



POLITÉCNICA

Escuela Técnica Superior de Ingeniería de
SISTEMAS INFORMÁTICOS
Universidad Politécnica de Madrid

PROYECTO FIN DE CARRERA

Solución Informática para gestión de Centros Sanitarios

Autor: Rodrigo Elvira Arana

Julio 2018

RESUMEN

Este escrito recoge toda la documentación del Proyecto Fin de Carrera **Solución Informática para gestión de Centros Sanitarios**.

Inicialmente se plantea la problemática a la que se enfrentan tanto servicios públicos como privados actualmente en cuanto a la necesidad de gestionar la información que requieren sus sistemas informáticos. Se realiza un estudio sobre el estado de la cuestión y se revisan algunos de los proyectos que se encuentran funcionando en estos momentos.

En la siguiente parte se propone una solución sencilla y funcional que cumpla un conjunto inicial básico de objetivos. El proyecto se divide en ciertos componentes y se estudia el desarrollo e implantación de cada uno de ellos. Se definen y programan la Aplicación, la Base de Datos, el motor de Base de Datos, la arquitectura del Sistema, el mecanismo de Replicación y el sistema de Publicación de Informes. La memoria describe los elementos realizados para el proyecto.

Por último se diseña un caso de prueba y se exponen las características de la simulación en ejecución. Durante el funcionamiento es posible interactuar con el sistema y acceder a la información almacenada. Empleando como interfaz la aplicación, un navegador WEB o una consola de administración de Bases de Datos, según funcionalidad y privilegios de usuario, se puede acceder a la información del Sistema. De igual manera permite la interacción con otros procesos que se puedan implantar en el Sistema Informático.

ÍNDICE

1. INTRODUCCIÓN

2. ESTADO DE LA CUESTIÓN

3. DESCRIPCIÓN DE LA SOLUCIÓN

4. ELEMENTOS DE LA SOLUCIÓN

4.1. Aplicación **MedicAp**

- 4.1.1. Diseño
- 4.1.2. Código
- 4.1.3. Implantación

4.2. Base de Datos **medicDB_fi**

- 4.2.1. Diseño
- 4.2.2. Código
- 4.2.3. Objetos
 - a. Tablas. Entidades y Relaciones
 - b. Vistas
 - c. Usuarios, procedimientos, índices, claves
- 4.2.4. Almacenamiento
- 4.2.5. Mantenimiento
 - a. Chequeo de Integridad
 - b. Reindexación y Actualización de Estadísticas
 - c. Sistema de Respaldo

4.3 Implantación

4.4 Sistema de Replicación

4.5 Sistema de Publicación de informes

5. EJECUCIÓN. CASO DE PRUEBA

6. DOCUMENTACIÓN. RECURSOS TÉCNICOS

1. INTRODUCCIÓN

Por mi formación y en mi trayectoria profesional he tenido la oportunidad de trabajar en varios proyectos como técnico de Sistemas Informáticos. En estos proyectos se aplican soluciones que integran diversas funcionalidades y que dan servicio a distintas aplicaciones. Como Proyecto Fin de Carrera me pareció adecuado crear una solución integral que incluyera todos los elementos implicados necesarios para ofrecer una funcionalidad completa.

En mi última etapa profesional participo en el proyecto de gestión sanitaria del grupo Quirónsalud como administrador de Bases de Datos en Telefónica. La solución informática es una solución de naturaleza dinámica en continua evolución que integra distintas funcionalidades. El sistema da soporte a todas las funcionalidades implementadas y tiene la capacidad de integrar nuevas que se van desarrollando según las necesidades del negocio. Además de la parte estrictamente funcional en la solución también se integran herramientas de mantenimiento. Estas herramientas informáticas se ocupan de gestionar la salud del sistema y de aportar estrategias de recuperación total o parcial en caso de desastre o migración a otras plataformas.

Para implementar un proyecto de tal envergadura se requiere la participación de varios equipos de técnicos que se dedican al desarrollo, implantación, proceso de renovación tecnológica y mantenimiento de la solución informática. Estos equipos técnicos han de coordinarse y debe existir un diálogo permanente entre ellos. Eso hace necesario disponer de un conocimiento compartido común sobre cada elemento partícipe de la solución global. Por ello parece interesante la realización de un sistema integral desde cero, obviamente significativamente simplificado pero al mismo tiempo funcional y completo.

Realizar e integrar todos los componentes de una solución técnica ofrece la oportunidad de abordar un mismo proyecto desde diferentes aspectos y de poder enfrentarse a la problemática subyacente en cada uno de ellos, aplicando soluciones basadas en el conocimiento y experiencia adquiridos durante mi formación y desarrollo profesional. Esos son el objetivo y la principal motivación por los que se crea y presenta este trabajo.

2. ESTADO DE LA CUESTIÓN

El dinamismo y la competitividad del sector sanitario hoy en día requiere que se propongan soluciones con la capacidad de integrar en una funcionalidad de gestión típica controles de tipo financiero que posibiliten un control o evaluación lo más actualizados posible de la situación económica del centro en particular y de la corporación a la que se encuentre adscrito en general. A los datos propios de cada institución acceden herramientas de naturaleza financiera que realizan consultas frecuentemente bastante costosas (cierres contables, informes de gastos, proyecciones de inversión, balances económicos) que en un momento dado pueden impactar en el rendimiento del sistema y degradar o incluso provocar la caída del servicio.

Elevadas esperas en las respuestas del sistema informático causan un efecto notorio en el funcionamiento rutinario de los centros, provocando molestias y retrasos en todas las actividades, desde consultas médicas hasta el control de inventarios y pedidos pasando por la gestión de información media de pruebas médicas. La caída del sistema desencadena la paralización del servicio sanitario hasta el punto en que tienen que ser definidos protocolos de emergencia con lápiz y papel para al menos poder dar respuesta para casos de Urgencias.

La criticidad a la que se enfrentan a diario sistemas de este tipo que funcionan de forma ininterrumpida obliga a buscar soluciones de alta disponibilidad con elevadas exigencias de rendimiento y capacidad. Una dirección en que se invierten muchos recursos es la de crear distintas redundancias de tipo activo/pasivo a distintos niveles que permitan la subsistencia del servicio ante caídas por problemas tanto de software como de hardware. Se buscan también mecanismos de balanceo de la actividad sobre el conjunto de recursos disponibles, mecanismos de paralelización de procesos, mecanismos de sincronización o programación de uso y mecanismos para priorización/discriminación de peticiones. Para incorporar alguna característica de este tipo al proyecto se implementa un mecanismo de replicación que permite derivar ciertas consultas en modo lectura a un segundo motor de Bases de Datos o segunda instancia que va aplicando las transacciones validadas en el motor o instancia principal.

Este grado de exigencia al sistema conlleva asociada lo que podríamos denominar una natural degradación por uso que obliga a incorporar funcionalidades encaminadas a mantener la información, su disponibilidad y accesibilidad, en el mejor estado de salud posible. Tareas de reorganización, de chequeo y de respaldo tienen que ser incorporadas a la actividad estrictamente propia del negocio y pasan a engrosar el total de carga del sistema. Necesariamente y de forma similar a como se hace con la metainformación, esta actividad

extra se ha de contemplar en el diseño, tiene que ser documentada y evaluada, y finalmente incluida como parte propia y necesaria del proyecto.

Un último aspecto ha de ser mencionado, aunque queda fuera del ámbito del problema que se aborda, es el de la protección de datos. Si los datos personales tienen que ser por principio conservados y protegidos con las máximas garantías; los datos médicos, por su enorme importancia, sensibilidad, coste y por el grave perjuicio personal que puede ocasionar su conocimiento o publicación por terceros, tienen que ser tratados con estricta regulación y control. Por razones de seguridad es necesario crear a distintos niveles mecanismos de protección de información del sistema y que se aplican tanto en el ámbito de los datos como en de los metadatos. Se definen políticas de seguridad, se crean controles de acceso, se cierran caminos/puertos y se encriptan los datos sensibles dentro de las tablas de las Bases de Datos. Ante la necesidad actual de tener que registrar, conservar y poner a disposición de facultativos, administrativos y gestores la información sobre cualquier evento relacionado con la actividad en un centro, individualmente o formando parte de una corporación sanitaria, se presentan en el mercado diversas plataformas y proyectos que tratan de ofrecer herramientas que lo posibiliten.

Estas propuestas tienen que cubrir unos objetivos mínimos comunes. Se tiene que ofrecer ante todo fiabilidad comprometiendo el rendimiento en la menor medida posible. Hay que tener en consideración que una herramienta con tantos puntos de conexión con la actividad que se realiza a diario en un centro sanitario puede incidir de manera notoriamente negativa en la calidad de su servicio médico. Las soluciones de naturaleza integral facilitan enormemente la supervisión y gestión de recursos de valiosa cuantía, tanto humanos como materiales.

La adecuada organización a la hora de programar una intervención quirúrgica, por poner un ejemplo en el que intervienen muchos factores, incide en la programación del personal sanitario que se requiere, de la disponibilidad de las dependencias (habitación, quirófano, sala de postoperatorio, cuidados intensivos), del material quirúrgico, maquinaria, pruebas de laboratorio. Afecta incluso a la actividad del servicio de comidas, que registra este entre otros aspectos y que influye finalmente también en los pedidos a proveedores. Afecta también al servicio de limpieza de instalaciones, que en esta situación tiene que resolver una petición especial para preparación de quirófanos. La criticidad de esta clase de sistemas hace adoptar medidas de prevención o protocolos de funcionamiento extraordinarios (en modo de lápiz y papel incluso) en caso de desastre en departamentos tan básicos como el de Urgencias.

En este punto queda claro que una solución eficiente y competitiva tiene la obligación de abarcar innumerables campos, o cuanto menos ofrecer capacidad de integración con otras

soluciones implantadas, y al mismo tiempo tiene que ofrecer esenciales garantías de robustez frente a situaciones por problemas accidentales y por ataques informáticos. Tiene que ofrecer un alto grado de resistencia e invulnerabilidad y dar una rápida respuesta en caso de pérdida total para regeneración de todo el sistemas desde cero. Por esta razón se deben complementar las tareas de copiado de seguridad en el propio sistema (ámbito de acción de la solución informática) con la realización recurrente de ejecuciones de volcado de información de copias de seguridad, ficheros, configuraciones y puntos de control en soportes externos.

Se pueden añadir o elegir otros recursos técnicos para enriquecer funcionalmente la solución informática. Una opción muy popular por sencillez, robustez y agilidad es la clusterización de los servicios con posibilidad de redundar toda la funcionalidad prácticamente de forma instantánea en uno o múltiples nodos. Esta herramienta dispone de capacidad incluso, y si así se configura, de asumir la responsabilidad de toma de decisiones ante situaciones de pérdida de recursos como comunicaciones o almacenamiento para efectuar balanceos entre nodos. Obviamente toda complejidad añadida implica necesariamente realizar un esfuerzo extra para coordinar y hacer compatibles las mejoras que se van incorporando.

En la actualidad la tecnología se está haciendo muy presente en los entornos sanitarios. Existe una gran profusión de propuestas en continuo desarrollo y se han implantado numerosas mejoras que no desde hace muchos años comienzan a consolidarse en el panorama del sector. El aporte de estas soluciones se diversifica y se hace presente en todos los ámbitos. Las facilidades que aportan son múltiples y contemplan todos los aspectos. Económicamente interesan por el gran ahorro de costes en recursos materiales y la optimización de los humanos. Además dotan de valor añadido a los grupos empresariales que las integran en sus sistemas. Por otra parte los costes derivados de su implantación, mantenimiento y desarrollo no son nada desdeñables. Las inversiones en tecnología, y en especial la de última generación, son altas. Su rentabilidad, debido a los múltiples factores que influyen, no está en ningún momento garantizada. Combinar soluciones implica multiplicar costes, una solución de redundancia impone por definición al menos la duplicación de los recursos mínimos requeridos para garantizar los servicios.

En este escenario la industria y el usuario demandan nuevos servicios. Servicio más rápidos y más versátiles. Que se emplean para aportar valor añadido a las empresas que los implantan y que los utilizan como aval de su alto grado de innovación y preparación en un sector tan crítico y competitivo como el sanitario. El éxito de cualquier propuesta tecnológica tiene que basarse sobre todo en la eficiencia, en el aprovechamiento de la disponibilidad de grandes tecnologías y su buen uso, hacer más con la tecnología disponible y reducir las ineficiencias.

La necesidad de utilizar software de asistencia médica comenzó en los años setenta cuando se vislumbraba la idea de utilizar ordenadores para ayudar en la formulación de diagnósticos médicos. Los primeros esfuerzos informáticos no despegaron en principio en parte porque los registros de los pacientes seguían siendo escritos a mano, y el proceso de transcribir dicha información a soporte digital muy engorroso. Con el tiempo la tecnología informática se ha desarrollado de forma asombrosa y los ordenadores son ahora parte integral de la medicina estándar, asumiendo el control de los registros e impulsando la calidad, la seguridad y la eficiencia en los sistemas de atención de la salud. Los sistemas de apoyo al diagnóstico también han cambiado, ya que se han vuelto mucho más rápidos y suelen vincularse directamente a los estudios, permitiendo a los clínicos evaluar rápidamente las pruebas y aprender más sobre diagnósticos potenciales. Se emplean algoritmos para vincular los síntomas con los diagnósticos y algunos incluso son capaces de extraer automáticamente información de los registros electrónicos actuales de los pacientes.

El error humano se ha intentado delimitar lo más posible desde el inicio de la Medicina. Existen trampas cognitivas en las que los médicos pueden caer al hacer un diagnóstico. Los estudios sugieren que los médicos se pueden quedar con el primer diagnóstico y dan mayor peso a cualquier información que lo refuerce, descartando la búsqueda de otras informaciones y datos. La cuestión motiva la investigación sobre sistemas de apoyo a la toma de decisiones clínicas. Incluso se consideran prácticas soluciones de presentación de listas de verificación únicamente para evitar saltar pasos clave en rutinas importantes. Escuelas de medicina, hospitales de enseñanza y otros centros de atención invierten en este tipo de asistencia informática.

