Universidad Zaragoza

Trabajo Fin de Grado

Monitorización web en tiempo real de alertas en entornos hospitalarios

Real-time web monitoring for alerts in hospital environments

Autor

Leticia Sánchez Romero

Director

Carlos Aisa Redondo

Ponente

Francisco Javier Zarazaga Soria

Grado en Ingeniería Informática
Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura

Junio 2023

This page is intentionally blank

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

D./Dª. _____,

con nº de DNI _____ en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) _____, (Título del Trabajo)

_____

_____

_____

_____

_____

_____,

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, _____

Fdo: _____

This page is intentionally blank

AGRADECIMIENTOS

Agradezco a Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc,

y especialmente a los alumnos que hacen plantillas de LaTeX.

This page is intentionally blank

Monitorización web en tiempo real de alertas en entornos hospitalarios

RESUMEN

Ibernex es una compañía especializada en el diseño, desarrollo e integración de soluciones y servicios tecnológicos destinados al sector socio-sanitario. Realizan soluciones para automatizar y digitalizar la atención y experiencia de residencias u hospitales.

Actualmente la compañía tiene desarrollada una aplicación que se encarga de la gestión al completo de distintas funcionalidades dentro del sector comentado anteriormente. Esta aplicación se conecta con otras aplicaciones según las necesidades que tienen los distintos clientes de Ibernex.

Las soluciones que se han desarrollado hasta el momento son soluciones orientadas a aplicaciones de escritorio. Sin embargo, la empresa considera neceario que una buena opción sea utilizar alguna de sus funcionalidades en una aplicación web de tal manera que por ejemplo se pueda tener dicha aplicación ejectuando en monitores en una residencia u hospital.

Por esta razón, en este proyecto se desarrolla, cómo método de prueba de cara a que la empresa pueda reutilizar la implementación que consiedere necesaria, la funcionalidad de monitorizar alertas en tiempo real.

This page is intentionally blank

Índice

This page is intentionally blank

Capítulo 1

Introducción

En este capítulo se presenta el contexto del trabajo, así como la motivación del problema concreto que se aborda. Se explica los objetivos y sus limitaciones, además de las herramientas de trabajo utilizadas. Por último, se explica el esquema general de la memoria del proyecto.

1.1. Contexto de trabajo

Ibernex es una empresa zaragonana especializada en el diseño, desarrollo e integración de soluciones y servicios tecnológicos destinados al sector sanitario y socio-sanitario. Estas soluciones se centran en automatizar y digitalizar la atención y experiencia de residencias u hospitales.

Además, Ibernex cuenta con un largo recorrido de la mano del grupo Pikolín y recientemente se han incorporado dos nuevos accionistas mayoritarios con la intención de potenciar su presencia en el mercado internacional. Actualmente, la empresa es líder en el sector en Iberia (España y Portugal) donde su principal core de negocio son los centros sociosanitaios, pero también alcanzó el 30 % de facturación en el mercado internacional durante el 2022, más concretamente en Latinoamérica potenciando la digitalización de los hospitales y ciudades de la salud.

Los clientes de la empresa, como se ha comentado anteriormente, son residencias u hospitales que quieren digitalizar el proceso de cuidado y atención de pacientes, además de otros procesos que puedan tener según sus necesidades.

La compañía realiza todo el proceso de construcción del producto.

– Por un lado, la fabricación del hardware que interactúa con el software. En

esta parte se encuentran los terminales que están en las habitaciones de los pacientes. Estos terminales son pantallas en las que se pueden realizar distintas acciones como por ejemplo disparar y codificar alarmas, registrar presencias de trabajadores, registrar tareas u otras actividades según el sector en el que se encuentren.

- Por otro lado, la implementación de Helpnex que es el software de gestión diseñado y desarrollado por Ibernex. Helpnex está diseñado para poder adaptarse a cada cliente ya sea por sus necesidades concretas, capacidades, como perfiles de trabajadores, entre otras. Además, las soluciones que ofrecen también se pueden integrar con otros sistemas de gestión como Resiplus (muy presente en residencias que quieren transformarse y digitalizarse, pero mantener su programa de gestión). La empresa se encuentra en continua innovación para seguir ofreciendo un sistema actualizado que ayude a optimizar la gestión de residencias y hospitales y así mejorar no solo la calidad de vida de los residentes y pacientes, sino también la labor de los trabajadores con el fin de que puedan invertir más tiempo en lo realmente importante, cuidar de las personas.

