

Trabajo Fin de Grado

Aplicación de evaluación de riesgos laborales
de un puesto de trabajo

Occupational risk assessment application for a
job

Autor

Sergio Alloza García

Directoras

M^a Carmen Aguilar Martín

Piedad Garrido Picazo

ESCUELA UNIVERSITARIA POLITÉCNICA DE TERUEL

2022

AGRADECIMIENTOS

Quiero aprovechar este apartado para dar las gracias, en primer lugar, a Carmen y a Piedad, mis tutoras del proyecto, con las que he intercambiado múltiples correos durante este tiempo, por aguantar todas mis dudas y preguntas y ayudarme a darles solución.

En segundo lugar, a mis amigos Tristán, Miguel Ángel, Nacho y Cristina, en los que me he apoyado para la realización de diversas partes del proyecto y con los que he compartido muchos de los años de carrera.

Y, por último, y no menos importante, a mis padres, que han estado siempre para mí, me han apoyado en los momentos más delicados y me han animado a seguir adelante.

RESUMEN

El proyecto consiste en la realización de una aplicación móvil para identificar, evaluar y proponer medidas preventivas a los riesgos laborales de un puesto de trabajo con Pantalla de Visualización de Datos (PVD).

El usuario final, ya sea técnico u operario de Prevención de Riesgos Laborales (PRL), así como cualquier persona interesada en evaluar y asegurar su puesto de trabajo con PVD, dispondrá de una aplicación con la que poder identificar los factores de riesgo de un puesto con PVD incluido el teletrabajo, evaluarlos y proponer una serie de medidas preventivas acordes a la valoración realizada. Esta información (evaluación de riesgos) se almacenará en una base de datos para su posterior consulta.

Para ello, se ha desarrollado una aplicación desarrollada mediante Android Studio, utilizando lenguaje Kotlin y un completo repositorio de información en Firebase, sin dejar de lado el seguimiento e incorporación de las normas y legislación vigente al respecto.

Palabras clave

A continuación, se muestra una serie de palabras o siglas de importancia que aparece a lo largo del presente documento, junto a su significado:

- *Evaluación de riesgos*
- *Trabajador usuario de PVD*
- *Teletrabajo*
- *App*
- *Firebase*

ABSTRACT

The project consists of the realization of a mobile application to identify, evaluate and propose preventive measures to the occupational risks of a job with a Data Display Screen (DDS).

The end user, whether they are an Occupational Risk Prevention (ORP) technician or operator, as well as anyone interested in evaluating and insuring their job with DDS, will have an application with which they can identify the risk factors of a job with DDS, including teleworking, evaluate them and propose a series of preventive measures according to the assessment made. This information (risk assessment) will be stored in a database for later consultation.

To do this, an application developed using Android Studio has been developed, using Kotlin language and a complete repository of information in Firebase, without neglecting the monitoring and incorporation of current regulations and legislation in this regard.

Keywords

Below is a series of important words or acronyms that appear throughout this document, along with their meaning:

- *Risk assessment*
- *DDS user worker*
- *Telecommuting*
- *App*
- *Firebase*

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS	VI
ÍNDICE DE TABLAS	IX
1. INTRODUCCIÓN Y OBJETIVOS.....	1
2. ESTADO DEL ARTE	2
3. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN.....	6
3.1. Análisis	6
3.1.1. Requisitos funcionales.....	6
3.1.2. Requisitos no funcionales.....	7
3.1.3. Diagrama de casos de uso	8
3.2. Diseño	8
3.2.1. Nombre y logo	8
3.2.2. Diagrama Entidad-Relación	9
3.2.3. Diagrama de clases	10
3.2.4. Prototipos de interfaz.....	11
3.2.5. Arquitectura de la aplicación.....	14
3.2.6. Paleta de colores	15
3.2.7. Diseño físico e Integración del SGBD.....	15
3.3. Implementación	20
3.3.1. Tecnologías utilizadas.....	20
3.3.2. Implementación de las actividades	21
3.3.3. Implementación de la BD	34
4. ACCESIBILIDAD Y USABILIDAD	38
4.1. Principios de usabilidad de Jakob Nielsen	38
5. LICENCIA SOFTWARE Y DOCUMENTAL.....	41
6. CONCLUSIONES	43
6.1. Trabajo futuro	43
7. REFERENCIAS BIBLIOGRÁFICAS	45
ANEXOS	48
Anexo I - Manual de usuario	48
Registrar usuario.....	49

Iniciar sesión	50
Visualizar la ayuda	51
Realizar evaluación	53
Guardar evaluación	55
Dar permisos a la aplicación	56
Visualizar evaluación	57
Borrar evaluación	59
Descargar evaluación.....	60
Visualizar FAQ.....	62
Cerrar sesión.....	64

ÍNDICE DE FIGURAS

FIGURA 01. METODOLOGÍA DE INGENIERÍA DEL SOFTWARE UTILIZADA.	6
FIGURA 02. DIAGRAMA DE CASOS DE USO DE PVD PREVENCIÓN.	8
FIGURA 03. LOGOS DE ALL PREVENTION Y PVD PREVENCIÓN.	9
FIGURA 04. DIAGRAMA E-R DE PVD PREVENCIÓN (DEREDITOR).	9
FIGURA 05. DIAGRAMA DE CLASES DE LA APLICACIÓN GENERADO MEDIANTE MODELIO OPEN SOURCE.....	11
FIGURA 06. PANTALLAS DE AUTENTICACIÓN Y REGISTRO.	12
FIGURA 07. PANTALLAS DE MENÚ PRINCIPAL Y REALIZAR EVALUACIÓN.	12
FIGURA 08. PANTALLAS DE RESULTADOS Y MIS EVALUACIONES.	13
FIGURA 09. PANTALLAS DE INFORMACIÓN DE MI EVALUACIÓN Y FAQ.....	13
FIGURA 10. DIAGRAMA DE LA ARQUITECTURA DE LA APLICACIÓN.	14
FIGURA 11. PALETA DE COLORES DE PVD PREVENCIÓN.	15
FIGURA 12. URL Y BOTÓN DE ACCESO A LA CONSOLA EN FIREBASE.....	16
FIGURA 13. BOTÓN DE AGREGAR PROYECTO EN FIREBASE.	16
FIGURA 14. BOTÓN DE CREAR PROYECTO EN FIREBASE.....	16
FIGURA 15. BOTONES DE AGREGAR APP Y DE ANDROID EN FIREBASE.	16
FIGURA 16. BOTÓN DE REGISTRAR APP EN FIREBASE.	17
FIGURA 17. BOTÓN DE DESCARGA DEL FICHERO JSON DE FIREBASE.	17
FIGURA 18. RECORRIDO DE LA INCLUSIÓN DEL ARCHIVO JSON EN ANDROID STUDIO.	17
FIGURA 19. BOTÓN DE SIGUIENTE EN FIREBASE.	18
FIGURA 20. UBICACIÓN DEL FICHERO BUILD.GRADLE (PROJECT) DEL PROYECTO EN ANDROID STUDIO.....	18
FIGURA 21. LÍNEAS DE CÓDIGO A INCLUIR EN EL ARCHIVO BUILD.GRADLE (PROJECT) EN ANDROID STUDIO.....	18
FIGURA 22. LÍNEAS DE CÓDIGO A INCLUIR EN EL ARCHIVO BUILD.GRADLE (APP) EN ANDROID STUDIO.....	19
FIGURA 23. BOTÓN DE SINCRONIZAR EN ANDROID STUDIO.....	19
FIGURA 24. BOTÓN DE SIGUIENTE EN FIREBASE.	19
FIGURA 25. MENSAJE DE AGREGACIÓN DE APP EXITOSA EN FIREBASE.	19
FIGURA 26. PESTAÑAS DE CÓDIGO, DISEÑO Y MEZCLA DE AMBOS DE ANDROID STUDIO.	21
FIGURA 27. SPLASH SCREEN DE PVD PREVENCIÓN.	21
FIGURA 28. CÓDIGO DE LA SPLASH SCREEN ASOCIADA A LA ACTIVIDAD DE AUTENTICACIÓN.....	22
FIGURA 29. CÓDIGO PARA LOS EVENTOS DE FIREBASE ANALYTICS EN LA ACTIVIDAD DE AUTENTICACIÓN.....	22
FIGURA 30. CÓDIGO DEL BOTÓN ACCEDER DE LA APLICACIÓN (ACTIVIDAD DE AUTENTICACIÓN).....	23

FIGURA 31. FRAGMENTO DE CÓDIGO DEL FICHERO BUILD.GRADLE A NIVEL DE APLICACIÓN.	23
FIGURA 32. CÓDIGO DEL BOTÓN ACCEDER MEDIANTE GOOGLE DE LA APLICACIÓN (ACTIVIDAD DE AUTENTICACIÓN).....	24
FIGURA 33. CÓDIGO DEL BOTÓN ACCEDER DE LA APLICACIÓN (ACTIVIDAD DE REGISTRO).	25
FIGURA 34. CÓDIGO DEL BOTÓN DE PREGUNTAS FRECUENTES (FAQ) DEL MENÚ PRINCIPAL.	26
FIGURA 35. CÓDIGO DEL BOTÓN DE VOLVER AL MENÚ PRINCIPAL (ACTIVIDAD DE AYUDA).	26
FIGURA 36. FRAGMENTO DE CÓDIGO DE LA LISTA DE VERIFICACIÓN (ACTIVIDAD DE REALIZACIÓN DE EVALUACIONES).....	27
FIGURA 37. LÍNEA DE CÓDIGO PARA INTERACTUAR CON LA BD (ACTIVIDAD DE RESULTADOS).....	28
FIGURA 38. FRAGMENTO DE CÓDIGO DE GUARDADO DE VARIABLES EN PREFERENCIAS (ACTIVIDAD DE RESULTADOS).	28
FIGURA 39. CÓDIGO DEL BOTÓN DE GUARDAR EVALUACIÓN (ACTIVIDAD DE RESULTADOS).....	29
FIGURA 40. FRAGMENTO DE CÓDIGO REFERENTE A LOS PERMISOS (ACTIVIDAD DE VISUALIZACIÓN DE EVALUACIONES).	29
FIGURA 41. CÓDIGO DE LA FUNCIÓN ONCREATE (ACTIVIDAD DE VISUALIZACIÓN DE EVALUACIONES).....	30
FIGURA 42. FRAGMENTO DE CÓDIGO DE LA FUNCIÓN SAVEPDF (ACTIVIDAD DE VISUALIZACIÓN DE EVALUACIONES).	30
FIGURA 43. CÓDIGO CORRESPONDIENTE A LOS BOTONES DE VISUALIZAR, BORRAR Y DESCARGAR EVALUACIÓN (ACTIVIDAD DE VISUALIZACIÓN DE EVALUACIONES).	31
FIGURA 44. FRAGMENTO DE CÓDIGO DE LA FUNCIÓN ONCREATE (ACTIVIDAD DE INFORMACIÓN DE LA EVALUACIÓN).	33
FIGURA 45. FRAGMENTO DE CÓDIGO AÑADIDO AL FICHERO BUILD.GRADLE A NIVEL DE APP ASOCIADO A LA PANTALLA DE PREGUNTAS FRECUENTES.	33
FIGURA 46. FRAGMENTO DE CÓDIGO DE LA FUNCIÓN ONCREATE (ACTIVIDAD DE PREGUNTAS FRECUENTES).	34
FIGURA 47. CÓDIGO DE LA SOBRESERITURA DE LA FUNCIÓN ONBACKPRESSED.....	34
FIGURA 48. PESTAÑA REGLAS DE LA BD.	35
FIGURA 49. REGLA REFERENTE A LA DURACIÓN DEL MODO DE PRUEBA DE LA BD.	35
FIGURA 50. INFORMACIÓN ALMACENADA EN LA BD DE PVD PREVENCIÓN.	36
FIGURA 51. PESTAÑA DE USUARIOS (FIREBASE AUTHENTICATION).	37
FIGURA A01. PANTALLA DE AUTENTICACIÓN DE PVD PREVENCIÓN.....	48
FIGURA A02. PANTALLA DE REGISTRO DE PVD PREVENCIÓN.	49
FIGURA A03. PANTALLA DE AUTENTICACIÓN DE PVD PREVENCIÓN (INICIAR SESIÓN).....	50
FIGURA A04. MENÚ PRINCIPAL DE PVD PREVENCIÓN.	50
FIGURA A05. MENÚ PRINCIPAL DE PVD PREVENCIÓN (VISUALIZAR LA AYUDA).	51
FIGURA A06. PANTALLA DE AYUDA DE PVD PREVENCIÓN.	51
FIGURA A07. INFORMACIÓN IMPORTANTE DE LA PANTALLA DE AYUDA DE PVD PREVENCIÓN.	52
FIGURA A08. MENÚ PRINCIPAL Y PANTALLA DE REALIZAR EVALUACIÓN DE PVD PREVENCIÓN.....	53
FIGURA A09. PANTALLAS DE REALIZAR EVALUACIÓN Y RESULTADOS DE PVD PREVENCIÓN.	54
FIGURA A10. PANTALLA DE RESULTADOS Y MENSAJE DE EVALUACIÓN GUARDADA DE PVD PREVENCIÓN.	55

FIGURA A11. PANTALLA DE SELECCIÓN DE PERMISOS DEL DISPOSITIVO.....	56
FIGURA A12. ACCESO A LOS ARCHIVOS DE LA PANTALLA DE SELECCIÓN DE PERMISOS.	57
FIGURA A13. MENÚ PRINCIPAL Y PANTALLA DE MIS EVALUACIONES DE PVD PREVENCIÓN.	57
FIGURA A14. PANTALLA DE VISUALIZACIÓN DE UNA EVALUACIÓN DE PVD PREVENCIÓN.....	58
FIGURA A15. PANTALLA DE MIS EVALUACIONES Y MENSAJE DE BORRAR EVALUACIÓN DE PVD PREVENCIÓN.	59
FIGURA A16. PANTALLA DE MIS EVALUACIONES Y MENSAJE DE DESCARGAR EVALUACIÓN DE PVD PREVENCIÓN.	60
FIGURA A17. CAMINO PARA ENCONTRAR LA EVALUACIÓN DESCARGADA EN UN DISPOSITIVO SAMSUNG GALAXY A40.....	61
FIGURA A18. EVALUACIÓN DESCARGADA MEDIANTE LA APLICACIÓN PVD PREVENCIÓN.	61
FIGURA A19. MENÚ PRINCIPAL DE PVD PREVENCIÓN.....	62
FIGURA A20. PANTALLA DE PREGUNTAS FRECUENTES (FAQ) DE PVD PREVENCIÓN.....	63
FIGURA A21. MENÚ PRINCIPAL DE PVD PREVENCIÓN.....	64

ÍNDICE DE TABLAS

TABLA 01. LISTADO DE APLICACIONES EXISTENTES RELACIONADAS CON LA PRL.....	3
TABLA 02. EVALUACIÓN DE DIVERSOS ASPECTOS DE LAS APLICACIONES RELACIONES CON LA PRL.	4
TABLA 03. TECNOLOGÍAS UTILIZADAS EN EL PROYECTO.	20

1. INTRODUCCIÓN Y OBJETIVOS

La idea de realizar este proyecto surgió de la búsqueda de la sinergia del campo de la informática, con otro terreno abordado en la carrera con el que, a priori, no parecía tener tanta relación: la Prevención de Riesgos Laborales [03].

Una vez planteada la idea, se decidió realizar una aplicación para móvil que englobase algunos de los conocimientos adquiridos durante la titulación, así como algunas tecnologías todavía pendientes de aprender y que pudieran ser útiles para cumplir los objetivos del proyecto.

Actualmente, la aplicación está exclusivamente orientada a abordar, dentro de la prevención, el ámbito de las pantallas de visualización de datos [12], es decir, los puestos de teletrabajo que, desgraciadamente, nos hemos visto obligados a aplicar en numerosos entornos de nuestra sociedad, debido a la situación de pandemia que se está viviendo.

El principal objetivo de este proyecto es la realización de una aplicación Android, sencilla de utilizar y que permita al usuario poder llevar a cabo una evaluación de su puesto de teletrabajo, en poco tiempo. Esta evaluación proporciona al usuario las medidas necesarias para conseguir un lugar de trabajo seguro, desde el punto de vista de la PRL, así como guardar dicha evaluación para poder consultarla en un futuro.

Aunque en un principio se pretendía orientar la aplicación al entorno de la empresa, los usuarios finales de la aplicación pueden ser tanto técnicos como operarios de PRL, o cualquier persona interesada en el tema de evaluar y asegurar su puesto de trabajo, incluso que tenga cierto interés en abordar el tema, ya que la aplicación está preparada para que cualquier persona sea capaz de utilizarla, posea o no conocimientos relacionados con los ámbitos de la PRL y la informática.

Es importante mencionar el cumplimiento de algunos puntos de los objetivos de desarrollo sostenible [25] aprobados por la Organización de Naciones Unidas en el año 2015, con la intención de emprender, como sociedad, un nuevo camino con el que mejorar la vida de todos, sin dejar a nadie atrás.

Algunas de las metas que cumple este proyecto, son, la cuarta del objetivo número cuatro (4.4, referente a la educación de calidad) y la octava del objetivo número ocho (8.8, referente al trabajo decente y crecimiento económico).

Para recoger lo anteriormente mencionado, se ha dividido el contenido en diversos apartados, de forma que se empieza tratando el estado del arte actual antes de detallar las fases de análisis, diseño e implementación de la aplicación. También se incluye un apartado de la accesibilidad y usabilidad del sistema, así como la licencia documental aplicada y una serie de conclusiones del proyecto una vez realizado. Se termina con la lista de referencias utilizadas y un manual de usuario (anexos) sobre el funcionamiento de la app.