Una dificultad que complica el avance tecnológico es la circunstancia de que cada hospital trabaja con diferentes sistemas y protocolos electrónicos, lo que hace imposible estandarizar los avances e instaurar cambios para evolucionar el sistema. Los médicos pueden ser reacios a emplear tiempo aprendiendo un sistema sin garantías de que sea el definitivo y necesite tener que volver a formarse en poco tiempo sin poder rentabilizar el tiempo y esfuerzos de dedicados en formación de herramientas informáticas. Se empezó a realizar hincapié en la evolución del diseño de interfaces hacia herramientas mucho más visuales y amigables basadas en la forma de trabajar de los médicos pero ahorrando ostensiblemente los tiempos de espera. Herramientas que muestran diagramas y referencias visuales, y que tienen la capacidad de reflejar cambios, registrarlos y mostrar sus progresiones. Información muy valiosa en el proceso de observación de la evolución de un paciente.

La Inteligencia Artificial también pasa a aplicarse como otro elemento de diagnóstico médico. Con los entornos de Big Data actuales se dedica mucho tiempo en el diagnóstico de imágenes y

muestras que son normales. Con nuevos protocolos de diagnóstico se han diseñado soluciones de ayuda para comparación de imágenes ubicadas en la red, lo que amplía el campo de decisión. El objetivo no es la sustitución de los profesionales, el objetivo es centrar el tiempo en detenerse en aquello que realmente es distinto. El problema es conseguir transformar los datos en conocimiento. La Inteligencia Artificial puede avanzar patrones, parámetros, históricos e incidir en lo diferente. Es una gestión similar a la que realiza un profesional a lo largo de su trayectoria laboral. Reproduce de alguna manera el proceso de formación natural.

Lo que hay que conseguir es que una solución administrativa efectiva y una herramienta potente al final tenga consecuencias no previstas como la pérdida de las relaciones sociales entre profesionales y pacientes. La Inteligencia Artificial, el desarrollo de procesos creativos para el uso de datos y el aprendizaje que acumulan las máquinas tiene que enfocarse en la predicción y lo que supone una acción y una recomendación. Los técnicos han de buscar que las máquinas predigan con exactitud pero sea el ser humano el que explique y decida sobre la acción propuesta. Máquinas y humanos tienen que convivir, pero teniendo claro que es fundamental mantener la inteligencia humana en el bucle, dentro del contexto social, clínico y personal. La capacidad de la Inteligencia Artificial para automatizar es muy valiosa y contribuye incluso en la conexión entre el profesional y el paciente. Facilitar costosos trámites, tener acceso a resultados de análisis desde la red o gestionar disponibilidad de consultas son algunas de las ventajas que se pueden ofrecer al usuario. De este modo se consigue el avance de que el servicio sanitario sea percibido de forma más amable. Como consecuencia se produce una mejora en las historias clínicas y la calidad de los datos aportados al sistema.

El enorme grado de implantación de funcionalidad computacional en el sector de la sanidad es un fenómeno que sucede y se manifiesta de forma universal. Estudios de todos los tipos se realizan y la asociación entre las disciplinas de Medicina e Informática continúa estableciéndose cada día con mayor solidez. Es un hecho y es inevitable la fase de la informatización de todos los sistemas, de una forma u otra y aunque sea de un modo más o menos rústico desde ya hace mucho tiempo los ordenadores forman parte del equipo básico de asistencia de cualquier docente. Para su propia formación profesional, para dispensar atención sanitaria y para facilitar su comunicación con otros expertos.

El reto actual se encamina en la dirección de la elaboración de planes estratégicos y estudios para mejorar en calidad y eficacia la atención. Se genera un debate social sobre la experiencia del paciente y se señalan los objetivos a resolver. Las entidades públicas, además de las iniciativas privadas, están tomando en consideración estas necesidades que demanda la sociedad. Se presenta como una necesidad primordial crear facilidades que permitan un intercambio ágil y seguro de la información entre diferentes entidades. Debe procurarse el

acceso a datos críticos personales de los pacientes si las circunstancias lo demandan, al mismo tiempo que garantizar la confidencialidad en su uso. Los sistemas, que ya se enfrentan a sus particulares problemáticas en lo referente a interacción entre sus propios componentes, además han de procurar mantener una comunicación segura y fluida con otros sistemas externos. Pruebas o analíticas de un determinado paciente registradas en un servicio de atención de salud deberían ser recuperables para su uso desde cualquier otro.

Una solución profesional implantada en varios centros es **Selene Hospitales**, desarrollado por la empresa Cerner. Su apuesta por la gestión sin papeles de los hospitales. Selene es un sistema que permite compartir la información clínica y administrativa de los pacientes a los médicos, enfermeras y otros usuarios autorizados. Los flujos de trabajo de la organización sanitaria se centran en el paciente, facilitando la continuidad asistencial y creando un entorno de trabajo colaborativo y de fácil uso. Sus características se asemejan a la aplicación propuesta en este proyecto, con las dimensiones y requerimientos de un sistema real y en funcionamiento. Se realiza un seguimiento del paciente a lo largo de su paso por el hospital: desde la gestión administrativa, la documentación de la historia, la realización peticiones electrónicas o la prescripción de medicación, hasta la recepción de resultados provenientes de sistemas externos. Además cuenta con un sistema de alertas que se activa en caso de duplicidades, alergias o interacciones medicamentosas. Ofrece accesibilidad desde dispositivos móviles conectados a la red inalámbrica del hospital para facilitar la atención a pie de cama o en los lugares en los que el acceso a un puesto fijo es complicado.

Aporta entre otras facilidades la aplicación informática **Selene Mobility**, en la cual a través de dispositivos móviles conectados a la red wifi del Hospital los profesionales sanitarios pueden consultar a pie de cama la historia clínica electrónica del paciente, los cuidados de enfermería y la administración de la medicación, así como solicitar pruebas diagnósticas, abrir órdenes de enfermería y registrar notas.

Como efecto negativo y difícilmente evitable de esta situación es la de la gran repercusión que un error puede llegar a tener y que puede desencadenar una situación repentina de pérdida de servicio. Las medidas adoptadas tratan de garantizar la restitución del servicio en un tiempo prudencial. A pesar de cumplir con los estándares establecidos, cualquier parada por mínima que sea puede llegar a provocar la detención del sistema con consecuencias desastrosas.

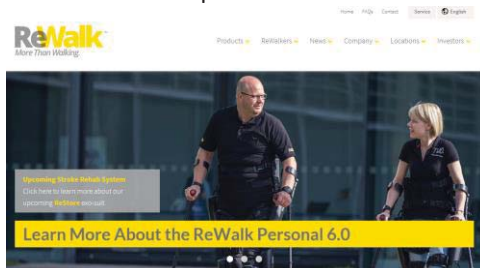
El grupo **Quirónsalud** se destaca en innovación tecnológica aplicada al sector de la salud. Se ha centrado en el desarrollo de nuevas herramientas tecnológicas que permitan tanto la mejora de la efectividad de sus centros hospitalarios como la facilidad de autogestión de los pacientes. A través de un exhaustivo análisis de los diferentes procesos asistenciales, se han desarrollado

nuevas estrategias de trabajo en red entre sus hospitales, y se han implantado nuevas herramientas de ayuda para los pacientes. Uno de estos proyectos ha sido la puesta en marcha de una aplicación móvil de gestión sanitaria que permite realizar teleconsultas de forma directa con los centros de atención primaria o residencias. Estas nuevas herramientas tecnológicas han supuesto una mejora considerable para todas las partes implicadas en la gestión sanitaria. Tanto para los pacientes, que cuentan con mayor accesibilidad a su información clínica con el portal del paciente, un espacio personal desde el que podrá acceder a sus pruebas diagnósticas, informes clínicos, gestionar sus citas y tener un seguimiento sin esperas ni desplazamientos, como para los propios centros mediante la mejora de los procesos internos de diagnóstico, tratamiento y seguridad del paciente en menos tiempo. Un proceso de rediseño que ha permitido la supresión de tareas y la mejora de la accesibilidad al sistema sanitario gracias a la integración de los procesos tecnológicos.

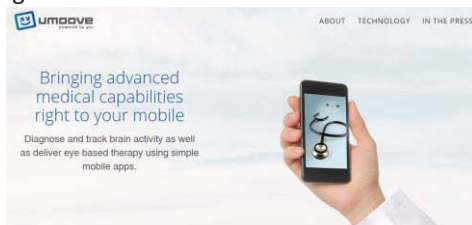
Otra área de gran importancia es la de formación , donde se crean espacios como el portal **univadis**, dirigido a profesionales sanitarios, que ofrece resúmenes diarios de las últimas investigaciones clínicas y noticias de la industria médica. Los artículos se adaptan a las especialidades específicas, lo que facilita que los usuarios se mantengan al día con la información más reciente y más relevante. Estos disponen de acceso a las tres secciones básicas: noticias, herramientas y referencias para ayuda a la toma de decisiones correctas en el momento oportuno. También suministra breves presentaciones interactivas y casos de expertos y compañeros.

Este mercado del mundo tecnológico promueve herramientas para monitorización, diagnóstico e incluso tratamiento, así como para establecer una gestión sanitaria más eficiente. Aparecen aplicaciones móviles, robots y simuladores complejos que conforman una situación de constante renovación en busca de conseguir la mejor asistencia sanitaria. Algunas propuestas que se destacaban en publicaciones del sector sanitario:

ReWalk. Sistema desarrollado para que personas con lesiones de médula espinal, puedan caminar gracias a un sistema de piernas artificiales. El usuario se coloca dichas piernas con la adecuada sujeción del dispositivo, y hace uso de unas muletas para equilibrar su cuerpo a la nueva situación. Se trata de una aplicación que funciona con un motor eléctrico con una batería recargable, ubicada en una mochila que el usuario lleva a su espalda.



uMoove. Software que sustituye el control del dispositivo móvil con las manos al control del mismo con los movimientos de los ojos, incluso de la cabeza. Este sistema recoge los movimientos del usuario y las traduce a movimientos en el teléfono. Personas con parálisis y enfermedades neurológicas podrían hacer uso de sus dispositivos. Además puede servir como herramienta para detectar enfermedades, cuyos diagnósticos son basados en los movimientos oculares.



Telesofia. Esta App tiene el objetivo de paliar la desinformación por parte del paciente tras el diagnóstico. Permite a los profesionales médicos a insertar información personalizada, a través de vídeos fáciles de comprender, que puede incluir desde las instrucciones que debe acometer el paciente, hasta información relevante sobre dicha enfermedad, o consejos de prevención. Así los pacientes cuentan con un punto de información al que recurrir en caso de no recordar alguna indicación o dudar de algún tipo de procedimiento o hábito posiblemente inconveniente.



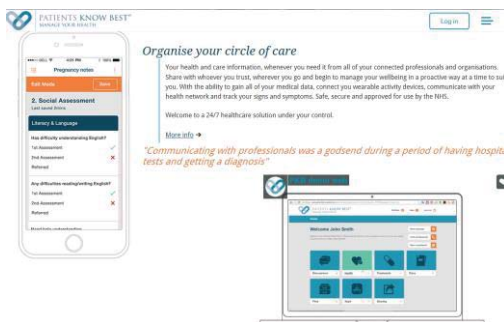
Surgical Theater. Esta herramienta utiliza la tecnología de las simulaciones de vuelo en la sala quirúrgica. Permite visualizar las zonas intervenidas por los cirujanos en imágenes 3D, a través de la fusión de resonancias magnéticas, tomografías y demás pruebas. El sistema tiene la doble función de ser utilizado para tanto cirujanos, como médicos en formación quirúrgica.



TotallyPregnant. La aplicación móvil recoge todas las funciones de las aplicaciones desarrolladas para embarazadas. Seguimiento de dieta, ejercicios durante y después del embarazo, rastreadores del embarazo. La función más revolucionaria es la de la inclusión de vídeos de animación del feto en 3D, así como álbumes de fotos. También ofrece vídeos informativos, consejos, foros de asesoramiento de expertos, compras y blogs

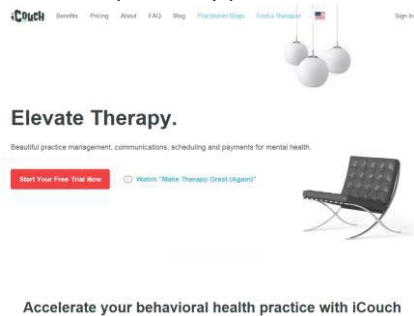


Patients Know Best. Plataforma que mantiene registros médicos en la nube, para que los pacientes y médicos tengan un control de sus datos. La herramienta pretende mejorar la calidad del cuidado de los médicos y enseñar a los pacientes a controlar su salud, así como mejorar la eficacia de los administradores. Un extenso repertorio de enfermedades puede encontrarse en la red y sirve para intercambio de tratamientos, así como del tipo de atención y diagnóstico realizados. Busca dar acceso desde cualquier lugar al conocimiento común compartido.



iCouch. Ofrece videoconferencias entre los profesionales de la salud mental y los pacientes. La facilidad de uso de esta aplicación elimina los obstáculos de algunas personas con este tipo de enfermedades y

permite que puedan obtener tratamiento desde cualquier punto del mundo en cualquier momento. Contribuye aportando un fácil acceso a la atención clínica por parte de aquellas personas que sufren enfermedades complicadas y particularmente alejadas de lugares donde poder ser asistidos.



