



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



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).

D./D^a. _____,

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, consectetur 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 necesario 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 ejecutando 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 considere necesaria, la funcionalidad de monitorizar alertas en tiempo real.

This page is intentionally blank

Índice

1. Introducción	1
1.1. Contexto de trabajo	1
1.2. Objetivos y limitaciones	2
1.3. Herramientas de trabajo	3
1.4. Esquema general de la memoria del proyecto	3
2. Análisis y diseño del sistema	5
2.1. Requisitos del sistema	5
2.2. Arquitectura software del sistema	5
2.3. Base de datos	6
2.4. Interfaz de usuario	6
3. Implementación	7
3.1. Implementación del frontend	7
3.2. Implementación del backend	7
4. Gestión del proyecto	9
4.1. Planificación del proyecto	9
4.2. Dedicación y dimensión del proyecto	9
5. Conclusiones	11
5.1. Conclusiones	11
5.2. Conocimientos adquiridos	11
5.2.1. Conocimientos técnicos	11
5.2.2. Conocimientos personales	11
5.3. Trabajo futuro	11
6. Bibliografía	13
Lista de Figuras	15

Lista de Tablas	17
Anexos	18
A. Alternativas arquitecturas	21
B. Decisión descarte SignalR	23
C. Decisión descarte JWT	25

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 sociosanitarios, 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 entre los que

se tienen los terminales que se encuentran 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.

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.

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 no identificadas (se pulsa un botón en la pantalla táctil del terminal) o identificadas (el trabajador pasa su tarjeta de identificación por el lector del terminal).
- **Alarmas:** son alertas que se disparan una única vez. Estas son generadas desde los terminales que se encuentran en las habitaciones y pueden ser codificadas por el primer trabajador que se ha identificado en ese terminal.

Este sistema web servirá para pequeños y grandes escenarios. Es decir será útil para por ejemplo 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.

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, se debe realizar el análisis de tecnologías y análisis y diseño del sistema en base a esas condiciones y teniendo en cuenta las necesidades que se podrán ver posteriormente en la sección de requisitos del sistema.

1.3. Herramientas de trabajo

TODO: ¿Explicar frameworks, tecnologías, entornos de desarrollo, editores de código fuente, git, tortoisegit, github, google met y chat/gmail para comunicación con empresa en remoto?

1.4. Esquema general de la memoria del proyecto

TODO: redactar una vez clara la estructura final de la memoria

Capítulo 2

Análisis y diseño del sistema

En este capítulo se expone el análisis de requisitos funcionales del sistema acordado con la empresa. A continuación se presenta la arquitectura software final que tiene el sistema y se explica lo relevante respecto a la base de datos que se utiliza para la aplicación. Por último se muestra la interfaz del usuario final.

2.1. Requisitos del sistema

2.1

ID	Requisito
RF-1	Un usuario deberá iniciar sesión en el sistema utilizando usuario, contraseña y puesto
RF-2	Un usuario podrá cerrar su sesión
RF-3	El sistema mostrará al usuario un carrusel con el listado de alertas y su información necesaria de la planta seleccionada
RF-4	El sistema mostrará al usuario el plano de la planta seleccionada
RF-5	El sistema mostrará al usuario el listado de plantas y permitirá seleccionar una de ellas
RF-6	El sistema mostrará al usuario el listado de alertas pendientes sin filtrado por planta con la información básica
RF-7	El sistema mostrará al usuario el número total de alarmas y presencias
RF-8	El sistema permitirá al usuario navegar en la aplicación mediante un menú lateral

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*.

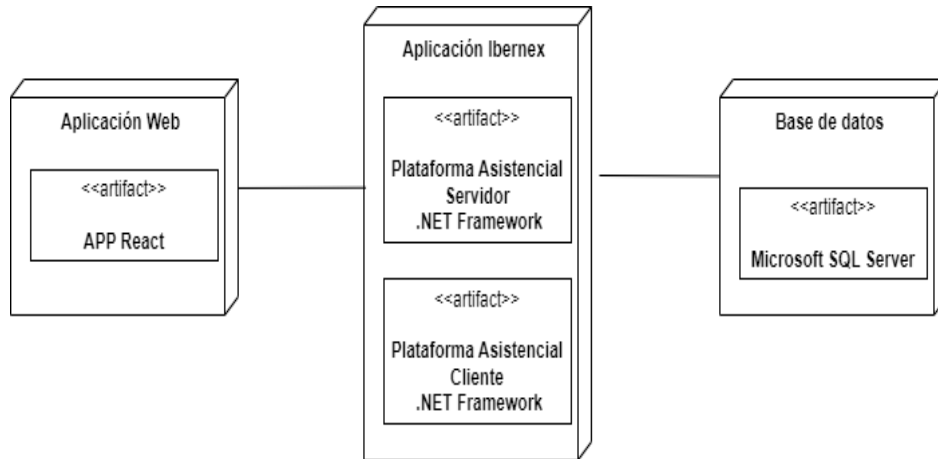


Figura 2.1: Diagrama de despliegue

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

TODO: Hacer referencia al Anexo donde se expliquen las alternativas que se plantearon

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 herramientas 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

4.1. Planificación del proyecto

TODO: Explicar cómo se planificó el proyecto desde el principio como lo puesto en la propuesta

4.2. Dedicación y dimensión del proyecto

TODO: tabla con tiempo final invertido aproximado a ciertas tareas del proyecto.

TODO: dimensión del proyecto qué y cómo ??

Capítulo 5

Conclusiones

5.1. Conclusiones

TODO: ¿Qué conclusiones principales deberían aparecer?

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] Package npm reconnecting-websocket. URL: <https://www.npmjs.com/package/reconnecting-websocket>.
- [2] Websocket browser api. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API.
- [3] websocket-sharp. URL: <http://sta.github.io/websocket-sharp/>.
- [4] Signalr. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/signalr>.

Lista de Figuras

2.1. Diagrama de despliegue	6
---------------------------------------	---

Lista de Tablas

2.1. Requisitos funcionales del sistema	5
---	---

Anexos

Anexos A

Alternativas arquitecturas

Anexos B

Decisión descarte SignalR

Anexos C

Decisión descarte JWT