



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**RiskReal: Desarrollo de un
portal web para comprobar
las soft skills**



Presentado por Marcos Ubierna Fernández
en Universidad de Burgos — 9 de julio de 2024
Tutor: Raúl Marticorena Sánchez



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D.Raúl Marticorena Sánchez, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Marcos Ubierna Fernández, con DNI 71313709R, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado RiskReal.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 9 de julio de 2024

Vº. Bº. del Tutor:

D. Raúl Marticorena Sánchez

Resumen

A la hora de determinar las cualidades que tienen las personas hay diferentes maneras, pero una de las más comunes es la utilización de las habilidades blandas “Soft Skills” para tener una idea del comportamiento sobre diferentes escenarios.

De esta manera se plantea la utilización de tests para poder evaluar las Soft Skills, así surge **RiskReal** que es un portal web desde la cual se pueden realizar dos tipos de **tests** diferentes para poder determinar las Soft Skills, uno se basa en escenarios y el otro en un cuestionario más general.

Lo que se plantea es tener un sistema en el cual, se pueden realizar varios test al disponer de un archivo que siga una estructura dada. Para poder resolver esto se ha tenido que hacer una extracción de que elementos utiliza el test principal, de esta manera podemos trabajar con los datos adecuados y darles forma para obtener un nuevo test que cumpla los valores iniciales.

Descriptores

Habilidades blandas, evaluación, página web, Python, Flask, Json.

Abstract

When it comes to determining the qualities that people have, there are different ways, but one of the most common is the use of “Soft Skills” to get an idea of their behavior on different scenarios.

In this way, the use of tests is proposed to evaluate the Soft Skills, thus **RiskReal** emerges, which is a web portal from which two different types of **tests** can be carried out to determine the Soft Skills, one is based on scenarios and the other on a more general questionnaire.

What is proposed is to have a system in which various tests can be carried out by having a file that follows a given structure. In order to solve this, we had to extract which elements the main test uses, in this way we can work with the appropriate data and shape it to obtain a new test that meets the initial values.

Keywords

Soft Skills, assessment, web page, Python, Flask, Json.

Índice general

Índice general	iii
Índice de figuras	iv
Índice de tablas	v
1. Introducción	1
1.1. Estructura de la Memoria	1
1.2. Estructura de los Anexos	2
2. Objetivos del proyecto	3
3. Conceptos teóricos	5
4. Técnicas y herramientas	9
5. Aspectos relevantes del desarrollo del proyecto	13
5.1. Elección de las herramientas	13
5.2. Desarrollo del proyecto	17
5.3. Problemas	17
6. Trabajos relacionados	19
7. Conclusiones y Líneas de trabajo futuras	21
Bibliografía	23

Índice de figuras

3.1. Tabla de Soft Skills	7
-------------------------------------	---

Índice de tablas

6.1. Tabla con diferencias entre RiskReals	19
--	----

1. Introducción

Una forma de poder determinar los conocimientos de cualquiera es con la realización de un test, de esta forma se vinculan las **Soft Skills** también denominadas como habilidades blandas las cuales nos muestran cómo se comporta en diferentes escenarios el ser humano. A partir de los valores obtenidos se puede realizar un estudio de cómo se comportara en diferentes escenarios y así tener una idea de que rol ejercerá en un equipo o como actuara ante imprevistos. Una vez dicho esto RiskReal es una página web desde la cual se pueden realizar diferentes cuestionarios o tests para comprobar las Soft Skills que tiene el usuario. Lo que se pretende con este proyecto es la implementación de una plataforma en la cual se pueden tener diferentes test y cuestionarios independientes para que los puedan realizar los diferentes usuarios registrados en la plataforma como de forma anónima. De otra manera los usuarios que tengan el rol de administrador tienen acceso al listado de los usuarios registrados y al listado de los resultados de los test, teniendo la opción de descargar los datos en un archivo “.csv” para tratar los datos.

1.1. Estructura de la Memoria

1. **Introducción.** Breve descripción sobre el proyecto.
2. **Objetivos del proyecto.** Diferentes objetivos que se persiguen con el proyecto
3. **Conceptos teóricos.** Explicación de los conceptos y términos en relación con el proyecto

4. **Técnicas y herramientas.** Listado de herramientas y bibliotecas utilizadas con una breve explicación.
5. **Aspectos relevantes.** Listado de los puntos más interesantes e importantes que han ido surgiendo durante el desarrollo del proyecto.
6. **Trabajos relacionados.** Exposición de trabajos y plataformas relacionadas con RiskReal.
7. **Conclusiones y Líneas de trabajo futuras.** Conclusiones respecto a los diferentes apartados del proyecto y diferentes mejoras que se pueden implementar al proyecto.