HelpAround. Para enfermos con diabetes, que necesitan utilizar herramientas de equipos médicos a diario. Conecta a los usuarios para que en caso de olvido o urgencia puedan acceder a dichos materiales. Responde a demandas para mediciones de glucosa como para establecer un suministro de insulina.



Doximity. Es una red social para médicos. Funciona como *LinkedIn* adaptado al campo de la medicina a través de la colaboración entre profesionales sanitarios en los casos clínicos más complicados. También ofrece una oportunidad para intercambiar opiniones, y consejos sobre tratamientos, productos sanitarios y gestión clínica de pacientes.



Fitocracy. Aplicación que funciona como una red social para el estímulo de actividades físicas como juegos. Se basa en el concepto de que muchas personas abandonan el ejercicio porque no ven el progreso con suficiente rapidez, o porque tienen que hacerlo solos y se encuentran suficientemente motivados.



Buen reflejo del interés despertado se encuentra en el campo de la cirugía, en la inclusión de mecanismos e incluso sistemas completos robóticos que asisten y mejoran sustancialmente las operaciones. Su precisión posibilita la realización de intervenciones menos invasivas y su ahorro en costes económicos y humanos. Las posibilidades ofrecidas por la realidad aumentada y la evolución de mecanismos y artilugios capaces de ejecutar cada vez acciones más minuciosas. El ahorro en molestias y tiempo que otorga a los pacientes estos avances también es real y también es valorado y medido, tiene una gran importancia y es un gran estímulo para el desarrollo en ciencia y tecnología en el campo de la Medicina.

Especial relevancia en los últimos años ha tenido el **robot quirúrgico Da Vinci**, con una consola ergonómica desde la que el cirujano opera sentado en el mismo quirófano o de forma remota. Próximo a la camilla se posiciona la torre de visión y el carro quirúrgico que incorpora brazos robóticos interactivos controlados desde la consola, en el extremo de los cuales se encuentran acopladas las distintas herramientas necesarias para operar. Optimiza el rango de acción de la mano humana perfeccionando los movimientos del cirujano, permite realizar movimientos altamente precisos en espacios muy reducidos. De esta manera se minimizan las posibilidades de error en relación a otros sistemas quirúrgicos con visiones del área anatómica en la que interviene en 2D. En contraposición, da Vinci ofrece una visión tridimensional de la zona intervenida, en que el cirujano gestiona la cámara de forma totalmente autónoma. Ofrece al profesional la posibilidad de poder contar con una visión superior en 3D, alineada entre la zona anatómica afectada y el instrumental, una posición única desde la que se puede trabajar de forma cómoda, intuitiva y precisa.



robot quirúrgico Da Vinci

Importantes logros son recogidos regularmente en artículos de publicaciones médicas y tienen su eco asimismo en la prensa nacional. Se recapitula información sobre la intervención a un niño con cirugía mínimamente invasiva asistida por el robot Da Vinci. Llevada a cabo con éxito

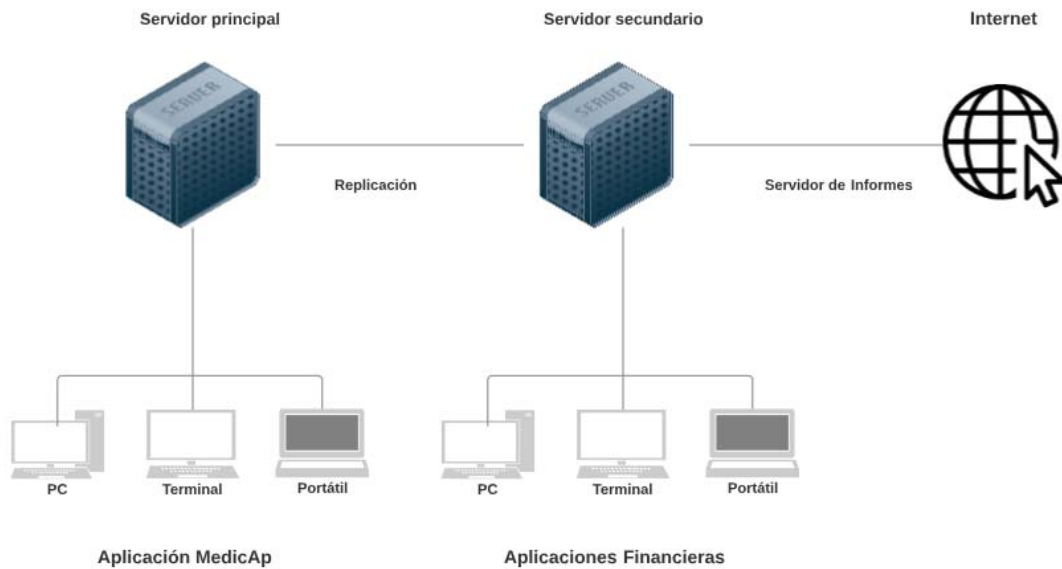
por un equipo de cirujanos pediátricos y urólogos, quienes han extirpado un riñón al paciente, de 9 años de edad, dado de alta sin complicaciones 48 horas después.

Más proyectos en robótica aparecen en el panorama del sector.

El proyecto **Senhance** presenta un robot a través del cual el cirujano siente y maneja lo que ve dentro del paciente moviendo sus ojos. Con características más peculiares se presenta el proyecto **Flex Robotic System**, que desarrolla un robot con forma de serpiente articulada que incorpora una cámara de alta definición e instrumentos quirúrgicos capaz de seguir las curvas del organismo. No todas las instalaciones sanitarias pueden incorporar sistemas tan costosos. Aparecen soluciones más económicas que implementan similares funciones, que aprovechan el conocimiento acumulado y que con un importante abaratamiento en recursos las convierten en opciones muy viables. Se menciona el robot **Broca**, que trata de incorporar la sensación de tacto, es decir, replica las percepciones que tiene el cirujano al enfrentarse a un tejido blando o duro. Otra propuesta que se presenta es **FlexDex Surgical**, que mejora las herramientas existentes disponibles para cirugías mínimamente invasivas. Se podría ayudar a cirujanos de hospitales remotos a realizar determinados procedimientos médicos muy precisos que actualmente solo pueden llevarse a cabo en grandes centros médicos. Este dispositivo es un instrumento para la mano especializado en pequeñas incisiones y suturas. Se coloca en el brazo del cirujano con su centro de rotación en el mismo punto que la muñeca del médico, de forma que opera como una extensión del brazo.

3. DESCRIPCIÓN DE LA SOLUCIÓN

El objeto del proyecto consiste en la realización de una **solución informática integral** para la gestión de un centro de atención sanitaria. Esta solución incluye todos los componentes que se requieren para su funcionamiento. La arquitectura de la solución queda reflejada en el siguiente esquema:



El conjunto de componentes que conforman la solución:

Aplicación [MedicAp](#). Programada en [C#](#), su interfaz permite el acceso y gestión de los datos del Centro Sanitario.

Base de Datos [medicDB_fi](#). Almacén de toda la información requerida del personal sanitario y los pacientes. Se programa en lenguaje [Transact-SQL](#).

Motor de Base de Datos de la herramienta [Microsoft SQL Server](#). Ejecutable que implementa el servicio de Base de Datos.

Servidor de Informes [SQL Server Reporting Services](#), servicio instalable para publicación de informes en la WEB.

Sistema de replicación mediante [trasvase de registros \(log shipping\)](#), basado en la restauración ordenada de copias de seguridad transaccionales.

Plan de mantenimiento, programación del conjunto de trabajos de revisión y mejora de la *salud* de la información.

Cada componente ocupa su espacio independiente en el sistema y participa según su rol en la construcción de la solución integral. La aplicación se conecta a la entidad Base de Datos, que funciona y es accesible gracias al motor, el programa ejecutado como servicio en el Sistema Operativo.

La replicación a su vez es un mecanismo del motor de Base de Datos que permite la propagación de los datos según conveniencia y restricciones. La existencia de esta copia diferida y presente como Base de Datos propia, con la limitación de no ser modificable en esta estancia, abre otras muchas opciones de soluciones informáticas. Con objeto de hacer uso de esta característica, se implementa un sistema de publicación de informes en entorno WEB. Mediante un navegador y tras completar la validación de seguridad se posibilita acceso a los contenidos establecidos por los administradores del sitio WEB. Se conforma una relación de objetivos específicos que ha de conseguir cumplir el sistema:

1. Posibilidad de Incorporar nuevos pacientes en el registro del centro de atención. Han de quedar almacenados sus datos personales, clínicos y bancarios.
2. Posibilidad para añadir personal con sus datos particulares e información de índole profesional.
3. Registro de control residencial y gestión de ocupación. Esta actividad conllevará imputaciones de gastos asociados en las cuentas de los pacientes.
4. Acceso al historial clínico de cada paciente. Se conservan los datos de fechas y facultativos.
5. Funcionalidad para realizar ingresos y altas hospitalarias de usuarios del centro sanitario.
6. Programador de consultas médicas para los facultativos. Por necesidad administrativa ha de permitir la cancelación de citas ya registradas.
7. Capacidad para la imputación de gastos por paciente. Cálculo de importes a cargar en cuenta.
8. Implementación de un mecanismo de replicación para balanceo de carga de consultas y recuperación ante desastre.
9. Funcionalidad para diseño, generación y publicación de informes de control financiero y de salud del sistema en entorno WEB.
10. Diseño de un plan de mantenimiento de información de la Base de Datos. Programación de tareas rutinarias.
11. Establecimiento de la estrategia de recuperación.
12. Control de seguridad para validación de acceso a las distintas funcionalidades que conforman la solución.

En el siguiente punto de este documento se abordan en detalle las características de cada elemento y se fija el rol que desempeñan dentro de la solución integral.

4. ELEMENTOS DE LA SOLUCIÓN

4.1. Aplicación **MedicAp**

En lenguaje de programación [C#](#) se genera el ejecutable **MedicAp**, que mediante su interfaz permite el acceso y gestión de datos. Ofrece funcionalidad para la consulta, inserción y actualización de información relativa al Centro Sanitario. El programa gestiona la administración de recursos materiales y humanos e integra un control financiero de gastos. Para la programación del código se emplea la herramienta de desarrollo [Microsoft Visual Studio](#).

4.1.1. Diseño

En la fase de diseño se aborda la cuestión de transformar los requerimientos y necesidades planteados en una solución práctica y funcional. Así mismo se establece como una prioridad proponer un modelo escalable que sea fácilmente ampliable y que permita continuar evolucionando con la incorporación de nuevas funcionalidades. También se marca el objetivo de proponer un diseño aplicable a entornos de diferentes características. Se establecen los siguientes hitos que resolver:

- Capacidad para incorporar nuevos pacientes en el centro creando nuevas entradas con su información particular. Opciones de búsqueda para acceso a datos personales y de historial clínico.
- Posibilidad para crear registros igualmente con personal sanitario. La interfaz permitirá añadir campos específicos como la especialidad y el número de colegiación.
- Deberá posibilitar el control de ocupación de habitaciones en la parte residencial. Sólo pacientes existentes en el registro podrán acceder a este servicio. Realizar el alta hospitalaria de un residente dejará libre el recurso a disposición del centro.
- Programación consultas sanitarias según disponibilidad de facultativos. También debe integrar la posibilidad de acceso al historial de citas del paciente, con registro de fecha y médico.
- Recopilación de información en tiempo real sobre los costes asociados a cada acción realizada en el sistema. La asignación de una habitación a un paciente y la concertación de una cita con determinado médico llevan asociados unos costes que deben ser computados, por tanto han de generarse sus asientos contables asociados a la fecha y a la actividad. Como el sistema permite la cancelación de citas y como no se quiere eliminar registros o

apuntes contables, se determina que esta acción en particular crea un registro con el mismo valor en negativo. Este control de costes es independiente del modo de pago o cobertura sanitaria del usuario y ha de ser tratado como corresponda al modelo de negocio.

La aplicación presenta una interfaz sencilla e intuitiva que resuelve la necesidad de introducir modificar y consultar datos del centro. Mediante menús interactivos presentados en varias ventanas se conecta con la Base de Datos y se desencadenan las transacciones que equivalen a las acciones seleccionadas.

A continuación se describe el aplicativo implementado. La navegación por las opciones presentes se lleva a cabo como respuesta a la activación de los distintos botones. Desde el menú inicio o desde el acceso directo creado en el Escritorio se invoca al ejecutable .exe ubicado en la ruta definida en la configuración. La interfaz diseñada y sus elementos, con las acciones desencadenantes que cada uno activa, se describen en el siguiente apartado.

0. Pantalla **Inicio**

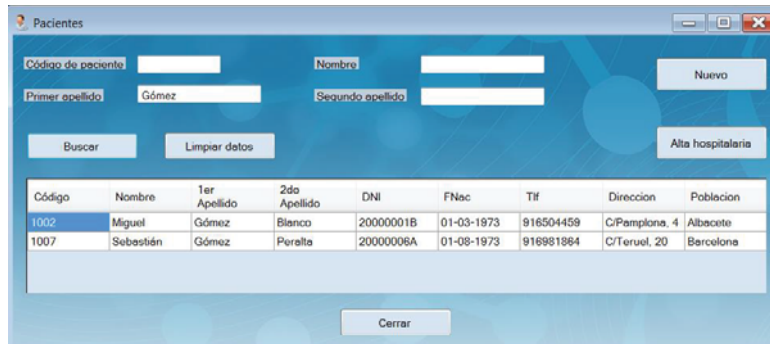
Al iniciar la ejecución del aplicativo **MedicAp** aparece una ventana emergente. La pantalla inicial de la aplicación presenta los diferentes recursos a los que se puede acceder para la gestión administrativa del **Centro Sanitario**.

Cada uno de los seis botones presentados nos abre una ventana al menú propio de cada contenido. El botón **Salir**, al igual que en el resto de ventanas emergentes, cierra la caja de diálogo.



