, y antes de un bloque o un comentario de línea.

4.2.2. Convenciones y estándares de nombres

- **Notación Camell:** El primer carácter de todas las variables y excepto el de la primera palabra, se escribe en mayúsculas y los otros caracteres en minúsculas.

```
proveedorCadenaConexion = new ProveedorCadenaConexion();
```

Figura 22. Ejemplo de la utilización de la notación Camell

- **Notación Pascal:** El primer carácter de todas las palabras que dan nombre a métodos y clases se escriben con mayúscula y el resto de los caracteres con minúscula.
- Usar el prefijo "I" con notación Pascal para las interfaces.

```
public class CapitalHumanoService : ICapitalHumanoService
{

}
```

Figura 23. Ejemplo de la utilización de la notación Pascal.

- Usar palabras entendibles y descriptivas para nombrar a las variables.

4.3. Pruebas

Para comprobar las funcionalidades de la aplicación, se realizaron pruebas durante todo el proceso de desarrollo del software. Se decidió adoptar la forma en que son efectuadas estas por la metodología XP, puesto que representa uno de los elementos fundamentales de la misma y establece bien las directivas para su empleo. Ello permitirá mejorar el sistema durante el desarrollo estrechando así el tiempo transcurrido entre la aparición de un error y su detección.

XP divide las pruebas del sistema en dos grupos, por un lado se realizan pruebas unitarias, diseñadas antes de comenzar el código, con el objetivo de verificar el mismo a medida que se vaya implementando; y por otro lado realiza pruebas de aceptación destinadas a evaluar si al final de una iteración se consiguió cumplir con las expectativas del cliente.

4.3.1. Pruebas unitarias

Estas pruebas se conciben antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema, además, aunque no generan artefactos y no son directamente

palpables para el cliente, son de vital importancia para el desarrollo de un proyecto. Entre las principales ventajas de la utilización de estas pruebas, se encuentran: brinda la posibilidad de saber cómo se está desarrollando el trabajo en la programación ofreciendo esto una retroalimentación. Permite saber si se puede agregar o no una determinada funcionalidad sin alterar el funcionamiento del sistema; así como la realización de cambios de forma segura. [26]

Haciendo uso del Visual Studio se logró llevar a cabo esta tarea, gracias a las facilidades que brinda este IDE en la realización de pruebas unitarias, las cuales se realizaron sobre cada una de las funcionalidades implementadas. Para aplicar la prueba se generaron primero los resultados esperados, en la Figura 21 se muestra un ejemplo del código creado para la prueba al método "*ObtenerPorId*" de la clase "*PersonaController*", donde se guarda en la variable "*actual*" el valor de la consulta al método, para luego comparar y/o evaluar el resultado arrojado con el esperado.

