



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por nombre alumno
en Universidad de Burgos — 19 de mayo
de 2024

Tutor: nombre tutor

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	9
Apéndice B Especificación de Requisitos	13
B.1. Introducción	13
B.2. Objetivos generales	13
B.3. Catálogo de requisitos	13
B.4. Especificación de requisitos	13
Apéndice C Especificación de diseño	15
C.1. Introducción	15
C.2. Diseño de datos	15
C.3. Diseño procedimental	15
C.4. Diseño arquitectónico	15
Apéndice D Documentación técnica de programación	17
D.1. Introducción	17
D.2. Estructura de directorios	17
D.3. Manual del programador	17

D.4. Compilación, instalación y ejecución del proyecto	17
D.5. Pruebas del sistema	17
Apéndice E Documentación de usuario	19
E.1. Introducción	19
E.2. Requisitos de usuarios	19
E.3. Instalación	19
E.4. Manual del usuario	19
Apéndice F Anexo de sostenibilización curricular	21
F.1. Introducción	21

Índice de figuras

Índice de tablas

A.1. Coste total del personal	9
A.2. Coste total del Software	10
A.3. Dependencias del proyecto y licencias bajo las cuales están regis- tradas	12
B.1. CU-1 Nombre del caso de uso.	14

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apartado se recoge el ciclo de vida del proyecto, detallando los aspectos más relevantes del mismo y como se han resuelto los diferentes problemas a lo largo de su desarrollo. Se presentarán secciones que muestran de manera cronológica la justificación de las decisiones tomadas.

Para llevar a cabo este seguimiento y planificación del proyecto se ha utilizado una metodología Scrum, que ha permitido un desarrollo ágil dividido en sprints de dos semanas cada uno. Al comienzo de cada sprint se establecen las tareas y objetivos a realizar durante ese periodo. Al final de cada sprint se realizan reuniones con los tutores para valorar los resultados obtenidos y definir nuevas tareas para el siguiente sprint. Para organizar las diferentes tareas se ha usado Gitlab que permite visualizar los diferentes estados de desarrollo de las tareas.

A.2. Planificación temporal

La propuesta del proyecto consistía en crear una aplicación android basada en blockchain que simplifica la contratación y la verificación a través de contratos inteligentes. Los requisitos principales del proyecto se pueden dividir en los siguiente puntos:

- **Tecnología Blockchain:** Utilizar la tecnología blockchain para desplegar contratos inteligentes que gestionen automáticamente los contratos laborales, desde su creación hasta su ejecución.

- **Contratos Inteligentes:** Implementar contratos inteligentes en Python, Vyper y Solidity.
- **Localización GPS:** Integrar tecnología GPS para permitir a los empleadores imponer zonas de trabajo específicas.
- **Identificación segura mediante dispositivo móvil:** Implementar autenticación biométrica y el escaneo de códigos QR.
- **Integración con Pagos:** Incorporar procesamiento de pagos dentro de la aplicación para facilitar transacciones rápidas y seguras
- **Desarrollo de Aplicación Móvil:** Diseñar una interfaz de usuario amigable para dispositivos móviles que facilite la creación de contratos, seguimiento y pago.

Una vez expuestos los requerimientos principales del proyecto, la etapa inicial del proyecto se basó en una exhaustiva investigación que permitió obtener un conocimiento detallado sobre las tecnologías y herramientas necesarias. Dado que inicialmente no contaba con conocimientos previos en desarrollo de aplicaciones móviles y tecnología blockchain, esta investigación fue crucial para identificar las mejores prácticas y soluciones en estos campos. Esta etapa de investigación y adaptación a las nuevas tecnologías tuvo una duración de aproximadamente 3 semanas, durante el cual se hizo especial énfasis en entender a fondo el funcionamiento de la blockchain y los contratos inteligentes. Este aprendizaje teórico se reforzó de manera práctica con diversos proyectos usando Truffle, completando videoTutoriales y realizando el curso interactivo "CryptoZombies", además de consultar numerosos artículos especializados. Estas actividades facilitaron la asimilación del nuevo lenguaje de programación y familiarización con el entorno blockchain.