2. ESTADO DEL ARTE

En este apartado se va a realizar un pequeño estudio de las diferentes aplicaciones relacionadas con la PRL que existen actualmente y que puedan tener cierto parecido con PVD Prevención. Para ello, lo primero es obtener una lista de dichas aplicaciones y añadir una breve descripción de éstas:

 <p>LABORALES Prevención de Riesgos Laborales (Team Fasabe)</p>	<p>Curso online que pretende dar nociones básicas de Prevención de Riesgos Laborales a todas aquellas personas que puedan estar interesadas.</p> <p>Prevención de Riesgos Laborales - Aplicaciones en Google Play</p>
 <p>Test Prevención Riesgo Laboral</p>	<p>Conjunto de tests de Prevención de Riesgos Laborales que incluye una gran cantidad de preguntas orientadas a preparar oposiciones, todas ellas basadas en exámenes oficiales.</p> <p>Test Prevención Riesgo Laboral - Apps on Google Play</p>
 <p>APP Test de Prevención L.P.R.L.</p>	<p>Conjunto de tests de la Ley 31/1995, del 8 de noviembre, de Prevención de Riesgos Laborales, organizado por capítulos e incluyendo un test final.</p> <p>App Test Prevención LPRL - Apps on Google Play</p>
 <p>Gestión Salud y Seguridad en el Trabajo</p>	<p>Lista de cursos, vídeos de youtube y PDF 's organizados referentes a la Prevención de Riesgos Laborales.</p> <p>GESTIÓN SALUD Y SEGURIDAD EN EL TRABAJO - Aplicaciones en Google Play</p>
 <p>Salud Laboral CCOO CYL</p>	<p>Aplicación que incluye un cuestionario para realizar un chequeo de un puesto de trabajo determinado, un apartado de preguntas frecuentes y otro de contacto para resolver las posibles dudas del usuario.</p> <p>Salud Laboral CCOO CYL - Aplicaciones en Google Play</p>


	Servicio de asesoramiento gratuito creado para ayudar al personal trabajador autónomo y a las empresas de hasta de 25 trabajadores a gestionar los riesgos laborales.
	Prevencion10 - Aplicaciones en Google Play

Tabla 01. Listado de aplicaciones existentes relacionadas con la PRL.

Como se puede observar, éstas son aplicaciones relacionadas con la PRL, aunque no todas ellas guardan semejanzas con PVD Prevención, siendo Salud Laboral CCOO CYL y Prevencion10, las más parecidas.

El siguiente paso será construir otra tabla en la que se relacionan estas aplicaciones con una serie de aspectos importantes que debería cumplir la aplicación en cuestión, tanto tecnológicos como técnicos relacionados con la PRL y la información que brinda al usuario.

A continuación, se van a explicar detalladamente todos los aspectos que se van a incluir, de una forma más resumida, en dicha tabla:

- Registro previo: Este apartado hace referencia a la existencia de un registro previo a la utilización de la aplicación por parte del usuario.
- Diseño agradable: La aplicación tiene un diseño que, por lo general, se hace agradable a la vista del usuario, de forma que se sienta atraído a seguir utilizándola.
- Interfaz intuitiva: La interfaz de la aplicación es lo suficientemente intuitiva para que el usuario que la utilice sea consciente, en todo momento, de dónde está y cómo puede acceder a las diferentes pantallas que se le presentan.
- Publicidad: La aplicación contiene publicidad cuando el usuario intenta acceder a determinadas funcionalidades de la misma. Por ejemplo, al intentar acceder a un documento o justo antes de obtener los resultados de un test previamente realizado.
- Información precisa: La aplicación ofrece al usuario la información estrictamente necesaria para conseguir su objetivo añadiendo, en ocasiones, apartados extra que le aporten un mayor valor.
- Resultados adecuados: La mayoría de las aplicaciones relacionadas con la PRL ofrecen al usuario la posibilidad de realizar test a los que asocian unos determinados resultados. Éstos deben adecuarse a las respuestas aportadas por el usuario y ser lo más precisos posible.

	Registro previo	Diseño agradable	Interfaz intuitiva	Publicidad	Información precisa	Resultados adecuados
	No	Sí	Sí	Sí	Sí	No aplica
	No	Sí	Sí	Sí	Sí	Sí
	No	No	No	Sí	Sí	No
	No	Sí	Sí	Sí	Sí	No aplica
	No	Sí	Sí	No	Sí	Sí
	Sí	Sí	Sí	No	Sí	Sí

Tabla 02. Evaluación de diversos aspectos de las aplicaciones relaciones con la PRL.

Como se ha mencionado previamente, las aplicaciones que guardan mayor semejanza con PVD Prevención son Salud Laboral CCOO CYL y Prevencion10, por lo que el principal objetivo será diferenciarse de ambas en la medida de lo posible.

Por un lado, Salud Laboral CCOO CYL ofrece un servicio de tests de prevención similar al de PVD Prevención, pero no le asocia unos resultados como tal, sino que, una vez se

termina de completar un test, brinda al usuario la posibilidad de contactar con los desarrolladores y ellos proporcionen una valoración adecuada.

Por otro lado, PVD Prevención es capaz de ofrecer al usuario unos resultados una vez completado el test, incluso ofreciendo la posibilidad de guardarlos bajo un título (por ejemplo: "Evaluación de mi oficina.") a los que podrá acceder posteriormente en el apartado de "Mis evaluaciones", en vez de volver a realizar el test correspondiente.

El principal competidor de PVD Prevención es Prevencion10. Es con la que más parecido guarda y, en líneas generales, la aplicación que mejor cumple todos los aspectos tratados en la anterior tabla.

Esta aplicación dispone de una opción sin registro, para ingresar en la aplicación como usuario invitado, pero, por otro lado, el acceso a las evaluaciones puede resultar algo tedioso, ya que el usuario debe rellenar unos pequeños formularios acerca de su empresa antes de evaluarla.

Esto abre un frente en el cual PVD Prevención puede ser capaz de competir con Prevencion10 porque, aunque es obligatorio el registro para su utilización, los campos a rellenar son mínimos (únicamente correo y contraseña) y se ofrecen las evaluaciones de puestos de trabajo al usuario de una manera rápida y sencilla, pudiendo acceder a los resultados obtenidos sin necesidad de una navegación profunda por la aplicación y de una forma fácil de interpretar.

3. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN

En primer lugar, se describe la metodología de ingeniería del software que se utiliza para la realización y puesta en marcha de la aplicación de este proyecto.

Su aplicación en el proyecto permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar.



Figura 01. Metodología de ingeniería del software utilizada.

En este caso, se ha decidido utilizar una metodología tradicional, en la que se distinguen las fases de planteamiento análisis, diseño, programación y pruebas, antes de la puesta en marcha de la propia aplicación, tal y como se observa en la figura anterior.

3.1. Análisis

La etapa de análisis en el ciclo de vida [02] del software corresponde al proceso mediante el cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

Dicho esto, se especifican las funciones que se podrán realizar dentro de la aplicación PVD Prevención y bajo qué restricciones, mediante los requisitos funcionales y no funcionales [13].

3.1.1. Requisitos funcionales

RF01. El sistema permite a los usuarios registrarse introduciendo su correo electrónico y contraseña.

RF02. El sistema permite a los usuarios registrarse mediante su cuenta de Google.

RF03. El sistema permite a los usuarios iniciar sesión utilizando su correo electrónico y contraseña.

RF04. El sistema permite a los usuarios cerrar sesión.

RF05. El sistema permite a los usuarios realizar evaluaciones de su puesto de trabajo, marcando las opciones que cumple el mismo.

RF06. El sistema permite a los usuarios obtener los resultados correspondientes a las evaluaciones de los puestos de trabajo que han analizado.

RF07. El sistema permite a los usuarios guardar dichos resultados dentro de la aplicación.

RF08. El sistema permite a los usuarios obtener una lista de todas las evaluaciones que han guardado dentro de la aplicación.

RF09. El sistema permite a los usuarios borrar una evaluación de la lista de evaluaciones de su cuenta en la aplicación.

RF10. El sistema permite a los usuarios descargar en formato PDF una evaluación de la lista de evaluaciones de su cuenta en la aplicación.

RF11. El sistema permite a los usuarios obtener los resultados de una evaluación que ha realizado previamente y ha guardado en la aplicación.

3.1.2. Requisitos no funcionales

RNF01. La aplicación puede ser utilizada desde la mayoría de los dispositivos Android (95% de ellos, aproximadamente).

RNF02. Se almacenan los datos de la aplicación en una base de datos Firebase.

RNF03. Existe un solo tipo de cuenta para usuarios de la aplicación, siendo posible el registro de la misma mediante correo y contraseña o Google.

RNF04. Cuando un usuario intenta registrar una nueva cuenta, se comprueba que los datos introducidos son correctos y no existen ya en la BD.

RNF05. Cuando un usuario se registra, la BD se actualiza automáticamente.

RNF06. Cuando un usuario guarda en su cuenta de la aplicación una evaluación realizada, se actualiza la BD.

RNF07. Cuando un usuario intenta eliminar una evaluación, previamente guardada en su cuenta de la aplicación, se comprueba que existe dicha evaluación.

RNF08. Cuando un usuario elimina una evaluación previamente guardada en su cuenta de la aplicación, se actualiza la BD.

3.1.3. Diagrama de casos de uso

A continuación, se presenta el diagrama de casos de uso, utilizado para especificar la comunicación y el comportamiento del sistema mediante su interacción con los usuarios u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

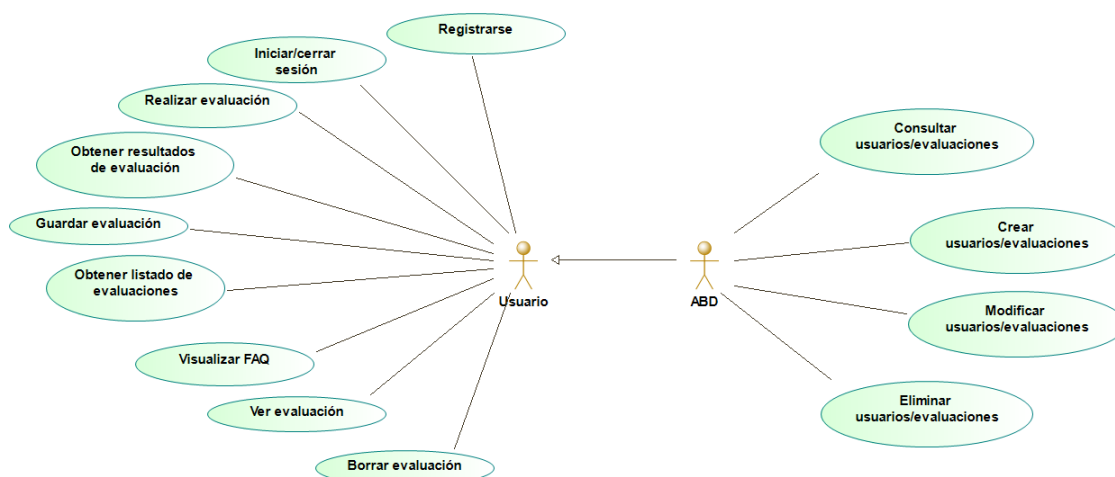


Figura 02. Diagrama de casos de uso de PVD Prevención.

3.2. Diseño

En esta fase se estudian las posibles alternativas de implementación para el sistema que se pretende construir y también se decide la estructura general que tendrá (diseño arquitectónico).

Los modelos que se utilizan en la fase de diseño representan las características del sistema que permitirán una implementación efectiva.

3.2.1. Nombre y logo

El primer nombre que recibió la aplicación fue All Prevention, pero se descartó por no hacer una referencia demasiado concreta al tema abordado (las pantallas de visualización de datos), por lo que finalmente se optó por PVD Prevención.

Además, se decidió obviar la idea de un nombre en inglés, dado que actualmente la aplicación solo está implementada en español.

A continuación, se muestran los diferentes logos de la aplicación, tanto el actual como el primero que tuvo. Los iconos [20] de la aplicación que se muestran en los dispositivos Android son similares a los logos, pero sin las palabras Prevention y Prevención, respectivamente.



Figura 03. Logos de All Prevention y PVD Prevención.
Fuente de elaboración propia.

A continuación, se van a explicar los diagramas Entidad-Relación y de clases de la aplicación, una colección de diagramas que ayudan a visualizar un programa de software.

3.2.2. Diagrama Entidad-Relación

En este apartado se incluye el diagrama de entidad-relación asociado al proyecto PVD Prevención, en el que se muestra la información relativa a la aplicación que se almacena en el sistema gestor de la base de datos (SGBD) de Firebase.

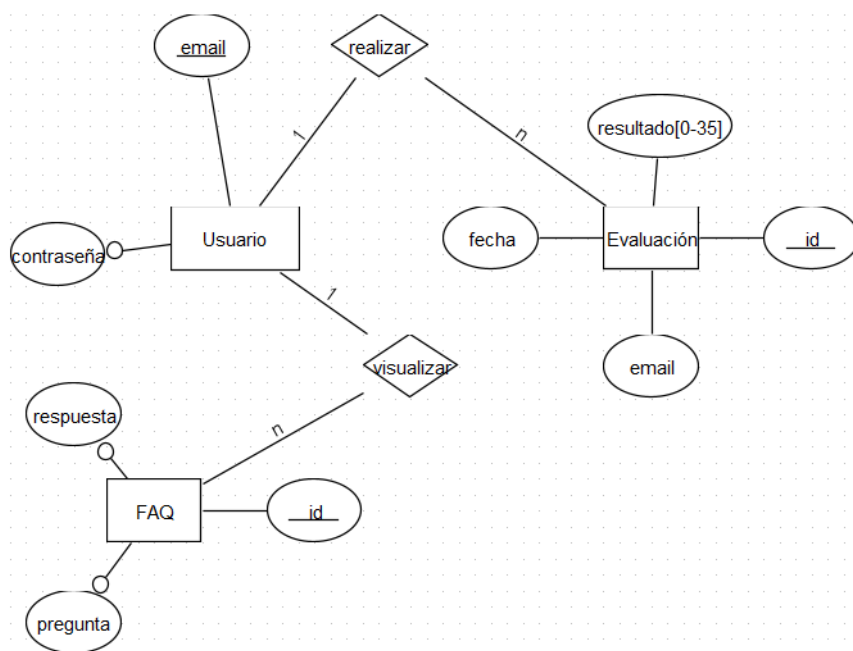


Figura 04. Diagrama E-R de PVD Prevención (DEREditor).

A continuación, se presenta el código autogenerated por el programa DEREditor [10] asociado al diagrama E-R anterior:

-- Generador Pseudocódigo / Lenguaje CRT-AT --

/* Entidad Usuario: */

Usuario (email varchar, contraseña:varchar)

Clave Primaria { email }

Valor No Nulo { contraseña }

/* Entidad Evaluación: */

Evaluación (id varchar, fecha:date, email:varchar, resultado[0-35]:varchar)

Clave Primaria { id }

/* Relacion realizar: */

realizar (email varchar, id varchar)

Clave Primaria { id }

Valor No Nulo { email }

Clave Ajena { email } hace referencia a Usuario

Clave Ajena { id } hace referencia a Evaluación

/* Entidad FAQ: */

FAQ (id integer, pregunta:varchar, respuesta:varchar)

Clave Primaria { id }

Valor No Nulo { pregunta }

Valor No Nulo { respuesta }

/* Relación visualizar: */

visualizar (email varchar, id integer)

Clave Primaria { id }

Valor No Nulo { email }

Clave Ajena { email } hace referencia a Usuario

Clave Ajena { id } hace referencia a FAQ

3.2.3. Diagrama de clases

UML [15] son las siglas de Unified Modeling Language o Lenguaje Unificado de Modelado. Se trata de un estándar para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

El diagrama de clases es una herramienta para comunicar el diseño de un programa orientado a objetos, permitiendo modelar las relaciones entre las entidades. En UML,

una clase es representada por un rectángulo que posee tres divisiones: Nombre de la clase, atributos que tiene y mensajes que entiende.

A continuación, se muestra una imagen del diagrama de clases de PVD Prevención realizado mediante el software Modelio Open Source [17].

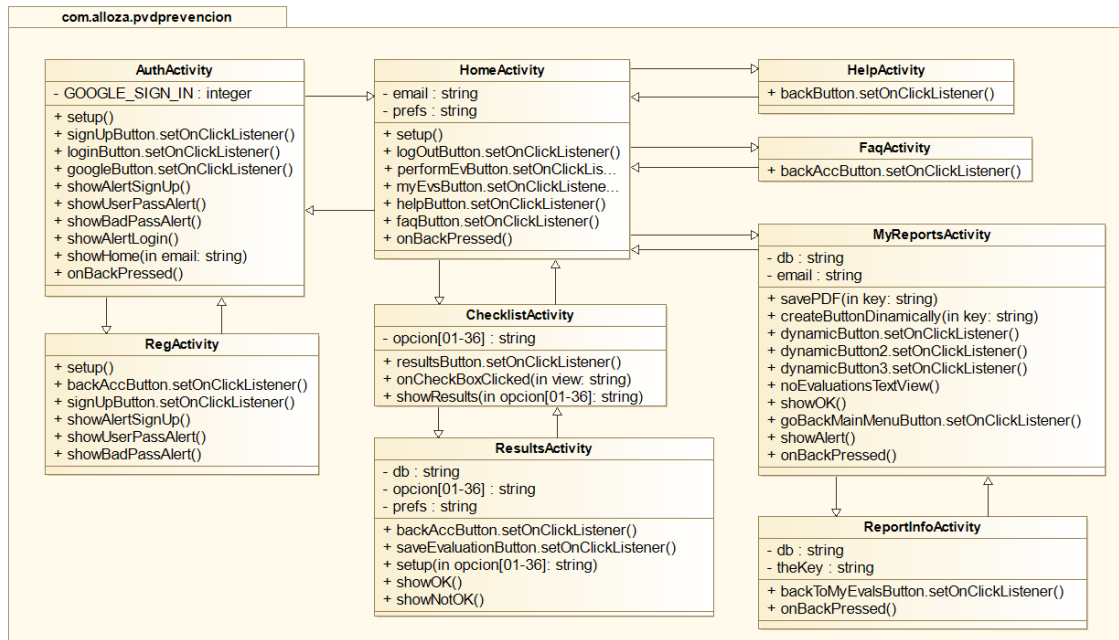


Figura 05. Diagrama de clases de la aplicación generado mediante Modelio Open Source.

3.2.4. Prototipos de interfaz

Dentro de la fase de diseño del proyecto, se ha realizado una serie de prototipos mediante el software Balsamiq Mockup [27].

Antes de la creación del proyecto, se llevan a cabo bocetos o dibujos para tener una imagen visual de cómo va a ser la interfaz de la aplicación.

Esto ayuda a que, una vez se realice la implementación, se pueda crear un proyecto lo más semejante posible a este boceto previamente realizado.