1. Pantalla **Pacientes**

Al acceder a la opción **Pacientes** se puede realizar una búsqueda sobre datos de pacientes registrados. Introduciendo en las casillas blancas el código, nombre y/o apellidos y accionado el botón **Buscar** se muestra un listado con las coincidencias, que es el resultado de ejecutar un *SELECT* con condiciones sobre la Base de Datos. La opción **Limpiar datos** vacía de contenido los casilleros de filtro.



The screenshot shows the 'Pacientes' application window. It features a search form with the following fields: 'Código de paciente', 'Nombre', 'Primer apellido' (with 'Gómez' entered), and 'Segundo apellido'. There are buttons for 'Nuevo', 'Buscar', 'Limpiar datos', and 'Alta hospitalaria'. Below the form is a table with the following data:

Código	Nombre	1er Apellido	2do Apellido	DNI	FNac	Tlf	Direccion	Poblacion
1002	Miguel	Gómez	Blanco	20000001B	01-03-1973	916504459	C/Pamplona, 4	Albacete
1007	Sebastián	Gómez	Peralla	20000008A	01-08-1973	916981884	C/Teruel, 20	Barcelona

At the bottom of the window is a 'Cerrar' button.

También es posible registrar un **nuevo paciente** o realizar un **Alta hospitalaria**. Cada acción sobre el botón correspondiente abre las siguientes ventanas:

1.a Pantalla **Nuevo Paciente**

En esta pantalla se realiza el ingreso de datos de pacientes en el sistema. Los datos del nuevo registro se rellenan en el formulario y el sistema verifica si son correctos para proceder con el nuevo registro al pulsar sobre la opción **Registrar**. Si es así se realiza el correspondiente *INSERT* en la tabla de pacientes invocado por el procedimiento **Realizar ingreso** que genera un nuevo y único código para el registro.

Así mismo con el ingreso de un nuevo paciente se crea una cuenta con los datos financieros asociados al paciente. Se crea un registro nuevo con saldo inicial igual a cero. En esa cuenta se descargarán los gastos generados por los servicios prestados y serán emitidos los cargos a las cuentas almacenadas en el sistema.

También aquí la opción **Limpiar datos** vacía de contenido las casillas de datos.



The screenshot shows the 'Nuevo Paciente' application window. It features a registration form with the following fields: 'Nombre', 'Fecha Nacimiento', 'Primer apellido', 'Teléfono', 'Segundo apellido', 'Dirección', 'DNI', and 'Población'. There are buttons for 'Registrar', 'Limpiar datos', and 'Cerrar'.

1.b Pantalla **Alta Hospitalaria**

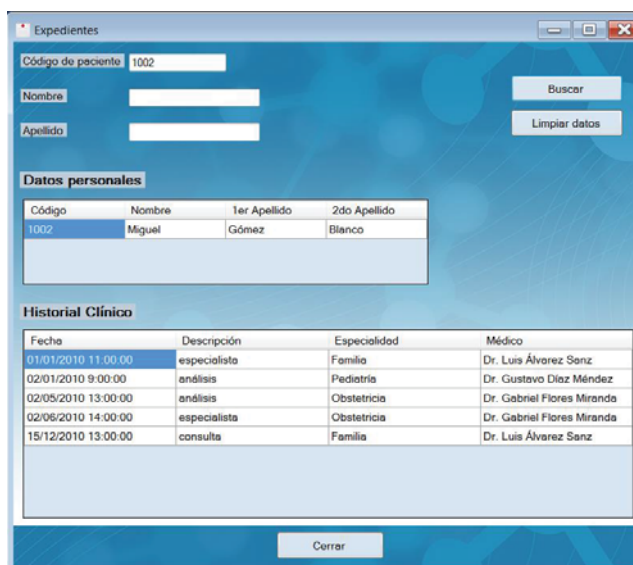
En la ventana para proceder a realizar el alta en el centro se introduce el código interno del paciente. Mediante una consulta de *DELETE* a la Base de Datos con el código del paciente, si se encuentra ingresado se libera la habitación, y en otro caso muestra el error por no encontrarse registrado como ingresado.

Con el alta hospitalaria se procede a registrar en el sistema el cargo generado por la estancia del paciente para posteriormente poder ser procesado y cargarse en su cuenta. Se abre una transacción que ejecuta un *DELETE* en la tabla de ocupación y un posterior *INSERT* en la tabla de cargos, generando un apunte asociado al código del paciente. En la tabla de cargos se distingue si este ha sido imputado (con valor *cargado=1*) o aún no.



2. Pantalla **Expedientes**

El sistema de **Expedientes** ofrece información con el **historial clínico** de cada paciente. La actividad del paciente en el **Centro Sanitario** se muestra ordenada temporalmente y se refleja por paciente el tipo de consulta por su descripción, la especialidad asociada y el médico que la atendió. Con un *SELECT* de búsqueda sobre código o nombre del paciente en la Base de Datos se presenta el listado de consultas ordenado por antigüedad. Aparece en este formulario de nuevo la opción para **Limpiar datos**, permitiendo realizar ulteriores consultas de iguales características.



3. Pantalla **Habitaciones**

El sistema informático implementa un sistema de control de alojamiento para pacientes en el centro. Cada **habitación** tiene un registro de estado y es asignada según disponibilidad. Se muestra información sobre el estado actual y el tiempo de ocupación. La estancia en una habitación conlleva un coste y este se registra en el sistema como un apunte de **gasto** en base al número de días una vez que se realiza el **Alta hospitalaria**.

Al rellenar con un dato válido de habitación o paciente y pulsar en el botón **Buscar** se lanza una consulta **SELECT** con el cruce de las tablas de ocupaciones y pacientes que muestra tanto datos personales como fecha y número de días de ingreso.



4. Pantalla **Personal Sanitario**

La herramienta dispone de funcionalidad para gestión de **Personal Sanitario**. Este menú redirige a las opciones de **Consulta de Datos** y **Nuevo Registro** para insertar nuevo personal.

Accionando los botones se activa la presentación de la ventana emergente seleccionada. Se explica su funcionamiento en las siguientes secciones.



4.a. Pantalla **Consulta de Datos**

Ventana para consulta de datos del **Personal Sanitario**. Es posible realizar búsquedas sobre cada campo del formulario y se listan los registros coincidentes. Indicando una especialidad se listan todos sus especialistas registrados en el centro.

Todos los sanitarios se encuentran registrados en el sistema. Las transacciones de consulta *SELECT* sólo se completan al establecer una conexión segura con la Base de Datos, utilizando la validación y el encaminamiento designados. Esta acción se ejecuta al incidir en el botón **Buscar**. Incluida aquí la opción para **Limpiar datos**.



The screenshot shows a window titled "Consulta de Datos" with a search form and a table of results. The form includes fields for "Código", "Nombre", "Primer apellido", "Segundo apellido", "Núm Colegiado", and "Especialidad". The "Código" field contains the value "105". There are "Buscar" and "Limpiar datos" buttons. The table below has the following data:

Código	Nombre	1er Apellido	2do Apellido	Nº Colegiado	Especialidad
105	Maria José	Farías	Figueroa	99105	Nefrología

A "Cerrar" button is located at the bottom of the window.

4.b. Pantalla **Nuevo Registro**

Ventana para añadir **Nuevo registro** de Personal Sanitario en el sistema. Son requeridos los datos personales así como el número de colegiado y la especialidad que ejerce. Al activarse **Insertar** se validan los datos y se ejecuta un comando *INSERT* en la tabla de facultativos. Se incluye **Limpiar datos**.



The screenshot shows a window titled "Registro" with a form for adding a new record. The form includes fields for "Nombre", "Primer apellido", "Segundo apellido", "Núm Colegiado", and "Especialidad". There are "Insertar" and "Limpiar datos" buttons. A "Cerrar" button is located at the bottom of the window.

5. Pantalla **Ingresos**

Los ingresos en las dependencias residenciales se pueden realizar relleno del formulario de **Ingresos**. En caso de realizar el ingreso de un paciente aún no existente en el sistema es posible crear el nuevo registro de paciente y proceder con su ingreso y asignación de habitación. El sistema verifica y asigna la primera habitación disponible al paciente. En caso de no haber dependencias posibles, no existir el código de paciente o este encontrarse ya ingresado el aplicativo indica el tipo de error. En otro caso lanza una transacción de *INSERT* sobre la tabla de ocupación de habitaciones.

Se ofrece la posibilidad de **Realizar Ingreso** de forma directa al activar el botón y haber relleno la casilla de código de un paciente. En este caso la aplicación envía un comando *UPDATE* en la tabla de ocupaciones actualizando la información de la habitación por estar desde ese momento ocupada. Lógicamente no podrá ser asignada a un paciente una habitación ya ocupada.

También es posible rellenar los datos de un nuevo paciente y con una sola activación del botón **Ingreso Nuevo Paciente** añadirlo al sistema como nuevo paciente asignándole en ese momento una habitación vacía.

Aceptados los datos como válidos se desencadenan consultas de *INSERT* en la tabla de pacientes y *UPDATE* en la tabla de ocupaciones. Se permite realizar varios registros de forma consecutiva sobre la Base de Datos al incluirse la facilidad **Limpiar datos**.

La imagen muestra una ventana de software titulada "Ingreso Hospitalario". En la parte superior izquierda hay un campo de texto etiquetado "Código de paciente". A la derecha de este campo están dos botones: "Realizar Ingreso" y "Limpiar datos". Debajo de esto, hay una sección titulada "Nuevo Paciente" que contiene un formulario con los siguientes campos: "Nombre", "Primer apellido", "Segundo apellido", "DNI", "Fecha Nacimiento", "Teléfono", "Dirección" y "Población". En la parte inferior izquierda de esta sección hay un botón "Ingreso Nuevo Paciente", y en el centro inferior de la ventana hay un botón "Cerrar".

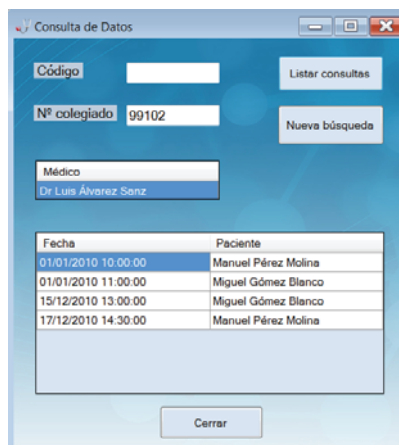
6. Pantalla Consultas

Nuevamente la aplicación ofrece el acceso a otras ventanas. En la sección de **Consultas** se gestiona la programación de citas. El formulario presenta las opciones de **Consulta de Datos**, **Nueva Cita** y **Eliminar cita**.



6.a. Pantalla Consulta de Datos

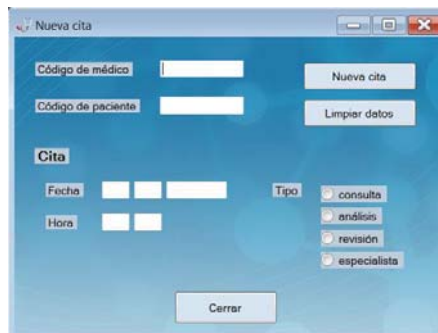
Al acceder a la opción **Consulta de Datos** se puede revisar la agenda de citas que tiene cada médico. La búsqueda requiere los valores del código interno o del número de colegiado del sanitario. Al pulsar **Listar consultas** se establece una conexión que lleva una instrucción *SELECT* que es resuelta por el motor de Base de Datos. Se muestra el listado asociado al médico con el nombre del paciente y el momento de la consulta sanitaria. Internamente cada cita conlleva un registro en las cuentas con el gasto asociado correspondiente. Se pueden efectuar posteriores búsquedas accionado **Nueva búsqueda** y refrescando los datos de entrada en los espacios en blanco.



6.b. Pantalla **Nueva Cita**

En el formulario para **Nueva Cita** se registran las consultas médicas en función de médico y paciente, fijando necesariamente una fecha y el tipo de cita (*consulta, análisis, revisión o especialista*). En la tabla de consultas de la Base de Datos se realiza una transacción *INSERT* con los valores introducidos por el personal del centro en los campos del menú.

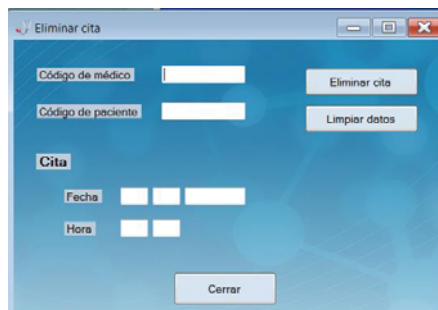
El sistema asimismo crea un apunte contable asociado a cada cita, también una transacción *INSERT*, con el gasto establecido por el centro para el servicio. Este gasto posteriormente será procesado y cargado en la cuenta del paciente. Se incluye **Limpiar datos** para crear nuevas consultas.



6.c. Pantalla **Eliminar Cita**

La ventana de **Eliminar Cita** quita la entrada de una consulta médica en función del identificador del médico o del identificador del paciente según una fecha asignada. Se incluye esta función por la clara necesidad de poder actualizar en tiempo real datos esenciales para el centro. Estas características dotan de dinamismo a la solución y reduce el tiempo dedicado a tareas periódicas de postprocesamiento de datos.

Para facilitar el seguimiento contable no se realiza *DELETE* sobre la tabla de gastos. El sistema ante la restricción de no poder ser borrado un apunte contable ya registrado, crea un nuevo apunte con un comando *INSERT* y un código de apunte único, y con valor del mismo gasto en negativo. El apunte del gasto posteriormente al procesarse se descuenta el importe en la cuenta del paciente. Se permite eliminación múltiple de citas con la acción **Limpiar datos**.