Los clientes que tiene la compañía actualmente trabajan mayormente con aplicaciones de escritorio. Pero Ibernex considera que sería una buena opción que en residencias u hospitales se pueda monitorizar alarmas mediante la visualización de estas en pantallas.

Para esto han considerado que una buena solución sería tener una aplicación web en la que se pueda visualizar en tiempo real las alertas de la institución.

En este contexto la parte del trabajo a realizar (que se explica con más detalle en la siguiente sección) es realizar modificaciones en la implementación de Helpnex y realizar el frontal, que será el que reciba las alertas en tiempo real, separado de la aplicación actual.

1.2. Objetivos y limitaciones

El objetivo del proyecto es desarrollar un sistema de información en forma de aplicación web que muestra la monitorización en tiempo real de alertas.

Estas alertas pueden ser de dos tipos:

- **Presencias**: son alertas que indican la presencia de un trabajador en una habitación. Estas presencias se muestran en los terminales y pueden ser de dos tipos:

    1. Identificadas: un trabajador pasa su tarjeta de identicación por el lector del terminal.
    2. No identificadas: mediante la pulsación de un botón en la pantalla táctil del terminal.

- **Alarmas**: son alertas generadas por los pacientes cuando hay una situación de emergencia desde los terminales que se encuentran en las habitaciones. Pueden ser codificadas por el primer trabajador que se ha identificado en ese terminal. El ciclo de vida de estas alarmas pasa por tres estados:

    1. Disparada: la alarma ha sido generada y queda pendiente de ser aceptada.
    2. Aceptada: un trabajador es consciente de la existencia de esa alarma mediante su identificación en el terminal y queda pendiente de ser atendida.
    3. Atendida: la alarma queda pendiente de ser codificada por el trabajador que la ha aceptado.

Este sistema web servirá para pequeños y grandes escenarios. Es decir, será útil para una residencia pequeña en la que el número de clientes, habitaciones y por tanto terminales sea reducido. O el caso contrario en el que se quiera monitorizar las alertas de un hospital de gran tamaño con un alto número de clientes y terminales.

El trabajo a realizar para llevar a cabo la construcción de la aplicación web consta de dos partes. En primer lugar, implementar un frontend completo desde cero separado de la aplicación actual de Ibernex, que recibirá las alertas en tiempo real y permitirá visualizar cierta información de estas alertas. En segundo lugar, el desarrollo de un backend que se realiza integrando código en Helpnex aprovechando algunas de las funcionalidades existentes y desarrollando las nuevas funcionalidades necesarias.

La limitación a la hora de realizar el proyecto es que al tener ya una aplicación desarrollada con ciertas tecnologías y modelos de datos, se debe realizar el análisis y diseño del sistema en base a esas condiciones y teniendo en cuenta las necesidades concretas de la empresa, que se podrán ver posteriormente en la sección de requisitos del sistema.

1.3. Herramientas de trabajo

Para desarrollar el proyecto descrito anteriormente es necesario trabajar con las herramientas que ya utiliza la empresa e introducir nuevas.

El backend se desarrolla con el lenguaje de programación *C#* utilizando *.NET Framework 4.8* ya que la aplicación de la empresa está implementada de esta forma. Además, como entorno de desarrollo se utilza *Microsof Visual Studio*.

El frontend se desarrolla con *React* y utilizando los lenguajes de programación *JavaScript*, *HTML* y *CSS* y teniendo *Visual Studio Code* como editor de código fuente.

Como software de control de versiones se utiliza *Git*. Para alojar el código del frontend se utiliza *GitHub*. Para alojar el código del backend se utiliza un repositorio de la empresa. Y para clonar el repositorio donde se aloja la aplicación Helpnex se utiliza el cliente *TotoiseGit*.

