

SISTEMA DE INFORMACIÓN PARA EL APOYO A LA PRÁCTICA MÉDICA DE LA ORTODONCIA IMPLEMENTADO EN TECNOLOGÍAS .NET

Saida Rocío Higuera Ramírez
Universidad Industrial de Santander, Bucaramanga, Santander, Colombia.
say2032@gmail.com

Mary Elizabeth Ramírez Sanguino
Universidad Industrial de Santander, Bucaramanga, Santander, Colombia
mary0605@gmail.com

Henry Argüello Fuentes
Universidad Industrial de Santander, Bucaramanga, Santander, Colombia
henarfu@uis.edu.co

ABSTRACT

With the completion of this project succeeded in creating a software tool that was developed using the advantages provided to us. .NET technologies for the realization of both desktop applications such as mobile, also achieving select and implement an agile methodology that would reduce times to achieve the objectives that were set to initially keep the updated according to the changing needs of the client and do not require large quantities of documents only necessary for project traceability.

Finally, a partition tests such as equivalent boundary value analysis and documentation that allowed us to detect and correct any errors and optimize tool.

Keywords

.NET, Agile Unified Process, Central Application, Mobile Application, Mobile, Devices, Wi-Fi.

RESUMEN

Con la realización de este proyecto se logró crear una herramienta software que fue desarrolla aprovechando las ventajas que nos brinda las Tecnologías .NET tanto para la realización de aplicaciones de escritorio como de móviles, también se logro seleccionar e implementar una metodología ágil que permitiera reducir los tiempos para alcanzar los objetivos que se trazaron al inicialmente, mantener los requisitos actualizados de acuerdo a las necesidades variables del cliente y que no exigiera realizar grandes cantidades de documentos solo los necesarios para llevar la trazabilidad del proyecto.

Finalmente se realizaron una pruebas tales como partición equivalente, análisis valor límite y de documentación que nos permitieron detectar y

corregir algunos errores de la herramienta y optimizarla.

Palabras Claves

.NET, Aplicación central, Aplicación Móvil, Dispositivos Móviles, Proceso Ágil Unificado (AUP), Sistema de Información, Wi-Fi.

1. INTRODUCCIÓN

Actualmente las ciencias de la SALUD están viendo lo importante e indispensable que son los sistemas de información para su práctica médica,

Los especialistas en el área de la ortodoncia plantean la necesidad de contar con una herramienta que les permita administrar la información de los pacientes en el momento que se precise. Pensando en esto el reto que se presenta es el de diseñar un sistema de información centrado en la ortodoncia que le permita al usuario de manera amigable gestionar la historia clínica de cada paciente.

Para dar solución a la problemática planteada inicialmente se investigó acerca de los diferentes lenguajes de programación, tales como Java y .NET¹, con los cuales se realizó un contraste entre los beneficios que cada uno de estos brinda, teniendo en cuenta que el objetivo principal del proyecto era el de realizar un sistema que contara con dos aplicaciones una para escritorio y otra móvil. Como resultado se obtuvo que las herramientas de .NET se adaptaban mejor a las exigencias del sistema.

De igual forma después de seleccionar las herramientas HW y SW fue necesario elegir la metodología con la cual desarrollaríamos el proyecto, esta metodología debía tener en cuenta la constante intervención de los usuarios y el corto plazo para la entrega de resultados.

Durante este proceso se evaluaron diferentes metodologías como XP (eXtremme Programming), RUP (Rational Unified Process), AUP (Agile Unified Process), siendo esta ultima la que se decidió usar en este proyecto, dado que es un modelo de proceso hibrido que captura algunas de las mejores prácticas de desarrollo de software y tiene como objetivo principal desarrollar un software de la mayor calidad que satisfaga las necesidades de los usuarios finales.

2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO

Microsoft SQL Server 2005[7]: Sistema empleado para la gestión de bases de datos, implementa un servicio escalable y disponible para el almacenamiento, procesamiento y seguridad de los datos.

Microsoft SQL Server Management Studio Express [8]: Herramienta de administración gráfica para SQL Server 2005 Express Edition.

Visual C Sharp 2008[9]: Lenguaje de programación utilizado para el desarrollo de la herramienta.

IDE Microsoft Visual Studio .Net 2008[9]: Entorno de desarrollo integrado utilizado en la realización del proyecto.

Microsoft .Net Framework 3.5 [9]: Paquete de rutinas empleado para la ejecución de aplicaciones .NET.

Microsoft .Net Compact Framework 3.5 [9]: Paquete de rutinas empleado para la ejecución de aplicaciones .NET en dispositivos móviles.