Este proceso de creación de la estructura de una interfaz es denominado wireframing.

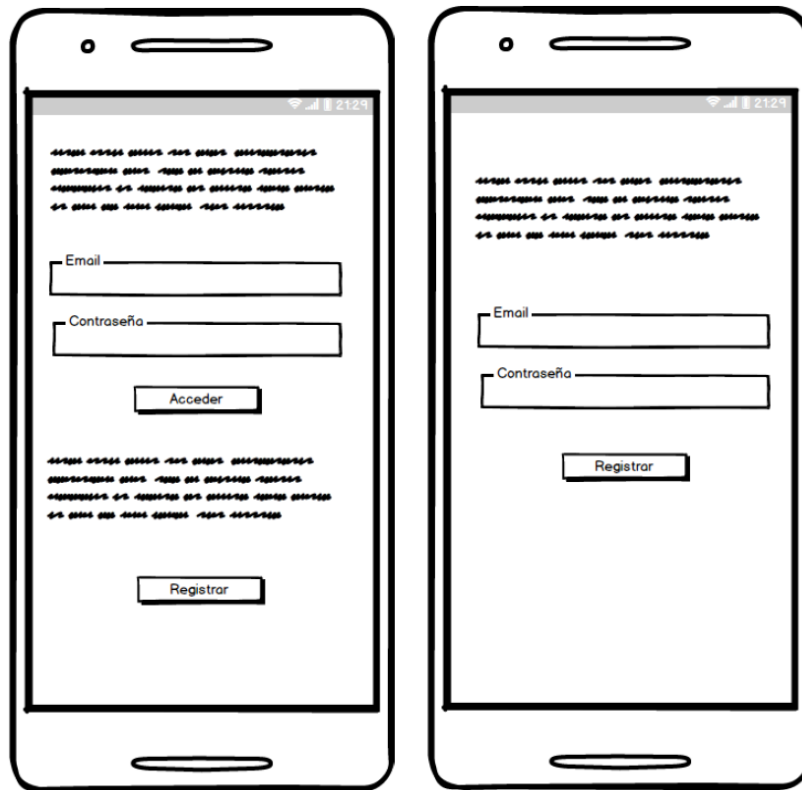


Figura 06. Pantallas de autenticación y registro.

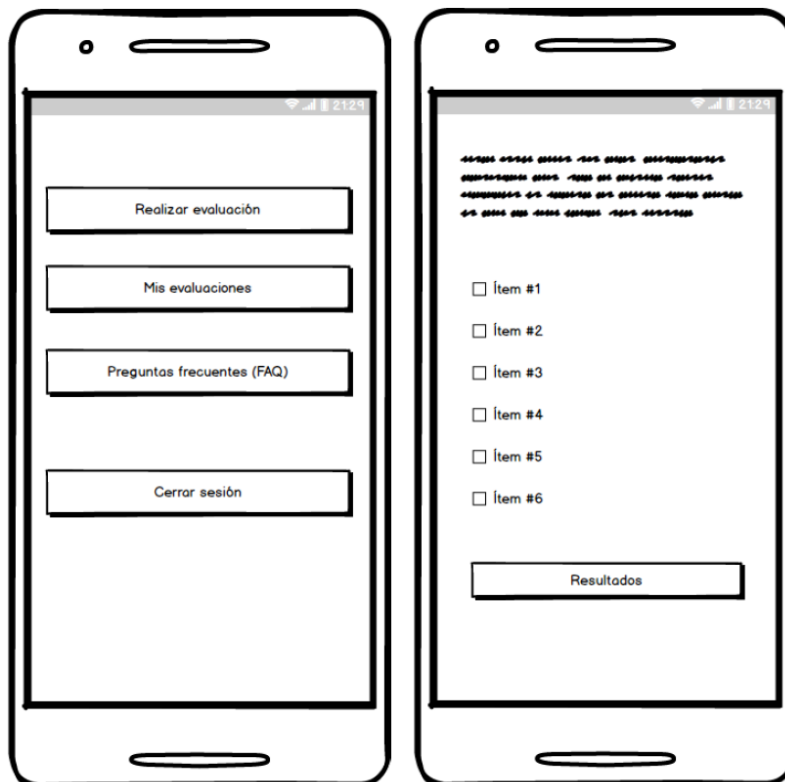


Figura 07. Pantallas de menú principal y realizar evaluación.

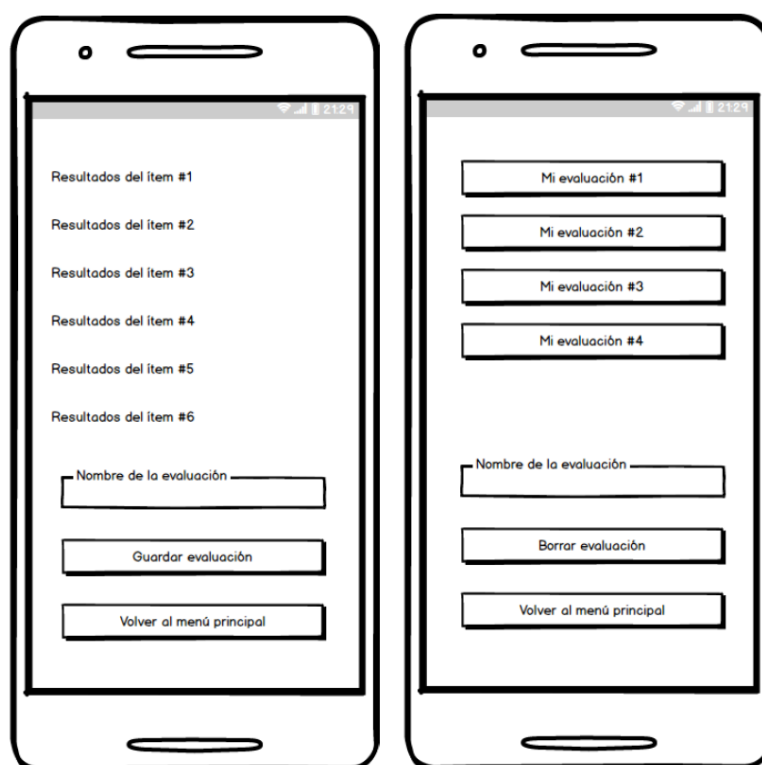


Figura 08. Pantallas de resultados y mis evaluaciones.

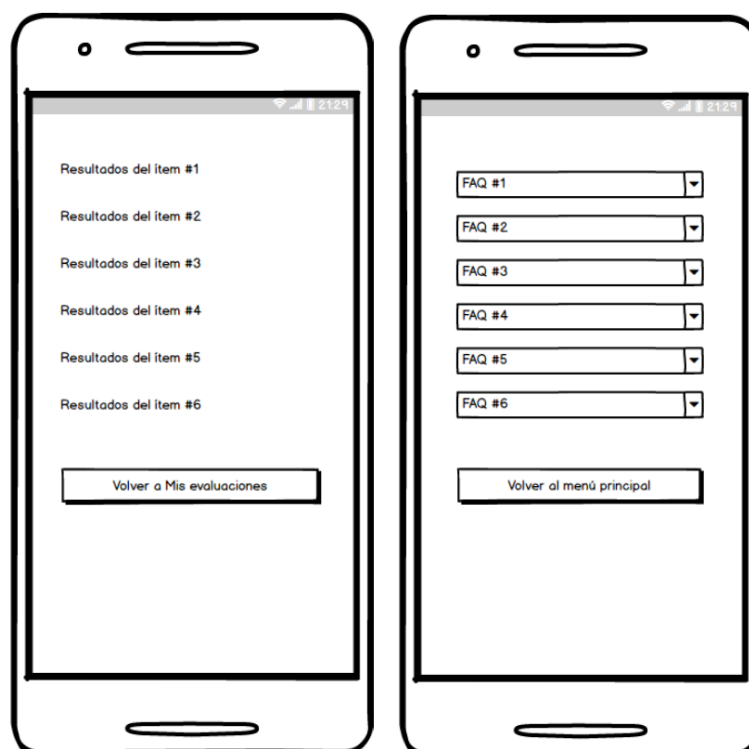


Figura 09. Pantallas de información de mi evaluación y FAQ.

Estos son los prototipos creados inicialmente en la fase de diseño para la interfaz de la aplicación PVD Prevención.

Como se verá posteriormente, en la fase de implementación se realizará una interfaz lo más fiel posible a los bocetos creados.

3.2.5. Arquitectura de la aplicación

La arquitectura de aplicaciones web [26] es un marco que se compone de interacciones y relaciones entre elementos de la aplicación, como interfaces de usuario, sistemas de middleware (software que brinda servicios y funciones comunes a las aplicaciones, además de lo que ofrece el sistema operativo) y bases de datos. Consta de un conjunto de componentes y una descripción de su interacción lógica. La idea de la arquitectura de aplicaciones web está asociada con la idea del navegador del usuario que activa una aplicación que puede ejecutarse en varios sitios web.

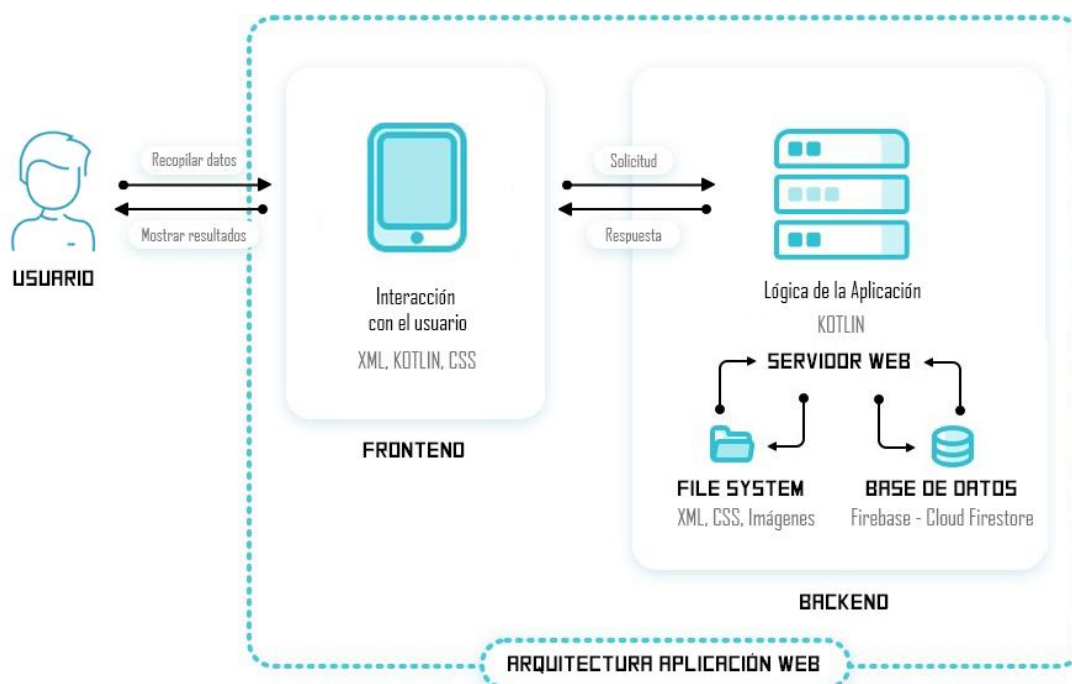


Figura 10. Diagrama de la arquitectura de la aplicación.

3.2.6. Paleta de colores

En este apartado se define la paleta de colores utilizada para la aplicación, de forma que la combinación de los mismos recrea cierta armonía visual al usuario que navega por las diferentes pantallas.

Se trata de un aspecto que hace referencia, dentro del campo de la informática, a la disciplina de la interacción persona-ordenador y, más concretamente, al apartado de usabilidad de la aplicación, que se explica más detalladamente en apartados posteriores de este documento.

Dicho esto, se presentan los colores utilizados en la aplicación, junto a su código hexadecimal:



Figura 11. Paleta de colores de PVD Prevención.

5CA6BF

Color utilizado para las barras de herramientas (Toolbar) y de estado (Status Bar) de la aplicación.

FFFFFF

Color utilizado como fuente del texto de los botones con fondo 003459 y como fondo de aquellos con fuente de texto 00171F.

003459

Color utilizado como fondo de los botones con fuente de texto FFFFFFFF.

00171F

Color utilizado como fuente del texto de los botones con fondo FFFFFFFF y de los textos (TextViews) de la aplicación.

3.2.7. Diseño físico e Integración del SGBD

Para la BD de datos asociada a la aplicación PVD Prevención se ha optado por Cloud Firestore [19], de Firebase.

Se trata de una BD muy sencilla, en la que se almacena la información referente a los usuarios registrados en la aplicación (correo electrónico y contraseña) y las evaluaciones realizadas por los mismos.

A continuación, se detalla el proceso de integración de la BD en el proyecto Android Studio y lenguaje de programación Kotlin.

Un dato que señalar es que el servicio que ofrece Firebase garantiza hasta diez mil operaciones al mes de autenticación en la aplicación gratuitas, por lo que no se debería aplicar coste alguno.

El primer paso es acceder a la web de Firebase (<https://firebase.google.com/>) e iniciar sesión mediante una cuenta de Google.



Figura 12. URL y botón de acceso a la consola en Firebase.

Una vez dentro, se pulsa en *Ir a la consola* [06], situado en la esquina superior derecha y, a continuación, en *Agregar proyecto*.



Figura 13. Botón de agregar proyecto en Firebase.

El siguiente paso es rellenar las tres hojas de formularios de datos que ofrece Firebase, incluyendo el nombre del proyecto, el id autogenerado (es posible modificarlo si se desea) y una serie de datos en caso de haber habilitado la opción de Google Analytics (herramienta que ofrece Firebase proporcionando un conjunto de analíticas sobre la aplicación). Se pincha en *Crear proyecto*.

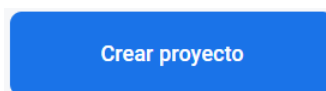


Figura 14. Botón de crear proyecto en Firebase.

Una vez creado el proyecto, se pulsa en *Agregar app* o, en caso de no aparecer este botón (cuando no se tiene todavía ninguna plataforma integrada), en el símbolo de Android.

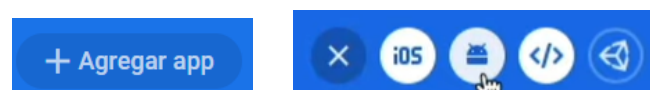


Figura 15. Botones de agregar app y de Android en Firebase.

Para realizar la conexión de la aplicación con la base de datos de Firebase, es necesario disponer de un proyecto de Android Studio.

A continuación, se requieren una serie de datos, algunos de los cuales se obtienen del proyecto previamente creado en Android Studio:

- Nombre del paquete de Android: es posible obtener el nombre de este paquete dentro del Android Manifest del proyecto Android (ejemplo de la línea de código: `package="com.fernando.aplicacion1">`).
- Sobrenombre de la app: es un campo opcional que se puede completar con el nombre que se le desee dar a la aplicación.

Se pulsa en *Registrar app*.

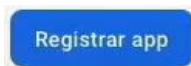


Figura 16. Botón de registrar app en Firebase.

El siguiente paso para realizar es la descarga del fichero JSON y agregar el mismo a la raíz del proyecto Android.



Figura 17. Botón de descarga del fichero JSON de Firebase.

Para ello, dentro de Android Studio se cambia a vista de proyecto (desplegable *Android, Project*), como se observa en las siguientes imágenes, y se añade el fichero JSON en la jerarquía de *app*. Para añadirlo, basta con copiar y pegar el archivo y pulsar *OK*.

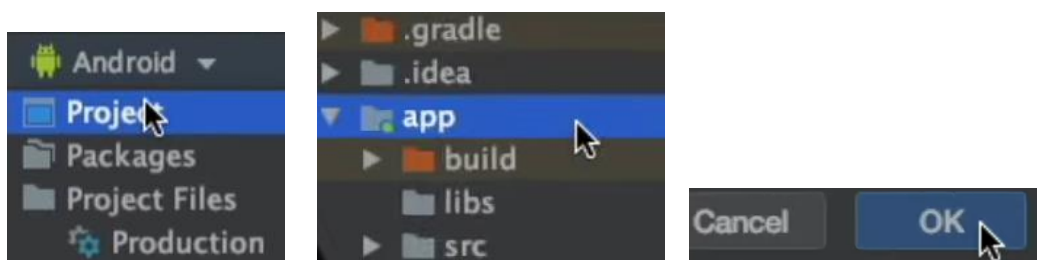


Figura 18. Recorrido de la inclusión del archivo JSON en Android Studio.

Se vuelve al asistente de la consola de Firebase y se pulsa en *Siguiente*.

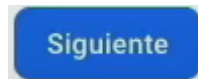


Figura 19. Botón de siguiente en Firebase.

El siguiente paso es añadir la dependencia a la librería de Firebase al proyecto Android. Para ello, se accede al archivo *build.gradle* a nivel de proyecto y se añaden las líneas de código resaltadas en las siguientes imágenes:

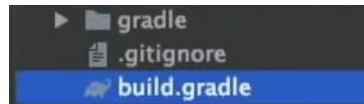


Figura 20. Ubicación del fichero build.gradle (project) del proyecto en Android Studio.

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.3'
    }
}

allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

Figura 21. Líneas de código a incluir en el archivo build.gradle (project) en Android Studio.

Es posible que las líneas de código *google()* ya aparezcan en los repositorios correspondientes, simplemente se comprueba que estén y, en caso negativo, se añaden como en la imagen.

Ahora se debe abrir el fichero de build-gradle, a nivel de app, para añadir las siguientes líneas de código:

```

apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // add the Firebase SDK for Google Analytics
    implementation 'com.google.firebase:firebase-analytics:17.2.2'
    // add SDKs for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}

```

Figura 22. Líneas de código a incluir en el archivo build.gradle (app) en Android Studio.

Una vez añadidas estas líneas de código, se pincha en sincronizar (*Sync now*, en la parte superior derecha) para aplicar los cambios realizados dentro del proyecto de Android Studio.

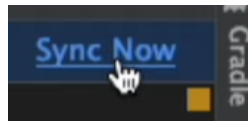


Figura 23. Botón de sincronizar en Android Studio.

Se vuelve a la guía de integración de Firebase y se pulsa en *Siguiente*.