Sprint 0

Este sprint se desarrolló entre los días 3 y 17 de Noviembre de 2023. Se realizaron las siguientes tareas y objetivos:

1. **Configuración repositorio:** Se configuró el repositorio en GitLab y se establecieron ramas principales basadas en el flujo de trabajo de GitFlow, incluyendo 'main' para la producción y 'develop' para el desarrollo

2. **Configuración entorno para la redacción de la memoria:** Se descargaron las plantillas y se configuró el software de redacción "Texmaker".
3. **Investigación de tecnologías para realizar una app móvil:** Se realizó un análisis comparativo de las plataformas de desarrollo móvil más populares. Se evaluaron criterios como el rendimiento, la facilidad de uso y la comunidad de desarrolladores.
4. **Diseñar la arquitectura del proyecto:** Se definieron los componentes principales del sistema y su interacción. Para facilitar la comprensión de la estructura y el flujo de datos se desarrollaron diagramas.

Sprint 1

Este sprint se desarrolló entre los días 17 de Noviembre y 1 de Diciembre de 2023. Se realizaron las siguientes tareas y objetivos:

1. **Investigación Smart Contracts:** Se evaluó la elección entre usar tokens fungibles o no fungibles para el desarrollo de los contratos, optándose finalmente por el estándar ERC-721, característico de los NFTs, debido a su adaptabilidad al proyecto. A partir de esta decisión fue necesario evaluar la versión adecuada para el compilador de solidity que asegure la compatibilidad con ciertas bibliotecas necesarias.
2. **Creación de un prototipo visual:** Se diseñó un prototipo visual inicial para la aplicación móvil utilizando herramientas como Figma y Canvas. El prototipo se centró en crear una interfaz intuitiva y atractiva que refleje las funcionalidades claves del proyecto.

Sprint 2

Este sprint se desarrolló entre los días 1 de Diciembre y 21 de Diciembre de 2023. Se realizaron las siguientes tareas y objetivos:

1. **Primera implementación del contrato inteligente:** Siguiendo el estándar ERC-721 se planteó una primera solución que incluyese las funcionalidades básicas de creación y transferencia de NFTs de manera segura y eficiente.
2. **Configuración entorno de desarrollo aplicación móvil:** Se configuró el entorno de desarrollo para asegurar la correcta conectividad

entre la aplicación móvil y el backend, representado por el blockchain y el contrato inteligente.

Sprint 3

Este sprint se desarrolló entre los días 21 de Diciembre de 2023 y 12 de Enero de 2024. Se realizaron las siguientes tareas y objetivos:

1. **Continuación con la implementación del contrato inteligente:** Se continuó con el desarrollo del contrato inteligente implementando funciones que manejasen la lógica de los pagos dentro de la app. También se programación la lógica para la correcta destrucción del contrato una vez ha transcurrido su ciclo de vida.
2. **Creación de pruebas unitarias para comprobar el correcto funcionamiento del contrato inteligente:** Para garantizar el correcto funcionamiento del contrato inteligente se realizaron test en Java utilizando las herramientas Ganache y Truffle. Ganache sirvió para simular un entorno blockchain local y Truffle sirvió para compilar, migrar y probar el contrato inteligente con los test creados.
3. **Implementar navegación en la aplicación móvil:** Se programaron las pantallas principales de la aplicación. Estas pantallas se integraron mediante un menú interactivo que facilita la navegación dentro de la app mejorando la experiencia del usuario.
4. **Tareas de ordenación de repositorio:** Se realizaron tareas de ordenación y limpieza en el repositorio del código. Incluyó la reorganización de directorios, la eliminación de archivos obsoletos y la estandarización de nombres de archivos y carpetas para mejorar la accesibilidad y el mantenimiento.
5. **Modificación del ReadMe:** Se actualizó el archivo ReadMe para reflejar los cambios recientes en el proyecto y proporcionar una guía más clara y detallada para los usuarios que quieran replicar el proyecto.