Microsoft Active Sync [9]: Herramienta empleada para el proceso de sincronización entre la aplicación móvil y la de escritorio.

3. METODOLOGÍA

Escoger la metodología de desarrollo fue uno de los temas que más ocupo tiempo de investigación en la etapa preliminar del trabajo, puesto que de ella dependía la forma de trabajo durante todas las etapas del desarrollo del proyecto.

Se quería usar una metodología que permitiera reducir los tiempos para lograr los objetivos propuestos y la constante intervención de los usuarios, se evaluaron metodologías ágiles como

XP (eXtremme Programming)[2], RUP (Rational Unified Process)[3], AUP (Agile

Unified Process)[4], siendo esta ultima la que se decidió usar en este proyecto.

AUP fue escogida por las características especificadas en la documentación, ya que permite mantener los requisitos actualizados de acuerdo a las necesidades variables del cliente, no exige realizar grandes cantidades de documentos solo se realizan los necesarios para llevar la trazabilidad del proyecto, está enfocada para grupos pequeños de trabajo donde los roles no son considerados como puestos solo como asignaciones que puede realizar varias personas y cada persona puede ser parte de varios roles.

4. ANALISIS Y DISEÑO

En el desarrollo de la fase de análisis se obtuvo el documento de Especificación de Requisitos Software (ERS) del sistema de Información, que sigue las recomendaciones del Estándar IEEE 830 en cuanto a la organización de los distintos apartados así como el contenido de cada uno de ellos, y es el resultado de toda la información recolectada en las reuniones realizadas con los usuarios del sistema durante la fase de análisis, también se obtuvo el modelo de casos de uso del negocio, -los casos de uso son funcionalidades implementadas relacionando la interacción del usuario-, los actores del sistema de información y los casos de uso en su formato extendido.

4.1 REQUISITOS DEL SISTEMA

4.1.1 Actores del Sistema

El listado de actores detectados para el sistema de Información Bandelette:

Tabla 1. Actores del Sistema

ACT1	Administrador General
ACT2	Médico Ortodoncista
ACT3	Auxiliar
ACT4	Secretaria

4.1.2 Requisitos de Interfaz Externo

Requerimientos de la Interfaz Externo tenidos en cuenta para la realización de la aplicación:

Tabla 2. Requisitos de Interfaz Externo

RI1	Ítems del Menú Principal de la aplicación de escritorio
RI2	Ítems del Menú Principal de la aplicación móvil
RI3	Ítems del sub-Menú de Administración
RI4	Ítems del sub-Menú Pacientes
RI5	Ítems del sub-Menú Agenda
RI6	Ítems del sub-Menú Informes
RI7	Formato de la Historia Clínica

Para ampliar al detalle de cada una de los requisitos de interfaz externos anteriormente nombrados se pueden consultar en [1].

4.1.3 Requisitos Funcionales

Los requerimientos funcionales resultantes del levantamiento de información son:

Tabla 3. Requisitos Funcionales

RF1	Administración de Usuarios
RF2	Gestión de la Historia Clínica
RF3	Gestión del Control del Tratamiento
RF4	Generación de Reportes
RF5	Mantenimiento del Sistema
RF6	Sincronización de Datos
RF7	Gestión de los datos personales del paciente
RF8	Administración de Dispositivos
RF9	Autenticación de usuarios
RF10	Gestión de Imágenes

Para poder ampliar más detalladamente de cada uno de los requisitos funcionales se pueden consultar en [1].

4.1.4 Requisitos No Funcionales

Los requerimientos no funcionales resultantes del levantamiento de información se dividen en requisitos de rendimiento y restricciones de diseño, estos se muestran a continuación:

4.1.4.1 Requisitos de Rendimiento

Tabla 4. Requisitos de Rendimiento

RNF1	Almacenamiento de los Datos
RNF2	Número de dispositivos que soportan la aplicación

Para poder ampliar más detalladamente de cada uno de los requisitos de rendimiento consultar en [1].

4.1.5 Restricciones de Diseño

Tabla 5. Restricciones de Diseño

RNF3	Visualización de la información
RNF4	Resolución del Dispositivo Móvil
RNF5	Diseño y Usabilidad de la aplicación

Para poder ampliar más detalladamente cada una de las restricciones de Diseño consultar en [1].