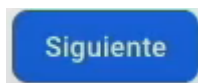


Figura 24. Botón de siguiente en Firebase.

De esta forma se completa la integración de Firebase a la aplicación.



Figura 25. Mensaje de agregación de app exitosa en Firebase.

En caso de querer utilizar la herramienta de Analytics mencionada previamente, es necesario esperar un período de tiempo de hasta veinticuatro horas para poder empezar a visualizar las diferentes gráficas. En este documento se obvia este apartado, dado que no se utilizan analíticas durante la realización del proyecto, por lo que se expone directamente la creación de la base de datos Cloud Firestore [08] necesaria para el almacenamiento de datos de los usuarios y los resultados de sus evaluaciones.

3.3. Implementación

Una vez establecidas las funciones que debe desempeñar la aplicación (fase de análisis) y esté decidida la organización de sus componentes (fase de diseño), inicia la fase de implementación.

Es fundamental haber comprendido el problema que se intenta resolver en el proyecto y haber aplicado los principios básicos de diseño que permitan construir una aplicación de calidad antes de escribir código o crear las tablas de la base de datos.

3.3.1. Tecnologías utilizadas

Para una correcta fase de implementación, se han de seleccionar las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para la aplicación que se pretende llevar a cabo.

Dicho esto, se presenta una tabla con las diferentes tecnologías utilizadas, así como una breve descripción de las mismas:

	Firestore de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móviles. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.
	Android Studio es el entorno de desarrollo integrado oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de código y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan la productividad en el desarrollo de apps para Android.
	Kotlin es un lenguaje de programación pragmático pensado para funcionar con Máquina Virtual de Java (JVM) y Android. Además, puede ser compilado a código fuente de Javascript y se caracteriza por una perfecta combinación de características claramente orientadas a la funcionalidad durante la programación, centrándose en la seguridad, la claridad y la interoperabilidad.

Tabla 03. Tecnologías utilizadas en el proyecto.

3.3.2. Implementación de las actividades

En este apartado se explica en profundidad el conjunto de actividades o pantallas correspondiente a la aplicación, así como los aspectos más relevantes de las mismas. Se puede visualizar el código completo de la aplicación en el repositorio *TFG_ALLOZA*, del usuario *sallozag* de GitHub (https://github.com/sallozag/TFG_ALLOZA.git).

En esta memoria, sólo se mencionan aquellos métodos, funciones, etc, de mayor importancia para la implementación de esta aplicación.

Cada actividad en Android Studio está asociada a un fichero xml en el que se trabaja con elementos de diseño que el usuario percibe en las diferentes pantallas. Aunque es posible trabajar estos ficheros de diseño desde el punto de vista de diseño (pestaña Design) dentro de Android Studio, moviendo y arrastrando elementos por la pantalla, en este proyecto se han realizado todos desde el código (pestaña Code).

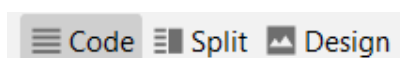


Figura 26. Pestañas de código, diseño y mezcla de ambos de Android Studio.

Actividad de autenticación

Actividad correspondiente al registro y el acceso a la aplicación por parte del usuario. Permite acceder a la aplicación introduciendo los campos de correo y contraseña o ir a la actividad de registro para poder darse de alta.

Incluye una splash screen, es decir, una pantalla de espera que aparece en el momento que abrimos la aplicación mientras ésta se carga.



Figura 27. Splash screen de PVD Prevención.

Por defecto, Android Studio no la proporciona, así que se ha creado una muy sencilla que muestra el logo de PVD Prevención y el de la Escuela Universitaria Politécnica de Teruel.

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@color/white"/>
    <item>
        <bitmap android:src="@drawable/pvd300"
            android:gravity="center"/>
    </item>
</layer-list>
```

Figura 28. Código de la splash screen asociada a la actividad de autenticación.

La splash screen [21] no se crea en la propia actividad de autenticación, sino dentro de la carpeta `res/drawable` en Android Studio, modificando el código predeterminado `selector` por `layer-list`, como se muestra en la figura.

En un principio se introdujo en el código un pequeño evento para posteriormente poder visualizar las gráficas y estadísticas referentes al mismo en la función de Analytics de Firebase, pero finalmente se decidió no utilizar dicha función. Igualmente, se mantiene el código ya que puede resultar de utilidad para futuras versiones de la aplicación.

```
// Eventos para Analytics
val analytics: FirebaseAnalytics = FirebaseAnalytics.getInstance(context, this)
val bundle = Bundle()
bundle.putString("message", "Integración de Firebase completa.")
analytics.logEvent(name: "InitScreen", bundle)
```

Figura 29. Código para los eventos de Firebase Analytics en la actividad de autenticación.

El mayor peso de la actividad de autenticación recae, sin duda alguna, sobre el código asociado al botón de Acceder (rellenando previamente los campos de correo y contraseña) de la aplicación.

```

loginButton.setOnClickListener { it: View!
    if (emailEditText.text.isNotEmpty() && passwordEditText.text.isNotEmpty()) {
        if (passwordEditText.text.length >= 6){

            FirebaseAuth.getInstance().signInWithEmailAndPassword(
                emailEditText.text.toString(),
                passwordEditText.text.toString()
            ).addOnCompleteListener { it: Task<AuthResult!>

                if (it.isSuccessful) {
                    showHome( email: it.result?.user?.email ?: "", ProviderType.BASIC)
                } else {
                    showAlertSignUp()
                }
            }
        } else {
            showBadPassAlert()
        }
    } else {
        showUserPassAlert()
    }
}
}

```

Figura 30. Código del botón Acceder de la aplicación (actividad de autenticación).

Para ello, se realizan las comprobaciones necesarias para asegurar que se puede realizar el acceso a la aplicación (correo electrónico existente, tamaño mínimo de contraseña...) correctamente y, en caso negativo, que el usuario obtenga un error por pantalla para poder solucionarlo.

Las líneas de código asociadas a Firebase, se consiguen incluyendo algunas implementaciones en el fichero build.gradle a nivel de aplicación en Android Studio, como se observa en la siguiente figura.

```

implementation platform('com.google.firebase:firebase-bom:29.0.3')
implementation 'com.google.firebase:firebase-analytics-ktx'
implementation 'com.google.firebase:firebase-auth-ktx'

```

Figura 31. Fragmento de código del fichero build.gradle a nivel de aplicación.

Estas implementaciones corresponden a la propia conexión del proyecto de Android Studio con el Firebase, así como a los servicios de analítica y autenticación de éste, y es posible que, cada cierto tiempo, sea conveniente actualizar su versión (29.0.3, en este caso) a una más actual.

La idea original fue introducir diferentes métodos de acceso a la aplicación, además del básico mediante usuario y contraseña, como Google, Facebook y otras redes sociales.

```

googleButton.setOnClickListener { it: View!

    //Configuración
    val googleConf = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
        .requestIdToken(getString(R.string.default_web_client_id))
        .requestEmail()
        .build()

    val googleClient = GoogleSignIn.getClient(activity, googleConf)
    googleClient.signOut()

    startActivityForResult(googleClient.signInIntent, GOOGLE_SIGN_IN)

}

```

Figura 32. Código del botón Acceder mediante Google de la aplicación (actividad de autenticación).

La versión actual de la aplicación incluye el método de Google, de forma que se puede llevar a cabo correctamente desde el propio dispositivo de pruebas y obteniendo un error relacionado con la seguridad SHA-1 en el resto de los dispositivos utilizados, como se puede observar en el código de la anterior figura (token `default_web_client_id`, en color rojo).

Se ha decidido mantener el botón de acceso mediante Google para la exposición de este proyecto, quedando como trabajo futuro el correcto funcionamiento del mismo en el resto de los dispositivos.

Para el fichero de layout (diseño) asociado a la actividad de autenticación, se han utilizado elementos básicos de diseño (Button o botones, EditText o cajas de texto editable, TextView o texto plano, etc) de forma estática, por lo que no se entra en mayor detalle, pudiendo visualizar el código completo de la aplicación.

Actividad de registro

Actividad correspondiente al registro en la aplicación por parte del usuario. Permite al usuario darse de alta en la aplicación introduciendo los campos de correo electrónico y contraseña.

```
signUpButton.setOnClickListener { it: View!
    if (emailEditText.text.isNotEmpty() && passwordEditText.text.isNotEmpty()) {
        if (passwordEditText.text.length >= 6) {

            FirebaseAuth.getInstance().createUserWithEmailAndPassword(
                emailEditText.text.toString(),
                passwordEditText.text.toString()
            ).addOnCompleteListener { it: Task<AuthResult!>

                if (it.isSuccessful) {
                    val text = "Su cuenta ha sido registrada con éxito."
                    val duration = Toast.LENGTH_LONG

                    val toast = Toast.makeText(applicationContext, text, duration)
                    toast.show()
                    val intent = Intent( packageContext: this, AuthActivity::class.java)
                    startActivity(intent)
                } else {
                    showAlertSignUp()
                }
            }
        } else {
            showBadPassAlert()
        }
    } else {
        showUserPassAlert()
    }
}
```

Figura 33. Código del botón Acceder de la aplicación (actividad de registro).

Lo más relevante de la actividad de registro es el código asociado al botón de Registrar, para el que previamente el usuario que quiere darse de alta en la aplicación debe haber introducido correctamente los campos de usuario y contraseña. Si no es el caso, como se puede observar en el código y de forma similar a como ocurre con el botón de Acceder, el usuario recibe por pantalla un mensaje con el error producido para que pueda darle solución (como, por ejemplo, que la contraseña introducida debe tener un mínimo de seis caracteres).

Una vez el usuario se ha registrado con éxito, aparece en pantalla un toast message, es decir, un mensaje que informa al usuario de su correcto registro en este caso, y desaparece al cabo de unos pocos segundos.

Todo el código asociado a las alertas (showUserPassAlert, por ejemplo) corresponde a simples dialogs (diálogos), mensajes para informar al usuario que se cierran pulsando en el botón de OK o Aceptar que los acompañan.

En cuanto al fichero de layout (diseño) asociado a esta actividad, de la misma forma que para la autenticación, se han utilizado elementos básicos de diseño de forma estática.

Actividad del menú principal

Actividad correspondiente al menú principal que se le ofrece al usuario en cuanto accede a la aplicación. Dispone de los botones para realizar una evaluación, visualizar las existentes guardadas en la app, cerrar la sesión actual y un apartado de preguntas frecuentes.

Además, en la esquina superior derecha hay un botón de ayuda por si el usuario tiene dudas sobre cómo navegar por las diferentes pantallas (actividades) de PVD Prevención.

Se trata de una de las actividades más sencillas de la aplicación, ya que el código de los diferentes botones redirige al usuario a la siguiente pantalla.

```
faqButton.setOnClickListener { it: View!  
  
    val intent = Intent( packageContext: this,FAQActivity::class.java)  
    startActivity(intent)  
  
}
```

Figura 34. Código del botón de Preguntas frecuentes (FAQ) del menú principal.

En la figura se muestra el código del botón [01] de preguntas frecuentes a modo de ejemplo, ya que el resto de los botones son casi idénticos.

El layout asociado a esta actividad vuelve a hacer uso de los elementos más básicos de diseño de forma estática, esta vez incluyendo las vistas de imagen (ImageView) para incluir algunas imágenes, como por ejemplo el logo de la aplicación.

Actividad de ayuda

Actividad que ofrece al usuario información sobre la aplicación, sus diferentes pantallas y cómo moverse entre ellas.

Al tratarse de una actividad que única y exclusivamente muestra información por pantalla al usuario (TextView), el código asociado corresponde al botón para volver al menú principal.

```
title = "Ayuda"  
  
backButton.setOnClickListener { it: View!  
  
    val intent = Intent( packageContext: this,HomeActivity::class.java)  
    startActivity(intent)  
  
}
```

Figura 35. Código del botón de volver al menú principal (actividad de ayuda).

Para comentar algún aspecto extra de esta actividad, se ha incluido en la anterior figura el código correspondiente al título que se le muestra al usuario en pantalla en cada una de las diferentes actividades (ayuda, en este caso).

En el layout asociado a esta actividad, aparece el elemento de vista (View), una línea utilizada como separador, para mostrar al usuario la información de una forma que le resulte más agradable a la vista.

Actividad de realización de evaluación

Actividad que incluye el listado de todos los aspectos pertenecientes a la PRL que se van a evaluar en la aplicación, junto con los botones de visualizar los resultados obtenidos y de volver al menú principal.

El código referente a esta actividad hace referencia casi exclusiva a la checklist [24] (lista de verificación) y a la información que será almacenada en la BD de la app según el usuario marque, o no, cada una de las casillas de verificación [05] asociadas a un aspecto determinado de la PRL.

```
fun onCheckboxClicked(view: View) {  
  
    if (view is CheckBox) {  
        val checked: Boolean = view.isChecked  
  
        when (view.id) {  
            R.id.checkbox1 -> {  
                opcion1 = if (checked) {  
                    getString(R.string.checkbox1OK)  
                } else {  
                    getString(R.string.checkbox1NOTOK)  
                }  
            }  
            R.id.checkbox2 -> {  
                opcion2 = if (checked) {  
                    "No es necesario aplicar medidas preventivas."  
                } else {  
                    "Debido a la variabilidad de los usuarios potenciales y ..."  
                }  
            }  
        }  
    }  
}
```

Figura 36. Fragmento de código de la lista de verificación (actividad de realización de evaluaciones).

En la anterior figura se observa la función asociada a la checklist, de forma que la aplicación guardará un resultado u otro dependiendo de si el usuario marca o no la casilla de verificación.

Si el usuario marca una casilla, no será necesario aplicar ninguna medida preventiva, mientras que si la deja vacía se aplicará una medida preventiva asociada al aspecto tratado en esa casilla de verificación.

Las variables checkboxOK y checkboxNOTOK son una serie de cadenas de caracteres (traducidas a texto plano en la app) almacenadas en el fichero string.xml de Android Studio.

El fichero de layout asociado a esta actividad incluye el elemento casilla de verificación o Checkbox como novedad, así como algunos de los elementos más básicos de diseño ya vistos en las anteriores actividades.

Actividad de resultados

Actividad que ofrece al usuario los resultados (en formato de texto plano) de la evaluación que acaba de realizar, con las medidas necesarias para aplicar sobre el puesto de trabajo analizado.

Además, se le ofrece la posibilidad de guardar dichos resultados para poder visualizarlos en un futuro.

Esta actividad requiere de la interacción con la base de datos de Firebase, por lo que es necesario añadir la línea de código de la siguiente figura dentro de la actividad actual.

```
class ResultsActivity : AppCompatActivity() {  
  
    private val db = FirebaseFirestore.getInstance()
```

Figura 37. Línea de código para interactuar con la BD (actividad de resultados).

En esta actividad se lleva a cabo la recuperación de las variables obtenidas a través de la checklist de la actividad anterior, así como su almacenamiento en un fichero de preferencias dentro de Android Studio para su posterior utilización a lo largo de esta actividad.

```
val prefs = getSharedPreferences("com.alloza.allprevention.PREFERENCE_FILE_KEY2", Context.MODE_PRIVATE).edit()  
prefs.putString("opcion1", opcion1)  
prefs.putString("opcion2", opcion2)
```

Figura 38. Fragmento de código de guardado de variables en preferencias (actividad de resultados).

Lo más destacable de la actividad de resultados es el funcionamiento del botón de Guardar evaluación, dónde se realiza la interacción previamente mencionada con la BD. Como se puede observar en la siguiente figura, en la colección de “informes” (correspondiente a las evaluaciones) de la BD, en su subcolección titulada con el nombre que proporciona el usuario al informe o evaluación en cuestión, se guarda una serie de campos que, más tarde, serán recuperados en otras actividades para volver a ser mostrados por pantalla cuando se soliciten.

Como trabajo futuro, tal y como se propone más adelante en este documento, se pretende implementar un botón de búsqueda de evaluaciones, de forma que resulta interesante guardar la fecha de creación (o guardado) de una evaluación en la BD para usarla como posible filtro de búsqueda y, aunque actualmente no se le da uso a esta información, queda almacenada en la base de datos.

```

saveEvaluationButton.setOnClickListener { it: View!

    if (nombreInforme.text.isNotEmpty()) {

        val email: String? = FirebaseAuth.getInstance().currentUser?.email
        val fecha: String? = SimpleDateFormat( pattern: "dd/MM/yyyy", Locale.getDefault())
            .format(System.currentTimeMillis())

        db.collection( collectionPath: "informes").document(nombreInforme.text.toString()).set(
            hashMapOf("id" to nombreInforme.text.toString(),
                "email" to email,"fecha" to fecha, "opcion01" to opcion1, "opcion02" to opcion2,
                "opcion03" to opcion3, "opcion04" to opcion4, "opcion05" to opcion5,
                "opcion06" to opcion6, "opcion07" to opcion7, "opcion08" to opcion8,
                "opcion09" to opcion9, "opcion10" to opcion10, "opcion11" to opcion11,
                "opcion12" to opcion12, "opcion13" to opcion13, "opcion14" to opcion14,
                "opcion15" to opcion15, "opcion16" to opcion16, "opcion17" to opcion17,
                "opcion18" to opcion18, "opcion19" to opcion19, "opcion20" to opcion20,
                "opcion21" to opcion21, "opcion22" to opcion22, "opcion23" to opcion23,
                "opcion24" to opcion24, "opcion25" to opcion25, "opcion26" to opcion26,
                "opcion27" to opcion27, "opcion28" to opcion28, "opcion29" to opcion29,
                "opcion30" to opcion30, "opcion31" to opcion31, "opcion32" to opcion32,
                "opcion33" to opcion33, "opcion34" to opcion34, "opcion35" to opcion35,
                "opcion36" to opcion36)
            )
        showOK()
    } else {
        showNotOK()
    }
}

```

Figura 39. Código del botón de Guardar evaluación (actividad de resultados).

Las funciones showOK y showNotOK son simples dialogs (diálogos), mensajes para informar al usuario que se cierran pulsando en el botón de OK o Aceptar que los acompañan.

El layout asociado a la actividad de resultados consta de los elementos básicos de diseño explicados en algunas de las actividades anteriores, como pueden ser View, TextView, EditText y Button.

Actividad de visualización de evaluaciones

Actividad que muestra al usuario un listado de las evaluaciones que ha guardado en la aplicación, junto a sus correspondientes botones para borrarlas o descargarlas.

De la misma forma que la actividad anterior, incluye la línea de código referente a interacciones con la BD de Firebase.