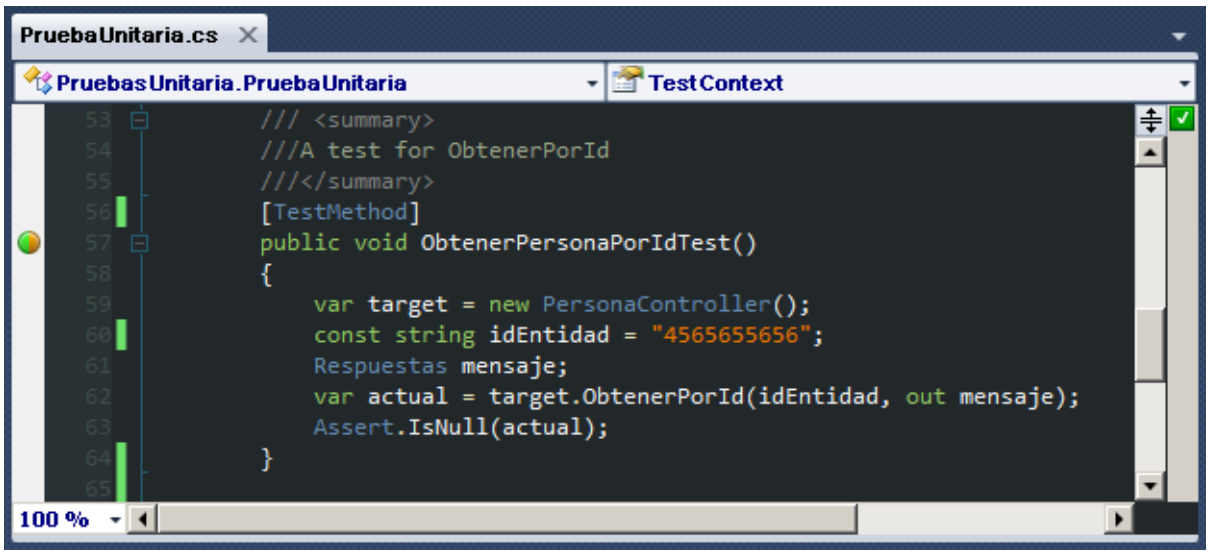

Figura 24. Código de una de las pruebas realizadas.

En la siguiente figura se muestra el resultado obtenido de la prueba anteriormente descrita.

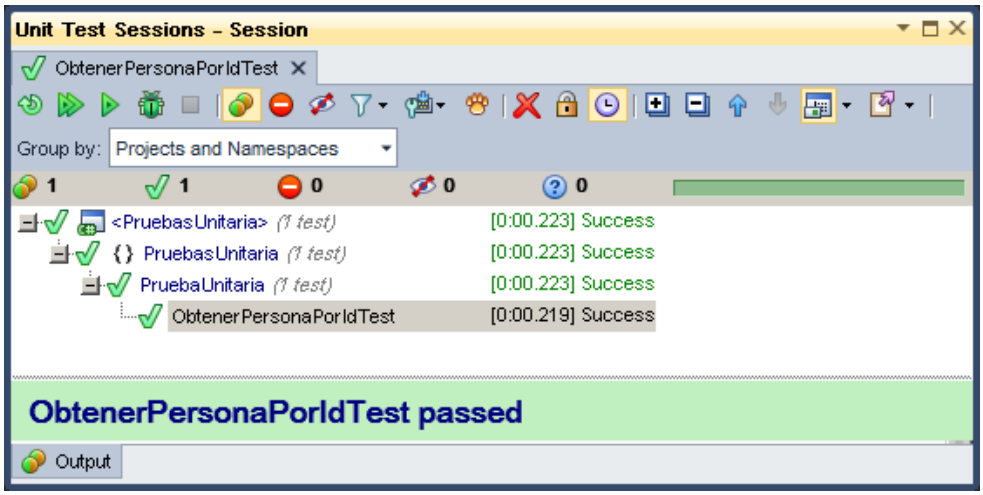

Figura 25. Resultado obtenido una vez aplicada la prueba unitaria

Con el empleo de este tipo de pruebas se garantizó que las funcionalidades exigidas por el cliente se implementaran correctamente, puesto que se comprobaron los resultados obtenidos al instante, disminuyendo el número de errores no detectados.

4.3.2. Pruebas de aceptación

Las funcionalidades seleccionadas para cada iteración son desarrolladas y probadas en un ciclo de iteración de acuerdo al orden preestablecido. Al comienzo de cada ciclo se planifica la iteración. Cada funcionalidad se traduce en tareas específicas de programación estableciéndoseles pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores.

Las pruebas de aceptación son llevadas a cabo por el cliente y usuarios finales de la aplicación, siendo probadas cada una de las funcionalidades exigidas y descritas con anterioridad. Son más importantes que las pruebas unitarias puesto que reflejan la satisfacción del cliente con el producto desarrollado. Se realizan a la par del desarrollo del sistema y se adaptan a los cambios que surjan, estableciendo así en las iteraciones la conclusión de una y el inicio de la siguiente. Luego de ser superadas las pruebas de aceptación puede considerarse que la aplicación se encuentra lista para utilizarse y realizar su respectivo despliegue. [26]

Como resultado de las pruebas de aceptación se obtienen artefactos descritos en tablas, estas contarán con los siguientes campos:

Código: identificador de la prueba y a su vez hará referencia al nombre de la misma.