4.1.2. Código

Se desarrolla la aplicación **MedicAp** en lenguaje [C#](#) mediante *Windows Forms* que contempla todas las acciones propias de la funcionalidad descrita. Se emplea la herramienta de desarrollo [Microsoft Visual Studio](#) que integra el código y el diseño de las ventanas de formularios que utilizan los operadores.

El lenguaje de programación orientado a objetos [C#](#) fue desarrollado por la empresa *Microsoft* como parte de la plataforma *.NET*. Es un lenguaje de programación diseñado para la infraestructura de lenguaje común. Su sintaxis básica deriva de *C/C++* y utiliza el modelo de objetos *.NET*. Como lenguaje de programación independiente permite su compilación y generación de ejecutables en diversas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

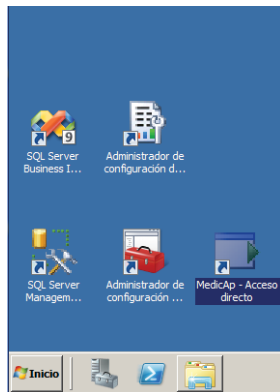
Los controles comunes que ofrece la plataforma *.NET* pueden ser editados para satisfacer los requerimientos de la solución. El programa permite compilar el código e incluye las funcionalidades de diseño de gráficos y publicación de soluciones. Las fuentes de los módulos escritos en lenguaje [C#](#) se archivan en ficheros de extensión *.cs*, adjuntados en la entrega. El flujo de ejecución del aplicativo se activa por las acciones del operador pulsando en los botones que se presentan en las ventanas. Las consultas a la Base de Datos se desencadenan también seleccionando las opciones apropiadas en cada ventana. Según sea requerido, para insertar, modificar y consultar información, y cumplimentando los campos de filtro, se lanzan las transacciones en lenguaje [Transact-SQL](#). La salida de estas instrucciones se sirve asimismo al operador, y en caso de no ser asumibles por alguna incorrección se muestra un mensaje de error con la indicación.

Destacan algunas características que hace adecuada la elección de este recurso. Para la construcción de aplicaciones la recolección de elementos no utilizados automáticamente el programa reclama la memoria ocupada por objetos no utilizados y no accesibles. También aporta el control de excepciones, proporcionando una visión estructurada para la detección de errores y la recuperación. La seguridad del lenguaje con el control de tipos hace imposible leer desde las variables sin inicializar, indizar matrices más allá de sus límites o realizar conversiones de tipos no comprobados.

Con el componente diseñador se facilita la integración en el ejecutable final de las características gráficas definidas en la fase de diseño. Las dimensiones, ubicación y color de cada objeto se configuran aquí. También los fondos de pantalla y las fuentes de los textos se definen en este entorno de desarrollo. Está fundamentado en componentes de software en forma de paquetes independientes y autodescriptivos de funcionalidad. Se presenta un modelo de programación con propiedades, métodos y eventos, que tienen atributos con información declarativa sobre los componentes.

El aplicativo se presenta en formato de ventanas, denominadas *Windows Forms*. El menú ofrecido en cada ventana aloja botones y/o campos para rellenar con información. Los botones desencadenan la apertura de nuevas ventanas, y su cierre, y envío de transacciones construidas según la opción seleccionada, que se aplican en la Base de Datos.

Al proyecto se aporta una interfaz sencilla e intuitiva. Las ventanas, o *Windows Forms*, quedan codificadas en archivos de extensión *.cs*. Cada ventana se corresponde con un fichero y juntos integran la solución en formato *.sln*, de la herramienta [Microsoft Visual Studio](#). Este mismo programa ofrece funcionalidad para publicación y gestión de versionado. La solución se configura con las identidades creadas y las variables de entorno del servidor para posteriormente publicarse y exportarse al entorno virtual. El sistema de ventanas de la aplicación es activado desde *Inicio* o mediante su acceso directo desde el *Escritorio*. Se adjunta al documento el código del programa implementado.



Acceso a aplicación y herramientas de Administración de SQLServer

4.1.3. Implantación

Con los elementos descritos de la solución de [Microsoft Visual Studio](#) se instala el programa en el sistema compatible de la plataforma *Wintel* de la máquina virtual creada a tal efecto con el programa [ORACLE VM VirtualBox](#).

El servidor que ejecuta el motor de base de datos debe ser configurado para aceptar solicitudes de servicio de diversas aplicaciones. Se ejecuta en Sistema Operativo [Windows Server 2008 R2 Standard](#). El SO ofrece la plataforma en que se instalan y configuran los servicios involucrados en la solución de Base de Datos. La capa de seguridad de la instancia de base de datos puede delegar parte de la seguridad en la validación de cuentas de dominio con el SO. Aparte de esto, la instancia ofrece acceso a cuentas propias de SQL en las que lógicamente hay que logarse con contraseña.

Se instalan los servicios *SQL Server*, *SQL Agent*, *SQL Browser*, *SQL Integration Services* y *SQL Reporting Services*. Forman parte de la solución integral del Sistema de Gestión de Bases de Datos. Se comenta su funcionalidad más adelante.

En el caso de estudio, para la interacción de la aplicación **MedicAp** con la Base de Datos **medicDB_fi** se requiere que se configure una cadena de conexión que permita gestionar un control de acceso para dar visibilidad a los datos a diferentes niveles. La ejecución puede ir certificada mediante seguridad integrada en el directorio del dominio o en el directorio local de la máquina. La interfaz se puede configurar dependiendo de los requerimientos y restricciones de cada política de Sistemas. De igual manera la solución permite la configuración para el acceso de otras aplicaciones y sistemas de información incluidos los de otras tecnologías. Se publican *drivers* y desarrollan protocolos para posibilitar este diálogo. La herramienta [SQL Configuration Manager](#) sirve para definir estos aspectos. Además el SO incluye programas como el gestor de conexiones ODBC que facilitan la comunicación entre procesos de distintas tecnologías.

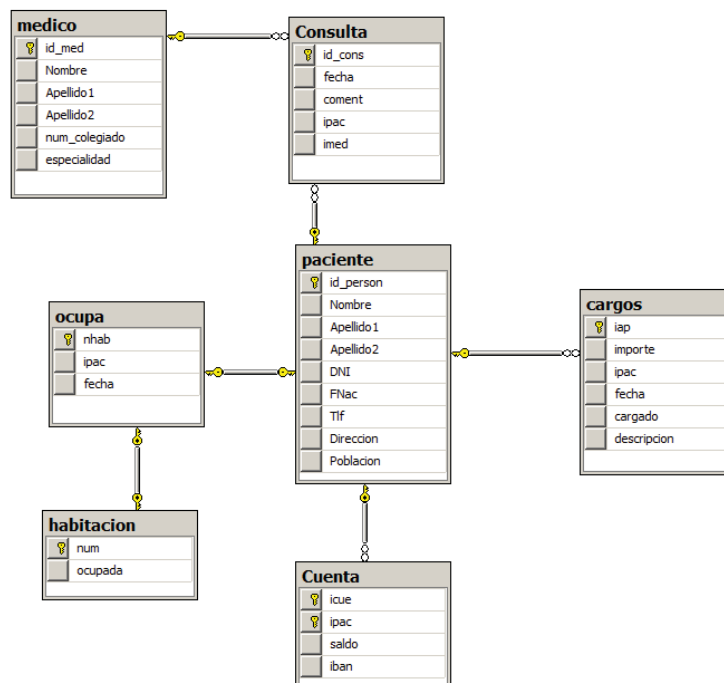
4.2. Base de Datos **medicDB_fi**

Es la entidad en que se ubica la información, el espacio en que se almacenan los datos de personal y pacientes. Se definen tablas, claves, índices, relaciones, vistas y procedimientos implicados en las distintas consultas y transacciones. A la Base de Datos se conectan las aplicaciones que recogen y posibilitan la actividad del centro.

Para esta solución se plantea la creación de una primera versión de base de datos de tipo relacional sencilla que recoja las funcionalidades planteadas y que permita su diálogo con otras entidades y sistemas. Igualmente debe ser escalable y facilitar la integración y aplicación de nuevas ampliaciones.

4.2.1. Diseño

Mediante el modelo E/R se define el diagrama de la Base de Datos. Las entidades y sus relaciones quedan reflejadas en el paso a tablas. En las tablas se detallan los atributos y sus tipos y características. Las restricciones por clave primaria y foránea garantizan la integridad de la información no permitiendo la incorporación de información incoherente con el modelo propuesto.



4.2.2. Código

La Base de Datos [medicDB_fi](#) una vez diseñada ha de ser llevada a código. Este proceso se realiza utilizando el lenguaje de estructurado de consultas SQL, [Transact-SQL](#), para la tecnología de Bases de Datos. *Transact-SQL* es la versión de SQL que implementa *Microsoft* y *Sybase*. Es un lenguaje de cómputo estandarizado para realizar búsquedas, alterar y definir bases de datos relacionales utilizando sentencias declarativas. *Transact-SQL* expande el estándar de SQL para incluir programación procedimental, variables locales, varias funciones de soporte para procesamiento de *strings*, procesamiento de fechas, matemáticas y otros tipos definidos. Es un lenguaje muy potente que permite definir casi cualquier tarea sobre la base de datos. Incluye características propias de cualquier lenguaje de programación que permiten definir la lógica necesaria para el tratamiento de la información: tipos de datos, definición de variables, estructuras de control de flujo, gestión de excepciones y funciones predefinidas.

Se puede utilizar como lenguaje embebido en aplicaciones desarrolladas en otros lenguajes de programación como Visual Basic, C o Java. Y por supuesto los lenguajes incluidos en la plataforma *.NET*. También es posible ejecutar directamente de manera interactiva desde editores de consultas como [SQL Server Management Studio](#).

Con las consultas o *queries CREATE* se originan tanto la Base de Datos como los objetos que la componen. La instrucción *CREATE DATABASE* es capaz de reservar el espacio en disco y componer la estructura de la información. Se identifican nombre y ficheros y su ubicación en el servidor además de otra serie de parámetros como formatos de fecha, juegos de caracteres especiales, modos de recuperación, etc. De igual forma se crean los objetos, invocando *CREATE <TABLE, VIEW, PROCEDURE, USER, INDEX, KEY>*. Este código se adjunta en la entrega del trabajo.

En la creación de tablas se indican en este lenguaje nombres, atributos y claves primarias. También las tablas de relaciones recogen la interactividad entre elementos o entidades de la Base de Datos. Para esto se declaran claves foráneas cuyos registros han de tener su correspondencia en otras tablas de la Base de Datos. Esta característica propia de las Bases de Datos Relacionales obliga a construir los objetos en un orden correcto puesto que por lógica no se pueden crear objetos que referencien a entidades aún no existentes.

En el caso de las inserciones o borrado de datos, y también con la eliminación de objetos, se tiene que seguir el mismo criterio. Los mecanismos que protegen la integridad del conjunto de información bloquean estas peticiones. Para insertar los datos iniciales en la Base de Datos se programan unas cargas de información. Y en el segundo caso se utiliza un script de *Transact-SQL* como en el caso de la generación de objetos que los elimina ordenadamente siguiendo un criterio de desapilamiento. Las peticiones de la aplicación a la Base de Datos también se realizan mediante consultas de lenguaje *Transact-SQL* o transacciones.

4.2.3. Objetos

Existen diversos objetos en la Base de Datos que posibilitan la manipulación de la información. Las tablas representan entidades con sus propios atributos, claves e índices (*paciente, medico, Cuenta, cargos, habitacion*) y relaciones también con sus propios atributos (*Consulta, ocupa*). Se crean vistas para crear conjuntos de datos, procedimientos para realizar búsquedas, inserciones, búsquedas y actualizaciones, y relaciones que implementan los mecanismos que garantizan la integridad del sistema informático mediante el uso de claves primarias y foráneas. Se detallan:

a. Tablas. Entidades y Relaciones

Las tablas de datos recopilan información sobre entidades y relaciones entre elementos de estas. Las entidades tienen atributos que corresponden a información en determinados formatos de texto. Se guardan cadenas de caracteres como los nombres o especialidades médicas, y fechas como las de nacimiento o de una cita clínica.

Para alcanzar un grado de normalización adecuado se declaran claves o códigos únicos para los registros de las tablas (claves primarias) que garantizan la no existencia de filas repetidas o inconsistentes con el modelo. Definiendo asimismo claves foráneas se pueden especificar entradas de relaciones entre las entidades. La definición de las tablas de la Base de Datos y sus atributos:

cargos - Registra los apuntes contables asociados a cada paciente con fecha, descripción y registro de cargo en cuenta:

cargos (ia int, importe int, ipac int, fecha date, cargado bit, descripcion varchar(50))

Consulta - Registra las consultas sanitarias asociadas a cada paciente y médico con fecha y descripción:

Consulta (id_cons int, fecha smalldatetime, coment varchar(50), ipac int, imed int)

Cuenta - Registra las cuentas de cada paciente. Se almacenan saldo, debido a los gastos imputados y cuenta bancaria en que efectuar los cargos:

Cuenta (icue int, ipac int, saldo int, iban nchar(30))

habitacion - Registra el estado de las dependencias de la residencia:

habitacion (num int, ocupada bit)

medico - Registra los datos de los facultativos del centro. Se recopila información del número de colegiado y la especialidad:

medico (id_med int, Nombre varchar(50), Apellido1 varchar(50), Apellido2 varchar(50), num_colegiado int, especialidad varchar(50))

ocupa - Registra los datos de ocupación de un paciente y la fecha de ingreso:

ocupa (nhab int, ipac int, fecha date)

paciente - Registra los datos de los pacientes, con su código interno personal:

paciente (id_person int, Nombre varchar(50), Apellido1 varchar(50), Apellido2 varchar(50), DNI nchar(10), FNac date, Tlf nchar(12), Direccion varchar(100), Poblacion varchar(50))

b. Vistas

Se definen dentro de la Base de Datos una serie de vistas que permiten crear conjuntos de datos en base a consultas, que se utilizan con frecuencia y facilitan el manejo de información y la creación de informes.