```

private val db = FirebaseFirestore.getInstance()

@RequiresApi(Build.VERSION_CODES.R)
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_my_reports)

    if (!Environment.isExternalStorageManager()) {
        val intent = Intent(Settings.ACTION_MANAGE_ALL_FILES_ACCESS_PERMISSION)
        startActivity(intent)
    }
}

```

Figura 40. Fragmento de código referente a los permisos (actividad de visualización de evaluaciones).

El fragmento de código de la figura corresponde a la selección de permisos del usuario cuando pulsa por primera vez el botón de Mis evaluaciones, y se le redirige a la pantalla de acceso a todos los archivos de su dispositivo. Una vez se le da permisos, no se vuelve a entrar en este condicional if.

```
db.collection( collectionPath: "informes").whereEqualTo( field: "email", email).get().addOnCompleteListener { task ->

    if (task.isSuccessful) {

        val list: MutableList<String> = ArrayList()
        var key: String
        var numEvals = 0

        for (document in task.result) {

            list.add(document.id)
            key = document.id
            createButtonDynamically(key)
            numEvals += 1
        }

        if (numEvals == 0){
            noEvaluationsTextView()
        }

        goBackMainMenuButton()
        Log.d(TAG, list.toString())
    } else {

        Log.d(TAG, msg: "Error a la hora de encontrar las evaluaciones. ", task.exception)
        showAlert()
    }
}
}
```

Figura 41. Código de la función onCreate (actividad de visualización de evaluaciones).

El código correspondiente a la función onCreate recorre la colección de informes (evaluaciones) de la base de datos, para mostrar todos aquellos correspondientes a la cuenta en PVD Prevención del usuario en cuestión.

En caso de no existir todavía ninguna evaluación, la aplicación mostrará un mensaje para alertar al usuario, de forma que pueda realizar y guardar alguna para, posteriormente, poder visualizarla en este apartado, así como borrarla o descargarla.

```
private fun savePDF(key: String) {

    val mDoc = Document()
    val mFileName = key
    val mFilePath = Environment.getExternalStoragePublicDirectory(Environment
        .DIRECTORY_DOWNLOADS).absoluteFile.toString() + "/" + mFileName + ".pdf"

    try {
        PdfWriter.getInstance(mDoc, FileOutputStream(mFilePath))

        db.collection( collectionPath: "informes").document(key).get().addOnSuccessListener { it: DocumentSnapshot!
            mDoc.open()
            mDoc.add(Paragraph( string: "EVALUACIÓN DE PREVENCIÓN DE RIESGOS LABORALES EN" +
                " ENTORNO DE TRABAJO CON PANTALLAS DE VISUALIZACIÓN: $key"))
            mDoc.add(Paragraph( string: "  "))
            mDoc.add(Paragraph(it.get("opcion01") as String?))
            mDoc.add(Paragraph( string: "  "))
            mDoc.add(Paragraph(it.get("opcion02") as String?))
        }
    }
```

Figura 42. Fragmento de código de la función savePDF (actividad de visualización de evaluaciones).

El fragmento de código de la función savePDF [04] muestra la creación de un documento PDF con la información almacenada en la colección del informe (evaluación) cuyo identificador es el seleccionado por el usuario al pulsar el botón de descarga asociado al mismo.

El mFilePath corresponde al directorio donde la aplicación descarga el documento PDF asociado a la evaluación en cuestión.

Dentro del try, se incluyen las líneas de código con la información correspondiente a la evaluación, desde la cabecera del documento de un entorno de trabajo que incluye pantallas de visualización, hasta las medidas preventivas asociadas a los problemas que presenta el mismo y las fuentes de información de PVD Prevención para concluir el documento.

A continuación, se muestra una figura con el código de los botones para visualizar una evaluación (dynamicButton), borrarla (dynamicButton2) y descargarla (dynamicButton3).

```
dynamicButton.setOnClickListener { it: View!
    val intent = Intent( packageContext: this, ReportInfoActivity::class.java)
    intent.putExtra( name: "clave", key)
    startActivity(intent)
}

dynamicButton2.setOnClickListener { it: View!

    val builder = AlertDialog.Builder( context: this)
    builder.setMessage("¿Está seguro que quiere borrar la evaluación $key?")
    builder.setPositiveButton( text: "Aceptar", listener: null)
    builder.setPositiveButton( text: "Aceptar") { dialog, which ->
        db.collection( collectionPath: "informes").document(key).delete()
        showOK(key)
    }
    builder.setNegativeButton( text: "Cancelar", listener: null)
    val dialog: AlertDialog = builder.create()
    dialog.show()
}

dynamicButton3.setOnClickListener { it: View!
    savePDF(key)
}
```

Figura 43. Código correspondiente a los botones de visualizar, borrar y descargar evaluación (actividad de visualización de evaluaciones).

El primer botón dirige al usuario a la pantalla de información de la evaluación cuyo identificador corresponde con la key (clave).

El segundo, muestra un mensaje (con opciones de aceptar y cancelar) de confirmación de borrado de la evaluación seleccionada.

En caso afirmativo (aceptar), se accede a la BD para borrar la subcolección correspondiente a dicho identificador (key) dentro de la colección de informes (evaluaciones) y se lo notifica al usuario por pantalla.

En caso negativo (cancelar), no ocurre nada.

El tercer botón llama a la función savePDF [14], explicada anteriormente, de forma que se descarga la evaluación seleccionada como archivo en formato PDF en el directorio Descargas del dispositivo del usuario, notificándose mediante un mensaje por pantalla.

El apartado más interesante de la actividad de visualización de evaluaciones corresponde a la creación de elementos de diseño (botones, en este caso) de forma dinámica, de forma que el layout correspondiente a esta actividad solamente incluye el LinearLayout (estructura dónde colocar los diferentes elementos de diseño de la actividad) y un par de textos planos (TextView).

Esta decisión se debe a que el número de elementos (evaluaciones) que contiene la colección de informes de un determinado usuario es desconocido hasta el momento en que se recorre, es decir, una vez se accede a la actividad de visualización de evaluaciones.

En este momento, se crea un listado de botones dinámico [16] en el que, a cada evaluación encontrada en el recorrido de la colección de la BD, se le asigna una tripla formada por los dynamicButton mencionados previamente, de visualización, borrado y descarga de la evaluación pertinente, respectivamente.

Para insertar correctamente la tripla de botones, se debe crear un LinearLayout horizontal previamente, también de forma dinámica (uno para cada tripla, y una tripla para cada evaluación existente almacenada en la BD de la app), así como ajustarlo de forma que se adapte a la dimensión de la pantalla del dispositivo, de la misma forma que se haría mediante xml, pero en este caso desde el código Kotlin de la actividad.

Actividad de información de la evaluación

Actividad que ofrece al usuario los resultados (en formato de texto plano) de la evaluación que previamente ha realizado y guardado, con las medidas necesarias para aplicar sobre el puesto de trabajo analizado.

Se trata de una actividad sencilla cuya única finalidad es mostrar por pantalla, mediante texto plano, la información almacenada en la BD de la evaluación seleccionada por el usuario.

```

private val db = FirebaseFirestore.getInstance()

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_report_info)

    val theKey = intent.getStringExtra( name: "clave")
    title = theKey

    if (theKey != null) {
        db.collection( collectionPath: "informes").document(theKey).get().addOnSuccessListener { it: DocumentSnapshot!
            medidaPrev1TextView.text = it.get("opcion01") as String?
            medidaPrev2TextView.text = it.get("opcion02") as String?
        }
    }
}

```

Figura 44. Fragmento de código de la función onCreate (actividad de información de la evaluación).

Para ello, se rescata el identificador con el nombre de la evaluación mediante la variable theKey y se accede a su colección de informes almacenados en la BD.

De esta forma, se obtienen las medidas preventivas asociadas a cada una de las casillas de verificación de la checklist (opcion01, opcion02, etc) y se muestran al usuario por pantalla como texto plano.

El layout asociado a esta actividad no incluye nuevos elementos de diseño, solo algunos de los básicos mencionados en las actividades anteriores, como TextView y Button.

Actividad de preguntas frecuentes (FAQ)

Actividad que muestra al usuario un listado (en formato de texto plano, pero expandible) de preguntas frecuentes que hacen referencia al ámbito de pantallas de visualización de datos y que no se recogen en los aspectos de la checklist de la aplicación.

Es una actividad sencilla, que muestra la información de las FAQ [11] rescatando el valor de variables recogidas en el fichero strings.xml de la app en Android Studio, por lo que no es necesario interactuar con la BD.

Un aspecto relevante, de cara al layout asociado, es la inclusión de una librería en el fichero build.gradle a nivel de app, para tener acceso a la implementación de un nuevo elemento de diseño: Expandable TextView [23] (texto plano expandible).

```

implementation 'com.ms-square:expandableTextView:0.1.4'

```

Figura 45. Fragmento de código añadido al fichero build.gradle a nivel de app asociado a la pantalla de preguntas frecuentes.

Este nuevo elemento muestra la pregunta frecuente junto a una fecha que, al pulsar en ella (o en el propio texto de la pregunta) se expande para mostrar la respuesta relacionada a la misma.

```

class FaqActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_faq)

        title = "Preguntas frecuentes (FAQ)"

        //getText instead of getString to apply bold, cursive filters...
        expand_text_view.text = getText(R.string.faqP1yR1)
        expand_text_view2.text = "¿Cómo deben ser las pausas y descansos para el personal..."
    }
}

```

Figura 46. Fragmento de código de la función onCreate (actividad de preguntas frecuentes).

La función termina con el código referente al botón de volver al menú principal, implementado de la misma forma que en actividades anteriores.

```

override fun onBackPressed() {

}

```

Figura 47. Código de la sobreescritura de la función onBackPressed.

Para finalizar, se ha añadido en varias de las actividades mencionadas en este documento la sobreescritura de la función onBackPressed [22], que corresponde a la función de volver a la anterior actividad cuando el usuario pulsa el botón de atrás de su dispositivo.

Como se puede observar en la anterior figura, la función no incluye código, de forma que el botón de atrás del dispositivo no redirige al usuario a la anterior actividad. Para navegar debe utilizar los botones proporcionados por la misma aplicación.

Esta decisión se ha tomado para evitar la posibilidad de que se produzcan determinados reenvíos de formularios, dando lugar a situaciones no deseadas. Por ejemplo, un usuario puede borrar una evaluación y pulsar el botón de volver hacia atrás de su dispositivo. Si no se sobreescribe la función para que dicho botón no haga nada, el usuario observará como se recarga la actividad de visualización de evaluaciones y volverá a aparecer en ella la evaluación que acaba de eliminar.

3.3.3. Implementación de la BD

En esta sección de la implementación se explica en detalle la creación de la base de datos de Firebase. Como se ha explicado en apartados previos, se ha decidido por una base de datos Cloud Firestore [07] (siendo un modelo de BD muy parecido a Realtime Database, segunda opción que ofrece Firebase).

Los pasos por seguir para su creación son los siguientes:

1. Se abre la consola de Firebase (<https://console.firebase.google.com/>).
2. Se pulsa en Firestore Database y en Crear base de datos.
3. Se elige entre modo de producción y modo de prueba.

El modo de producción limita por defecto el número de operaciones de escritura que se pueden realizar, mientras que el de prueba habilita todas las operaciones durante treinta días para poder llevar a cabo un proceso completo de desarrollo de la aplicación. Para este proyecto se ha elegido el modo de prueba.

4. Se crean automáticamente las reglas asociadas a la BD (pueden modificarse en cualquier momento).

Estas reglas de seguridad de acceso a la BD se pueden modificar, una vez creada la BD, pulsando en la pestaña de Reglas, para que, por ejemplo, sólo un tipo de usuario pueda realizar determinadas operaciones.

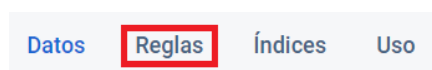


Figura 48. Pestaña Reglas de la BD.

Además, es posible modificar los treinta días iniciales que ofrece el modo de prueba, siempre y cuando se sigan cumpliendo los requisitos en cuanto al número de operaciones soportada por la BD sin necesidad de pagar por el servicio.