4.2 CASOS DE USO DEL SISTEMA

A continuación se muestran los casos de uso que definen las funcionalidades que el usuario: Médico Ortodoncista para realizar

la administración de usuarios desde la aplicación de PC:

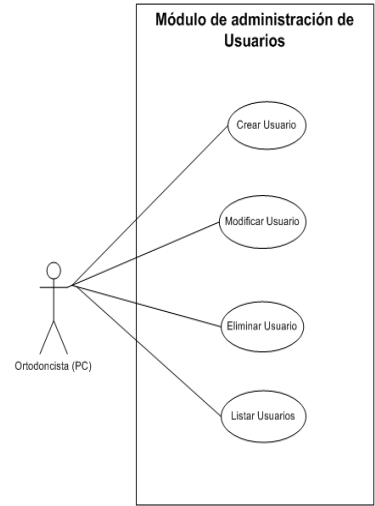


Figura 1. Diagrama de caso de uso para el modulo de Administración de Usuarios.

Para poder ampliar más detalladamente en los casos de uso que se tuvieron en cuenta para la realización del sistema consultar en [1].

4.3 ARQUITECTURA DE LA APLICACIÓN

Para la selección de la arquitectura de la aplicación se evaluaron varias arquitecturas pero finalmente las autoras del presente proyecto decidieron basarse en la arquitectura de 3 capas, dado que en este tipo de arquitectura la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC, el código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular, la separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

5.4 MODELO DE DATOS

Después de realizar la recolección de requisitos y los casos de uso del sistema, se paso a la fase de diseño donde se realizo el modelo de datos del sistema, que se muestra a continuación.

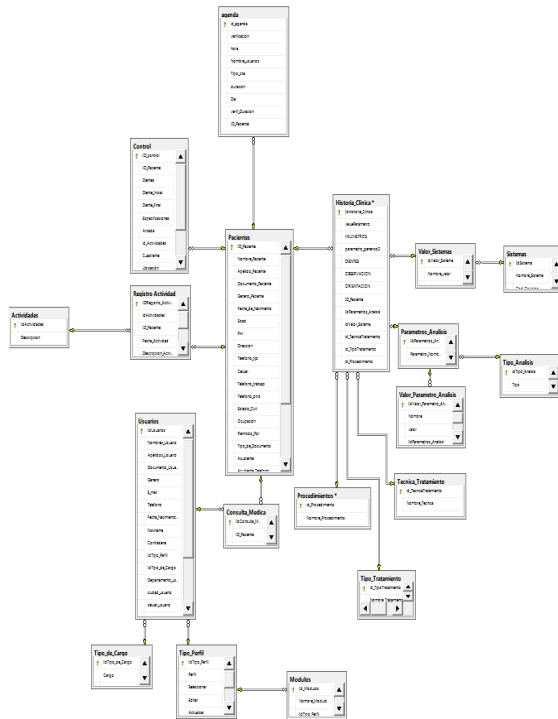


Figura 2. Modelo de Datos del Sistema de Información *BADELETTE*

5. RESULTADO FINAL

Como prototipo final se obtuvo una herramienta software que está compuesta por dos aplicaciones, una aplicación central para ordenadores de escritorio, y otra aplicación móvil.

El sistema de información lleva pronombre Bandedette y se ajusta a los requisitos del usuario brindándoles a éstos un acompañamiento total al momento de ejercer su profesión.

Para poder lograr lo anterior el sistema cuenta con dos formas de acceso a los datos:

Acceso cableado

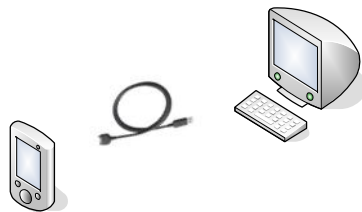


Figura 3. Acceso a datos cableado.

El acceso cableado se administra en la aplicación de escritorio y envía la información almacenada en el sistema de la aplicación central a la móvil y viceversa para lograr que los datos siempre estén actualizados.

Acceso en línea

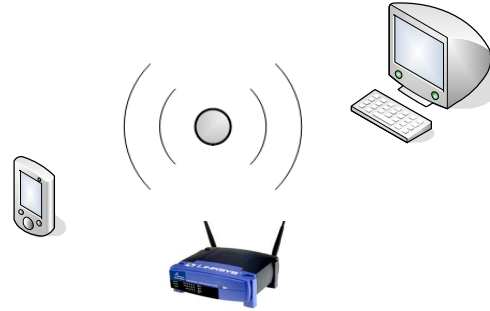


Figura 4. Acceso a datos WiFi.

Si el consultorio del usuario cuenta con un sistema inalámbrico de Internet, éste tiene la ventaja de poder acceder a la base de datos central y así manipular la información sin necesidad de tenerla almacenada localmente en el dispositivo.

5.1 PRUEBAS AL SISTEMA.

Para supervisar el funcionamiento del sistema fue necesario realizar ciertas pruebas durante y al final de la etapa de desarrollo del sistema con el fin de encontrar defectos y errores que el sistema pudiera presentar.

