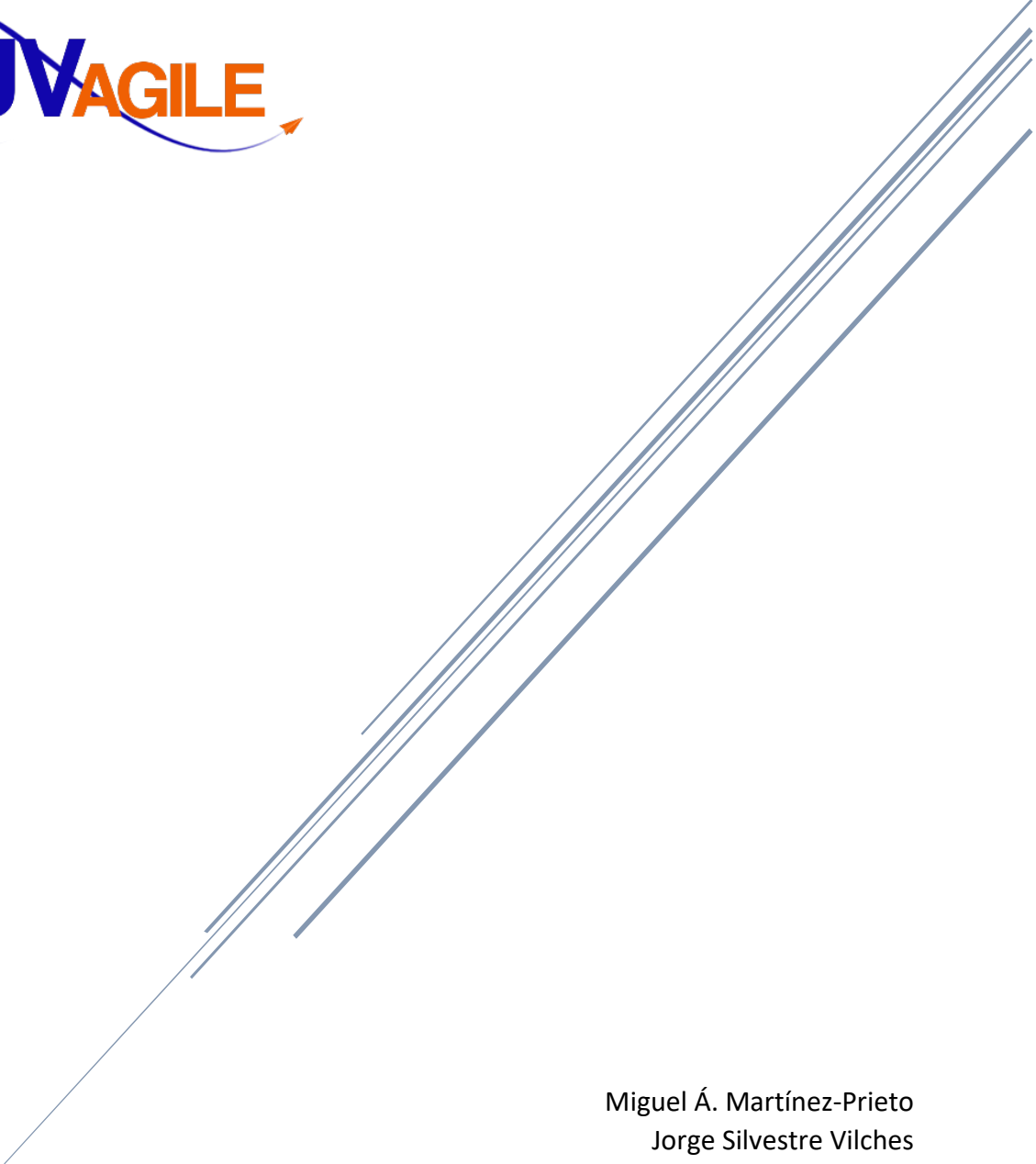


TRABAJO FIN DE ESTUDIOS ÁGIL

Guía de referencia para la realización de TFGs / TFMs con
UVAGILE

A series of four parallel, light blue diagonal lines extending from the bottom left towards the top right of the page.

Miguel Á. Martínez-Prieto
Jorge Silvestre Vilches

Versión 0.9
Septiembre, 2020

Índice de contenidos

1. General.....	2
1.1. Bienvenid@	2
1.2. Soft skills.....	3
1.3. Trabajo Diario.....	3
1.4. Acuerdos de Trabajo (<i>Working agreement</i>).....	4
2. Incrementos	5
2.1. Incrementos	5
2.2. Presentación.....	5
2.3. Memoria.....	6
2.4. Código Fuente	6
3. Espacio de trabajo.....	7
4. Herramientas digitales	8
4.1. Herramientas Digitales.....	8
4.2. Tablero de Proyecto	8
4.3. Canal de Comunicación	9
4.4. Control de Versiones	9
4.5. Otras herramientas	9

1. General

1.1. Bienvenid@

En primer lugar, queremos darte la bienvenida a este proyecto académico que vas a iniciar. Muchas gracias por haber confiado en nosotros para que formemos parte de él. Esperamos poder ayudarte todo lo posible, además de aprender contigo. Desde este momento, somos parte del mismo equipo, así que vamos a trabajar duro para conseguir el mejor proyecto posible. Por falta de ilusión no será ;).