```
request.time < timestamp.date(2022, 7, 25);
```

Figura 49. Regla referente a la duración del modo de prueba de la BD.

5. Se selecciona la región en la que se ubica la BD.

La región hace referencia al punto físico dónde se va a encontrar el servidor de la BD. Para este proyecto, se ha seleccionado Europa. En la propia documentación de Firebase se puede comprobar este apartado:

<https://firebase.google.com/docs/projects/locations>

Una vez completados estos sencillos pasos, ya se ha creado una base de datos vacía, a la que se puede acceder pulsando en la pestaña Datos.

En este punto, es posible interactuar con la BD directamente desde la consola de Firebase, de forma que se pueden crear colecciones, documentos y campos, pero en este proyecto se realizan todas las operaciones desde el código.

A continuación, se muestra una figura en la que se puede observar la información que almacena actualmente la BD de PVD Prevención.

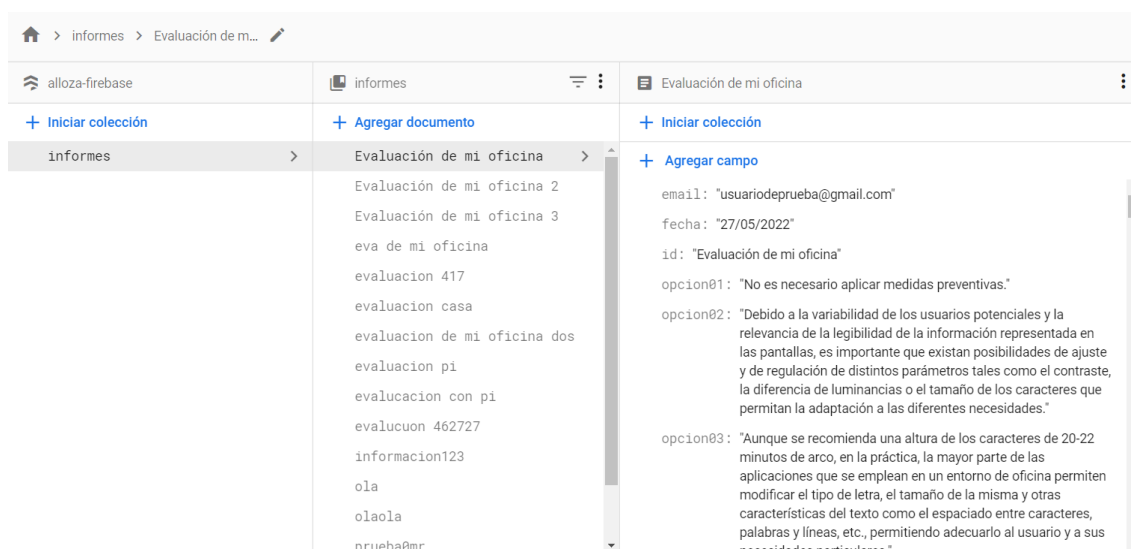


Figura 50. Información almacenada en la BD de PVD Prevención.

La colección de informes hace referencia al conjunto de evaluaciones que se realizan dentro de la aplicación por parte de los usuarios. Cada una de ellas tiene un documento con el nombre (o título) que los usuarios le han dado y una serie de campos que se comentan a continuación:

- email: hace referencia al correo electrónico del usuario al que pertenece la evaluación seleccionada.
- fecha: información que no se trata actualmente, pero que queda implementada para el trabajo futuro de incluir una búsqueda de evaluación utilizando este parámetro como posible filtro.
- id: identificador que recoge el nombre o título de la evaluación a la que está asociada. No es necesario incluirlo, pero a lo largo de la realización de la programación del documento ha sido de ayuda disponer de este parámetro.
- opcionXX: medida preventiva asociada a cada una de las casillas de verificación de la checklist que previamente ha realizado un determinado usuario. Parámetro almacenado para su posterior visualización en los resultados, evaluaciones guardadas y descargas de archivos PDF.

En un principio se pensó en la implementación de la actual colección de informes (evaluaciones) como subcolección de una colección principal de usuarios (con sus atributos correo y contraseña encriptada), pero debido al apartado exclusivo para las autenticaciones de usuarios que se registran en la app que ofrece Firebase, se decidió incluir los usuarios solamente como un campo (email) dentro del documento de los informes.

La siguiente figura muestra el contenido de la autenticación de usuarios en Firebase [09], principal razón por la que se ha tomado la decisión de no incluir los usuarios en la Firestore Database creada previamente:

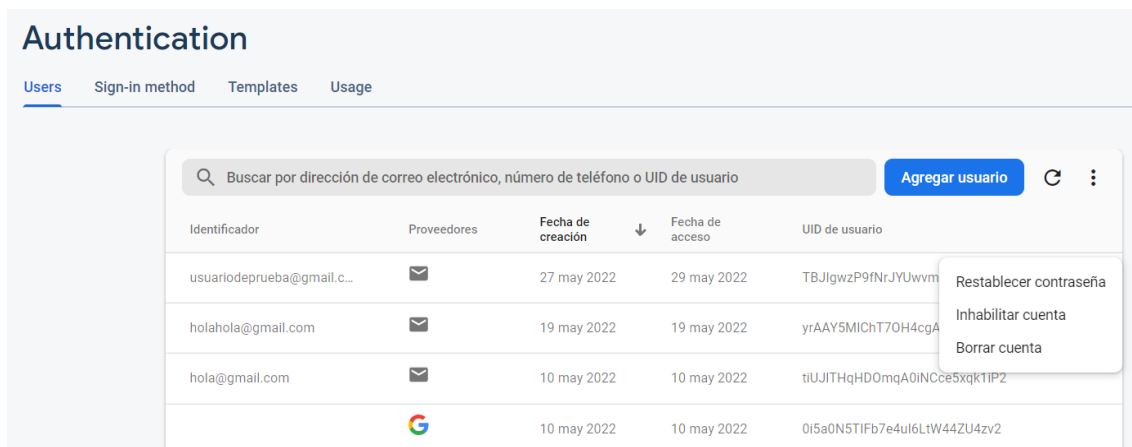


Figura 51. Pestaña de usuarios (Firebase Authentication).

Como se puede observar, Firebase almacena la información de los usuarios que se registran en la aplicación, quedando registrado el correo utilizado, el proveedor (en este caso, correo electrónico y contraseña o cuenta de Google), la fecha de creación de la cuenta y un UID asociado al usuario.

El usuario registrado mediante una cuenta de Google ha sido censurado en la figura por temas de privacidad, pero queda correctamente registrado su parámetro Identificador (correo) en Firebase.

De esta forma, queda almacenada de información de los usuarios registrados en PVD Prevención de una forma clara, pudiendo realizar operaciones con ellos desde la misma consola de Firebase (y desde la propia app en un futuro), sin necesidad de añadirlos a la Firestore Database desde el código Kotlin de la aplicación en Android Studio.

Este documento pone a disposición del usuario un manual de ayuda que explica cómo interactuar y moverse entre las diferentes pantallas de la aplicación (ver anexo I).

4. ACCESIBILIDAD Y USABILIDAD

En cuanto a la accesibilidad del sistema [18], no se ha podido realizar una evaluación adecuada, debido a que no se dispone de los medios necesarios para llevarla a cabo, es decir, pruebas reales de laboratorio con muestras de población por género, edad, etc.

Por otro lado, es posible aplicar diversos tests para realizar una valoración de la usabilidad de la aplicación. En este documento, se van a aplicar los diez principios de usabilidad de Nielsen [28].

Jakob Nielsen es una de las mayores eminencias mundiales en el campo de la usabilidad. Tanto es así, que en 1990 estableció un decálogo sobre buenas praxis a nivel de usabilidad que, hoy en día, sigue siendo el mayor referente para ofrecer la mejor experiencia de usuario posible en un entorno digital.

4.1. Principios de usabilidad de Jakob Nielsen

1. Visibilidad del estado del sistema

El sistema (web, app o cualquier otro producto digital) debe siempre mantener informado al usuario de lo que está ocurriendo.

PVD Prevención informa al usuario mediante mensajes de las acciones que va realizando él en la aplicación. Para ello utiliza los Toast, mensajes que aparecen y permanecen en pantalla unos segundos, así como texto plano en la aplicación.

2. Relación entre el sistema y el mundo real

El sitio web o aplicación tiene que utilizar el lenguaje del usuario, con expresiones y palabras que le resulten familiares. Además, la información debe aparecer en un orden lógico y natural.

Los errores que PVD Prevención ofrece al usuario son claros y utilizan texto plano con la solución al problema para que él mismo sea capaz de aplicarla.

3. Control y libertad del usuario

En caso de elegir alguna opción del sitio web por error, el usuario agradecerá disponer de una “salida de emergencia” para abandonar el estado no deseado en que se halla. Debe poder deshacer o repetir una acción previamente realizada.

PVD Prevención no dispone actualmente de una opción para deshacer una acción. La única acción que aplica este principio es la de borrar una evaluación, en la que el usuario podría arrepentirse de borrarla, pero en su defecto recibe un mensaje de confirmación de borrado de la evaluación por pantalla para evitar los casos en los que pulse el botón por equivocación.

4. Consistencia y estándares

Es importante establecer convenciones lógicas y mantenerlas siempre. El usuario no tiene por qué saber que diferentes palabras, situaciones o acciones significan lo mismo.

Al tratarse de una aplicación sencilla, no dispone de menú desplegable que contenga los distintos apartados de la aplicación. Si en un futuro se amplía, introduciendo nuevos apartados, debería incluir este menú en la parte superior.

5. Prevención de errores

Ayuda al usuario a que no caiga en un error.

La aplicación ofrece en cada pantalla una pequeña descripción de ésta (en texto plano) para ayudar al usuario a entenderla y desenvolverse correctamente en ella, para evitar la aparición de errores.

6. Reconocimiento antes que recuerdo

Debemos hacer visibles acciones y opciones para que el usuario no tenga que recordar información entre distintas secciones o partes del sitio web o aplicación.

Actualmente PVD Prevención no dispone de texto junto a determinados iconos dentro de la aplicación (la papelera para borrar, la interrogación para la ayuda...) para reforzar el mensaje de la utilidad de éstos.

7. Flexibilidad y eficiencia de uso

Los aceleradores o atajos de teclado, por ejemplo, pueden hacer más rápida la interacción para usuarios expertos, de tal forma que el sitio web o aplicación sea útil tanto para usuarios básicos como avanzados.

Al tratarse de una aplicación sencilla, no tiene sentido aplicar ningún tipo de atajo para realizar una acción dentro de la aplicación, es suficiente con los botones de que dispone la misma para volver a una pantalla anterior o al menú principal.

En un futuro, se podría incluir la opción de borrar o descargar varias evaluaciones a la vez para ahorrar algo de tiempo en caso de disponer de mucha información (evaluaciones) de un mismo usuario en la BD.

8. Estética y diseño minimalista

Las páginas no deben contener información innecesaria. Cada información extra compite con la información relevante y disminuye su visibilidad.

PVD Prevención ofrece al usuario un diseño minimalista que incluye lo justo y necesario para cumplir su función y objetivos. Es posible que contenga información poco (o nada) relevante en algunas pantallas para los usuarios más experimentados en aplicaciones de este tipo, por lo que en un futuro se podría incluir un botón de ayuda en cada una de las pantallas para limpiarlas todavía más, dejando esa ayuda a disposición de los usuarios más novatos en la aplicación.

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores

Los mensajes de error se deben entregar en un lenguaje claro y simple, indicando en forma precisa el problema y sugerir una solución constructiva al problema.

Los formularios disponen de un control de errores que refleja al usuario, por medio de mensajes, cuál ha sido la causa del error y qué pueden hacer para solucionarla por sí mismos.

10. Ayuda y documentación

Aunque es mejor que el sitio web o aplicación pueda ser usado sin ayuda, puede ser necesario proveer cierto tipo de ayuda. En este caso, la ayuda debe ser fácil de localizar, especificar los pasos necesarios y no ser muy extensa.

La aplicación ofrece en cada pantalla una pequeña descripción de ésta y cómo operar en ella, a modo de referencia, pero también incluye un botón de ayuda en el menú principal con información más extensa de cada una de las pantallas y sus funcionalidades.

Como se ha comentado en apartados anteriores de este mismo test, en un futuro se podría implementar la idea de incluir un apartado de ayuda en cada pantalla (mediante un icono de una interrogación) para evitar recargar la pantalla a los usuarios más experimentados con información no necesaria.

Estos son los diez principios de la usabilidad que se deben respetar si se quiere garantizar la mejor experiencia para los usuarios de la aplicación.

En un principio pueden parecer muy obvios, pero es recomendable tenerlos en cuenta durante el diseño e implementación de la app, así como aplicarlos sobre la misma una vez esté terminada, para comprobar que el sistema cumple con las expectativas.

En el caso de PVD Prevención, se observa que cumple la mayoría de los principios de Nielsen, dejando algunos de ellos como trabajo futuro de la aplicación.

5. LICENCIA SOFTWARE Y DOCUMENTAL

Copyright © 2022, Sergio Alloza García

Este documento, así como sus anexos son liberados bajo los términos de una licencia GFDL.

Se permite la redistribución, copia y/o modificación de estos documentos bajo los términos de la GNU Free Documentation License, versión 1.3 o cualquier versión posterior publicada por la Free Software Foundation.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation.



El plugin liberado es software libre: puedes redistribuirlo o modificarlo bajo los términos de la GNU General Public License publicada por la Free Software Foundation en su versión 3 o cualquier versión posterior.

Este programa es distribuido con el propósito de que sea útil, pero CON NINGUNA GARANTÍA, incluso sin garantía de COMERCIALIZACIÓN o APTITUD PARA CUALQUIER PROPÓSITO CONCRETO.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.



ESTE SOFTWARE ES SUMINISTRADO POR SERGIO ALLOZA GARCÍA Y CUALQUIER GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, PERO NO LIMITANDO A: LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN Y APTITUD PARA UN PROPÓSITO PARTICULAR, SON RECHAZADAS. EN NINGÚN CASO SERGIO ALLOZA GARCÍA O COLABORADORES SERÁN RESPONSABLES POR NINGÚN DAÑO DIRECTO, INDIRECTO, INCIDENTAL, ESPECIAL, EJEMPLAR O CONSECUCIONAL (INCLUYENDO, PERO NO LIMITADO A: LA ADQUISICIÓN O SUSTITUCIÓN DE BIENES O SERVICIOS, LA PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS; O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) O POR CUALQUIER TEORÍA DE RESPONSABILIDAD, YA SEA POR CONTRATO, RESPONSABILIDAD ESTRICTA O AGRAVIO (INCLUYENDO NEGLIGENCIA O CUALQUIER OTRA CAUSA) QUE SURJA DE CUALQUIER MANERA DEL USO DE ESTE SOFTWARE, INCLUSO SI SE HA ADVERTIDO DE LA POSIBILIDAD DE TALES DAÑOS.

Tampoco hay que olvidar que este documento, por defecto, está al amparo de la licencia 7.3, por su inclusión en el Repositorio Institucional de Documentos de la Universidad de Zaragoza: ZAGUAN.



6. CONCLUSIONES

En líneas generales, las conclusiones a las que he conseguido llegar tras completar el proyecto son positivas.

Si bien es cierto que en un principio no estaba tan convencido de que fuese una buena idea la combinación de ambos ámbitos, tecnológico y de PRL, o de mi preparación para afrontar este proyecto de final de carrera, tanto el recorrido de la realización de la aplicación y del presente documento, como los resultados que los acompañan han sido, a nivel personal, satisfactorios.

La parte más orientada a la Prevención de Riesgos Laborales del proyecto me ha servido como refuerzo a los contenidos de su asignatura de cuarto curso vistos durante la titulación, además de encontrar una forma de aplicarlos en un proyecto tecnológico, una aplicación Android en este caso.

La parte más tecnológica del TFG me ha ayudado a ver cómo se pueden enlazar los conocimientos aprendidos durante estos años de universidad y afianzarlos, pasando por las diferentes áreas de conocimiento de la informática y sus asignaturas asociadas, como pueden ser Bases de datos, Comercio electrónico o Programación, entre otras.

Por otro lado, he observado la importancia de estructurar bien un proyecto, dividirlo en diferentes partes y asignar un determinado intervalo de tiempo a cada una, así como realizar una buena presentación de éste, consiguiendo ofrecer un producto (aplicación móvil) lo más profesional posible.

6.1. Trabajo futuro

A continuación, se expone una serie de apartados referentes a la aplicación que han podido quedar incompletos, o que podrían tener una versión mejorada, así como otros que todavía no se han implementado, de forma que quedan como trabajo futuro para PVD Prevención.

- Las evaluaciones realizadas en la aplicación, como indica el nombre de la misma, están orientadas a analizar los entornos de trabajo que incluyen pantallas de visualización de datos.
Sería interesante, de cara al futuro, poder ampliar la aplicación de forma que incluya diferentes tipos de tests, orientados a las diferentes áreas en que se puede aplicar la PRL. En dicho caso, se podría barajar de nuevo la posibilidad de adoptar el nombre All Prevention, que hacía más referencia a la PRL que solo a las pantallas de visualización.