1.2. Estructura de los Anexos

- A. **Plan del proyecto:** Plan de proyecto software que estudia la planificación y viabilidad del proyecto.
- B. **Especificación de Requisitos:** Catálogo de los requisitos funcionales y no funcionales del proyecto.
- C. **Especificación de Diseño:** Diseño general de los distintos apartado del sistema así como su justificación.
- D. **Manual de Programador:** Manual con toda la información importante que permita a otro programado retomar el proyecto rápidamente.
- E. **Manual de Usuario:** Manual de usuario que explora todas las características de la plataforma web.
- F. **Anexo de sostenibilización curricular:** Sostenibilidad del proyecto respecto a las ODS.

2. Objetivos del proyecto

En este apartado se proporciona un listado con los diferentes objetivos que se quieren alcanzar con el proyecto.

1. Realizar una página web funcional que responda correctamente.
2. Implementar dos tipos de forma de evaluarse con el fin de tener diferentes tipos de maneras de evaluar diferentes conceptos.
3. Implementación de una manera de moverse entre las diferentes preguntas/cuestiones.
4. Tener la opción de realizar diferentes tests y cuestionarios entre los que estén disponibles.
5. Usar correctamente las sesiones de Flask, con el fin de separar inicios de un mismo usuario en diferentes navegadores/dispositivos.
6. Manejar adecuadamente las diferentes librerías utilizadas como sqlalchemy y migrate.
7. Utilizar la metodología **Kanban** durante el desarrollo del proyecto.
8. Implementar el rol administrador con funcionalidades diferentes respecto a los demás.
9. Utilizar Github para integrar el proyecto y su control de versiones.
10. Implementar que la aplicación se adapte a diferentes dispositivos.

3. Conceptos teóricos

En este apartado se definen algunos términos y conceptos relevantes sobre el proyecto.

Soft Skills

Las habilidades blandas [9], competencias blandas o habilidades suaves son una combinación de habilidades sociales, habilidades de comunicación, rasgos de la personalidad, actitudes, atributos profesionales, inteligencia social e inteligencia emocional, que facultan a las personas para moverse por su entorno, trabajar bien con otros, realizar un buen desempeño y, complementándose con las habilidades duras, conseguir sus objetivos. Las habilidades blandas son un cúmulo de rasgos productivos de la personalidad que caracterizan las relaciones de una persona en un medio. Estas habilidades pueden incluir autoestima, comunicación, elocuencia, hábitos personales, empatía, gestión del tiempo, trabajo en equipo y liderazgo. Una definición basada en la revisión de artículos al respecto considera "habilidades blandas" un término paraguas para habilidades con tres elementos funcionales clave: interpersonales, sociales y profesionales. Las habilidades blandas se pueden categorizar en diferentes como estas:

1. **Comunicación:** capacidad de expresarse claramente hablando y escribiendo, y de hacer presentaciones en público.
2. **Cortesía:** buenos modales, etiqueta, respeto, decir «por favor» y «gracias».
3. **Flexibilidad:** adaptabilidad, disposición al cambio y a la formación continua, aceptación de lo nuevo, se le puede enseñar.

4. **Integridad:** sinceridad, moralidad, valores personales, honradez.
5. **Habilidades interpersonales:** agradable, con sentido del humor, amistoso, hospitalario, empático, con autocontrol, paciencia, sociabilidad, calidez, habilidades sociales.
6. **Actitud positiva:** optimista, entusiasta, ánimo a sus compañeros, feliz, seguro.
7. **Profesionalidad:** bien vestido, buen aspecto, compostura.
8. **Responsabilidad:** responsable, fiable, termina el trabajo, quiere hacerlo bien, es consciente, tiene recursos, autodisciplina y sentido común.
9. **Trabajo en equipo:** coopera, se lleva bien con otros, es agradable, anima, ayuda a quien lo necesita.
10. **Actitud hacia el trabajo:** trabaja duro, está disponible, es leal, puntual, tiene iniciativa, dedicación, motivación y falta poco al trabajo.
11. **Creatividad** [8]: capacidad de crear nuevas ideas o conceptos, de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales.
12. **Resolución de problemas** [17]: la conclusión de un proceso más amplio que tiene como pasos previos la identificación del problema y su modelado.
13. **Ser organizado:** llevar a cabo unos pasos de manera seguida non un propósito aprovechando el tiempo lo mas posible.