Funcionalidad: nombre de la funcionalidad de la Pila de Tareas del Producto a la que hace referencia la prueba a realizar.

Nombre: nombre de la prueba a realizar.

Descripción: descripción de la prueba a realizar.

Condiciones de Ejecución: condiciones que deben cumplirse al llevar a cabo la prueba.

Entradas / Pasos de Ejecución: descripción de cada uno de los pasos a seguir en el desarrollo de la prueba, tiene en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.

Resultado esperado: breve descripción del resultado que se espera obtener con la prueba realizada.

Evaluación de la prueba: evaluación emitida sobre la prueba realizada. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:

- **Satisfactoria:** el resultado de la prueba es el esperado.
- **Parcialmente satisfactoria:** el resultado no es completamente el esperado por el cliente o usuario de la aplicación y mostrando resultados erróneos o fuera de contexto.
- **Insatisfactoria:** el resultado de la prueba realizada genera un error de codificación en la aplicación, se muestran elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad quede invalidada.

A continuación se expone uno de los casos de pruebas de aceptación llevados a cabo según propone XP (para ver el resto dirigirse al Anexo 1):

Tabla 6. Prueba de aceptación Autenticar usuario

Prueba de Aceptación	
Código: PAF1.15	**Funcionalidad:** *Autenticar usuario*
Nombre: Autenticar usuario	
Descripción: Una vez ejecutada la aplicación se mostrará un formulario para introducir el usuario y clave de la persona que va a interactuar con el sistema. Si la persona no se encuentra registrada en el sistema no podrá acceder al mismo, por otro lado, en caso de estarlo y de ser correctos los valores introducidos, el sistema muestra la interfaz que le corresponde al usuario autenticado según el rol que posea.	

Condiciones de Ejecución: Ninguna
Entradas / Pasos de Ejecución: El usuario se autentica en el sistema.
Resultado esperado: Se espera que se muestre la interfaz correspondiente al rol del usuario.
Evaluación de la prueba: Satisfactoria

Aplicadas las pruebas de aceptación, el cliente para el cual se desarrolló el sistema informático pudo comprobar el cumplimiento satisfactorio de cada una de las funcionalidades demandadas, quedando de esta forma la aplicación apta para su empleo.

Conclusiones del Capítulo

- Se obtuvieron los artefactos generados por las metodologías empleadas, como es el caso de la pila de tareas de las iteraciones (Scrum) y las pruebas de aceptación (XP), que permitieron dirigir el proceso de implementación de las funcionalidades del sistema así como la comprobación de las mismas.
- Se establecieron estándares de codificación a emplear en la implementación del sistema informático obteniéndose una mejor organización del código generado.
- Se obtuvieron resultados satisfactorios al aplicar diferentes tipos de pruebas al sistema informático desarrollado, quedando el mismo listo para su uso.

CONCLUSIONES GENERALES

Con la realización de este trabajo investigativo se arribó a las siguientes conclusiones:

- Se realizó un estudio del estado del arte sobre diferentes sistemas informáticos existentes dirigidos a la gestión de información de recursos humanos y medios tecnológicos, comprobándose que dichos sistemas no engloban las especificidades del cliente en su totalidad, de ahí la necesidad del desarrollo de la solución propuesta.
- Se identificaron las herramientas y tecnologías a emplear en la construcción del sistema.
- Se capturaron los principales requerimientos funcionales según las necesidades del cliente y se determinaron las tareas de implementación a desarrollar.
- Se implementó SICREL, un sistema informático que permite la gestión de información de recursos humanos y medios tecnológicos de los laboratorios pertenecientes al centro ISEC.
- Al realizar pruebas unitarias y de aceptación, se obtuvieron resultados satisfactorios según los indicadores de evaluación, quedando demostrado que el sistema informático propuesto reúne las condiciones tecnológicas para su uso.