- La aplicación, actualmente, está implementada solamente para dispositivos móviles Android, por lo que su inclusión en dispositivos iOS sería un aspecto a tener en cuenta.

- Como se ha mencionado previamente, en el apartado de implementación, el acceso a la aplicación mediante una cuenta de Google sólo es posible desde el dispositivo que se ha utilizado para las pruebas de este proyecto. Se debe corregir el error que existe en el código de la actividad de autenticación para conseguir el acceso mediante su cuenta de Google para el resto de los usuarios de la aplicación.
- En la actividad de visualización de evaluaciones (dónde además es posible borrarlas y descargarlas), sería conveniente incluir una barra de búsqueda para los casos en que un usuario disponga de numerosas evaluaciones guardadas, de forma que se pudiese utilizar la fecha de guardado de una evaluación almacenada en la BD como posible filtro de búsqueda.
Además, podría resultar útil incluir una opción de borrado masivo, para los usuarios interesados en eliminar más de una evaluación en una sola operación.
- Actualmente, aunque la aplicación solicita al usuario el mínimo de datos necesarios para registrarse (correo y contraseña), no le ofrece la posibilidad de modificarlos ni existe un apartado de contraseña olvidada o correo de recuperación de ésta.
- En cuanto al apartado de la accesibilidad, sería adecuado implementar alguna funcionalidad como, por ejemplo, un sistema de audio o lector de pantalla, de cara a los posibles usuarios de la aplicación que puedan presentar determinadas discapacidades.

7. REFERENCIAS BIBLIOGRÁFICAS

- [01] AGARWAL, ANIRUDH, 2017, Android - How to achieve setOnClickListener in Kotlin?. *Stack Overflow* [online]. 2017. [Accessed 14 June 2022]. Available from: <https://stackoverflow.com/questions/44301301/android-how-to-achieve-setonclicklistener-in-kotlin>
- [02] BERZAL, FERNANDO, 2003, El ciclo de vida de un sistema de información. *Elvex.ugr.es* [online]. 2003. [Accessed 14 June 2022]. Available from: <https://elvex.ugr.es/idbis/db/docs/lifecycle.pdf>
- [03] BOE, 1997, BOE-A-1997-8671 Real Decreto 488/1997, de 14 de abril, sobre disposiciones mínimas de seguridad y salud relativas al trabajo con equipos que incluyen pantallas de visualización. *Boe.es* [online]. 1997. [Accessed 14 June 2022]. Available from: <https://www.boe.es/buscar/act.php?id=BOE-A-1997-8671>
- [04] CODIGOFACILITO CHANNEL, 2017, Generar PDF en Android. *Youtube.com* [online]. 2017. [Accessed 14 June 2022]. Available from: <https://www.youtube.com/watch?v=FIleHM7u4IY>
- [05] DEVELOPERS, ANDROID, 2022, Casillas de verificación | Desarrolladores de Android | Android Developers. *Android Developers* [online]. 2022. [Accessed 14 June 2022]. Available from: <https://developer.android.com/guide/topics/ui/controls/checkbox?hl=es-419>
- [06] FIREBASE CONSOLE, GOOGLE 2022, Firebase Console. *Console.firebase.google.com* [online]. 2022. [Accessed 14 June 2022]. Available from: <https://console.firebase.google.com/>
- [07] FIREBASE, GOOGLE 2022, Choose a Database: Cloud Firestore or Realtime Database | Firebase Documentation. *Firebase* [online]. 2022. [Accessed 14 June 2022]. Available from: <https://firebase.google.com/docs/database/rtdb-vs-firestore>
- [08] FIREBASE, GOOGLE, 2022, Get data with Cloud Firestore | Firebase Documentation. *Firebase* [online]. 2022. [Accessed 14 June 2022]. Available from: <https://firebase.google.com/docs/firestore/query-data/get-data#kotlin+ktx>
- [09] FIREBASE, GOOGLE, 2022, Get Started with Firebase Authentication on Android | Firebase Documentation. *Firebase* [online]. 2022. [Accessed 14 June 2022]. Available from: <https://firebase.google.com/docs/auth/android/start>
- [10] GARRIDO, PIEDAD, COLL, MANUEL and MARTÍNEZ, FRANCISCO J., 2005, DerEditor4GL: Software para la docencia en el diseño de Bases de Datos. *Bioinfo.uib.es* [online]. 2005. [Accessed 14 June 2022]. Available from: <http://bioinfo.uib.es/~joemiro/aenui/procJenui/Jen2005/gadere.pdf>

- [11] INSST, 2022, Trabajo con pantallas de visualización. *Preguntas técnicas frecuentes* [online]. 2022. [Accessed 14 June 2022]. Available from: <https://www.insst.es/documentacion/material-tecnico/preguntas-tecnicas-frecuentes-gestion-prl/riesgos-ergonomicos/trabajo-con-pantallas-de-visualizacion>
- [12] INSST, O.A., M.P., 2021, Guía técnica para la evaluación y prevención de los riesgos relativos a la utilización de equipos con pantallas de visualización. *Insst.es* [online]. 2021. [Accessed 14 June 2022]. Available from: <https://www.insst.es/documents/94886/789467/Gu%C3%ADa+t%C3%A9cnica+para+la+evaluaci%C3%B3n+y+prevenci%C3%B3n+de+los+riesgos+relativos+a+la+utilizaci%C3%B3n+de+equipos+con+pantallas+de+visualizaci%C3%B3n.pdf>
- [13] ISO 830, 2008, Especificación de Requisitos según el estándar de IEEE 830. *Fdi.ucm.es* [online]. 2008. [Accessed 16 June 2022]. Available from: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
- [14] KB CODER CHANNEL, 2021, PDF Generate In Android Studio | Kotlin | Android Tutorials. *Youtube.com* [online]. 2021. [Accessed 14 June 2022]. Available from: <https://www.youtube.com/watch?v=oQATh2B29QE>
- [15] KIMMEL, PAUL, 2008, *Uenicmlk.edu.ni* [online]. 2008. [Accessed 14 June 2022]. Available from: <http://www.uenicmlk.edu.ni/img/biblioteca/Ing%20en%20sistema%20Manual%20de%20UML%20Kimmel.pdf>
- [16] MÁRQUEZ, ERICK, 2016, Botones dinámicos, Android Studio. *Stack Overflow en español* [online]. 2016. [Accessed 14 June 2022]. Available from: <https://es.stackoverflow.com/questions/17931/de-qu%C3%A9-manera-puedo-hacer-botones-din%C3%A1micos-sin-tener-que-escribir-todos-y-cada>
- [17] MODELIO, 2011, Modelio Open Source - UML and BPMN free modeling tool. *Modelio Open Source* [online]. 2011. [Accessed 14 June 2022]. Available from: <https://www.modelio.org/>
- [18] MORA, SERGIO, 2006, Accesibilidad Web: Española. *Accesibilidadweb.dlsi.ua.es* [online]. 2006. [Accessed 14 June 2022]. Available from: <http://accesibilidadweb.dlsi.ua.es/?menu=espanola>
- [19] MOUREDEV BY BRAIS MOURE CHANNEL, 2020, Base de datos Firestore con Firebase. *Youtube.com* [online]. 2020. [Accessed 14 June 2022]. Available from: <https://www.youtube.com/watch?v=t5yyc1XfQrs>
- [20] MOUREDEV BY BRAIS MOURE CHANNEL, 2021, Crear icono app Android Studio. *Youtube.com* [online]. 2021. [Accessed 14 June 2022]. Available from: <https://www.youtube.com/watch?v=ol6WgrHwsul>
- [21] MOUREDEV BY BRAIS MOURE CHANNEL, 2019, Splash Screen Android Studio. *Youtube.com* [online]. 2019. [Accessed 14 June 2022]. Available from: <https://www.youtube.com/watch?v=ksaaMt8Lo6U>

- [22] MVS, RF, 2016, Bloquear al usuario el botón atrás. *Stack Overflow en español* [online]. 2016. [Accessed 14 June 2022]. Available from: <https://es.stackoverflow.com/questions/15925/como-hacer-para-bloquear-al-usario-el-bot%C3%B3n-atr%C3%A1s-en-mi-app>
- [23] N PRABHU, TANU, 2020, Expandable Text View Using Kotlin and Android Studio. *Medium* [online]. 2020. [Accessed 14 June 2022]. Available from: <https://levelup.gitconnected.com/expandable-text-view-using-kotlin-and-android-studio-2b70ff04c0e9>
- [24] N PRABHU, TANU, 2019, To-Do List Application Using Kotlin Using Android Studio. *Medium* [online]. 2019. [Accessed 14 June 2022]. Available from: <https://medium.com/@tanunprabhu95/to-do-list-application-using-kotlin-using-android-studio-546e74ac75aa>
- [25] ONU, 2015, Objetivos y metas de desarrollo sostenible. *Desarrollo Sostenible* [online]. 2015. [Accessed 14 June 2022]. Available from: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>
- [26] SHARMA, POOJA, 2021, Reasons to treat web application architecture seriously. *Cynoteck* [online]. 2021. [Accessed 14 June 2022]. Available from: <https://cynoteck.com/es/blog-post/reasons-to-treat-web-application-architecture-seriously/>
- [27] TALENT, ISDI, 2014, Balsamiq la herramienta para hacer prototipos de proyectos. *ISDI Digital Talent* [online]. 2014. [Accessed 14 June 2022]. Available from: <https://www.isdi.education/es/blog/balsamiq-herramienta-para-realizar-prototipos-de-tus-proyectos>
- [28] URARTE, ELOY, 2019, Los 10 principios de usabilidad de Nielsen. *Welcome to UX* [online]. 2019. [Accessed 14 June 2022]. Available from: <https://welcometoux.com/ux/10-principios-usabilidad-nielsen/>

ANEXOS

Anexo I - Manual de usuario

El manual de usuario (o guía de usuario) es el apartado del documento destinado a dar asistencia a las personas que utilizan la aplicación PVD Prevención.

Para ello se ofrecen a continuación, de forma detallada, una serie de pasos que ayudará al usuario a navegar por la aplicación y desenvolverse mejor en cada una de las actividades, así como un conjunto de imágenes asociadas a las explicaciones para entender mejor su funcionamiento y poder dar una solución a los problemas encontrados más comunes.

La siguiente figura muestra la pantalla de autenticación de PVD Prevención, que es la primera que se le presenta al usuario al abrir la aplicación.

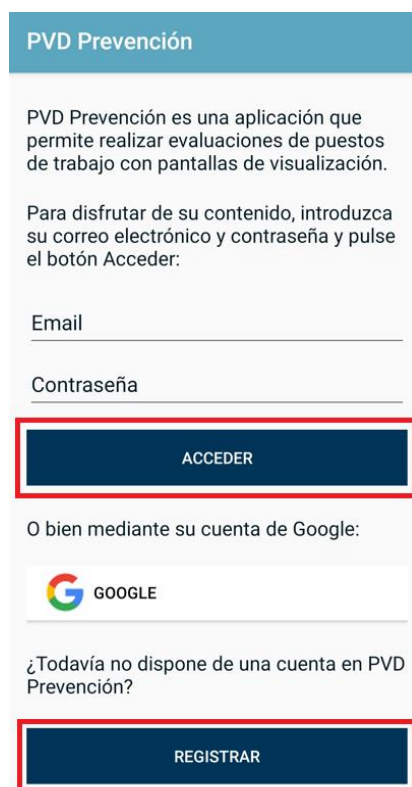
La imagen muestra la interfaz de usuario de la aplicación PVD Prevención. En la parte superior, un encabezado azul contiene el texto "PVD Prevención". Debajo, un párrafo describe la aplicación: "PVD Prevención es una aplicación que permite realizar evaluaciones de puestos de trabajo con pantallas de visualización." Seguido de otra línea de texto: "Para disfrutar de su contenido, introduzca su correo electrónico y contraseña y pulse el botón Acceder:". Hay dos campos de entrada: "Email" y "Contraseña". Debajo de estos campos, el botón "ACCEDER" está resaltado con un recuadro rojo. A continuación, se ofrece la opción "O bien mediante su cuenta de Google:" con un botón que incluye el logo de Google y la palabra "GOOGLE". En la parte inferior, se pregunta "¿Todavía no dispone de una cuenta en PVD Prevención?" y el botón "REGISTRAR" también está resaltado con un recuadro rojo.

Figura A01. Pantalla de autenticación de PVD Prevención.

En esta pantalla el usuario puede acceder en caso de disponer de una cuenta, o registrarse en caso de no tener una todavía, pulsando los botones de Acceder (después de rellenar los campos pertinentes) o Registrar, respectivamente, como se indica en la anterior figura.

Registrar usuario

Una vez se presiona el botón de Registrar en la pantalla de autenticación, el usuario accede a la pantalla de Registro de la aplicación.

The figure displays two versions of the registration screen. Both screens have a blue header with the title 'Registro'. The left screen shows the initial state with instructions: 'Para llevar a cabo el registro en la aplicación, introduzca los campos de correo electrónico y contraseña y pulse el botón Registrar:'. Below this are two input fields, 'Email' and 'Contraseña', and a dark blue button labeled 'REGISTRAR'. The right screen shows the form after some input. The 'Email' field contains 'usuariodeprueba@gmail.com' and the 'Contraseña' field contains six dots. The 'REGISTRAR' button is highlighted with a red rectangular border.

Figura A02. Pantalla de registro de PVD Prevención.

Para registrar una cuenta en PVD Prevención, se rellenan los campos de email (correo electrónico) y contraseña (mínimo seis caracteres) y se pulsa en el botón de Registrar. En caso de existir algún error en el registro, se le notificará al usuario por pantalla mediante un mensaje (ejemplo: “Ese correo electrónico ya está registrado en PVD Prevención. Por favor, introduzca uno nuevo”).

Si, por el contrario, el registro se efectúa correctamente, el usuario visualiza por pantalla un mensaje de “Registro realizado con éxito” y es redirigido a la pantalla de autenticación para poder acceder a la aplicación con su nueva cuenta.

Iniciar sesión

Para iniciar sesión, es necesario haber realizado previamente el apartado de registrar usuario.

The image displays two identical screenshots of the 'PVD Prevención' login interface. Each screen has a blue header with the text 'PVD Prevención'. Below the header, there is a paragraph: 'PVD Prevención es una aplicación que permite realizar evaluaciones de puestos de trabajo con pantallas de visualización.' followed by another paragraph: 'Para disfrutar de su contenido, introduzca su correo electrónico y contraseña y pulse el botón Acceder:'. Below the text are two input fields labeled 'Email' and 'Contraseña'. At the bottom of each screen is a dark blue button labeled 'ACCEDER'. In the left screenshot, red boxes highlight the 'Email' and 'Contraseña' fields. In the right screenshot, a red box highlights the 'ACCEDER' button.

Figura A03. Pantalla de autenticación de PVD Prevención (Iniciar sesión).

Se introducen los campos de email y contraseña registrados previamente en la aplicación y se pulsa en el botón de Acceder.

Una vez se accede a la aplicación, se presenta el Menú Principal de PVD Prevención.

The image shows the 'Menú principal' (Main Menu) of the PVD Prevención application. It has a blue header with the text 'Menú principal'. Below the header, there is a logo consisting of a laptop screen displaying 'DVD' and the word 'PREVENCIÓN' below it. In the top right corner, there is a square icon with a question mark. Below the logo, there are four menu items: 'REALIZAR EVALUACIÓN', 'MIS EVALUACIONES', 'PREGUNTAS FRECUENTES (FAQ)', and 'CERRAR SESIÓN'. The 'CERRAR SESIÓN' item is highlighted with a dark blue background.

Figura A04. Menú Principal de PVD Prevención.

Visualizar la ayuda

Para visualizar la ayuda, es necesario haber realizado previamente el apartado de iniciar sesión.



Figura A05. Menú Principal de PVD Prevención (Visualizar la ayuda).

Basta con pulsar el botón de la interrogación de la esquina superior derecha para entrar en la ayuda que ofrece PVD Prevención sobre la aplicación.

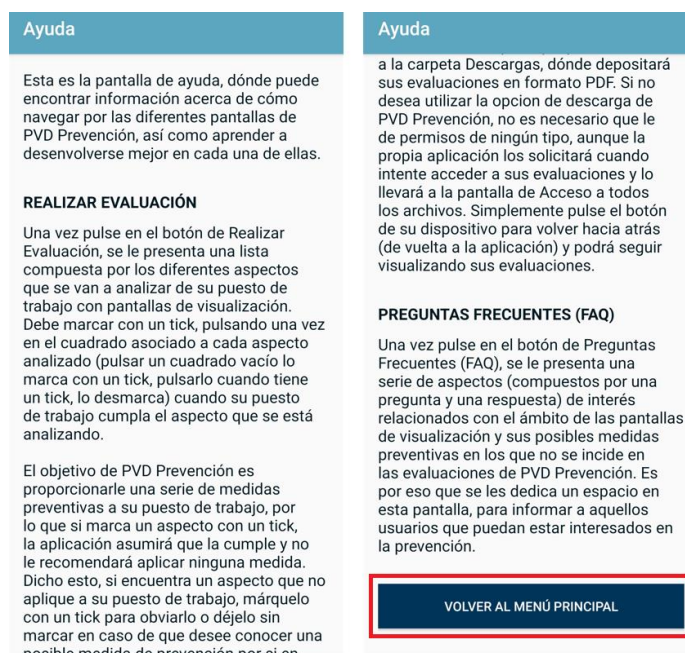


Figura A06. Pantalla de ayuda de PVD Prevención.

La pantalla de ayuda ofrece al usuario información sobre la aplicación y cómo desenvolverse en ella, de forma similar a este manual de usuario, pero más brevemente. La información explica, sobre todo, el uso de los botones del menú principal y qué se puede hacer en las pantallas a las que te dirige la aplicación al pulsarlos. Se trata de una pantalla meramente informativa, con un botón en la parte inferior para volver al menú principal.