Estas pueden ser utilizadas para poder hacer un cribado de un gran grupo de personas y así poder realizar una agrupación con diferentes integrantes que tengas diferentes atributos como el de líder, trabajador, creativo, espontaneo, observador, etc. ; o por otra parte hacer una selección para un nuevo proyecto a partir de una semejanza a los conocimientos deseados. [4]

SOFT SKILLS	
HABILIDAD DE COMUNICACION	Comunicación verbal y no verbal Escucha activa Public Speaking
LIDERAZGO	Trabajo en equipo Toma de decisiones Assunción de riesgos Negociación Motivación Colaboración
HABILIDADES PERSONALES	Aprender a aprender Gestión del estado emocional Resiliencia Adaptabilidad/ flexibilidad Entusiasmo Empatía
CREATIVIDAD	Solución de problemas Pensamiento crítico y creativo Innovación

Figura 3.1: Tabla de Soft Skills

4. Técnicas y herramientas

Sobre este apartado se describirán las diferentes herramientas y técnicas utilizadas para la realización del proyecto teniendo una continuidad y eficiencia.

Github

Github [14] es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza para alojar los diferentes cambios que se han ido implementado en un repositorio, que es lo que vendría ser un sitio en la nube donde guardar los proyectos de desarrollo software.

En mi caso utilizo Github a modo de tener un control de las diferentes versiones que voy implementando y a su vez como gestor de las diferentes tareas a realizar durante los Sprints.

Zube

Zube es una extensión que se implementa sobre los repositorios de Github, con Zube utilizamos la metodología Kanban [15] que es un método para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecarguen los miembros del equipo.

Para el proyecto he utilizado Zube con el fin de controlar el plazo de los Sprints y asignar las diferentes tareas “issues” sobre este mismo.

Overleaf

Overleaf [16] es un editor LaTeX colaborativo basado en la nube que se utiliza para escribir, editar y publicar documentos. Sigue un enfoque para la redacción científica y técnica de manera simple y compartida.

Overleaf ha sido utilizado para el desarrollo de la memoria y anexos del proyecto.

Visual Studio Code

Visual Studio Code [18] es un editor de código fuente de código abierto el cual tiene soporte con Git para poder desarrollar un proyecto desde Github, además tiene soporte para la gran mayoría de lenguajes de programación. Ha sido clave para desarrollar el código del proyecto y darle una utilidad.

Docker

Docker [12] es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que contenedores independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

Como bien se explica sirve para contener la aplicación una vez ya tenemos preparado el entorno y funciona correctamente, de esta manera siempre que se disponga de Docker Desktop en el caso de tener una imagen del proyecto y un contenedor correspondiente a esta, se podrá ejecutar la aplicación de una forma más sencilla.

ChatGPT

ChatGPT [11] es una aplicación de chatbot de inteligencia artificial desarrollado en 2022 por OpenAI que se especializa en el diálogo, ha sido entrenada y es capaz de resolver cuestiones útiles respecto cualquier tema. Ha sido utilizado para la búsqueda de algún error producido durante el desarrollo del proyecto.

Flask

Flask [13] es un framework minimalista escrito en Python que permite crear aplicaciones web, es la base sobre la cual mi proyecto se desarrolla. Hay otras alternativas como Django o Streamlit, las cuales eran opciones válidas para el desarrollo pero por cuestiones de comodidad al haber utilizado anteriormente Flask y su uso durante el tiempo abalan un amplio abanico de posibilidades respecto al desarrollo.

SqlAlchemy

SqlAlchemy [3] es una librería dependiente de Flask la cual es utilizada para el manejo de las bases de datos. Tiene soporte con un gran número de BBDD diferentes, al cabo de elegir sobre qué BBDD trabajar al solo disponer de dos tablas con pocos campos tome la decisión de utilizar SQLite, ya que este no utiliza un sistema de cliente-servidor disminuyendo la latencia, pero en su contraparte tiene limitado el valor de datos que puede manejar, lo cual en mi caso no es un problema al trabajar con pocos datos.