Debido a las fechas de vacaciones de Navidad, a los compromisos académicos y la gran demanda de los issues, no se logró completar lo previsto para este sprint, por lo que se decidió en la reunión del 12 de Enero ampliar 2 semanas más este sprint.

Sprint 4

Este sprint se desarrolló entre los días 26 de Enero y 9 de Febrero de 2024. Se realizaron las siguientes tareas y objetivos:

1. **Integración de Metamask:** Se integró la billetera Metamask para facilitar las transacciones en la aplicación. Esto incluyó configurar la autenticación y las firmas de transacciones a través de la popular billetera de criptomonedas, permitiendo a los usuarios interactuar de manera segura a través de la extensión de navegador de Metamask.
2. **Documentación de tecnologías utilizadas:** Tras el progresivo avance del proyecto y el amplio repertorio de frameworks y librerías utilizadas, se plasmaron y explicaron todas ellas en su correspondiente apartado en la memoria.

Sprint 5

Este sprint se desarrolló entre los días 9 de Febrero y 23 de Febrero de 2024. Se realizaron las siguientes tareas y objetivos:

1. **Creación de la guía de instalación:** Se elaboró una guía de instalación detallada para facilitar la configuración inicial del sistema por parte de los usuarios. Recogiendo las instrucciones paso a paso para la instalación del software, configuración del entorno y la conexión con las dependencias necesarias.
2. **Documentación del contexto teórico:** Se empezó a recoger en la memoria todos los fundamentos teóricos que fundamentan este proyecto, proporcionando una base sólida que ayude a entender el marco en el que se desarrolla la aplicación.
3. **Implementación de pantallas informativas:** Se implementaron dos pantallas clave en la aplicación móvil: la pantalla "Wallet", que permite al usuario la conexión con la billetera y muestra los datos de la cuenta seleccionada en la billetera Metamask, y la pantalla "Info", que incluye información actualizada sobre el precio del Ethereum, junto con un gráfico interactivo y una calculadora para conversiones.
4. **Implementar funcionalidad modificar contrato:** Se extendieron las funciones del contrato inteligente para permitir modificar un contrato existente. Esto incluyó la adición de métodos para actualizar

parámetros del contrato y funciones que permiten a los usuarios adaptar el contrato a necesidades cambiantes, asegurando así una mayor flexibilidad.

5. **Creación base de datos para el almacenamiento de las cuentas de los usuario:** Se implementó una base de datos utilizando Firebase de Google, aprovechando la amplia gama de funcionalidades que incluye. De esta forma se permitía a los usuarios iniciar sesión con su cuenta google, Facebook y Github entre muchas otras, facilitando un proceso de autenticación rápido y seguro, además de almacenar de manera eficiente los datos de las cuentas de los usuarios.

Sprint 6

Este sprint se desarrolló entre los días 23 de Febrero y 8 de Marzo de 2024. Se realizaron las siguientes tareas y objetivos:

1. **Modularizar código:** Debido a que los smart contracts en la Ethereum Virtual Machine (EVM) están sujetos a una restricción de tamaño para prevenir ataques de denegación de servicio, el contrato que me encontraba desarrollando se encontraba en el límite de tamaño permitido, impidiendo futuras ampliaciones. Para resolver esto, se optó por modularizar el contrato inteligente en diferentes componentes. Esto no solo evitó la necesidad de desplegar múltiples contratos, lo cual hubiera sido más costoso en términos de gas, sino que también ayudó en la organización del código.
2. **Validación de datos de entrada en Smart Contract:** Aunque inicialmente el contrato inteligente ya incluía algunas comprobaciones básicas en el código Solidity y la aplicación también verificaba las entradas, se realizaron mejoras significativas en estas validaciones para reforzar la seguridad.
3. **Ampliación funcionalidad SmartContract:** Se añadió código adicional al contrato inteligente para permitir que usuarios adicionales, aparte del dueño original, pudieran tener permisos de gestión del contrato. Esto fue diseñado para facilitar la colaboración permitiendo que usuarios autorizados puedan realizar cambios sin depender el propietario inicial.
4. **Modificación lógica contrato:** Se modificó la lógica del contrato para permitir la creación de contratos sin asignar inicialmente un

trabajador. Esta funcionalidad permitió mostrar contratos disponibles en una especie de "tablón de ofertas de trabajo", donde las personas en búsqueda de empleo pueden encontrar y seleccionar contratos que coincidan con sus habilidades y expectativas.