RECOMENDACIONES

- Extender el sistema a otros centros de la universidad.
- Extender el sistema de manera que pueda ser empleado en las diferentes facultades regionales pertenecientes a la universidad.
- Publicar el presente trabajo para que sirva de apoyo a investigaciones futuras.

REFERENCIAS BIBLIOGRÁFICAS

1. Alicante, E.P.S.d.l.U.d. *Servicio de Informática de la EPS*. 2012 [cited 2013 16 Enero]; Available from: https://maktub.eps.ua.es/servicios/laboratorios/visualizar/informalistalab.phtml.
2. ASSETS. *Sistema de Gestión Integral*. 2012 [cited 2013 17 Enero]; Available from: http://www.assets.co.cu/assets.asp.
3. Datazucar. *Datazucar*. 2013 [cited 2013 18 Enero]; Available from: http://www.datazucar.cu/?p=1.
4. Palacio, J., *Flexibilidad con Scrum*. 2008: SafeCreative. 183.
5. Canós, J.H., P. Letelier, and M.C. Penadés, *Método logías Ágiles en el Desarrollo de Software.* Universidad Politécnica de Valencia, 2010.
6. Calderón, S.D.A. and J.C.V. Rebaza, *Metodologías Ágiles*, in *Facultad de Ciencias Físicas y Matemáticas*. 2010, Universidad Nacional de Trujillo: Trujillo. p. 37.
7. Palacio, J., *ScrumManager: Gestión de proyectos*. 2008: SafeCreative.
8. Aguanno, K., PMP®, and MAPM, *Managing Agile Projects*. Primera ed, Ontario, Canada: Multi-Media Publications Inc.
9. Hernán, S.M., *Diseño de una Metodología Ágil de Desarrollo de Software.*, Universidad de Buenos Aires.
10. Kniberg, H., *Scrum y XP desde las trincheras. Cómo hacemos Scrum*. C4Media ed. 2007: InfoQ.
11. Nagel, C., et al., *Professional C# 4 and .NET 4*. 2010, Indianapolis: Wiley Publishing, Inc.
12. Mackey, A., *Introducing .NET 4.0 With Visual Studio 2010. A fast-track introduction to the new features of .NET 4.0, Visual Studio 2010, and their supporting technologies*. 2010, New York: Paul Manning.
13. Johnson, B., P. Madziak, and S. Morgan, *Microsoft .NET Windows Communication Foundation*. 2009, Washington: Microsoft Press.
14. Sharp, J., *Windows® Communication Foundation 4 Step by Step*. 2010, California: O'Reilly Media, Inc. .
15. MacDonald, M., *Pro WPF in C# 2010:Windows Presentation Foundation in .NET 4.0*. 2010.
16. Klein, S., *Pro Entity Framework 4.0*. 2010, New York: Paul Manning.
17. Pialorsi, P. and M. Russo, *Programming Microsoft® LINQ in Microsoft .NET Framework 4*. 2010, California: O'Reilly Media, Inc.

18. Mayo, J., *Microsoft ® Visual Studio® 2010A Beginner's Guide*. 2010, New York: The McGraw-Hill Companies, Inc.
19. Date, C.J., *Introducción a los Sistemas de bases de datos*. SÉPTIMA ed, México: Addison Wesley Longman, Inc.
20. DuBois, P., *MySQL Developer's Library*. Cuarta ed. 2009, Indianapolis: Pearson Education, Inc.
21. Llorente, C.d.l.T., et al., *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0* 2010: Microsoft Ibérica S.R.L.
22. Krafzig, D., K. Banke, and D. Slama, *Enterprise SOA: Service-Oriented Architecture Best Practices*: Prentice Hall PTR.
23. Delgado, A., *Metodología para desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)*, Montevideo.
24. Hall, G.M., *Pro WPF and Silverlight MVVM Effective Application Development with Model-View-ViewModel*. 2010, New York: Apress.
25. Larman, C., *UML y Patrones Introducción al análisis y diseiño orientado a objetos*: PRENTICE HALL.
26. Jurado, C.B., *Diseño Agil con TDD*. 2010: SafeCreative.