Migrate

Migrate [2] es una librería dependiente de Flask la cual es utilizada para tener guardado los diferentes cambios que se realizan a las tablas, de la misma manera se encarga de generar las tablas y de guardar los datos en estas. Se complementa perfectamente con SqlAlchemy y al no tener una base de datos en un servidor, si utilizamos migrate tenemos la opción de guardar la instancia de la BBDD y tener acceso a ella en cualquier momento.

Bootstrap

Bootstrap [10] es un framework multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Ha sido utilizado para el apartado estético de la página web utilizando una plantilla acompañándose de un archivo “.css” y de modificaciones propias de Bootstrap.

5. Aspectos relevantes del desarrollo del proyecto

En este apartado indicare las elecciones que he tomado, el transcurso del proyecto y como se han abordado algún problema relevante.

5.1. Elección de las herramientas

Una vez se empieza a trabajar hay que poner los cimientos, por lo que hay que realizar una selección cuidadosa con los elementos que se va a realizar el proyecto. Por eso con la ayuda de esta [herramienta web](#) para comparar aspectos de herramientas con el mismo propósito he podido hacer una elección cómoda.

Visual Studio Code

Para la elección del editor de texto tenemos varias opciones como pueden ser “PyCharm,, Atom, Komodo y VSCodium” todas estas opciones son válidas pero si tenemos que elegir una sería PyCharm la cual no he escogido debido al haber ya trabajado repetidamente con Visual Studio Code y estar más familiarizado, en parte para las necesidades que necesito no hay ninguna extensión que tenga que implementar que no venga ya incluida con Visual Studio Code. Diferencias entre Visual Studio Code y PyCharm utilizadas para la selección [7]:

- **Interfaz de usuario:** PyCharm ofrece una interfaz de usuario más completa y con más funciones, que ofrece una experiencia de IDE rica y profesional con herramientas avanzadas de depuración y creación

de perfiles. Por otro lado, Visual Studio Code presenta una interfaz de usuario liviana y minimalista, que se centra en la simplicidad y la extensibilidad, con una amplia gama de extensiones disponibles y opciones de personalización.

- **Edición de código:** PyCharm ofrece una potente asistencia de codificación y finalización de código inteligente, brindando sugerencias útiles e importaciones automáticas para módulos de Python. También incluye funciones especializadas para Django, Flask y otros marcos. Si bien Visual Studio Code también brinda finalización de código y sugerencias, es posible que requiera extensiones adicionales para igualar el nivel de inteligencia de código que brinda PyCharm.
- **Herramientas y complementos integrados:** PyCharm incluye un conjunto completo de herramientas integradas para el control de versiones (por ejemplo, Git), marcos de prueba, bases de datos y desarrollo web. Ofrece una integración perfecta con herramientas populares de Python, como entornos virtuales y administradores de paquetes. Visual Studio Code, por otro lado, depende en gran medida del uso de complementos y extensiones para proporcionar funcionalidades similares, lo que permite a los desarrolladores personalizar su entorno con las herramientas que elijan.
- **Rendimiento:** PyCharm está basado en la plataforma IntelliJ, que puede consumir una cantidad significativa de recursos del sistema y generar tiempos de inicio e indexación más lentos en comparación con Visual Studio Code. Visual Studio Code, al ser un IDE liviano, generalmente ofrece un rendimiento más rápido y tiempos de inicio más rápidos.
- **Comunidad y ecosistema:** PyCharm cuenta con una comunidad consolidada y madura con una amplia documentación y foros de soporte. Es ampliamente utilizado en la comunidad de desarrollo de Python, lo que se traduce en una gran cantidad de recursos y tutoriales disponibles. Visual Studio Code, desarrollado por Microsoft, se beneficia de su amplia comunidad y ecosistema, y ofrece una amplia gama de extensiones, temas e integraciones impulsados por la comunidad con otras herramientas de Microsoft.
- **Precios:** PyCharm ofrece ediciones comunitarias tanto gratuitas como de pago. Las versiones de pago ofrecen funciones adicionales y soporte para el desarrollo web y empresarial. Visual Studio Code, por otro

lado, es completamente gratuito y de código abierto, lo que lo convierte en una opción atractiva para los desarrolladores que prefieren una solución rentable.