Las pruebas que se realizaron al sistema fueron las siguientes:

5.1.1 Prueba de Partición Equivalente

La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Este tipo de prueba trata el dominio de valores de entrada. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada [5].

Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

1. Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos no válidas.

2. Si una condición de entrada requiere un valor específico, se define una equivalencia válida y dos no válidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una no válida.
4. Si una condición de entrada es lógica, se define una clase de equivalencia válida y una no válida.

A partir de estas directrices se realizaron las clases de prueba de partición equivalente para cada elemento de los datos de entrada, los resultados de estas pruebas se pueden visualizar en la siguiente tabla:

Condición de entrada	Condición de entrada Válida	Condición de entrada inválida
Longitud de Contraseñas	1) ≥ 6	2) < 6
Longitud de documento	3) > 0	4) ≤ 0
Longitud de Nombre	5) > 0	6) ≤ 0
Edad	7) ≥ 1	8) < 0
Genero	9) M 10) F	10) Ninguno
Tipo de Documento	11) TI 12) CC 13) CE	14) Ninguno
Número de Teléfono	15) $= 0$ 16) ≥ 7	17) < 7
Número de Celular	18) ≥ 10	19) < 10
Longitud Nickname	20) ≥ 4	21) < 4
Sufijo es	22) Numérico	23) Otro valor
Sufijo es	24) Alfanumérico	25) Otro valor
Duración de la Cita	26) 15 min 27) 30min 28) 45 min	29) Ninguno

Tabla 6. Prueba Partición Equivalente

Código C.E Inválida	Salida
2)	El sistema muestra al usuario una alerta
4)	El sistema impide escribir valores negativos.
6)	El sistema notifica al usuario a través de una alerta
8)	El sistema impide escribir valores negativos.
10)	El sistema muestra al usuario una alerta.
14) 20)	El sistema le notifica al usuario a

22) 24)	través de una alerta
26)	El sistema impide ingresar datos que no sean numéricos
28)	El sistema impide ingresar datos que no sean alfanuméricos
32)	El sistema impide ingresar valor diferentes a 15min, 30min ó 45 min

Tabla 7. Prueba Equivalente

Para ver las especificaciones de esta prueba más a fondo remítase a [1].

5.1.2 Prueba: Análisis Valor Límite

Esta prueba complementa a la prueba de valor equivalente, pero en lugar de centrarse solo en los valores de entrada también el AVL el AVL obtiene casos de prueba también para el campo de salida [6].

Las directrices para esta prueba son las siguientes:

1. Si una condición de entrada especifica un rango delimitado por los valores a y b, se deben diseñar casos de prueba para los valores a y b, y para los valores justo por debajo y justo por encima de a y b, respectivamente.
2. Si una condición de entrada especifica un número de valores, se deben desarrollar casos de prueba que ejerciten los valores máximo y mínimo. También se deben probar los valores justo por encima y justo por debajo del máximo y del mínimo.
3. Aplicar las directrices 1 y 2 a las condiciones de salida. Por ejemplo, supongamos que se requiere una tabla de «temperatura / presión» como salida de un programa de análisis de ingeniería. Se deben diseñar casos de prueba que creen un informe de salida que produzca el máximo (y el mínimo) número permitido de entradas en la tabla.
4. Si las estructuras de datos internas tienen límites preestablecidos (por ejemplo, una matriz que tenga un límite definido de 100 entradas), hay que asegurarse de diseñar un caso de prueba que ejercite la estructura de datos en sus límites.

Para esta prueba se obtuvieron los resultados que se muestran en la siguiente tabla:

	C.P Válida	C.P Inválida
Días del	1) 1 2) 30	2) -1 3) 31

Mes*		
Días del Mes**	4) 1 5) 31	6) -1 7) 32

*Para los meses de Febrero, Abril, Junio, Septiembre, Noviembre.

**Para los meses de Enero, Marzo, Mayo, Julio, Agosto, Octubre, Diciembre.

Código C.P Inválida	Salida
2)	El sistema impide escribir números negativos
3)	Para los meses que tienen 30 días el sistema impide escribir números mayores de 30
6)	El sistema impide escribir números negativos
7)	Para los meses que tienen 31 días el sistema impide escribir números mayores de 31

Tabla 8. Prueba Análisis Valor Límite.

5.1.3 Prueba a la Documentación y facilidades de ayuda.

Esta prueba es tan importante como las pruebas del desempeño de la aplicación o del código fuente. De la documentación depende el entendimiento del funcionamiento del sistema y que el usuario pueda sacar un mayor provecho de la misma.