BIBLIOGRAFÍA

1. **Suehring, Steve.** MySQL Bible. New York : Wiley Publishing. ISBN: 0-7645-4932-4.

2. **Pialorsi, Paolo y Russo, Marco.** Programming Microsoft LINQ in Mcrosoft .NET Framework 4. . California : OTSI, 2010. 978-0-735-64057-3.

3. **Palacio, Juan.** Flexibilidad con Scrum. s.l. : sefecreative, 2007.

4. **Nagel, Christian.** C# 4 and .NET 4. s.l. : Wiley Publishing. 978-0-470-50225-9.

5. **Molpeceres, Alberto.** Procesos de desarrollo: RUP, XP y FDD. s.l. : javaHispano, 2002.

6. **Miers, Derek y White, Stephen A.** Guía de Referencia y Modelado BPMN. s.l. : Future Strategies. 978-0-9819870-3-3..

7. **Mayo, Joe.** Microsoft Visual Studio 2010. A Beginner´s Guide. s.l. : Mc Graw Hill, 2010. 978-0-07-166896-5.

8. **Martín, Enrique.** BPM y SOA.Alineando SOA y BPM. s.l. : Systems Iberia.

9. **MacDonald, Matthew.** Pro WPF in C# 2010. Windows Presentation Foundation in .NET 4. . . New York : Apress, 2010. 978-14302-7204-5.

10. **Letelier, Patricio y Penadés, Mª Carmen.** Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Valencia : Universidad Politécnica de Valencia.

11. **Kniberg, Henrik.** Scrum y XP desde las trincheras. s.l. : InfoQ, 2007. 978-1-4303-2264-1.

12. **Klein, Scott.** Pro Entity Framework 4.0. New York : Apress. 2010. 978-1-4302-0648-4.

13. **Esposito, Dino.** Programming Microsoft ASP.NET 4. Washington : Microsoft Press, 2011. 978-0-7356-4338-3.

14. **DuBois, Paul.** MySQL. s.l. : Developer's Library, 2009. ISBN-13: 978-0-672-32938-8.

15. **Delgado, Andrea, González, Laura y Piedrabuena, Federico.** Desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture). s.l. : Universidad de la República, Facultad de Ingeniería, Instituto de Computación.

16. **de la Torre Llorente, César,.** Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0. . s.l. : Krasis Press, 2010. ISBN: 978-84-936696-3-8..

17. **Cibraro, Pablo.** Professional WCF 4. Windows Communication Foundation whit .NET 4. . Indianapolis : Wiley Publishing, 2010. 978-0-470-16846-2.

18. **Ble Jurado, Carlos.** Diseño Agil con TDD. s.l. : safeCreative, 2010. ISBN:978-1-4452-6471-4.

19. **Amaro Calderón, Sarah Dámaris y Valverde Rebaza, Jorge Carlos.** Metodologías Ágiles. Trujillo : Universidad Nacional de Trujillo, 2007.

20. Introducción a los sistemas de bases de datos . Massachusetts. : Addison Wesley Longman, Inc. 968-444-419-2.

21. **Larman, Craig .** UML y Patrones Introducci6n a1 an6lisis y diseiio orientado a objetos. MCxic : Prentice Hall, Inc. 0-13-748880-7.

22. **Mackey, Alex.** Introducing .NET 4.0 With Visual Studio 2010. New York : Springer-Verlag New York, Inc., 2010. 978-1-4302-2456-3 .

23. **Jennings, Roger.** Professional ADO.NET 3.5 with LINQ and the Entity Framework. Indianapolis : Wiley Publishing, Inc., 2009. 978-0-470-18261-1.