GitHub

En cuanto a la selección de la plataforma donde alojar el proyecto GitHub es muy común, tras realizar diferentes proyectos en esta plataforma ya me encuentro familiarizado y no he tenido ninguna intención de realizar el alojamiento en otras plataformas. Por otra parte la integración de la metodología Kanban por parte de Zube resulta más sencillo el manipular las tareas. Diferencias entre GitHub y GitLab utilizadas para la selección [6]:

- **Alojamiento e implementación:** GitHub es principalmente una plataforma basada en la nube que ofrece alojamiento para repositorios Git. Proporciona un servicio totalmente administrado, donde los usuarios pueden crear y administrar sus repositorios directamente en los servidores de GitHub. GitLab, por otro lado, ofrece una solución SaaS basada en la nube (GitLab.com) y una solución alojada por los propios usuarios (GitLab Community Edition). Esto significa que los usuarios tienen la flexibilidad de elegir entre alojar sus repositorios en la plataforma en la nube de GitLab o implementar GitLab en su propia infraestructura.
- **Precios y licencias:** GitHub ofrece opciones gratuitas para repositorios públicos y planes pagos para repositorios privados, mientras que GitLab ofrece una Community Edition gratuita y una Enterprise Edition paga.
- **Conjunto de características:** GitHub es conocido por su sólido ecosistema de integración, que incluye una amplia gama de herramientas y servicios de terceros para la integración continua, la implementación y la gestión de proyectos. GitLab, por otro lado, tiene como objetivo proporcionar una plataforma DevOps más completa con características integradas como canalizaciones de CI/CD, integración con Kubernetes y un registro de contenedores Docker integrado.
- **Comunidad y colaboración:** GitHub cuenta con una comunidad de desarrolladores grande y activa, lo que la convierte en una plataforma popular para proyectos de código abierto. Ofrece funciones que facilitan la colaboración y la contribución, como solicitudes de incorporación de cambios, revisiones de código y seguimiento de problemas. GitLab

también cuenta con una comunidad en crecimiento y su naturaleza de código abierto fomenta la colaboración. Además, GitLab enfatiza el concepto de "Flujo de GitLab", que promueve un flujo de trabajo más integrado y optimizado para el desarrollo, la colaboración y la implementación. Esto incluye funciones como canales de CI/CD integrados y una interfaz consolidada para administrar el código, los problemas y la gestión de proyectos.

Flask

Para la realización de la página web una manera sencilla de empezar en el caso de no haber realizado ninguna anteriormente, es Flask por lo cual entre la propia ayuda que te ofrece la documentación de flask y el gran abanico de librerías que ofrece son suficientes para el proyecto que tenía en mente. Diferencias entre GitHub y GitLab utilizadas para la selección [5]:

- **Tipo de framework:** Django es un framework web de alto nivel y de pila completa que sigue el patrón de diseño Modelo-Vista-Plantilla (MVT), y ofrece funciones integradas como autenticación, interfaz de administración y ORM. Por otro lado, Flask es un framework web ligero que se basa en Werkzeug y Jinja2 y ofrece flexibilidad a los desarrolladores para elegir las herramientas y bibliotecas con las que quieren trabajar.
- **Escalabilidad:** Django es más adecuado para aplicaciones más grandes debido a sus características y convenciones integradas que ayudan a gestionar proyectos complejos de manera eficiente. Flask, al ser más minimalista, es más adecuado para proyectos más pequeños o aplicaciones que requieren un enfoque más personalizado.
- **Soporte y ecosistema de la comunidad:** Django tiene una comunidad más grande y madura en comparación con Flask, y ofrece una amplia gama de paquetes de terceros, tutoriales y documentación que pueden ayudar a los desarrolladores a crear e implementar aplicaciones más rápido. Flask, por otro lado, tiene una comunidad más pequeña, pero ofrece un ecosistema más liviano y flexible que permite a los desarrolladores tener más control sobre sus proyectos.
- **Curva de aprendizaje:** Django tiene una curva de aprendizaje más pronunciada en comparación con Flask debido a sus características integrales y convenciones que los desarrolladores deben comprender y seguir. Flask, al ser más simple y minimalista, tiene una barrera de

entrada más baja, lo que facilita que los principiantes comiencen a crear aplicaciones web.