Por otro lado, este proyecto se está desarrollando de tal forma que la estudiante trabaja para la empresa de forma remota, estando la sede de la empresa en Zaragoza, España y la estudiante en Kaunas, Lituania. Por esta razón se tienen también en cuenta que las herramientas de trabajo utilizadas para mantener la comunicación con la empresa son *Gmail*, *Google Chat* y te *Google Meet* para realizar las reuniones con el tutor en la empresa.

1.4. Esquema general de la memoria del proyecto

La estructura seguida en este documento es la siguiente:

Previamente a esta sección se encuentran los agradecimentos y un resumen completo del proyecto realizado. A continuación el índice que resume esta sección.

En este mismo capítulo se encuentran tres secciones distintas, además de esta misma. La primera, es el contexto del trabajo donde se explica qué empresa es Ibernex, cómo trabaja y quiénes son sus clientes, además de exponer cómo son sus soluciones y cómo encaja este proyecto en la empresa. La segunda, son los objetivos y limitaciones con los cuales se expone detalladamente qué es lo que se va a realizar y qué es lo que se va a conseguir al final y con qué limitaciones. Y por último, se exponen las

herramientas de trabajo utilizadas durante el desarrollo del proyecto.

En el segundo capítulo se detalla el análisis y diseño de la aplicación web. En la primera sección se encuentran los requisitos funcionales del sistema acordados con la empresa. En la segunda, se explica la arquitectura software del sistema con ayuda de un diagrama. En la tercera, se expone la información relevante a los modelos y base de datos que utiliza la empresa utilizados en el contexto de este proyecto. Y por último se puede observar la interfaz de usuario de la aplicación web.

En el tercer capítulo se pueden encontrar dos secciones en las que se proporciona de manera más detallada diversos aspectos de la implementación realizada para el frontend y backend respectivamente.

En el cuarto capítulo se explica cómo se ha gestionado el proyecto, comentando la planificación inicial y final junto con un análisis explicativo de los posibles cambios.

En el quinto capítulo se exponen las conclusiones del proyecto y se explica qué conocimientos se han adquirido, técnicamente y personalmente. Además, se proporciona información acerca de una continuación del trabajo en el futuro.

Por último se puede encontrar detallada la bibliografía, la lista de figuras, la lista de tablas y los distintos anexos con explicaciones más detalladas de diferentes aspectos, que se han citado a lo largo del documento.

Capítulo 2

Análisis y diseño del sistema

En este capítulo se expone el análisis de requisitos funcionales, la arquitectura software, la base de datos y la interfaz del usuario.

2.1. Requisitos del sistema

Seguidamente (véase tabla 2.1) se presentan, en modo de tabla, los principales requisitos del sistema que se han acordado con la empresa.

ID	**Requisito**
RF-1	Un usuario iniciará sesión en el sistema utilizando usuario, contraseña y puesto.
RF-2	Un usuario podrá cerrar su sesión.
RF-3	El sistema permitirá al usuario navegar en la aplicación mediante un menú lateral.
RF-4	El sistema mostrará al usuario un listado de plantas del edificio.
RF-5	El sistema mostrará al usuario el número total de alarmas y presencias activas en cada planta.
RF-6	El sistema permitirá al usuario elegir una planta y ver la información de alertas correspondiente a esa planta.
RF-7	El sistema mostrará al usuario un carrusel con las alertas de la planta seleccionada.
RF-8	El sistema mostrará al usuario la información del paciente de las alarmas activas y el lugar y momento en el que se han disparado.
RF-9	El sistema mostrará al usuario la información del trabajador de las presencias identificadas activas y el lugar y momento en el que se han producido.
RF-10	El sistema mostrará al usuario el plano de la planta seleccionada.
RF-11	El sistema destacará al usuario en el plano las habitaciones que tengan alguna alerta en activo.
RF-12	El sistema mostrará al usuario el listado de alarmas pendientes, sin filtrado por planta, indicando el tipo de alarma.
RF-13	El sistema mostrará al usuario el número total de alarmas y presencias.