24**. Krafzig, D., K. Banke, and D. Slama**, *Enterprise SOA: Service-Oriented Architecture Best Practices*: Prentice Hall PTR.

25. **Mayo, J.,** *Microsoft® Visual Studio® 2010A Beginner's Guide*. 2010, New York: The McGraw-Hill Companies, Inc.

26. **Date, C.J.**, *Introducción a los Sistemas de bases de datos*. SÉPTIMA ed, México: Addison Wesley Longman, Inc.

27. **Aguanno, K**., PMP®, and MAPM, *Managing Agile Projects*. Primera ed, Ontario, Canada: Multi-Media Publications Inc.

28. **Hernán, S.M.,** *Diseño de una Metodología Ágil de Desarrollo de Software.*, Universidad de Buenos Aires.

29. **Hall, G.M.,** *Pro WPF and Silverlight MVVM Effective Application Development with Model-View-ViewModel*. 2010, New York: Apress.

30. **Sharp, J.,** *Windows® Communication Foundation 4 Step by Step*. 2010, California: O'Reilly Media, Inc. .

ANEXOS

Anexo 1. Pruebas de aceptación realizadas al sistema

En el presente anexo se presentarán las pruebas de aceptación realizada a cada una de las funcionalidades del sistema.

Tabla 7. Prueba de aceptación Insertar Docente

Prueba de Aceptación	
Código: PAF1.1	**Funcionalidad:** Insertar docente
Nombre: Insertar docente	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de insertar docente, en la pestaña Gestionar Docente. Al seleccionar la opción Insertar se mostrará una interfaz con los elementos para realizar la inserción de un docente. El administrador introduce el nombre del docente, da clic en insertar y automáticamente el sistema procede a inserta los datos referentes al nuevo docente, mostrando un mensaje con el resultado de la acción realizada.	
Condiciones de Ejecución: Ninguna	
Entradas / Pasos de Ejecución: - Se introduce el nombre del docente nuevo. - Se da clic en Insertar - El sistema muestra un mensaje notificando el resultado de la acción	
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada	
Evaluación de la prueba: Satisfactoria.	

Tabla 8. Prueba de aceptación Modificar Docente

Prueba de Aceptación	
Código: PAF1.2	**Funcionalidad:** Modificar docente
Nombre: Modificar docente	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de modificar docente, en la pestaña Gestionar Docente. Al seleccionar la opción Modificar se mostrará una interfaz con los elementos para realizar la modificación de un docente determinado. El administrador selecciona el docente a modificar, automáticamente el sistema muestra los datos referentes al docente, permitiendo la modificación de estos por parte del usuario.	

Condiciones de Ejecución: Debe haberse insertado previamente al menos un docente
Entradas / Pasos de Ejecución: - Se selecciona el docente - El sistema muestra los datos correspondientes - El usuario modifica los datos, luego da clic en el Modificar. - El sistema muestra un mensaje notificando el resultado de la acción
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada
Evaluación de la prueba: Satisfactoria.

Tabla 9. Prueba de aceptación Eliminar Docente

Prueba de Aceptación	
Código: PAF1.3	**Funcionalidad:** Eliminar docente
Nombre: Eliminar docente	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de eliminar docente, en la pestaña Gestionar Docente. Al seleccionar la opción Eliminar se mostrará una interfaz con una lista de los docentes existentes. El administrador da clic en el Eliminar del elemento de la tabla y automáticamente el sistema procede a eliminar el docente del sistema, mostrando un mensaje con el resultado de la acción realizada.	
Condiciones de Ejecución: Debe haberse insertado previamente al menos un docente	
Entradas / Pasos de Ejecución: - Se da clic sobre el elemento Eliminar de la lista de docentes correspondiente al docente sobre el que se desee realizar la operación - El sistema muestra un mensaje notificando el resultado de la acción	
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada	
Evaluación de la prueba: Satisfactoria	