Esta prueba se puede dividir en dos fases. La primera fase, la revisión e inspección, examina el documento para comprobar la claridad editorial. La segunda fase, la prueba en vivo, utiliza la documentación junto al uso del programa real.

La prueba de documentación se realizó a un grupo de potenciales usuarios del sistema que comprende secretarías, auxiliares y Médicos de Ortodoncia.

Al terminar con las pruebas se realizó una pequeña encuesta de donde se obtuvieron los siguientes resultados:

1. ¿En general como le pareció el desarrollo de las actividades?

Fácil	Normal	Difícil
30%	50%	20%

Tabla9. Resultados porcentuales Pregunta 1

2. ¿Que tanto le fue necesario recurrir a la ayuda?

Pocas veces	Casi Siempre	Siempre
20%	60%	20%

Tabla10. Resultados porcentuales Pregunta 2

3. ¿Cómo califica la ayuda para la aplicación de escritorio?

Escasa	Buena	Excelente
10%	40%	60%

Tabla11. Resultados porcentuales Pregunta 3

4. ¿Cómo califica la ayuda para la aplicación móvil?

Escasa	Buena	Excelente
6%	50%	44%

Tabla12. Resultados porcentuales Pregunta 4

5. ¿Cómo le pareció el ingreso de datos en la aplicación de escritorio?

Fácil	Normal	Difícil
47%	50%	3%

Tabla13. Resultados porcentuales Pregunta 5

6. ¿Cómo le pareció el ingreso de datos en la aplicación móvil?

Fácil	Normal	Difícil
10%	40%	50%

Tabla14. Resultados porcentuales Pregunta 6

7. Califique la facilidad de navegación en la aplicación

Fácil	Normal	Difícil
30%	50%	20%

Tabla15. Resultados porcentuales Pregunta 7

8. ¿Cómo le parecieron los mensajes de alerta de la aplicación?

Oportunos	Incómodos	Confusos
78%	14%	8%

Tabla16. Resultados porcentuales Pregunta 8

6. CONCLUSIONES

El éxito en el desarrollo de esta Herramienta se debió en gran parte a la aplicación constante y rigurosa de la Metodología AUP (Proceso Ágil Unificado), la cual permitió la constante mejora de la herramienta debido a que la metodología es flexible a las intervenciones de los usuarios.

La versatilidad y productividad de las herramientas .NET usadas como Visual Studio 2008 y Microsoft .Net Compact Framework , permitió el desarrollo de una aplicación móvil y de escritorio de gran calidad y madurez, además de que estas herramientas cuentan con una curva de aprendizaje bastante corto y con una gran cantidad de material de aprendizaje.

La arquitectura de Windows Mobile está diseñada para ofrecer un ambiente fuerte y protegido para una solución móvil, ésta arquitectura cubre los tres principales componentes de seguridad en una red ya que abarca las consideraciones de seguridad en el dispositivo, en el servidor y en las comunicaciones.

7. REFERENCIAS

- [1] HIGUERA, Saida, RAMÍREZ, Mary, DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA PARA LA REPRESENTACIÓN DE DATOS MÉDICOS Y SU ADQUISICIÓN A TRAVÉS DE DISPOSITIVOS MÓVILES APLICADO AL ÁREA DE LA ORTODONCIA, Universidad Industrial de Santander 2010.
- [2] KENDALL, Kenneth, KENDALL, Julie, ANALISIS Y DISEÑO DE SISTEMAS, Edición 6, PEARSON EDUCACIÓN, México, 2005
- [3] KRUCHTEN, Philippe, THE RATIONAL UNIFIED PROCESS AN INTRODUCTION, Third Edition, PERSON EDUCATION, Boston, 2004
- [4] LATTANZE, Anthony, ARCHITECTING SOFTWARE INTENSIVE SYSTEMS, TAYLOR & FRANCIS GROUP, 2009.
- [5] PRESSMAN, Roger, Ingeniería del Software, Quinta Edición, Consultor Editorial, México, 2002.
- [6] PIATTINI, Mario, Análisis y diseño de Aplicaciones Informáticas de Gestión Una perspectiva de Ingeniería del Software, Alfa omega - Ra Ma, México D.F, 2004.
- [7] HARRY, Andy, Microsoft c# Programming for the Absolute Beginner, Premier press, EEUU, 2002.
- [8] HORNER, Mark, Pro .NET 2.0 Code and Design Standards in C#, Torville Pty Ltd, 3rd Edition, 2006.
- [9] Microsoft Tech [Web línea] disponible en: <<http://technet.microsoft.com/es-co/default.aspx>> [consulta:05-01-2010]