En este manual de usuario, se exponen las acciones que se pueden realizar dentro de la aplicación en el orden lógico en que la mayoría de los usuarios debería llevarlas a cabo. El hecho de que la ayuda se sitúe justo después del inicio de sesión se debe a que incluye un apartado importante, como se observa en la figura A07, que el usuario debería consultar antes de acceder a la pantalla de evaluaciones guardadas en la app, tal y como se explica posteriormente (en el apartado Visualizar evaluaciones).

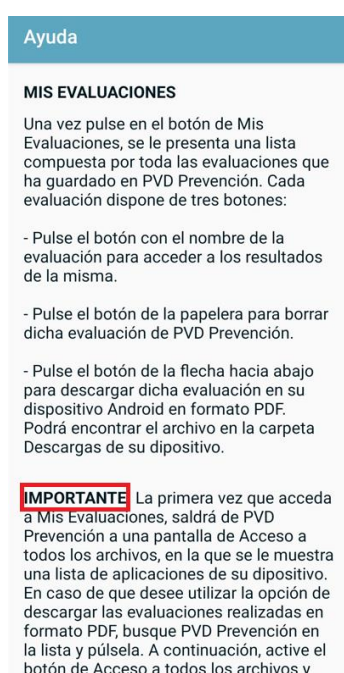


Figura A07. Información importante de la pantalla de ayuda de PVD Prevención.

Realizar evaluación

Para realizar una evaluación es necesario haber realizado previamente el apartado de iniciar sesión.

The image shows two side-by-side panels from a mobile application. The left panel, titled 'Menú principal', contains a logo with 'DVD' on a laptop screen and the word 'PREVENCIÓN' below it. Below the logo is a red-bordered button labeled 'REALIZAR EVALUACIÓN'. Underneath this button are three more options: 'MIS EVALUACIONES', 'PREGUNTAS FRECUENTES (FAQ)', and a dark blue button labeled 'CERRAR SESIÓN'. The right panel, titled 'Realizar evaluación', contains instructions at the top: 'Marque con un tick las opciones que considere que cumple el lugar se dispone a analizar.' and 'En caso de no aplicar una opción a su puesto de trabajo, márquela para no obtener la medida preventiva asociada en los resultados.' Below this is a section 'Consideraciones generales' with one checked option: 'La utilización del equipo es segura, no es una fuente de riesgo por sí mismo.' The bottom section is titled 'Pantalla' and contains six options with checkboxes: 'Los caracteres de la pantalla están bien definidos y tienen un tamaño suficiente.' (unchecked), 'El espacio entre caracteres y entre renglones es adecuado.' (checked), 'La imagen de la pantalla es estable y no se observan destellos, centelleos ni otras inestabilidades.' (unchecked), 'Se puede ajustar la luminosidad y el contraste entre los caracteres y el fondo de la pantalla.' (checked), 'La pantalla es orientable e inclinable.' (unchecked), and 'No se observan reflejos ni reverberaciones molestas.' (unchecked).

Figura A08. Menú principal y pantalla de Realizar Evaluación de PVD Prevención.

En el menú principal de la aplicación, se pulsa en el botón de Realizar evaluación, como se observa en la figura A08, para así acceder a su pantalla correspondiente.

En dicha pantalla, al usuario se le presenta un listado de aspectos para analizar de su puesto de trabajo con pantallas de visualización, junto a una casilla de verificación que se puede marcar o desmarcar pulsando en ella.

El usuario debe revisar toda la lista y marcar con un tick todas aquellas opciones que cumpla su puesto de trabajo. Tal y como se indica en la propia app al acceder a la pantalla de Realizar evaluación, en caso de que no aplique una opción al puesto de trabajo analizado (por ejemplo, en caso de no existir un portadocumentos) es preferible marcar la opción, puesto que PVD Prevención ofrece una posible medida a un problema que, en dicho caso, no existe. De esta forma, el usuario evita recibir en los resultados de la evaluación realizada una medida para ese problema inexistente.

Realizar evaluación

☐ Forman parte del espectro visible, con reducidas a niveles insignificantes.

Humedad

☐ El nivel de humedad ambiental es aceptable.

Interconexión persona/ordenador

☐ El programa está adaptado a la tarea.

☐ El programa es fácil de utilizar y se adapta a los conocimientos y a la experiencia de los usuarios.

☐ Se informa a los trabajadores y se consulta con sus representantes sobre la existencia de posibles dispositivos de control.

☐ El sistema (software) proporciona indicaciones sobre su desarrollo.

☐ El sistema (software) muestra la información en un formato y a un ritmo adaptado a los trabajadores.

☐ Se aplican los principios de la ergonomía al tratamiento de la información.

RESULTADOS

Resultados

Consideraciones generales

No es necesario aplicar medidas preventivas.

Pantalla

Debido a la variabilidad de los usuarios potenciales y la relevancia de la legibilidad de la información representada en las pantallas, es importante que existan posibilidades de ajuste y de regulación de distintos parámetros tales como el contraste, la diferencia de luminancias o el tamaño de los caracteres que permitan la adaptación a las diferentes necesidades.

Aunque se recomienda una altura de los caracteres de 20-22 minutos de arco, en la práctica, la mayor parte de las aplicaciones que se emplean en un entorno de oficina permiten modificar el tipo de letra, el tamaño de la misma y otras características del texto como el espaciado entre caracteres, palabras y líneas, etc., permitiendo adecuarlo al usuario y a sus necesidades particulares.

Idealmente, una pantalla de visualización no debe presentar ningún tipo de artefacto visual que dificulte la legibilidad: falta de uniformidad de luminancia, de color o de contraste, distorsiones geométricas (para filas o columnas de texto), defectos en la pantalla, parpadeo, efecto muaré, etc. Por ejemplo: en relación con la estabilidad de la imagen, la pantalla se debería ver libre de parpadeos por al menos el 90% de los usuarios.

Para que los símbolos en pantalla se visualicen adecuadamente, deben tener suficiente contraste

Figura A09. Pantallas de Realizar evaluación y Resultados de PVD Prevención.

Una vez el usuario ha completado la lista, pulsa el botón de la parte inferior de la pantalla para obtener los resultados de la evaluación, como muestra la figura A09. La pantalla de resultados ofrece al usuario medidas de prevención asociadas a todos aquellos aspectos que no cumple el puesto de trabajo analizado. En caso de cumplirlos, muestra el mensaje “No es necesario aplicar medidas preventivas”.

Guardar evaluación

Para guardar una evaluación es necesario haber realizado previamente el apartado de realizar evaluación.

Cabe la posibilidad de que el usuario que realiza una evaluación solamente quiera conocer algunas medidas que podría aplicar para corregir ciertos aspectos que su puesto de trabajo todavía no cumple, pero lo interesante de PVD Prevención es el hecho de poder guardarlas (o incluso descargarlas, como se explica más adelante en este documento), de forma que pueda visualizar los resultados en cualquier momento.

The figure consists of two side-by-side screenshots of a mobile application interface, specifically the 'Resultados' (Results) screen. Both screens display the same text content, which includes a paragraph about the system's adaptability and a section for saving the evaluation. The left screenshot highlights three elements with red rectangular boxes: the text input field containing 'Evaluación de mi oficina', the dark blue button labeled 'GUARDAR EVALUACIÓN', and another dark blue button labeled 'VOLVER AL MENÚ PRINCIPAL'. The right screenshot shows the same interface but with a white modal dialog box overlaid in the center. The dialog box contains the message 'Se ha guardado la evaluación correctamente. Para visualizarla, vaya a Mis Evaluaciones.' and a green button labeled 'ACEPTAR'.

Figura A10. Pantalla de Resultados y mensaje de evaluación guardada de PVD Prevención.

Para guardar una evaluación, hay que rellenar el cuadro con un título, como por ejemplo “Evaluación de mi oficina”, así como se observa en la figura A10. Una vez introducido, se pulsa el botón de Guardar evaluación. Si se ha efectuado el guardado con éxito, aparecerá un mensaje en pantalla. Se pulsa en Aceptar y la app te redirige al usuario al menú principal, desde donde podrá ir a la pantalla de visualizar las evaluaciones.

Dar permisos a la aplicación

Para dar permisos a la aplicación es necesario haber realizado previamente el apartado de realizar evaluación.

La primera vez que se accede a Mis evaluaciones (y cada vez que se acceda cuando todavía no se le ha dado permisos a la app), PVD Prevención redirige al usuario a la pantalla de selección de permisos de su dispositivo.

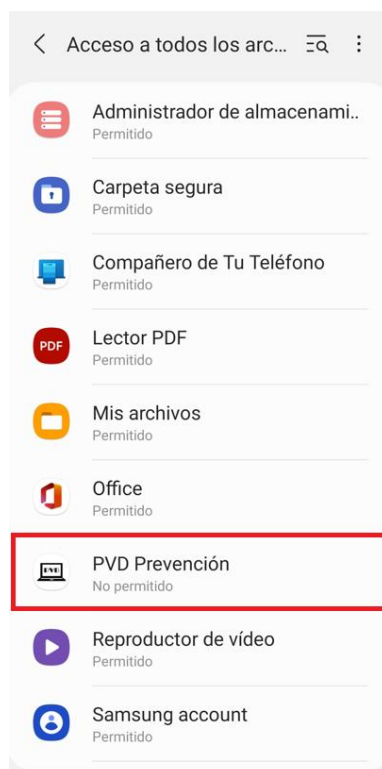


Figura A11. Pantalla de selección de permisos del dispositivo.

En esa lista de aplicaciones, se busca PVD Prevención y se pulsa en ella. Una vez dentro, aparecerá una pantalla con un botón de encendido-apagado. Se pulsa el botón (apagado) para encenderlo y darle, de esta manera, permisos a la aplicación para que pueda acceder al directorio Descargas del dispositivo y mostrar ahí las evaluaciones descargadas.

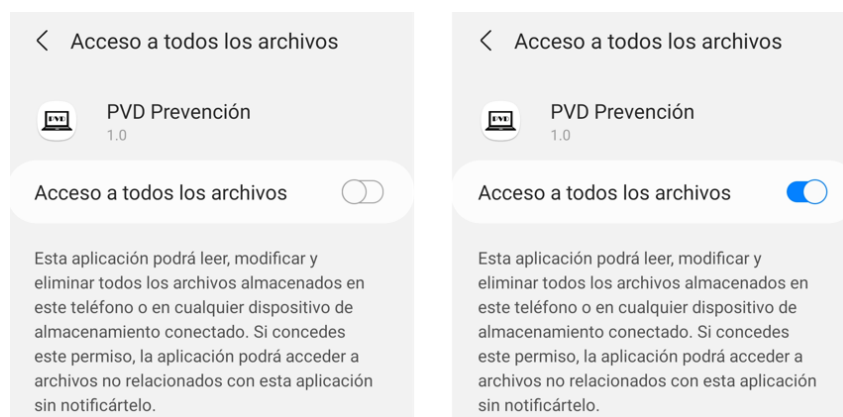


Figura A12. Acceso a los archivos de la pantalla de selección de permisos.

A continuación, se puede abandonar la pantalla de permisos del dispositivo y volver a la aplicación, la cual ya no volverá a pedir permisos al usuario, permitiéndole acceder a todo su contenido sin tener que volver a salir de la app.

Visualizar evaluación

Para visualizar una evaluación es necesario haber realizado previamente el apartado de realizar evaluación.



Figura A13. Menú principal y pantalla de Mis evaluaciones de PVD Prevención.

El usuario debe pulsar el botón de Mis evaluaciones en el menú principal para poder visualizarlas. Una vez pulsado, se le presenta una lista de todas las evaluaciones guardadas en la aplicación, así como un botón para borrarlas (papelera) y otro para descargarlas).

A continuación, se pulsa en el nombre (o título) de una de las evaluaciones de la lista para visualizar su contenido, como se observa en la figura A13 (en este caso, en “Evaluación de mi oficina 2”).

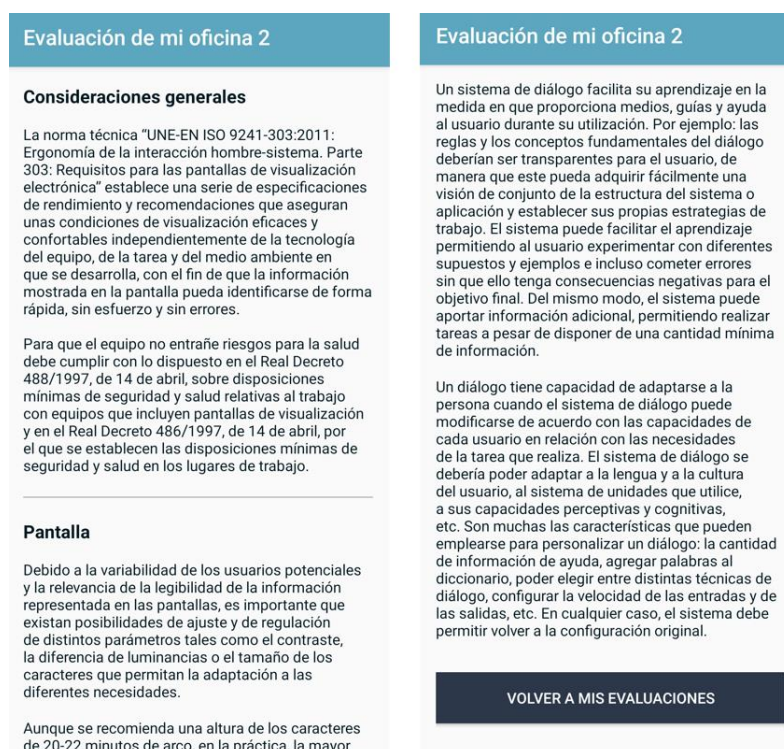


Figura A14. Pantalla de visualización de una evaluación de PVD Prevención.

La figura A14 corresponde a la pantalla a la que accede el usuario después de pulsar en el nombre de una evaluación. Como se puede observar, los resultados de la evaluación han sido almacenados en la aplicación para que el usuario tenga acceso a ellos en cualquier momento. En la parte inferior de la pantalla hay un botón para volver a la pantalla del listado de evaluaciones, para poder seguir interactuando con ellas.

Borrar evaluación

Para borrar una evaluación es necesario haber realizado previamente el apartado de realizar evaluación.

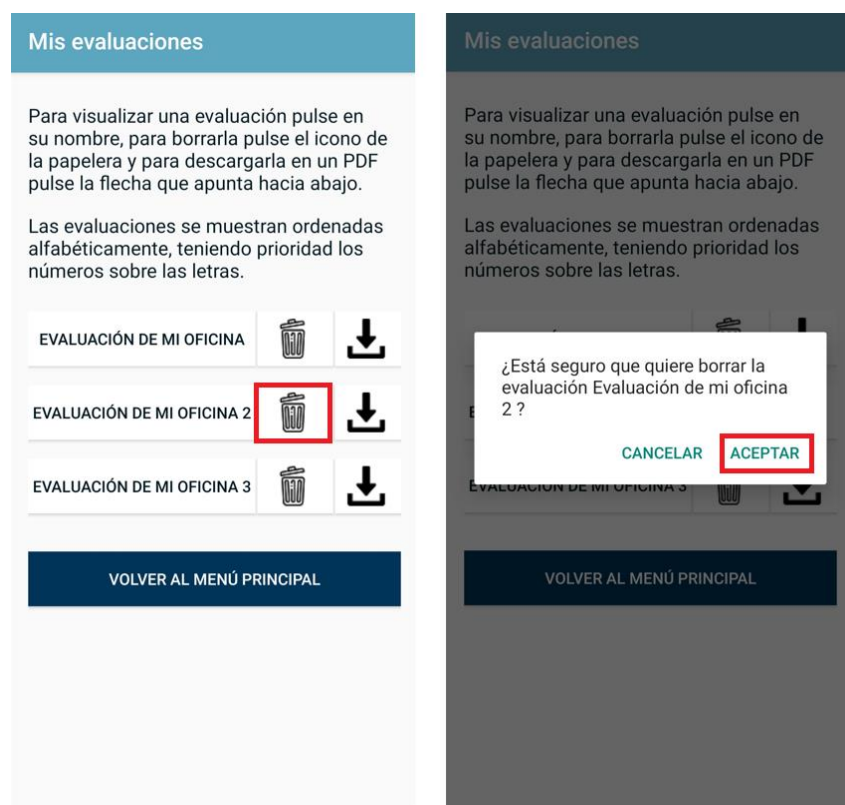


Figura A15. Pantalla de Mis evaluaciones y mensaje de borrar evaluación de PVD Prevención.

Para borrar una evaluación, se pulsa el botón de la papelera situado a la derecha de dicha evaluación. A continuación, se pulsa el botón de Aceptar y aparecerá un mensaje de que la evaluación ha sido eliminada correctamente y se actualiza la lista de evaluaciones, de forma que ya no aparece la evaluación eliminada.

Descargar evaluación

Para descargar una evaluación es necesario haber realizado previamente los apartados de dar permisos a la app y realizar evaluación.

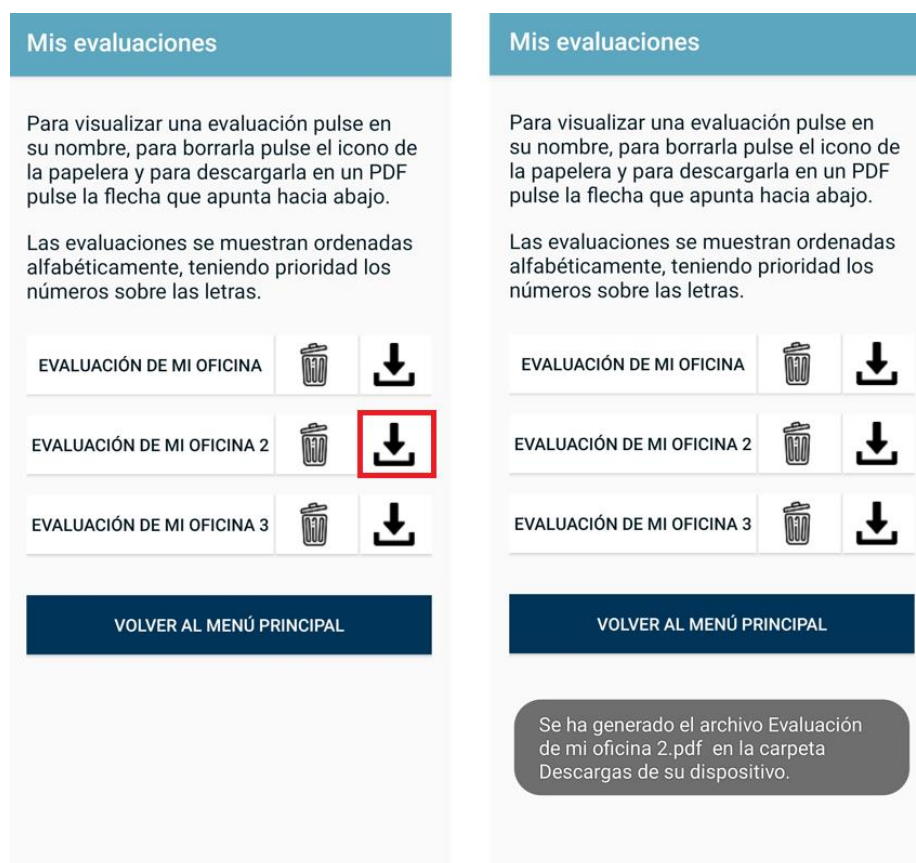


Figura A16. Pantalla de Mis evaluaciones y mensaje de descargar evaluación de PVD Prevención.

Para descargar una evaluación, se pulsa el botón de la flecha apuntando hacia abajo situado a la derecha del botón de la papelera asociado a dicha evaluación. Si previamente se han concedido los permisos de almacenamiento a la aplicación, a continuación, aparecerá un mensaje de que la evaluación ha sido descargada correctamente e informa al usuario del directorio donde puede encontrarla: la carpeta Descargas de su dispositivo.

La figura A17 muestra la evaluación descargada en un dispositivo Samsung Galaxy A40. Es posible que si el usuario dispone de otro dispositivo el directorio sea ligeramente distinto (mismo directorio, pero con otro nombre, por ejemplo), pero igualmente le servirán de guía las imágenes.



Figura A17. Camino para encontrar la evaluación descargada en un dispositivo Samsung Galaxy A40.

Una vez ubicado, solamente es necesario disponer de un lector de pdf en el dispositivo, así como cualquier otra aplicación alternativa que permita visualizar archivos de este formato y abrirlo.

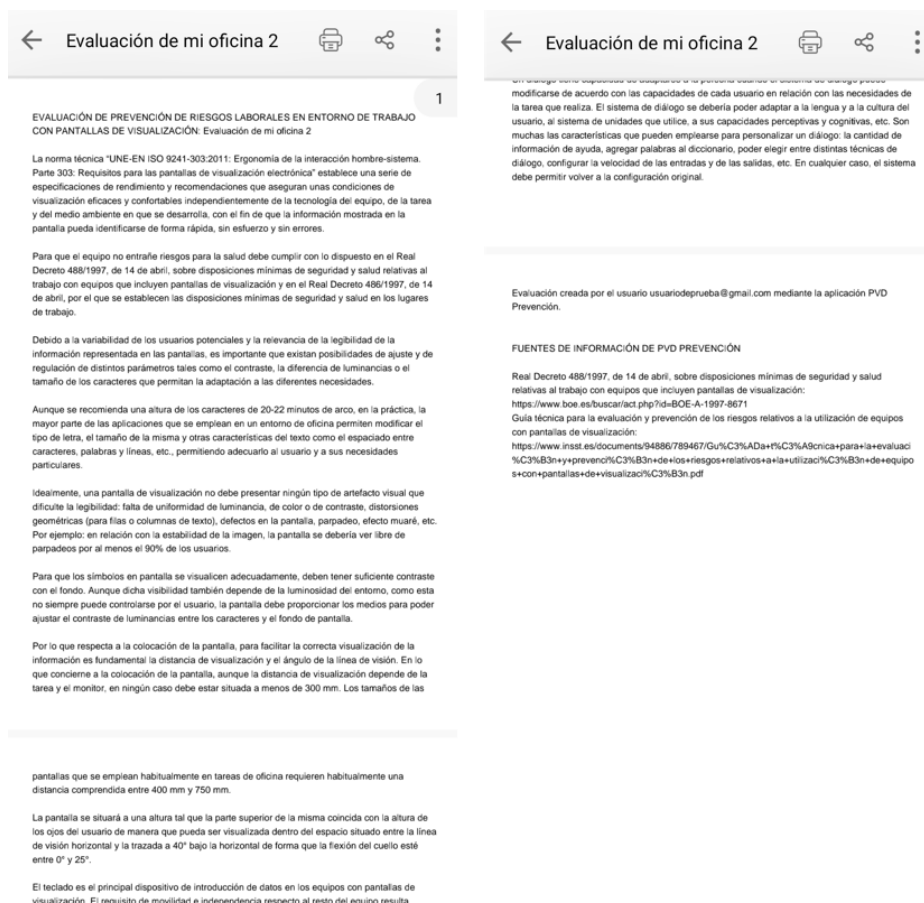


Figura A18. Evaluación descargada mediante la aplicación PVD Prevención.

Visualizar FAQ

Para descargar una evaluación es necesario haber realizado previamente el apartado de iniciar sesión.



Figura A19. Menú principal de PVD Prevención.

Acceder a la pantalla de preguntas frecuentes (FAQ) es tan sencillo como pulsar el botón de Preguntas frecuentes en el menú principal de la app. Una vez dentro, aparece una pequeña lista de preguntas junto a sus correspondientes respuestas. Para visualizarlas, basta con pulsar en el texto en **negrita** correspondiente a la pregunta (cómo se puede ver en la figura A20), así como pulsar en la flecha apuntando hacia abajo situada a la derecha del texto.

Al tratarse de un desplegable, se puede ocultar la respuesta de la misma forma, pulsando en el mismo texto o en la flecha que lo acompaña, esta vez apuntando hacia arriba.

Preguntas frecuentes (FAQ)	Preguntas frecuentes (FAQ)
<p>¿Cuál es el tiempo máximo de permanencia que no se debería sobrepasar ante una pantalla de visualización?</p> <p>▼</p>	<p>¿Cuál es el tiempo máximo de permanencia que no se debería sobrepasar ante una pantalla de visualización?</p> <p>▼</p>
<p>¿Cómo deben ser las pausas y descansos para el personal que trabaja con equipos con pantalla de visualización?</p> <p>▼</p>	<p>¿Cómo deben ser las pausas y descansos para el personal que trabaja con equipos con pantalla de visualización?</p> <p>La naturaleza de las pausas y descansos dependerán de las características de las tareas concretas que se realicen. Algunos aspectos a considerar respecto a las pausas son los siguientes:</p> <ul style="list-style-type: none"> - Las pausas deberían ser introducidas antes de que sobrevenga la fatiga. - El tiempo de las pausas no debe ser recuperado aumentando, por ejemplo, el ritmo de trabajo durante los períodos de actividad. - Resultan más eficaces las pausas cortas y frecuentes que las pausas largas y escasas. Por ejemplo,
<p>¿Cómo se debe ubicar la pantalla en un puesto que utiliza un equipo con PVDs?</p> <p>▼</p>	
<p>¿Quién se considera que es trabajador con equipos con pantallas de visualización?</p> <p>▼</p>	
<p>¿Qué medidas preventivas se pueden adoptar en los puestos con ordenador?</p>	

Figura A20. Pantalla de Preguntas frecuentes (FAQ) de PVD Prevención.

Cerrar sesión

Para cerrar sesión es necesario haber realizado previamente el apartado de iniciar sesión.



Figura A21. Menú principal de PVD Prevención.

En el menú principal de la aplicación, se pulsa el botón de Cerrar sesión, lo que redirige al usuario a la pantalla de inicio de PVD Prevención, dónde puede volver a iniciar sesión o registrarse con una cuenta diferente.