- **Enrutamiento de URL:** En Django, el enrutamiento de URL se realiza a través de una configuración de URL predefinida mediante expresiones regulares, lo que lo hace más estructurado y organizado. Flask, por otro lado, utiliza decoradores para definir rutas, lo que proporciona una forma más flexible e intuitiva de enrutar URL en la aplicación.
- **Motor de plantillas:** Django viene con un motor de plantillas integrado que sigue los principios DRY (Don't Repeat Yourself), lo que permite a los desarrolladores reutilizar el código de las plantillas de manera eficiente. Flask, por otro lado, utiliza Jinja2 como su motor de plantillas predeterminado, lo que proporciona una sintaxis más liviana y fácil de usar para el diseñador para crear plantillas.

5.2. Desarrollo del proyecto

Para el desarrollo he seguido unas pautas que han sido claves, las cuales consisten en:

1. Plantear la idea a fondo con un propósito adicional al deseado para abarcar posibles cambios.
2. Realizar un primer código con el cual se logre lo deseado y después adaptarlo para obtener la funcionalidad deseada.
3. Realizar una prueba de todo el programa para depurar y comprobar que no haya ningún error.

5.3. Problemas

Los problemas siempre aparecen y como no podía ser durante el desarrollo de la aplicación surgieron diferentes errores los cuales se han ido desarrollado.

Un problema que tuve fue la visualización de caracteres erróneos cuando el texto disponía de alguna tilde o símbolo de interrogación, al estar trabajando con todos los ficheros en “UTF-8” comprobándolo con el propio “Visual Studio Code” o el “notepad++” y de asegurar que el texto del html estuviese codificado también adecuadamente, no tenía el porqué del error, pero tras

darle algunas vueltas se me ocurrió la idea de que no estuviese tratando el fichero correctamente y de esta manera lo solucione al especificar como quería abrir el fichero en cuestión para obtener los datos ya así codificados en “UTF-8”.

Durante la implementación del usuario invitado para poder realizar la evaluación sin estar registrado, me encontré con que en un principio al solo tener la evaluación de un test al añadir el modelo de los cuestionarios tuve el dilema de cómo poner en común al invitado, por lo que para poder solucionarlo cree una función intermedia en la cual comprobaba si estaba logeado algún usuario en la sesión así pudiendo utilizar el mismo invitado en común para luego llamar a la función correspondiente dependiendo si se quiere hacer un test o cuestionario.

A la hora de utilizar Docker para crear un contenedor de la aplicación me surgió un error en el cual las imágenes correspondientes al test no se muestran dando como fallo que no se encuentran, el problema en cuestión resultaba en que las imágenes no se encontraban en el destino pero viendo los archivos del contenedor las imágenes si se encontraban, lo cual para solucionarlo probé diferentes maneras de llegar a la ruta pero al final el error en si es que los propios archivos tenían la extensión **.PNG** mientras utilizaba la extensión **.png** como ruta para cargar las imágenes, como docker genera una máquina virtual con un sistema operativo de Linux, al tratar los archivos las extensiones son importantes, por lo que para solucionarlo tuve que cambiar la extensión de la ruta que apunta a las imágenes del json a **.PNG**.

6. Trabajos relacionados

El caso claro sobre un trabajo relacionado es el RiskReal original [1] el cual realizo la universidad de Burgos en colaboración con la Unión Europea por lo que vamos a ver las diferentes aportaciones que nos ofrecen las dos versiones.

Funcionalidad	Original	Mi proyecto
Registro	Sí	Sí
Login	Sí	Sí
Opción de Invitado	Sí	Sí
Cambiar la selección	Solo cuestionario	Ambos
Opcion de anterior y siguiente en los cuestionarios/tests	No	Sí
Indicador pregunta	Solo cuestionario	Ambos
Salir del cuestionario/test	Sí	No
Cambio de contraseña	Sí	Sí
Rol de administrador	ND	Sí
Consulta de los usuarios	ND	Sí
Consulta de los resultados	ND	Sí
Opción de descarga de los resultados	ND	Sí
Disponibilidad a todos los cuestionarios/tests	No	Sí
Añadir nuevos cuestionarios/tests	No	Sí
Tiempo entre las transacciones	Medio	Bajo

Tabla 6.1: Tabla con diferencias entre RiskReals

Hay varios aspectos que no puedo comprobar respecto al no ser un administrador, lo cual lo tomare como una clara ventaja respecto a la original, a parte la funcionalidad que ofrecen los diferentes cuestionarios y test en mi caso se mantienen pero en el caso del original no tienen las mismas funcionalidades. Hay algunas disponibilidades que no se comparten como poder salir en mitad de la realización de un cuestionario/test lo cual es una línea de trabajo futura. En cuanto a la disponibilidad de todos los cuestionarios/tests me refiero a que en mi caso el botón desplegable te muestra todos aquellos respectivamente a su categoría, es decir en el original solo se puede hacer un cuestionario y un test.