Tabla 10. Prueba de aceptación Insertar Proyecto

Prueba de Aceptación	
Código: PAF1.4	**Funcionalidad:** Insertar proyecto
Nombre: Insertar proyecto	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de insertar proyecto, en la pestaña Gestionar Proyecto. Al seleccionar la opción Insertar se mostrará una interfaz con los	

elementos para realizar la inserción de un proyecto. El administrador introduce el nombre del proyecto, da clic en insertar y automáticamente el sistema procede a inserta los datos referentes al nuevo proyecto, mostrando un mensaje con el resultado de la acción realizada.
Condiciones de Ejecución: Ninguna
Entradas / Pasos de Ejecución: - Se introduce el nombre del proyecto nuevo. - Se da clic en Insertar - El sistema muestra un mensaje notificando el resultado de la acción
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada
Evaluación de la prueba: Satisfactoria.

Tabla 11. Prueba de aceptación Modificar Proyecto

Prueba de Aceptación	
Código: PAF1.5	**Funcionalidad:** Modificar proyecto
Nombre: Modificar proyecto	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de modificar proyecto, en la pestaña Gestionar Proyecto. Al seleccionar la opción Modificar se mostrará una interfaz con los elementos para realizar la modificación de un proyecto determinado. El administrador selecciona el proyecto a modificar, automáticamente el sistema muestra los datos referentes al proyecto, permitiendo la modificación de estos por parte del usuario.	
Condiciones de Ejecución: Debe haberse insertado previamente al menos un proyecto	
Entradas / Pasos de Ejecución: - Se selecciona el proyecto - El sistema muestra los datos correspondientes - El usuario modifica los datos, luego da clic en el Modificar. - El sistema muestra un mensaje notificando el resultado de la acción	
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada	
Evaluación de la prueba: Satisfactoria**.**	

Tabla 12. Prueba de aceptación Eliminar Proyecto

Prueba de Aceptación	
Código: PAF1.6	**Funcionalidad:** Eliminar proyecto

Nombre: Eliminar proyecto
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de eliminar proyecto, en la pestaña Gestionar Docente. Al seleccionar la opción Eliminar se mostrará una interfaz con una lista de los proyecto existentes. El administrador da clic en el Eliminar del elemento de la tabla y automáticamente el sistema procede a eliminar el proyecto del sistema, mostrando un mensaje con el resultado de la acción realizada.
Condiciones de Ejecución: Debe haberse insertado previamente al menos un proyecto
Entradas / Pasos de Ejecución: - Se da clic sobre el elemento Eliminar de la lista de proyectos correspondiente al proyecto sobre el que se desee realizar la operación - El sistema muestra un mensaje notificando el resultado de la acción
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada
Evaluación de la prueba: Satisfactoria

Tabla 13. Prueba de aceptación Insertar Laboratorio

Prueba de Aceptación	
Código: PAF1.7	**Funcionalidad:** Insertar laboratorio
Nombre: Insertar laboratorio	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de insertar laboratorio, en la pestaña Gestionar Laboratorio. Al seleccionar la opción Insertar se mostrará una interfaz con los elementos para realizar la inserción de un laboratorio. El administrador selecciona el docente en el cual se encuentra el laboratorio e introduce el número de este, da clic en insertar y automáticamente el sistema procede a inserta los datos referentes al nuevo laboratorio, mostrando un mensaje con el resultado de la acción realizada.	
Condiciones de Ejecución: Ninguna	
Entradas / Pasos de Ejecución: - Se selecciona el docente y se introduce el número del laboratorio nuevo. - Se da clic en Insertar - El sistema muestra un mensaje notificando el resultado de la acción	
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada	
Evaluación de la prueba: Satisfactoria.	