Un aspecto a valorar en el funcionamiento del motor de Base de Datos es el control que realiza basándose en las estadísticas de uso. Crea planes de ejecución en función de los costes en las consultas que se realizan y del número de registros de las tablas que se cruzan. Esta evaluación continua aconseja la preferencia de emplear de forma recurrente el mismo conjunto de consultas. Con las vistas y los procedimientos se facilita la reutilización de estructuras de búsqueda y por tanto la depuración de las mejores estrategias para recorrer las tablas. Por ello se procura descartar en la medida de lo posible el uso de consultas para búsquedas específicas o *ad hoc*.

Cargos_x_mes -	Cargos_x_mes_cargados
Cargos_x_mes_pendientes	Cargos_x_Paciente
Cargos_x_Paciente_procesado	Consulta_x_Medico
Consulta_x_Paciente	Medico_x_Especialidad
Medico_x_Nombre	Pacientes_x_Nombre
Saldo_x_Paciente	Tarjeta_x_Paciente
Total_Cargos	Vista_habitaciones

c. Usuarios, procedimientos, índices, claves

usuarios, que acceden con distintos grados de privilegios. Se crean usuarios de dominio para cuentas de servicio (*\sql*, *\sqlagent*) y para administración de la Base de datos (*\Administrador*). También se crea el usuario SQL *user_read*, que se valida directamente y se utiliza por la aplicación para establecer conexiones a la Base de Datos.

procedimientos, para crear transacciones determinadas pero con distintos literales se definen algoritmos que toman valores de entrada utilizados en las búsquedas y modificaciones. *Alta_Hosp* actualiza las tablas de ocupación de habitaciones cuando un paciente deja una libre; *Ingreso* asigna una habitación a un paciente; *Nueva_Cita* asienta una consulta solicitada al sistemas; *Realizar_Ingreso* muestra el número de habitación libre seleccionado para la próxima orden de ocupación.

índices, las claves primarias de las tablas por definición son índices *clusterizados* (registros almacenados en disco aprovechando la proximidad espacial).

claves, además de las primarias están las claves foráneas que se definen en base a las primeras y soportan la integridad del conjunto de información.

4.2.4. Almacenamiento

La abstracción del concepto Base de Datos requiere un espacio o recursos físicos donde poder ser implementada. En el caso de la tecnología que se utiliza la información se almacena en ficheros de dos tipos: el fichero de datos y el fichero de log.

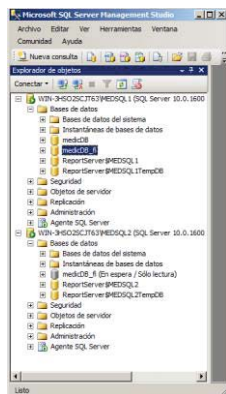
La generación de la Base de Datos se produce mediante una consulta ejecutada contra el motor de Base de Datos utilizando el lenguaje *Transact-SQL*. El servicio del motor tiene que ser configurado con cuentas locales o de dominio que puedan en este caso acceder al sistema de ficheros para manipular ficheros y directorios. Aparte de los ficheros de las Bases de Datos de usuario, han de existir los ficheros de las Bases de Datos de sistema: *master*, *msdb*, *model* y *tempdb*. Es importante ubicar adecuadamente estos ficheros, y por tanto estas Bases de Datos, en el servidor. Existen mecanismos de cifrado y seguridad que pueden hacer su exportación y reproducción en otras máquinas algo inviable.

Los valores de rendimiento de acceso a disco de los distintos puntos de montaje presentes tienen una incidencia crítica en el funcionamiento general del sistema. Tal cuestión hace recomendable utilizar en el caso de la *tempdb*, base de datos temporal, dispositivos rápidos sobre todo en entornos de alto volumen de transaccionalidad. En el proceso de instalación del servicio de motor de Base de Datos se configuran, entre otros aspectos, las rutas y accesibilidades de los archivos en el sistema de ficheros.

También se requiere almacenamiento, normalmente elegido de menor *calidad* por sus menores exigencias de rendimiento, para realizar las tareas de copia de seguridad. El diseño de la herramienta *SQL Server* se ha optimizado con el objetivo de establecer puntos de recuperación en ejecución que facilitan la realización de copias de respaldo. La gestión de los recursos que se le conceden al proceso priorizan dinámicamente las exigencias de uso. El balanceo de carga de trabajo se evalúa de forma constante y se aplican los ajustes necesarios. En la configuración del motor de Base de Datos se puede elegir el modo y el grado de paralelismo a la hora de resolver las consultas requeridas.

El fichero de datos en este caso es único aunque la herramienta permite y dado el caso recomienda la existencia de otros ficheros. Este fichero, típicamente con extensión *mdf*, recopila la información propia de la solución como la información interna o *metadata* de la misma Base de Datos: diseño (diagrama, relaciones, roles), configuración (idioma, ficheros, propietario) y objetos (tablas, usuarios, procedimientos, vistas, índices y restricciones). Los ficheros secundarios suelen tener extensión *ndf*, pueden ser agrupados y se pueden utilizar para aprovechar facilidades de paralelismo a la hora de realizar operaciones IO con el uso de varios dispositivos, incluso de distintas características. Otra facilidad aportada es la de ordenación de información, trabajando en combatir la alta fragmentación, y la de ordenación o ubicación interna de objetos, tanto de objetos en páginas como de ordenación de los mismos registros dentro de las tablas que los contienen (declaración y reorganización de índices, estudio de estadísticas, chequeo de integridad).

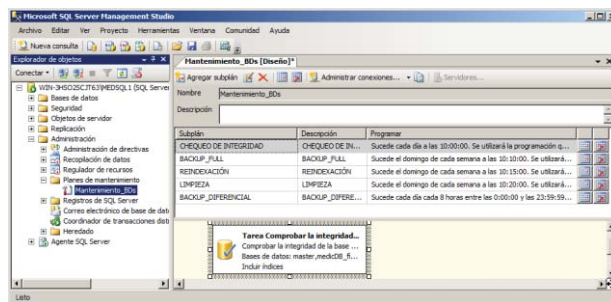
El fichero de log o fichero de transacciones, típicamente *ldf*, tiene la función de recoger la actividad que incide en la Base de Datos. Todas las transacciones validadas quedan guardadas en este fichero. Configurada la Base de Datos en modo de recuperación *completo* ofrece la posibilidad de restaurarla a un estado previo en un momento determinado. De esta forma, el procedimiento de restauración parte de una copia de seguridad completa a la que secuencialmente se le aplican restauraciones parciales aplicando las transacciones que se van recogiendo en la actividad diaria. Este mismo mecanismo permite, y es el que se ha implementado en este proyecto, crear un replicación de trasvase de registros o *log shipping*. Esta situación deja la BD de réplica en modo de lectura ya que las transacciones sólo son aplicables en el nodo principal y recogidas posteriormente en el secundario.



*Instancias de BDs
Principal y secundaria*

4.2.5. Mantenimiento

La solución incluye una estrategia de mantenimiento a nivel de bases de datos tanto de usuario como de sistema. Se elabora un **Plan de mantenimiento** compuesto por un conjunto de trabajos que se definen y programan, que consta de chequeo de integridad y reorganización de datos, y de creación y ciclado de copias de seguridad tanto de tipo completo como de tipo transaccional. En la vida de una Base de Datos de tipo relacional la continua realización de inserción, actualización y borrado de datos conlleva asociada una degradación o fragmentación a nivel de almacenamiento, tanto interno como externo, que pueden afectar a su rendimiento. Esta situación hace necesaria la implementación de un conjunto de mecanismos basados en la proximidad espacial de los datos dentro de las tablas que se ubican en páginas, el aprovechamiento de los recursos de memoria asignados (en este caso el motor de Bases de Datos SQL Server instalado en un entorno *Wintel* tiene un nivel de privilegio que permite bloquear y gestionar páginas de memoria).



Plan de mantenimiento

- a. **Chequeo de Integridad.** Tarea que revisa la integridad del modelo y la coherencia de la información de las Bases de Datos. Se programa una revisión diaria en horario de baja actividad. Revisa estructuras de registros e índices, ofrece opciones de evaluación y reparación de inconsistencias.
- b. **Reindexación y Actualización de Estadísticas.** En base al uso previsto de las Bases de Datos y como parte de la fase de diseño se definen los índices, con los que se busca la organización más óptima de los datos en función de la actividad transaccional, del tipo de consultas y sus incidencias. La entrada y salida continua de registros refrescan datos de uso y estos se utilizan posteriormente en búsqueda de optimizar los recursos según los algoritmos que definen los planes de ejecución de las instrucciones.

Ejecuciones de reindexado pueden, y habitualmente lo hacen, incidir negativamente en el rendimiento del motor de Base de Datos. Son transacciones costosas que provocan la activación de mecanismos de bloqueo responsables de garantizar el acceso exclusivo a páginas de memoria que se están modificando. Es aconsejable trasladar a un horario de actividad valle estas peticiones, que no dejan de ser movimientos de datos, y finalmente por tanto modificaciones en la información.

- c. **Sistema de Respaldo.** Para poder recuperar el sistema a un estado previo determinado ya sea por una situación de *disaster recovery*, para realizar una migración entre plataformas o para poder implementar el mecanismo de replicación se programan una serie de trabajos que realizan, copian entre recursos y restauran de forma total o parcial la Base de Datos con la que trabaja el Sistema Informático del Centro Sanitario.

Se propone una estrategia de copia de seguridad completa diaria, con dos copias de seguridad diferenciales y copias transaccionales cada dos minutos. Así se garantiza recuperación de situación a lo sumo dos minutos de un punto determinado (siendo en muchos casos posible precisar bastante más).

Utilizando esta misma funcionalidad se crea una solución de réplica que permite derivar consultas de tipo lectura consiguiendo así descargar de actividad al sistema. Existen herramientas de control financiero que demandan gran cantidad de procesamiento de información y con este tipo de mejoras se refleja en la calidad de servicio que se provee. También la replicación permite crear un servicio de publicación y creación de informes en WEB.

4.3. Implantación

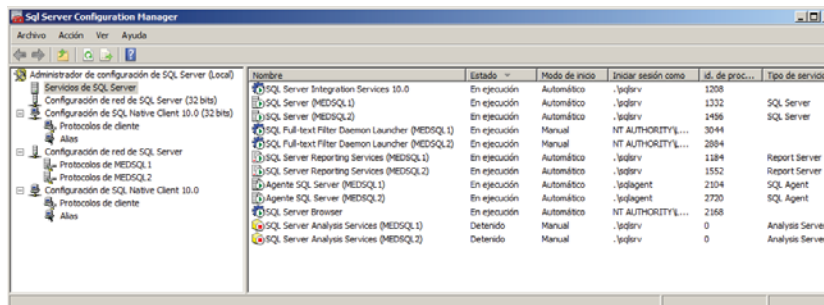
Se crea un servidor virtual con la herramienta [VirtualBox](#). Ejecutando en el Sistema Operativo [Windows Server](#) se instala el Motor de Base de Datos [Microsoft SQL Server](#).

Los servicios o demonios son procesos que se ejecutan de forma continua y que requieren validación según la configuración de seguridad del servidor y las políticas implantadas a niveles locales y de dominio. Los servicios de SQL Server se configuran para iniciarse automáticamente y se crean cuentas con privilegios avanzados sobre las que ejecutarse.

Se instalan los servicios de motor de Base de Datos [SQL Server](#), del agente para la programación de tareas, [SQL Agent](#), para mantenimiento y ejecución de funcionalidades recurrentes desatendidas. También se integran un explorador de motores, [SQL Browser](#), un gestor de paquetes de integración (como son las tareas de mantenimiento) denominado [SQL Integration Services](#), y el componente de configuración de servicios y protocolos de comunicaciones [SQL Configuration Manager](#).

El Administrador de configuración es un complemento de [Microsoft Management Console](#) para configurar los protocolos de red utilizados por SQL Server y para administrar la configuración de conectividad de red de los equipos cliente de SQL Server.

En la captura anexa se muestra la configuración de los servicios: el modo de inicio y el usuario que ejecuta (en este caso usuarios de ámbito local). En el caso de estudio se definen dos instancias de Bases de Datos en que se crean las Bases de Datos de usuario y sistema en que se aloja toda la información.



Los servicios de Bases de datos de esta tecnología:

[SQL Server](#), el motor de Base de Datos que se ejecuta como un proceso. Provee y almacena los datos a los programas que se conectan a la Base de Datos. Proporciona acceso controlado y procesamiento de transacciones rápido para cumplir con los requisitos de las aplicaciones consumidoras de datos. Sirve asimismo para crear bases de datos relacionales para el procesamiento de transacciones en línea o datos de procesamiento analíticos en línea. Se pueden crear tablas para almacenar datos y objetos de base de datos para ver, administrar y proteger los datos.

[SQL Agent](#), para mantenimiento y ejecución de funcionalidades recurrentes desatendidas. Como servicio de *Microsoft Windows* ejecuta tareas administrativas programadas, denominadas trabajos o *jobs* en SQL Server. Los trabajos contienen uno o más pasos y cada paso contiene su propia tarea. Se puede ejecutar según una programación, como respuesta a un evento específico o a petición. Tiene un control de errores que si encuentra un problema, el *Agente SQL Server* puede registrar el evento y notificarlo. Puede enviar notificaciones con la salida de la ejecución mediante correo electrónico o definiendo y utilizando operadores.