7. Conclusiones y Líneas de trabajo futuras

Para finalizar se expondrán las conclusiones finales obtenidas a la realización del proyecto y por otra parte las líneas de trabajo futuras.

Conclusiones

A medida que se avanzaba en el proyecto los objetivos marcados se realizaban satisfactoriamente en su mayoría aun al no haber podido realizar alguna funcionalidad de la manera deseada. La navegación entre las pestañas se hace intuitiva y no es pesado lo cual nos proporciona una página web que es accesible y no necesita tener una guía a lado para poder interactuar con ella satisfactoriamente. Al realizar el proyecto se ha obtenido una perspectiva que no se tenía respecto a cómo desarrollar una página web y diferentes funcionalidades que debería tener de base. Como resumen la realización de este proyecto pone a prueba la utilización de diferentes aspectos enseñados en la carrera, además que se hace un planteamiento único que resuelve los diferentes problemas que van surgiendo a medida.

Líneas de trabajo futuras

Aunque se tenga una funcionalidad correcta y unas funcionalidades satisfechas, siempre es posible realizar mejoras para tener un producto mejor.

- Mejora de la lógica de cambio de contraseña aplicando una doble verificación al enviar un mensaje al correo asociado a la cuenta que desea cambiar la contraseña.

- Implementación de una opción que te permita modificar los cuestionarios/tests o crear nuevos desde la propia aplicación web.
- Implementación de un nuevo rol “editor” que sea el encargado del apartado anterior.
- Poder hacer un filtrado sobre los diferentes campos respecto a las tablas de usuarios y los resultados.
- Creación de unas pruebas unitarias para comprobar un funcionamiento correcto sin tener que realizarlo a mano.

Bibliografía

- [1] Riskreal. <https://app.riskreal.eu/>. [Internet; acedido 2-marzo-2024].
- [2] Flask. Flask-migrate. <https://flask-migrate.readthedocs.io/en/latest/>. [Internet; acedido 24-junio-2024].
- [3] Flask. Flask-sqlalchemy. <https://flask-sqlalchemy.palletsprojects.com/en/3.1.x/>. [Internet; acedido 24-junio-2024].
- [4] softskillsacademy. Soft skills: un conjunto de características interpersonales. <https://www.softskillsacademy.es/soft-skills-un-conjunto-de-caracteristicas-interpersonales/>, 2018. [Internet; acedido 3-julio-2024].
- [5] stackshare. stackshare: Flask vs django. <https://stackshare.io/stackups/flask-vs-django>. [Internet; acedido 23-febrero-2024].
- [6] stackshare. stackshare: Github vs gitlab. <https://stackshare.io/stackups/github-vs-gitlab>. [Internet; acedido 23-febrero-2024].
- [7] stackshare. stackshare: Pycharm vs visual studio code. <https://stackshare.io/stackups/pycharm-vs-visual-studio-code>. [Internet; acedido 23-febrero-2024].
- [8] Wikipedia. Creatividad — wikipedia, la enciclopedia libre, 2023. [Internet; acedido 3-julio-2024].
- [9] Wikipedia. Soft skills — wikipedia, la enciclopedia libre, 2023. [Internet; acedido 3-julio-2024].

- [10] Wikipedia. Bootstrap (framework) — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].
- [11] Wikipedia. Chatgpt — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].
- [12] Wikipedia. Docker (software) — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 1-julio-2024].
- [13] Wikipedia. Flask — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].
- [14] Wikipedia. Github — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].
- [15] Wikipedia. Kanban (desarrollo) — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].
- [16] Wikipedia. Overleaf — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].
- [17] Wikipedia. Resolución de problemas — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 3-julio-2024].
- [18] Wikipedia. Visual studio code — wikipedia, la enciclopedia libre, 2024. [Internet; acedido 24-junio-2024].