



UNIVERSIDAD  
DE BURGOS



fundación  
miradas

**Entrega preliminar Trabajo Final de Grado.  
November 14, 2022**

**Grado en Ingeniería Informática**

**Curso 2022/2023**

**Alumno:** *Pablo Ahíta del Barrio*

**Tutor:** *Pedro Luis Sánchez Ortega*

## Contents

<b>1</b>	<b>Trabajos de final de grado de referencia</b>	<b>3</b>
<b>2</b>	<b>Diferencias entre <i>Azure</i> y las aplicaciones de bases de datos locales</b>	<b>4</b>
<b>3</b>	<b>Edición del informe</b>	<b>7</b>
3.1	Formato del informe . . . . .	7
3.2	Comprobación de la ortografía del informe . . . . .	7

# 1 Trabajos de final de grado de referencia

Los trabajos final de grado de referencia que se han utilizado son los siguientes:

- **Prueba de concepto *Azure Monitor*:** Es un Trabajo Final de Grado que trata sobre aplicaciones APM (*Application Performance Monitor*), que son herramientas que diagnostican el rendimiento de las aplicaciones, con la finalidad de encontrar fallos en el programa, cuellos de botella, e incluso poder solucionarlos, evitando degradaciones. La funcionalidad utilizada en este TFG es *Azure Monitor*, el cual es la herramienta de este estilo proporcionada por Microsoft, cuyo objetivo es averiguar si el proyecto puede ser utilizado por una organización que tiene dos entornos completamente diferentes: un entorno de nube *Azure* donde se encuentra su sitio web y que podrá albergar en el futuro alguno de los servicios que ofrece, y otro entorno en el sitio donde residen el resto de los servicios.
- **Desarrollo de una plataforma de tratamiento y streaming de vídeo para difusión de la cultura utilizando instancias de *Azure*:** Es un Trabajo Final de Grado que trata sobre una herramienta que es utilizada por artistas de cine independientes para darse a conocer a través de publicidad en diferentes dispositivos, ya sea a través de aplicaciones móviles o a través de navegadores. *Azure* es utilizado como soporte para la base de datos y la posterior gestión de transacciones, lo que se busca para el proyecto de la Fundación Miradas.
- **Servicios en la nube con *Microsoft Azure* : desarrollo y operación de una aplicación Android con DevOps:** Este proyecto trata sobre el desarrollo de una aplicación de Android con almacenamiento en la nube, utilizándose para el almacenamiento, visualización y procesamiento de fotografías. Dicho proyecto utiliza *Azure* para desplegar la aplicación en la nube, con ayuda de DevOps para poder alargar el ciclo de vida de la misma. Puede ser de utilidad también para poder desarrollar la aplicación tanto a nivel de frontend como a nivel de backend.
- **JIZT. Generación de resúmenes abstractivos en la nube mediante Inteligencia Artificial:** JIZT es un servicio de generación automática de resúmenes basado en la corriente *Cloud Native*, que se basa en los principios de los sistemas escalables, elasticidad y agilidad. Dicho servicio es sustentado por una arquitectura de micro-servicios dirigido por eventos, garantizando la alta disponibilidad del servicio, aparte de los tres principios mencionados con anterioridad. Dicha aplicación es multiplataforma, por lo que consume la API REST del servicio en la nube, donde cualquier usuario dispone de los resúmenes que desee.
- **UBUNurse:** Este proyecto consiste en una aplicación multi-dispositivo el cual almacena en la nube registros sobre la evaluación de la atención domiciliar por parte de dicho personal hacia un determinado paciente.

El procedimiento a seguir consiste en elegir un paciente en la lista de la cual dispone el enfermero, posteriormente se elige el test a realizar y por último se obtienen los resultados de la realización de dicho test. Este software pretende automatizar dicho proceso para mejorar la eficiencia del personal sanitario y también para mejorar el proceso de evaluación de cada paciente.

- **Machine learning mediante *Microsoft Azure*: una aplicación sobre real-state:** En este proyecto se trata más a fondo las herramientas de las cuales dispone *Azure*, las cuales son utilizadas para la creación de elementos de machine learning, los cuales explican el funcionamiento de *Azure* con fines estadísticos. Se utiliza en dicho proyecto *Azure Machine Learning* con los datos empíricos de una inmobiliaria estadounidense para la predicción y posterior clasificación del valor de las viviendas, comparando posteriormente con otros modelos de clasificación y de regresión. A pesar de no ser un proyecto de bases de datos, se le da mucho hincapié a los gráficos y a la muestra de resultados, lo que puede servir de utilidad para la parte final de muestra de resultados.
- ***Trello*. How to embrace remote work:** *Trello* es la herramienta mediante la cual se ha realizado el seguimiento de todos los sprint entre el tutor y el alumno. Dicha herramienta permite realizar el envío de la documentación requerida para el desarrollo de este proyecto de una manera bastante eficiente.
- **Desarrollo de un Bot en la plataforma *Azure* para ayudar en el aprendizaje del lenguaje de programación C:** Este proyecto consiste en el desarrollo de un bot mediante *Azure* para ayudar al alumnado de 1º curso del Grado en Ingeniería Electrónica y Automática de la Universidad Politécnica de Cartagena al aprendizaje del lenguaje de programación C. Aparte de eso trata también sobre las herramientas de Inteligencia Artificial de *Azure*, como *LUIS (Language Understanding)*, aparte de que el software de dicho bot tiene soporte de incorporación de diferentes idiomas, lo que es un factor de gran importancia para el desarrollo de la app. Se ha escogido este trabajo de final de grado puesto que se trata de un proyecto bastante completo en cuanto a contenido a sacar de él y en cuanto a estructuración del contenido del mismo.