Tabla 2.1: Requisitos funcionales del sistema

2.2. Arquitectura software del sistema

Para ver las relaciones entre el software y su entorno (dónde se despliega y ejecuta) se utiliza una vista de distribución estilo despliegue. Se puede ver en la *Figura 2.1*.

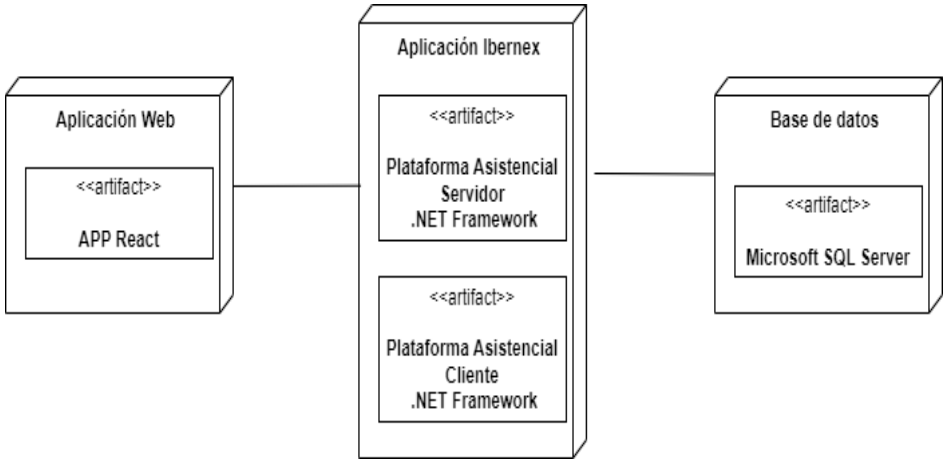

Figura 2.1: Diagrama de despliegue

La arquitectura completa respecto a lo que ha estado relacionado con el proyecto es la que se ve en la figura. No obstante lo que se ha implementado ha sido completamente la aplicación web y por otro lado se ha modificado y añadido funcionalidad a la Plataforma Asistencial Servidor de la aplicación de Ibernex.

El ámbito del despliegue del sistema será en una red de área local (LAN) ya que serviría para una residencia u hospital en el que se instalaría el sistema configurando la base de datos correspondiente a la información de los pacientes del centro.

TODO: Terminar de explicar la arquitectura final del sistema consultando con Carlos para hacerlo correctamente respecto a la parte de Ibernex

La arquitectura aquí expuesta es la elegida entre las distintas opciones barajadas que se pueden ver con detalle en el Anexo A.

2.3. Base de datos

TODO: explicar qué tablas de la base de datos utilizo de su sistema y adjuntaré diagramas creados con la aplicación Microsoft SQL Server Management.

2.4. Interfaz de usuario

TODO: Poner la interfaz final del usuario cuando esté terminada

Capítulo 3

Implementación

3.1. Implementación del frontend

EL frontend de la aplicación web se realiza con React utilizando los lenguajes de programación *javascript*, *HTML*, y *CSS*.

Para la comunicación del frontend con el backend mediante WebSockets se utiliza el paquete *npm reconnecting-websocket* [1]. Se decide utilizar este frente a otros ya que reconecta automáticamente si la conexión se cierra por alguna razón y es compatible con *WebSocket Browser API* [2]

TODO: explicar organización de ficheros en frontend indicando que cierta división de los módulos hace que la organización del código queda bien repartida teniendo las funcionalidades para poder ver a simple vista implementado ???

TODO: Decisiones de implementación que tengo anotadas y problemas u opciones surgidas respecto a la implementación del frontend

3.2. Implementación del backend

El backend de la aplicación se realiza con *.NET Framework 4.8* ya que como se ha explicado en secciones anteriores la implementación es añadir funcionalidad a la aplicación existente.