Los siguientes objetivos que se exponen a continuación, no han podido ser completados y han sido pospuestos hasta el sprint 10. Hasta ahora, la aplicación se ha estado probando de manera local en una pestaña del navegador por razones de eficiencia y comodidad. Sin embargo, al intentar realizar las pruebas con un teléfono móvil para funcionalidades que requieren ejecución nativa, como el escaneo de códigos QR o el acceso a la ubicación, se han encontrado errores debido a incompatibilidades con las bibliotecas y el sistema operativo Android. Por tanto, estos problemas se han tenido que posponer hasta que se solucione el problema de compatibilidad de la app con la biblioteca web3.

1. **Implementación de códigos QR:** La integración de códigos QR, diseñada para facilitar transacciones rápidas y seguras mediante escaneo, se ha pospuesto debido a problemas técnicos encontrados durante las pruebas en dispositivos móviles.
2. **Inclusión ubicación a los parámetros del contrato:** Del mismo modo que con los códigos QR, por la imposibilidad de probar los servicios de ubicación en un dispositivo móvil, se ha tenido que posponer.
3. **Creación de un filtro para buscar contratos:** Este issue ha sido pospuesto ya que depende directamente de la incorporación de la ubicación como parámetro para realizar el filtro.
4. **Creación de eventos para registrar alertas en los contratos:** Debido a la importancia de solucionar el problema de despliegue en teléfonos móvil, este issue ha pasado a segundo plano.

Sprint 7

Este sprint se desarrolló entre los días 8 de Marzo y 22 de Marzo de 2024. Se realizaron las siguientes tareas y objetivos:

1. **Resolver error compatibilidad app android y biblioteca web3:** Este issue tuvo prioridad máxima y se le dedicó únicamente para

el. Al ejecutar la aplicación en un dispositivo móvil aparecía un problema de compatibilidad significativo entre la biblioteca web3 y el motor JavaScript Hermes utilizado en React Native. Este problema incapacitaba cualquier avance para el que se requiriese las funcionales nativas del dispositivo móvil.

2. **adaptar aplicación web a aplicación móvil:** La aplicación en su primera etapa de desarrollo fue hecha desde un entorno web. Por lo que una vez solucionado el problema anterior que impedía desplegar el proyecto en un dispositivo móvil, fue necesario enfrentarse a diversos desafíos adicionales al adaptar características específicas de la web para funcionar en un entorno móvil.

Después de varias horas de esfuerzo, se logró identificar y corregir el error más importante hasta el momento. Además de cumplir con los requisitos establecidos para este sprint, también se consiguió implementar la funcionalidad para mostrar códigos QR con los detalles del contrato, el cual era uno de los objetivos que habían quedado pendientes del sprint anterior.

Sprint 8

Este sprint se desarrolló entre los días 22 de Marzo y 5 de Abril de 2024. Se realizaron las siguientes tareas y objetivos:

Este sprint se centró principalmente en resolver desafíos relacionados con la migración de la aplicación web a móvil, especialmente en lo que respecta a la conectividad con la billetera externa Metamask.

1. **Importar cuentas de Ganache a la app móvil de Metamask:** Esto representaba inicialmente una fase intermedia entre las pruebas locales y el despliegue en una red real. Por errores de conexión se llegó a la conclusión de que no era posible, por lo que se decidió omitir este paso ya que su eliminación no afecta el progreso hacia el entorno de producción.
2. **Gestionar conexión de la billetera desde la app:** Directamente conectar con Metamask resultó inviable debido a restricciones técnicas, por lo que se evaluó para adaptar el sistema para utilizar Wallet Connect junto con las librerías wagmi y viem, permitiendo la conexión con más de 300 billeteras diferentes

A.3. Estudio de viabilidad

El estudio de viabilidad es un análisis crítico realizado para evaluar la posibilidad y conveniencia de llevar a cabo un proyecto específico. Para determinar si este proyecto es viable, se van a desarrollar la viabilidad económica y la viabilidad legal.