## 2 Diferencias entre *Azure* y las aplicaciones de bases de datos locales

Las diferencias que tiene *Azure* con respecto a las aplicaciones de gestión de bases de datos a nivel local, como es el caso de *Microsoft Access* o *OpenOffice Database*, son las siguientes:

- En las aplicaciones de bases de datos locales es preciso utilizar un ordenador con un fichero *.accdb* en *Microsoft Access* o un fichero *.odb* en *OpenOffice Database*, el cual aloje todos los registros de los indicadores y sus respectivas incidencias, cuya difusión depende de la cantidad de personas que tengan ese fichero. En cambio, con Microsoft Azure, la difusión es más sencilla puesto que no se necesita un fichero en cada uno de los dispositivos, ya que al alojarse los registros de los datos y de las correspondientes incidencias en la nube, permite de mejor manera la implementación en diferentes dispositivos, aparte de que solo el administrador tiene acceso a la base de datos.
- En una aplicación que se apoya en *Azure* es necesaria una conexión a internet para poder cargar los datos correctamente al servidor, al igual que para realizar todas las demás operaciones en las que esté involucrada la base de datos, por lo tanto si estuviese caído el servidor donde se ha implementado Azure, no se tendría el comportamiento esperado en la aplicación. En cambio, en el caso de la base de datos en *Microsoft Access* o en *OpenOffice Database*, como el comportamiento de la base de datos depende de que esté alojado su correspondiente fichero *.accdb* o *.odb* en una cantidad determinada de dispositivos, no se tendría esta problemática, salvo en el hipotético caso en que ninguno de los dispositivos que cuenten con el fichero de la base de datos se encuentre operativo.
- En el caso de implementar una base de datos en *Azure*, se puede implementar de mejor manera en una aplicación como la de la Fundación Miradas, puesto que se espera que dicha aplicación reciba y envíe transacciones del cliente, que es el encargado de la Fundación Miradas que evalúa el correcto cumplimiento de los indicadores y de sus respectivas incidencias o el representante de la asociación de ayuda a la discapacidad que está siendo evaluada que comprueba los diferentes resultados realizados de las diferentes test, hacia el servidor que se encarga de alojar cada uno de los datos de las asociaciones evaluadas y de cada uno de los mencionados diagnósticos, proporcionando inmediatez y automatización en el proceso de muestra de resultados para todos los interesados mencionados con anterioridad. En cambio, esta tarea es más tediosa en la implementación original de la aplicación en Access, puesto que los resultados obtenidos sólo se mostrarían de forma inmediata en el equipo que corre dicha aplicación, por lo que sería necesario transportar el informe resultante manualmente, ya sea mediante un dispositivo de almacenamiento físico externo o por correo electrónico.
- *Azure*, al estar enfocada a alojar grandes cantidades de datos por parte de empresas, no dispone de una versión gratuita de forma permanente, sino que sólo unos cuantos servicios son gratuitos, mientras que hay ciertos que también son gratuitos, pero únicamente durante doce meses, mientras que otros se obtienen mediante las diferentes suscripciones de las que dispone *Azure*. En cambio, sucede lo contrario con las aplicaciones de bases de datos a nivel local, como *Microsoft Access*, cuya versión gra-

tuita se puede encontrar en OneDrive como aplicación web, como con *OpenOffice Database*, el cual forma parte del paquete de ofimática de libre distribución *OpenOffice*. En el caso del paquete de *Microsoft Office*, donde se incluye *Microsoft Access*, también es un software de pago el cual dispone de licencias desde un mes hasta los doce.