Para la comunicación con el frontend mediante WebSockets se utiliza la librería *websocket-sharp* [3] en su versión *1.0.3.0*. Se decide utilizar esta librería ya que en otra de las funcionalidades de la aplicación existente se utiliza aunque en una versión anterior, y se cree conveniente utilizar algo similar de cara a que internamente en la

empresa puedan entender mejor la implementación o reutilizar el código. La instalación de esto se realiza mediante el propio Visual Studio, que se ha comentado la sección de herramientras de trabajo que es el entorno de desarrollo utilizado para implementar el backend, utilizando la opción de administrar paquetes NuGet e instalando el nombrado.

En un primer lugar se propone realizar la comunicación del backend con el frontend utilizando *SignalR* [4] pero finalmente se descarta por incompatibilidad con la aplicación actual y simplicidad en la arquitectura software. Se puede ver más información referente a esta decisión en el Anexo B.

TODO: Decisiones de implementación respecto a ficheros y funciones que tengo anotadas y problemas u opciones surgidas respecto a la implementación del backend

Capítulo 4

Gestión del proyecto

TODO: Esta sección no la subdividiría. Pon una planificación inicial, la final, y comenta el porqué de las variaciones

Capítulo 5

Conclusiones

5.1. Conclusiones

TODO: Explicar lo que has hecho. Básicamente decir que lo que ibas a hacer lo has hecho

5.2. Conocimientos adquiridos

5.2.1. Conocimientos técnicos

TODO: nuevas tecnologías, nuevos entornos de trabajo

5.2.2. Conocimientos personales

TODO: organización, primer trabajo en una empresa del relacionada con los estudios, y trabajo en remoto y lo que conlleva, dificultades.

5.3. Trabajo futuro

TODO: explicar que la aplicación desarrollada puede ser utilizada por Ibernex para reutilizar código o como prueba/base por si quieren sacar a producción trabajar con sistemas web finalmente ya que me explicaron que esto era como una pequeña prueba que querían hacer

Capítulo 6

Bibliografía

[1] Librería reconnecting-websocket. URL: `https://www.npmjs.com/package/reconnecting-websocket` (Fecha de último acceso: 15/04/2023).

[2] WebSocket Browser API. URL: `https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API` (Fecha de último acceso: 15/04/2023).

[3] Librería websocket-sharp. URL: `http://sta.github.io/websocket-sharp/` (Fecha de último acceso: 28/04/2023).

[4] SignalR. URL: `https://dotnet.microsoft.com/en-us/apps/aspnet/signalr` (Fecha de último acceso: 13/04/2023).

[5] Librería microsoft/signalr. URL: `https://www.npmjs.com/package/@microsoft/signalr` (Fecha de último acceso: 10/04/2023).

[6] Librería react-img-mapper. URL: `https://www.npmjs.com/package/react-img-mapper` (Fecha de último acceso: 1/03/2023).

Lista de Figuras

Lista de Tablas

Anexos

Anexos A

Alternativas arquitecturas

Anexos B

Decisión descarte SignalR

SignalR [4] es una biblioteca de código abierto que simplifica la adición de funcionalidad web en tiempo real a las aplicaciones.

En un principio SignalR es la opción a utilizar en la implementación propuesta por la empresa para establecer la comunicación entre el backend y el frontend.

La decisión de descartar SignalR se toma en el momento en el que se observa que no es compatible con .NET Framework 4.8 (y sí lo es con ASP.NET Core) y durante la evaluación de las distintas opciones de arquitecturas software para el sistema, explicadas con detalle en el Anexo A.

La razón de esta decisión es que, para utilizar SignalR, la arquitectura elegida debería de haber sido la que contiene un middleware entre el frontend y backend, de tal forma que este middleware estuviese implementado con ASP.NET Core para que fuese compatible. Con esta arquitectura la comunicación entre middleware y frontend podría haber sido totalmente compatible utilizando en React la librería *microsoft/signalr* [5]. Pero para comunicar con el backend se debería de haber utilizado en la aplicación Helpnex otra librería para websockets como *websocket-sharp* [3] que es la finalmente escogida a utilizar teniendo en cuenta la arquitectura final elegida.

Por lo que se toma la decisión teniendo en cuenta utilizar o no SignalR y la elección de una arquitectura más compleja o más simple y que sea la más rentable en el momento para la empresa.

Anexos C

Decisión descarte JWT