Viabilidad económica

La viabilidad económica de un proyecto evalúa si los beneficios económicos previstos justifican los costos involucrados. Se considerarán los costes de recursos humanos, el material empleado y el Software usado

Coste de personal

Para realizar una estimación de los costos del proyecto, vamos a acotar el calculo a una duración de 9 meses, que ha sido el tiempo empleado en realizar el proyecto. Por otro lado, como solo ha habido una persona desarrollando (el alumno), se va a considerar un único trabajador para los cálculos Partiendo de un salario base promedio de un programador junior en España de 21000 brutos anuales, podemos estimar que el coste total personal sería:

Concepto	Coste (€)
Salario bruto (anual)	21000
Retención del IRPF (12 %)	2520
Seguridad Social (28,3 %)	5943
Salario neto (anual)	12537
Salario neto (mensual)	1044,75
Coste total en los 9 meses	15750

Tabla A.1: Coste total del personal

Coste hardware

El coste del hardware para la realización del proyecto únicamente ha sido necesario contar con un ordenador para el desarrollador. El ordenador utilizado para el desarrollo tiene una antigüedad de 7 años, por lo que ya ha sido amortizado y se desprecia el costo del ordenador.

Coste SoftWare

Las herramientas de software empleadas en este proyecto son principalmente gratuitas; sin embargo, los servicios de Firebase e Infura presentan ciertas restricciones en sus versiones sin costo que podrían limitar la operatividad al escalar la aplicación. FiraBase en su versión gratuita ofrece 1 GiB de almacenamiento, que bien es un limitante, y soporta hasta 50000 usuarios activos mensuales. Por otro lado, Infura proporciona 100000 peticiones diarias gratuitamente.

Considerando un escenario de total éxito en el lanzamiento de la aplicación, donde se estimen unos 200.000 usuarios activos mensuales, las necesidades podrían requerir una gran ampliación de almacenamiento, lo cual vendría acompañado de un gran volumen de operaciones sobre la base de datos. Para Infura se contemplaría la necesidad de escalar al plan más avanzado, que permite hasta 5 millones de peticiones diarias. Los costos, inicialmente en dólares han sido convertidos a euros a 0,92€

Concepto	Coste (€)
Infura	917,90
190.500.000 operaciones lectura	113
999 GiB Almacenamiento	165,22
53.100.000 operaciones escritura	86,74
200.000 usuarios activos	633,35
Coste total mensual	1916,21

Tabla A.2: Coste total del Software

Beneficios

El proyecto no se ha realizado con el objetivo de obtener un beneficio económico. Este producto ha sido desarrollado con el fin de implementar una herramienta para abordar los desafíos del mercado laboral, especialmente en sectores afectados por la informalidad como la agricultura y los servicios domésticos. Se considera por tanto que este proyecto tiene un carácter social, destinado a mejorar las condiciones laborales y a promover la equidad en el trabajo.

Se considera que la financiación del proyecto debería provenir principalmente de inversores interesados en la optimización y reducción de costos

del proceso de contratación. Adicionalmente, el gobierno español representa un potencial inversor clave, ya que podría promover activamente el uso de esta aplicación para fomentar prácticas laborales adecuadas y combatir la economía sumergida.

Viabilidad legal

Esta sección se centra en analizar los aspectos legales asociados con el desarrollo y distribución de una solución basada en blockchain. Los puntos claves que incluyen el análisis de las licencias de software y el cumplimiento de las leyes vigentes.