Este proyecto se organizará y desarrollará de acuerdo con la metodología de enseñanza ágil **UVagile**, desarrollada en el ámbito del grupo **datAi**, de la Universidad de Valladolid. Esta metodología está fuertemente inspirada en los marcos de trabajo ágiles (y, más concretamente, en Scrum) y plantea una organización iterativa e incremental del proyecto, en la que al final de cada *sprint* se entregará un incremento del producto en desarrollo (entendiendo como tal, un prototipo que ofrezca un subconjunto completo de los requisitos, que siempre incluirá la memoria técnica del mismo). Cada *sprint* planifica una serie de reuniones periódicas, que buscan que todos seamos conscientes del avance del proyecto y que podamos trabajar conjuntamente en la resolución de cualquier bloqueo.

Antes de comenzar el proyecto, tómate unos minutos en leer este pequeño cuaderno, porque seguro que te ayudará a comprender la organización, contenidos y dinámica de uso de este espacio de trabajo. Cada uno de sus directorios está explicado en una sección del cuaderno, mientras que los diferentes contenidos de cada uno de ellos se explican como páginas independientes (dentro de cada sección).

Actualmente, estamos en la página correspondiente al directorio *raíz*, que cuenta con los siguientes ficheros:

- **soft-skills.xlsx** (fichero Excel para el registro de tus **soft skills**).
- **work-diary.xlsx** (fichero Excel para la gestión del **trabajo diario** en el proyecto).
- **working-agreement.docx** (fichero Word que establece los **acuerdos de trabajo** de nuestro equipo).

Este directorio dos subdirectorios:

- **incrementos** (organiza cada uno de los **incrementos** generados a lo largo del proyecto).
- **workspace** (define tu propio **espacio de trabajo**, donde podrás organizar la información de acuerdo con tus necesidades).

Finalmente, también encontrará una sección dedicada a introducir y motivar las **herramientas digitales** que utilizaremos en este proyecto.

1.2. Soft skills

Las **soft skills** son habilidades relacionadas con el desarrollo personal, que no dependen de un ámbito temático o disciplinario específico, sino que aparecen en todos los dominios de la actuación profesional y académica. Por lo tanto, son competencias innatas a cada uno de nosotros, que debemos conocer, trabajar y explotar tanto en nuestro trabajo como en nuestra vida cotidiana.



La metodología UVagile busca potenciar algunas de estas competencias, para lo cual primero tenemos que ser conscientes de en qué estado se encuentran. Nuestra propuesta es utilizar la herramienta proporcionada desarrollada en el proyecto SkillsMatch¹ para hacer una evaluación exhaustiva de tus competencias al principio y al final del proyecto, y así observar su evolución durante estos meses.

Para ello, deberás completar el cuestionario que se propone para cada competencia y obtener una evaluación cuantitativa de todas ellas. Una vez finalizado, copia los resultados de cada competencia en la columna izquierda de la hoja de cálculo, indicando la fecha en la que has completado la información. Al final del proyecto vuelve a realizar el mismo ejercicio (las preguntas cambian cada vez que se realiza el cuestionario) y, nuevamente, copia los valores obtenidos en la hoja de cálculo (columna derecha), indicando la fecha de realización.

1.3. Trabajo Diario

La **gestión del tiempo** es una de las *soft skills* más relevante en el ámbito laboral (y también en el académico) y trasciende a la planificación y la organización de nuestro trabajo diario. La gestión del tiempo tiene que ver con nuestra capacidad para resolver las tareas planificadas en el tiempo del que disponemos para ello y, en el caso de un proyecto académico como este, la capacidad de poder alcanzar todos los objetivos planteados en los ECTS planificados para ello.



Gestionar el tiempo es una tarea compleja, sobre todo porque no siempre somos conscientes del tiempo que invertimos en nuestro trabajo. Por este motivo, te proponemos utilizar la hoja Excel **work-diary.xlsx** (disponible en el directorio raíz del espacio de trabajo) para que vayas registrando las diferentes actividades que realices durante el proyecto, indicando la fecha y el *sprint* al que pertenecen, así como su tipo de actividad (si echas en falta algún tipo de actividad, indícanoslo) y cualquier otro comentario sobre la tarea que te ayude a saber qué trabajo hiciste en esa tarea.