La base de datos de sistema llamada *msdb* es donde reside la información que posibilita su funcionamiento: trabajos, programaciones y roles. Los roles fijan niveles de lector, ejecutor y administración de tareas. Se controla el acceso al *Agente SQL Server* para los usuarios que no son miembros del rol de administrador. Además de estos roles fijos de base de datos, los subsistemas y los servidores proxy ayudan a los administradores de bases de datos a garantizar que cada paso de trabajo se ejecuta con los permisos mínimos necesarios para realizar la tarea.

[SQL Browser](#) se ejecuta como otro servicio de Windows Server. *SQL Browser* escucha las solicitudes entrantes de recursos de *Microsoft SQL Server* y proporciona información acerca de las instancias de SQL Server instaladas en el equipo. *SQL Browser* permite examinar una lista de los servidores disponibles, conectarse a una instancia del servidor o a los extremos de la conexión de administrador dedicada. Para cada instancia de motor de Base de Datos el servicio *SQL Browser* proporciona el nombre de la instancia y el número de versión.

Al iniciarse la instancia se asigna un puerto TCP/IP al servidor si el protocolo TCP/IP está habilitado. Si el protocolo de canalizaciones con nombre está habilitado, la escucha se hace en una canalización con nombre específica. Esa instancia específica utiliza dicho puerto, o canalización, para intercambiar datos con las aplicaciones cliente. Puesto que solo una instancia de SQL Server puede utilizar un puerto o una canalización, se asignan números de puerto y nombres de canalizaciones diferentes para las instancias con nombre. De forma predeterminada, cuando están habilitados, las instancias con nombre están configurados para utilizar puertos dinámicos, por lo que se asigna un puerto disponible cuando se inicia *SQL Server*. También se puede asignar un puerto determinado a la instancia de SQL Server. Cuando inicia *SQL Browser* reclama el puerto UDP 1434. *SQL Browser* lee el Registro, identifica todas las instancias de SQL Server en el equipo y registra los puertos y las canalizaciones con nombre que utilizan. Cuando un servidor tiene dos o más tarjetas de red, *SQL Browser* devuelve el primer puerto habilitado que encuentra.

[SQL Integration Services](#) es una plataforma para la creación de soluciones de transformación e integración de datos. Es adecuado para resolver complejos problemas empresariales mediante la copia o descarga de archivos, el envío de mensajes de correo electrónico como respuesta a eventos, la actualización de almacenamientos de datos, la limpieza y minería de datos, y la administración de objetos y datos del motor de Base de Datos de *SQL Server*. Los paquetes pueden funcionar en solitario o

junto con otros paquetes para resolver complejas funcionalidades. *Integration Services* puede extraer y transformar datos de diversos orígenes como archivos de datos XML, archivos planos y orígenes de datos relacionales y cargarlos en uno o varios destinos. Contiene un variado conjunto de tareas y transformaciones integradas, herramientas para la creación de paquetes y el servicio *Integration Services* para ejecutar y administrar los paquetes. Las herramientas gráficas posibilitan crear soluciones sin uso propiamente de código. También se puede programar el amplio modelo de objetos de *Integration Services* para crear paquetes mediante programación y codificar tareas personalizadas y otros objetos de paquete.

[SQL Reporting Services](#) es el servicio que posibilita la publicación de los resultados de búsquedas predefinidas de consulta a tablas de datos mediante informes previamente diseñados en un espacio WEB creado a tal efecto. Se configura igual que los otros servicios con un usuario determinado y en el modo manual o automático. Se describe con más detenimiento en el apartado de *Sistema de Publicación de informes*.

Parte esencial del sistema se sostiene precisamente utilizando información propia del proceso en bases de datos denominadas de sistema y que en el caso de esta tecnología son: *master*, *msdb*, *model* y *tempdb*.

La base de datos [master](#) registra toda la información de sistema de un sistema SQL Server: los metadatos, las cuentas de inicio de sesión, los extremos, los servidores vinculados y la configuración del sistema. *master* es la base de datos que registra la existencia de las demás bases de datos, la ubicación de los archivos de las bases de datos y la información de inicialización de SQL Server, y por tanto, no puede iniciarse si no está disponible.

La base de datos [model](#) se utiliza como plantilla para todas las bases de datos creadas en una instancia de SQL Server. Se crea nueva cada vez que se inicia el motor. Todo el contenido de *model*, incluidas las opciones de base de datos, se copia en la base de datos nueva. Por esto siempre debe existir en un sistema SQL Server .

La base de datos [msdb](#) es la utilizada por *SQLAgent* para programar alertas y trabajos. Otras características como *SQL Server Management Studio*, *Service Broker* y Correo electrónico también usan esta base de datos.

La base de datos [tempdb](#) es un recurso global disponible para todos los usuarios conectados a la instancia de SQL Server. Contiene objetos de usuario temporales creados explícitamente, procedimientos almacenados temporales, variables de tabla, tablas devueltas en funciones con valores de tabla o cursores. También objetos internos que se crean mediante el motor de base de datos, como tablas de trabajo para almacenar resultados intermedios para colas, cursores, ordenaciones y almacenamiento temporal de objetos grandes. Las operaciones de *tempdb* se registran de forma mínima, por lo que las transacciones se pueden revertir. *tempdb* se vuelve a crear cada vez que se inicia

SQL Server , de forma que el sistema siempre se inicia con una copia limpia de la base de datos. Las tablas y los procedimientos almacenados temporales se quitan automáticamente en la desconexión y ninguna conexión permanece activa cuando se cierra el sistema. Por tanto, no debe guardarse información de una a otra sesión de SQL Server y tampoco se permite realizar operaciones de copia de seguridad y restauración.

SQL Server emplea también los denominados [planes de ejecución](#) para la optimización de consultas. Tienen la capacidad de evaluar el costo de ejecución de las instrucciones y consultas específicas en SQL Server. También se pueden mostrar de forma gráfica los métodos de recuperación de datos que usa el optimizador de consultas. Se distingue el plan de consultas estimado que es generado sin realmente haber lanzado la consulta y se basa en una estimación del comportamiento esperado; y el real, que se genera después de que una consulta sea ejecutada. Este es más confiable ya que está basado en la ejecución real. También provee más información y estadísticas, por lo que es mucho más práctico al resolver problemas.

La consola de Administración ofrece acceso a cada espacio de objetos (tablas, usuarios, trabajos, etc) para posibilitar su creación, consulta y modificación de propiedades. Ofrece herramientas para gestión de procesos como el Monitor de Actividad, o *Activity Monitor*, con detalles de usuario, naturaleza de la operación, host de origen y consumo de CPU y operaciones de I/O; informes predefinidos para chequeo de recursos control, como espacios usados por objetos o relación de consultas costosas; y el monitor de trabajos, con acceso a estado e historial de ejecuciones de las tareas programadas.

4.4. Sistema de Replicación

Implementado con el método de [trasvase de registros](#) (*log shipping*). Partiendo de la restauración de una copia de seguridad completa se van restaurando copias de seguridad transaccionales que recogen a intervalos definidos de tiempo toda la actividad que se va produciendo en la Base de Datos Principal. La Base de Datos Secundaria queda en modo lectura y permite que gran parte de las consultas de datos puedan ser redirigidas a un Motor de Base de Datos alternativo. Este mecanismo posibilita balancear la carga de actividad del sistema y descargar el procesamiento de consultas costosas del motor principal. También ofrece una alternativa de recuperación del sistema de Base de Datos de un modo muy eficiente configurando, en el momento que sea necesario, la Base de Datos Secundaria para que asuma el rol de Base de Datos Principal, minimizando el tiempo de corte de servicio y la pérdida de datos en situaciones de desastre.

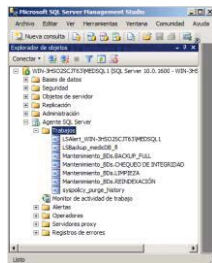
Ofrece la posibilidad de realizar este envío de transacciones desde la base de datos principal de una instancia del servidor principal a una o varias bases de datos secundarias en instancias independientes del servidor secundario. Las copias de seguridad del registro de transacciones se aplican a cada una de las bases de datos secundarias de forma individual. En una tercera instancia de servidor opcional, denominado servidor de supervisión, se registra el historial y el estado de las operaciones de copias de seguridad y restauración y, opcionalmente, se activan alertas si estas operaciones no se producen según lo programado. De esta forma proporciona la solución de recuperación ante desastres en una o más bases de datos secundarias, cada una en una instancia independiente de SQL Server.

El hecho de que haya acceso limitado de solo lectura a las bases de datos secundarias durante el intervalo entre los trabajos de restauración permite montar el mecanismo de publicación de informes que se explica en la siguiente sección del documento.

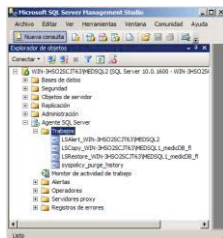
Por último reseñar que es configurable asimismo el retraso o *delay* definido por el usuario entre el momento en que el servidor principal realiza una copia de seguridad del registro de la base de datos principal y el momento en que los servidores secundarios deben restaurar la copia de seguridad de registros. Ajustando este valor de retraso se adecúa el sistema a los requerimientos de la solución. Además, en caso de realizarse un cambio accidental en la Base de Datos, el retraso puede permitirle recuperar los datos aún sin modificar de una base de datos secundaria antes de que el cambio se refleje en ella. Hay que indicar que en el trasvase de registros no se realiza automáticamente la conmutación por error del servidor principal al servidor secundario. Si la base de datos principal deja de estar disponible cualquiera de las bases de datos secundarias se puede poner en línea manualmente. Con este mecanismo la base de datos secundaria transita entre los estados de *SOLO LECTURA* y *RESTAURANDO* y por tanto se puede aprovechar esta circunstancia de diversas formas. Como ya se ha indicado se aplica en esta propuesta en el caso de la publicación de informes

También se configuran la frecuencia de copiado, el tiempo de retención de copias y el margen de alerta que avisa de posibles problemas en la replicación. Puede adaptarse para que una instancia opcional de SQL Server realice el seguimiento de todos los detalles del trasvase de registros. Una vez asignado el servidor de supervisión no puede modificarse sin quitar primero el trasvase de registros. En este caso la supervisión del estado de réplica se realiza desde las dos instancias implicadas. Se revisan momento en que se realizó por última vez una copia de seguridad del registro, cuando fue copiada en el destino de acceso compartido, el tiempo de restauración de los archivos de copia de seguridad en los servidores secundarios e información acerca de las alertas de error de copiado. Los trabajos de copias de seguridad son gestionados por el servicio [SQL Agent](#), que lleva a cabo las operaciones y registra el historial en el servidor local y el servidor de supervisión. También elimina los archivos de copia de seguridad y la información de historial antiguos.

Tareas involucradas en el servidor principal: copia de seguridad, copia de archivos, alerta.



Tareas en el servidor secundario (modo lectura/restaurando): trabajo de restauración, alerta.



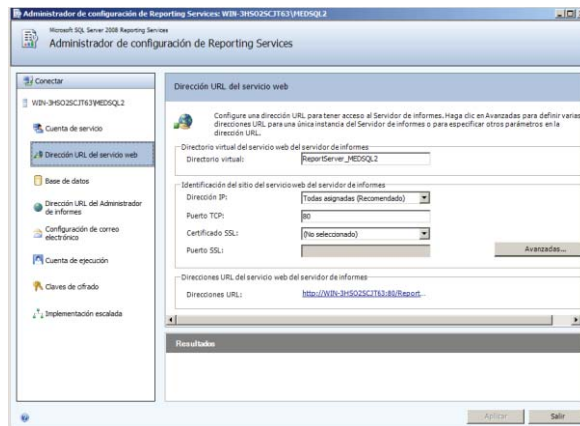
La funcionalidad interna de la consola de administración ofrece un informe ya definido de supervisión del estado del trasvase de registros.



Informe de estado de Traslase de Registros

4.5. Sistema de Publicación de informes

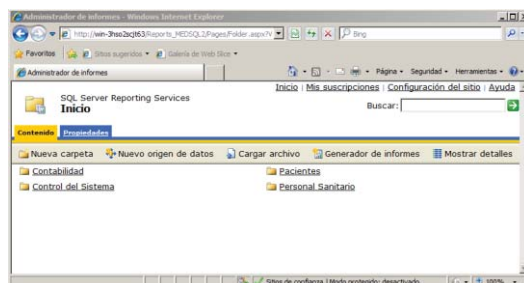
El Servidor de Informes se instala y configura con la herramienta [SQL Server Reporting Services](#). Permite la gestión de la información en el servidor de informes. Se crea un espacio o *site* para publicación WEB de distintas consultas e informes de la Base de Datos. Los distintos informes de este proyecto se diseñan, implementan y publican con la herramienta [SQL Server Business Intelligence Development Studio](#), posibilitando una interacción directa y dinámica para diseñar, implementar y cargar informes en el Sistema.



Consola de Administración del sitio WEB

La administración de contenidos se realiza mediante el Directorio de Informes. Después de validarse el usuario accede a esta interfaz que permite publicar informes en entornos WEB. Los ficheros *.rdl*, codificados con *SQL Server Business Intelligence Development Studio*, contienen las definiciones del informe y se cargan desde esta funcionalidad.

Cada informe puede utilizar una cadena conexión propia o una común definida previamente, siempre según los privilegios de administración concedidos. Existen diversos perfiles de seguridad, que buscan garantizar tanto el origen de la información como la protección con que tiene que ser gestionada cada clase de información.



Directorio de Informes

A continuación se adjuntan algunos Informes creados para uso de los operadores del sistema. Se dispone de la información con el retardo configurado en el sistema de réplica. Los informes sólo solicitan transacciones de consulta, o en modo *lectura*, lo que los hace idóneos para ser utilizados en sistemas secundarios. Además es habitual que en momentos puntuales sean empleados con notable profusión o sean creados informes que requieran importantes exigencias de recursos. Toda esa actividad delegada deja de tener que ser atendida por la Base de Datos primaria, incidiendo positivamente de forma importante en el sistema. A continuación se presentan ejemplos de informes diseñados.