- En *Azure* se garantiza la seguridad de los datos proporcionados gracias al cifrado de los datos en reposo, el cual se produce en tres niveles:
  - **A nivel de almacenamiento en el servidor** el servicio *Azure Storage*, el cual se encarga del almacenamiento de los datos, realiza un encriptado del servicio *SSE*, concretamente los datos se cifran y descifran de forma transparente mediante el cifrado *AES* de 256 bits, uno de los cifrados de bloques más sólidos que hay disponibles, y son compatibles con *FIPS 140-2*.
  - **A nivel de cliente**, la correspondiente biblioteca de *Azure Blob Storage* usa *AES* para cifrar los datos del usuario. Hay dos versiones de cifrados de cliente disponibles en la biblioteca de cliente:
    - \* La versión 2 utiliza el modo *Galois/Contador (GCM)* con *AES*.
    - \* La versión 1 utiliza el modo *Cipher Block Chain (CBC)* con *AES*.
  - **A nivel de disco duro del sistema operativo**, permite cifrar los discos del sistema operativo y los discos de datos usados por una máquina virtual *IaaS*. Dicho proceso se encarga de realizarlo *Azure Disk Encryption*, el cual ayuda a custodiar y proteger los datos con el objetivo de cumplir los compromisos de cumplimiento y seguridad. Usa la característica *DM-Crypt* de Linux para proporcionar cifrado de volumen tanto a los discos de datos como a los del sistema operativo de máquinas virtuales (VM) de *Azure* y se integra con *Azure Key Vault* para ayudarle a controlar y administrar las claves y los secretos del cifrado de disco.

En cambio, con las bases de datos locales, la seguridad depende de quien disponga el fichero correspondiente y del correcto uso que tenga del mismo.

## 3 Edición del informe

En cuanto a la edición del informe se han tenido en cuenta diferentes aspectos para poder escoger el formato que debe tener informe, como por ejemplo la personalización del documento, la preocupación por los márgenes de cada una de las páginas, el control de la ortografía, la introducción de fórmulas matemáticas, de tablas, de imágenes y de gráficos y el acabado profesional de este informe. Por tanto, dichos aspectos son los expuestos en los sub-apartados de este punto.

### 3.1 Formato del informe

El editor de documentos que se ha decidido utilizar para realizar dicho informe es *LaTeX* por encima de editores de documentos convencionales como *OpenOffice Write* o *Microsoft Word*. El motivo por el cual se ha decidido utilizar *LaTeX* es por su gran cantidad de funcionalidades y por la libertad con la que se puede diseñar el documento en todos los aspectos mencionados con anterioridad. Para poder crear los documentos pdf en (*LaTeX*) es necesaria la instalación de un compilador. Existen diferentes opciones para poder realizar la compilación del documento, como es el caso de *Overleaf*, el cual es un editor en línea de código en *LaTeX*, de uso compartido y con compilador integrado. En el caso de este informe, se ha utilizado el compilador *MikTeX*, que es un compilador de libre distribución el cual permite instalar todos los paquetes que el usuario necesite para su documento. Como editor de *LaTeX* se ha utilizado *Microsoft Visual Studio Code*. Se ha escogido ese editor por encima del editor predefinido de *MikTeX* (de nombre *TexWorks*), puesto que es un editor bastante más completo, ya que se puede editar el código fuente de la aplicación además del propio documento, puesto que soporta la gran mayoría de los lenguajes de programación más utilizados de la actualidad, aparte de dar soporte al propio *LaTeX* y a su compilador *MikTeX*. Para cambiar el tamaño de la fuente de forma automática, se utilizan los comandos de cambio de tamaño de texto, como `\normalsize` y `\footnotesize`.

### 3.2 Comprobación de la ortografía del informe

Gracias al editor *Microsoft Visual Studio Code* es posible agregar complementos que faciliten el soporte a todo tipo de lenguajes y que faciliten también un buen uso para todo tipo de usuarios. Uno de estos complementos es el que se ha utilizado para facilitar la comprobación de la ortografía de todo tipo de lenguajes, entre ellos *LaTeX*, el cual recibe el nombre de *Code Spell Checker*, para el idioma español, el cual se instala de una manera muy sencilla en *Visual Studio Code*, presionando el botón de *Instalar* dentro de la página web del *Marketplace*, el cual está disponible también para su búsqueda dentro del propio editor. Posteriormente, para activarlo se puede hacer de diferentes maneras:

1. El primer método consiste en presionar el botón *F1* y posteriormente escribir el comando *Enable Spanish Spell Checker Dictionary* y se presiona

el botón *Enter* para que empiece a trabajar.

2. El segundo método consiste en ir al menú de la barra superior de tareas *Ver* y posteriormente al sub-menú *Paleta de comandos*. Posteriormente se escribe el comando *Enable Spanish Spell Checker Dictionary* y se presiona el botón *Enter* para que empiece a trabajar.

La versión original de este complemento realiza las comprobaciones ortográficas para el idioma [inglés](#), el cual se instala en *Visual Studio Code* de la misma manera, presionando el botón de *Instalar* dentro de la página web del *Marketplace*. Al tener instalada la versión en español, que es una extensión de la versión original, no precisa instalarlo después de la versión en español.