Licencias de software

En el proyecto se emplea diversas bibliotecas y frameworks que están regidos bajo licencias de software específicas, que definen los términos y condiciones para su uso, modificación y distribución. A continuación, se va a listar las diferentes licencias de cada dependencia de mi proyecto, en orden de menos restrictiva a mas restrictiva:

- **Licencia MIT:** No impone restricciones significativas, permitiendo uso comercial, modificación, distribución y uso privado.
- **Licencia Apache 2.0:** Similar a la licencia MIT, pero también proporciona una protección explícita contra patentes.
- **Términos de servicio de Google:** Específicos para los productos de Google, no son licencias de software per se, pero regulan cómo se pueden usar los servicios.
- **GNU GPL:** Permite uso comercial, modificación, distribución y uso privado, pero cualquier versión modificada debe también ser libre.
- **GNU GPL-3.0:** Similar a la GPL pero con términos adicionales para cerrar algunas brechas legales en ciertos escenarios.

Dependencia	Licencia
React Native	Licencia MIT
Expo	Licencia MIT
Firebase	términos de servicio de Google
Solidity	Licencia GPL-3.0
Remix	Licencia GPL-3.0
Truffle	Licencia MIT
Ganache	Licencia MIT
MetaMask	Licencia GPL-3.0
WalletConnect	Licencia MIT
Node.js	Licencia MIT
OpenZeppelin	Licencia Apache 2.0
web3	Licencia MIT
ethers	Licencia MIT

Tabla A.3: Dependencias del proyecto y licencias bajo las cuales están registradas

Regulación legal

Debido a que este proyecto implicará la gestión de datos personales, es fundamental cumplir con el Reglamento General de Protección de Datos (RGPD) de la UE y la legislación española correspondiente, como la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD). Esto implica garantizar la protección adecuada de los datos personales y obtener consentimientos claros por parte de los usuarios para usar sus datos.

Actualmente no hay ninguna legislación vigente en España en cuanto a los contratos inteligentes. Sin embargo, estos debería de cumplir con el código civil en términos de formación de contratos, capacidad para contratar y consentimiento. Por su puesto, la aplicación deberá cumplir con la legislación laboral española, asegurando que los contratos creados cumplan con las normativas de trabajo, incluyendo las mínimas garantías y derechos para los trabajadores.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En esta sección se hace referencia a los requerimientos que debe cumplir el software para satisfacer las necesidades del cliente e identificar los componentes necesarios para entregar un producto adecuado.

B.2. Objetivos generales

A lo largo de la memoria ya se han definido distintos objetivos del proyecto, pero estos pueden ser resumidos en los siguientes puntos:

- **Tecnología Blockchain:** Uso de la tecnología blockchain como base para la creación de un sistema de contratación descentralizado y seguro. Se busca garantizar la transparencia e inmutabilidad de los datos, permitiendo que todas las transacciones queden registradas de forma segura.
- **Contratos Inteligentes:** Desarrollo de contratos inteligentes que aseguran el cumplimiento de acuerdos laborales. Estos contratos deben de ser capaz de autoejecutarse en función de las condiciones preestablecidas por las partes. El principal objetivo es reducir la necesidad de intermediarios.
- **Identificación segura mediante dispositivo móvil:** Implementación de métodos de autenticación y verificación seguros utilizando

dispositivos móviles, como biometría o códigos QR. Se pretende asegurar que solo los usuarios autorizados puedan acceder a un contrato y realizar operaciones dentro del sistema.

- **Desarrollo de Aplicación Móvil:** Creación de una aplicación móvil que sirva de interfaz para interactuar con los contratos inteligentes y la blockchain, escondiendo toda la complejidad al usuario. La aplicación debe de estar diseñada para ser intuitiva, con el objetivo de que sea fácilmente utilizable para personas con cualquier nivel de habilidades tecnológicas.

B.3. Catálogo de requisitos

B.4. Especificación de requisitos

CU-1	Ejemplo de caso de uso
Versión	1.0
Autor	Alumno
Requisitos asociados	RF-xx, RF-xx
Descripción	La descripción del CU
Precondición	Precondiciones (podría haber más de una)
Acciones	<ol style="list-style-type: none"> 1. Pasos del CU 2. Pasos del CU (añadir tantos como sean necesarios)
Postcondición	Postcondiciones (podría haber más de una)
Excepciones	Excepciones
Importancia	Alta o Media o Baja...

Tabla B.1: CU-1 Nombre del caso de uso.

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf.

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.