Información sanitaria:

Núm. Habitación	Apellidos	Nombre
1002	Sánchez	Ojeda Martín
1003	Molina	Fariás Sara
1004	Herrera	Correa Victoria
1005	Núñez	Flores Gabriela
1006	Flores	Ramos Gabriel
1007	Marthez	Castillo Samuel
1008	Cabrera	Medina Samantha
1009	Díaz	Quiroga Diego
1010	Pérez	Ledesma Nicolás
1011	Ferreira	Herrera Mía
1012	Luna	Acosta Elena
1013	Benitez	Coronel Adrián

Informe de ocupación en la Residencia:

importe	fecha	descripción	cargado
100	28/10/2010	Ortopedia	Falso
300	13/11/2010	Rehabilitación	Falso
210	16/11/2010	Consulta	Falso

Informe de los cargos de cada paciente

Información sobre estado del sistema:

Backup Start Date	type	physical device name
07/07/2018 16:58:02	L	C:\MSDQ\Servidor\medicDB_20180707145902.bk
07/07/2018 16:58:00	L	C:\MSDQ\Servidor\medicDB_20180707145800.bk
07/07/2018 16:57:00	L	C:\MSDQ\Servidor\medicDB_20180707165700.bk
07/07/2018 16:56:01	L	C:\MSDQ\Servidor\medicDB_20180707145601.bk
07/07/2018 16:55:03	L	C:\MSDQ\Servidor\medicDB_20180707145503.bk
07/07/2018 16:54:00	L	C:\MSDQ\Servidor\medicDB_20180707145400.bk

Informe de trabajo de copia de seguridad

last restored file	last copied date	last restored date	latency
medicDB_20180624191700.bk	24/06/2018 21:18:02	24/06/2018 21:18:02	1

Informe de estado de la réplica

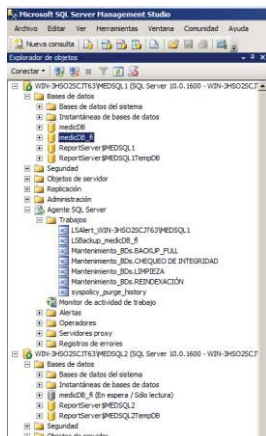
nombre	descripción	estado	fecha	horas
Mantenimiento_BD_BACKUP_DIFERENCIAL	No hay ninguna descripción.		0	0
Mantenimiento_BD_CHEQUEO DE INTEGRIDAD	Chequeo de integridad.	El trabajo concluyó correctamente. El trabajo fue invocado por Programar 17 (Mantenimiento_BD_Suscripción). El último paso ejecutado fue 1 (DESCRIBIR DETALLES).	20180617	500
Mantenimiento_BD_BACKUP_FULL	Copia de seguridad completa.	El trabajo concluyó correctamente. El trabajo fue invocado por Programar 18 (Mantenimiento_BD_BACKUP_FULL). El último paso ejecutado fue 1 (SHOW FULL).	20180617	1001
Mantenimiento_BD_REINDEXACION	Reindexación y actualización de estadísticas.	El trabajo concluyó correctamente. El trabajo fue invocado por Programar 19 (Mantenimiento_BD_REINDEXACION). El último paso ejecutado fue 1 (REINDEXACION).	20180617	1500
Mantenimiento_BD_LIMPIEZA	Limpieza de historial y copias antiguas.	El trabajo concluyó correctamente. El trabajo fue invocado por Programar 20 (Mantenimiento_BD_LIMPIEZA). El último paso ejecutado fue 1 (LIMPIEZA).	20180617	2000

Informe de estado de mantenimiento

5. EJECUCIÓN. CASO DE PRUEBA

Para la evaluación de la propuesta se implementa una solución integral y se instala en un servidor virtual creado para tal propósito. Utilizando el programa [ORACLE VM VirtualBox](#) se crea la máquina en la que se instala el Sistema Operativo [Windows Server 2008 R2 Standard](#) para asumir el rol de servidor. Ejecutando bajo el SO se instalan los servicios necesarios de [Microsoft SQL Server Enterprise](#) que proveerán las funcionalidades de Bases de Datos comentadas previamente.

En la misma máquina se instala la aplicación **MedicAp** que ofrece la interfaz operativa. Esta solución de ventanas establece conexiones con la Bases de Datos cuando tiene que consultarla. Las operaciones realizadas trabajan sobre los registros guardados en las tablas. En los casos de las operaciones de búsquedas no se realizan modificaciones, en las otras operaciones se ejecutan comandos para inserción, modificación y borrado de información. Su uso es restringido al personal del centro. La aplicación tiene su ámbito de acción exclusivamente en la Base de Datos principal. La Base de Datos secundaria atiende otras peticiones y esto permite balancear la carga de trabajo requerida para el conjunto de la solución (en este caso se reserva para la consulta de informes financieros y de estado del servicio). Ambas Bases de Datos están implementadas en sus propios contenedores, en este caso instancias independientes de motor de Bases de Datos *SQLServer*. Por tanto existe toda la estructura descrita anteriormente para cada Base de Datos **medicDB_fi**, con sus Bases de Datos de sistema y de usuario propias, y lógicamente toda la estructura de ficheros que necesita para implementarlas. Igualmente tiene sus propios usuarios, cuentas de servicio, trabajos, almacenamiento, historial, etc.



*Consola de Administración de SQLServer
Instancias principal y secundaria*

Para implementar la replicación, que normalmente se configura entre servidores remotos, se utiliza una segunda instancia ejecutando en el mismo servidor que a efectos prácticos se comporta como un sistema de base de datos independiente con su propia configuración y

recursos. La comunicación entre las dos instancias se configura para que sean accesibles remotamente mediante un protocolo elegido. El trasvase de ficheros de copia transaccional se realiza mediante un recurso compartido con permisos de acceso concedidos a las cuentas de servicio que lanzan los servicios de los motores de bases de datos.

Para empezar, y una vez que se encuentra todo el entorno en funcionamiento, se crea la Base de Datos primaria en la instancia principal. Acto seguido se realiza una copia de seguridad completa a disco en un fichero, y este se copia en el recurso compartido accesible por las dos instancias de *SQLServer* (los servicios de motor de Bases de Datos). Luego se restaura la Base de Datos dejándola en modo restauración o *restoring* (modo *NO_RECOVERY*), que impide ser accedida y en espera de ponerse en modo funcional o de recibir la siguiente copia de seguridad transaccional (si se aplicase una restauración completa simplemente sería una sobrescritura de Base de Datos).

Inicialmente, una vez creadas las Bases de Datos y el mecanismo de replicación, las tablas se encuentran vacías, sin datos. A este estado se puede volver en cualquier momento (se facilita el script en el fichero *DROP_ALL.sql*). Tanto los scripts de creación como de eliminación y los de carga de actividad al estar escritos en *Transact-SQL* son transacciones como el resto, lo que quiere decir que son propagadas a la instancia secundaria de igual forma.

La carga de datos iniciales en la instancia principal se realiza con los scripts de *Transact-SQL* de creación preparados y ejecutados en el orden adecuado. Para ello se invoca el programa de ejecución por lotes *CARGA_INICIAL.bat* que establece conexiones y carga archivos *.sql*, ubicados en *..\entrada* conteniendo el código SQL. Las salidas de las diferentes ejecuciones se redirigen al directorio *..\salida* para consultar posibles errores en el proceso de carga. Si todo se realiza adecuadamente se carga información de actividad referida a 4 meses. Existen registros de médicos y pacientes. También de citas realizadas, de habitaciones ocupadas y de cargos imputados a los pacientes. En este estado se ya puede interactuar con la Base de Datos **medicDB_fi** mediante la aplicación **MedicAp**. La funcionalidad descrita en las secciones previas ya se encuentra activa y el sistema preparado para recibir comandos.

Al iniciarse el servidor los servicios *de SQLServer* comienzan a ejecutar y las Bases de Datos continúan con su actividad desde el último momento en que fueron paradas. Se leen los datos de las Bases de Datos de sistema y las configuraciones definidas por los administradores se aplican en el sistema. Las Bases de Datos de usuario conectan con sus ficheros de datos y transacciones y entran en modo *ONLINE*.

Con el inicio del servicio *SQLAgent* comienzan a realizarse las tareas de mantenimiento siguiendo la programación establecida: las propias de mantenimiento (*chequeo, reindexación,*

copia de respaldo y estadísticas) y las que posibilitan la replicación entre instancias (*copia de ficheros, restauraciones y alertas*).

La ejecución del servicio *SQL Reporting Services* se encarga de publicar los informes creados por los gestores y hacerlos accesibles mediante un navegador para contenidos WEB.

6. DOCUMENTACIÓN. RECURSOS TÉCNICOS

Herramienta de Virtualización

ORACLE VM VirtualBox

<http://www.oracle.com/us/technologies/virtualization/virtualbox/overview/index.html>

Sistema Operativo

Windows Server 2008 R2 Standard

<https://www.microsoft.com/es-es/cloud-platform/windows-server>

Motor de Base de Datos

Microsoft SQL Server Enterprise

<https://www.microsoft.com/en-us/sql-server>

<https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2>

Servidor de Informes

SQL Server Reporting Services -

[https://msdn.microsoft.com/es-es/library/ms159106\(v=sql.120\).aspx](https://msdn.microsoft.com/es-es/library/ms159106(v=sql.120).aspx)

Herramientas de Administración

SQL Server Management Studio

<https://docs.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms>

SQL Server Configuration Manager

<https://docs.microsoft.com/es-es/sql/relational-databases/sql-server-configuration-manager>

Herramientas de Desarrollo

Lenguaje C#

<https://docs.microsoft.com/es-es/dotnet/csharp>

Transact-SQL

<https://docs.microsoft.com/es-es/sql/t-sql/language-reference>

Microsoft Visual Studio

<https://visualstudio.microsoft.com>

Microsoft SQL Server Express

<https://www.microsoft.com/es-es/sql-server/sql-server-editions-express>

SQL Server Business Intelligence

<https://www.microsoft.com/es-es/sql-server/sql-business-intelligence>

REFERENCIAS

- Humanismo - IA* - <http://isanidad.com/104694/humanismo-e-inteligencia-artificial>
- Plan Estratégico de Sistemas y Tecnologías* - <http://isanidad.com/32482/sanidad-impulsara-el-plan-estrategico-de-sistemas-y-tecnologias-de-la-informacion-para-mejorar-en-calidad-y-eficacia>
- Experiencia del paciente* - <http://isanidad.com/115432/puede-convertirse-en-datos-la-experiencia-del-paciente>
- Selene Hospitales* - <http://www.cerner.com/Soluciones/Sistemas-de-Informacion/Cerner-Selene/Selene-Hospitales>
<http://www.computing.es/infraestructuras/noticias/1105873001801/falla-sistema-selene-18-hospitales-de-madrid.1.html>
https://elpais.com/ccaa/2018/06/05/madrid/1528185386_352250.html
- Selene Mobility* - <https://www.cerner.com/Soluciones>
- Grupo Quirónsalud* - <https://www.quironsalud.es>
- Portal Univadis* - <https://www.univadis.es>
- Compañías tecnológicas mercado* - <http://isanidad.com/33900/las-10-companias-tecnologicas-que-están-revolucionando-el-mercado>
- ReWalk* - <http://rewalk.com>
- uMoove* - <http://www.umoove.me>
- Telesofia* - <https://www.telesofia.com>
- Surgical Theater* - <http://www.surgicaltheater.net>
- TotallyPregnant* - <https://play.google.com/store/apps/details?id=com.totallypregnant.android>
- Patients Know Best* - <https://www.patientsknowbest.com>
- iCouch* - <https://icouch.me>
- HelpAround* - <http://helparound.co>
- Doximityh* - <https://www.doximity.com>
- Fitocracy* - <https://www.fitocracy.com>
- robot Da Vinci* - https://www.intuitivesurgical.com/products/davinci_surgical_system
<https://campussanofi.es/2017/08/31/los-hospitales-espanoles-apuestan-por-el-robot-da-vinci>
<http://isanidad.com/115835/el-hospital-clinico-san-carlos-protagoniza-la-primera-intervencion-a-un-nino-con-el-robot-da-vinci>
- Senhance* - <https://transenterix.com>
<http://www.medicalexpo.es/prod/transenterix/product-84433-818957.html>
- Flex Robotic System* - <https://medrobotics.com/gateway/flex-robotic-system>
- Broca* - <http://www.proyecto-broca.es>
- FlexDex Surgical* - <https://flexdex.com>
- motor de Bases de Datos* - <https://docs.microsoft.com/es-es/sql/database-engine/sql-server-database-engine-overview>
- planes de ejecución* - [https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175913\(v=sql.105\)](https://docs.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/ms175913(v=sql.105))
- SQLAgent* - <https://docs.microsoft.com/es-es/sql/ssms/agent/sql-server-agent>
- SQLBrowser* - <https://docs.microsoft.com/es-es/sql/tools/configuration-manager/sql-server-browser-service>
- SQL Integration Services* - <https://docs.microsoft.com/es-es/sql/integration-services/sql-server-integration-services>
- tabla master* - <https://docs.microsoft.com/es-es/sql/relational-databases/databases/master-database>
- tabla model* - <https://docs.microsoft.com/es-es/sql/relational-databases/databases/model-database>
- tabla msdb* - <https://docs.microsoft.com/es-es/sql/relational-databases/databases/msdb-database>
- tabla tempdb* - <https://docs.microsoft.com/es-es/sql/relational-databases/databases/tempdb-database>
- trasvase de registros* - <https://docs.microsoft.com/es-es/sql/database-engine/log-shipping/about-log-shipping-sql-server>