Tabla 14. Prueba de aceptación Modificar Laboratorio

Prueba de Aceptación	
Código: PAF1.8	**Funcionalidad:** Modificar laboratorio
Nombre: Modificar laboratorio	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de modificar laboratorio, en la pestaña Gestionar Laboratorio. Al seleccionar la opción Modificar se mostrará una interfaz con los elementos para realizar la modificación de un laboratorio determinado. El administrador selecciona el proyecto, acto seguido selecciona el laboratorio a modificar, automáticamente el sistema muestra los datos referentes al laboratorio, permitiendo la modificación de estos por parte del usuario.	
Condiciones de Ejecución: Debe haberse insertado previamente al menos un laboratorio	
Entradas / Pasos de Ejecución: - Se selecciona el docente - Se selecciona el laboratorio - El sistema muestra los datos correspondientes - El usuario modifica los datos, luego da clic en el Modificar. - El sistema muestra un mensaje notificando el resultado de la acción	
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada	
Evaluación de la prueba: Satisfactoria**.**	

Tabla 15. Prueba de aceptación Eliminar Laboratorio

Prueba de Aceptación	
Código: PAF1.9	**Funcionalidad:** Eliminar laboratorio
Nombre: Eliminar laboratorio	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de eliminar laboratorio, en la pestaña Gestionar Laboratorio. Al seleccionar la opción Eliminar se mostrará una interfaz en la cual debe seleccionar el docente donde se encuentra el/los laboratorios a eliminar, automáticamente el sistema muestra una lista con los laboratorios existentes en dicho docente. El administrador da clic en el Eliminar del elemento de la tabla y automáticamente el sistema procede a eliminar el proyecto del sistema, mostrando un mensaje con el resultado de la acción realizada.	
Condiciones de Ejecución: Debe haberse insertado previamente al menos un laboratorio	

Entradas / Pasos de Ejecución: - Se selecciona un docente - El sistema muestra automáticamente un listado con los laboratorios pertenecientes a dicho docente. - Se da clic sobre el elemento Eliminar de la lista de laboratorios correspondiente al laboratorio sobre el que se desee realizar la operación - El sistema muestra un mensaje notificando el resultado de la acción
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada
Evaluación de la prueba: Satisfactoria

Tabla 16. Prueba de aceptación Insertar persona

Prueba de Aceptación	
Código: PAF2.0	**Funcionalidad:** Insertar persona
Nombre: Insertar persona	
Descripción: Una vez autenticado un administrador, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de insertar persona, en la pestaña Gestionar Proyecto. Al seleccionar la opción Insertar se mostrará una interfaz con los elementos para realizar la inserción de una persona. El administrador introduce el solapín de la persona, da clic en insertar y automáticamente el sistema procede a inserta los datos referentes al nuevo proyecto, mostrando un mensaje con el resultado de la acción realizada.	
Condiciones de Ejecución: Ninguna	
Entradas / Pasos de Ejecución: - Se introduce el solapín de la persona nueva. - Se da clic en Insertar - El sistema muestra un mensaje notificando el resultado de la acción	
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada	
Evaluación de la prueba: Satisfactoria.	

Tabla 17. Prueba de aceptación Registrar entrada a persona interna

Prueba de Aceptación	
Código: PAF2.1	**Funcionalidad:** Registrar entrada a persona interna

Nombre: Registrar entrada a persona interna
Descripción: Una vez autenticado un técnico de laboratorio, este tendrá acceso a las funcionalidades que le corresponde entre ellas está la de registrar la entrada de personas internas, en la pestaña Registrar entrada. Al seleccionar la opción Registrar Entrada Persona Interna se mostrará una interfaz con los elementos para realizar el registro de una persona interna que accede al laboratorio. El técnico introduce el solapín de la persona, da clic en Marcar Entrada y automáticamente el sistema procede a registrar los datos referentes al nuevo proyecto, mostrando un mensaje con el resultado de la acción realizada.
Condiciones de Ejecución: Ninguna
Entradas / Pasos de Ejecución: - Se introduce el solapín de la persona nueva. - Se da clic en Marcar Entrada - El sistema muestra un mensaje notificando el resultado de la acción
Resultado esperado: Se espera un mensaje indicando el resultado de la acción realizada
Evaluación de la prueba: Satisfactoria.