Este documento no es una forma de controlar el tiempo que trabajar en el proyecto, si no sólo un "diario de a bordo" que te permita ser consciente del esfuerzo que inviertes y te ayude, de cara a tu futuro profesional, a planificar mejor el esfuerzo que puedes dedicar a cada uno de los tipos de tareas que se pueden presentar en un proyecto de desarrollo.

¹ <https://beta.skillsmatch.eu/assessment>

1.4. Acuerdos de Trabajo (*Working agreement*)

Como se indicaba inicialmente, haber iniciado este proyecto significa que ya somos miembros del mismo equipo de trabajo y que compartimos un mismo objetivo, que no es más que consolidar el mejor producto posible, de acuerdo con los objetivos del proyecto.



Todo equipo requiere unas "normas básicas" que nos ayuden a convivir y a colaborar en este proyecto de una forma satisfactorio para todos. Estas normas abordarán diferentes dimensiones, desde la disponibilidad de cada uno de nosotros, a los horarios o las condiciones en las que se celebrarán las diferentes reuniones que tengamos que llevar a cabo.

Nuestra propuesta inicial se presenta en el documento **working-agreement.docx** (disponible en el directorio raíz de el espacio de trabajo). Revisala cuidadosamente y coméntanos todo aquello que pueda ser difícil de cumplir para ti y lo replanteamos de forma que nos encaje bien a todos. Este **working agreement** no es una imposición, si no un acuerdo consensuado en el que todos nos sintamos reflejados y que todos podamos cumplir fácilmente.

El contenido del *working agreement* no es estático, si no que está abierto a cualquier cambio que sea necesario para adaptarnos a cualquier circunstancia futura que impida que alguno de los miembros del equipo pueda cumplirlo. No obstante, cualquier cambio deberá ser también consensuado, sin ningún tipo de imposición por ninguna de nosotros.

2. Incrementos

2.1. Incrementos

Este directorio contiene cada uno de los incrementos que se generen durante el proyecto. Recuerda que al final de cada *sprint* se generará un nuevo incremento, que contendrá una "versión funcional" del producto en desarrollo (que debe satisfacer los objetivos planificados en el alcance del *sprint*).



Cada incremento se nombrará como: `aaaa-mm-dd-Sx`, donde "yyyy-mm-dd" indicará la fecha de entrega del incremento (en formato año-mes-día) y "x" referenciará el identificador del *sprint* actual. Dentro de este directorio se depositarán los siguientes contenidos:

- **project.pptx** (fichero PowerPoint que contendrá la **presentación** del proyecto preparada para la *review*).

Además, se consideran (al menos) dos subdirectorios:

- **report** (que contendrá todos los documentos que consoliden la memoria técnica del proyecto).
- **src** (que contendrá el **código fuente** desarrollado para implementar los requisitos abordados en el *sprint* actual).

2.2. Presentación

La **presentación** es un elemento fundamental para comunicar el avance del proyecto durante la *review* que se celebrará al final de cada *sprint*. No hay restricciones respecto al contenido y organización de esta presentación, pero sí debería proporcionar información detallada sobre el estado actual del proyecto, destacando los avances consolidados desde el *sprint* anterior, analizando las dificultades y bloqueos encontrados y las propuestas de mejora de cara al siguiente *sprint*.



Esta presentación irá evolucionando a lo largo del proyecto, de forma que en cada incremento proporcionará una descripción más detallada de los resultados obtenidos y el trabajo realizado, hasta alcanzar el final del proyecto. El objetivo es que la presentación que se haga en la *review* del último *sprint* sea la presentación que se utilice en el acto de defensa y que la propia *review* sea un ensayo de la presentación, destinada a afinar detalles finales y consolidar una propuesta de calidad para el acto de defensa.

2.3. Memoria

La **memoria técnica** del proyecto organiza y detalla todas las decisiones tomadas para consolidar el resultado final, incluyendo una revisión del contexto en el que se desarrolla (desde una perspectiva actual), un análisis del estado del arte relacionado y una explicación pormenorizada de todos los aspectos técnicos considerados.



Es importante comenzar con la redacción de la memoria desde el primer *sprint* del proyecto e ir actualizándola, iterativa e incrementalmente, durante todo él, reflejando todos los avances que se hayan consolidado en el *sprint*. De esta forma, evitaremos uno de los "males comunes" de estos trabajos fin de estudios: "el atracón final de escritura", que termina resultando en memorias pobremente estructuradas y con una redacción de poca calidad que, generalmente, desluce el trabajo realizado.

Por este motivo, cada incremento debe presentar una versión actualizada de la memoria técnica, que incluirá todo lo relacionado con el trabajo realizado en el *sprint* actual, así como una versión revisada de los contenidos entregados en la memoria correspondiente incluida en el incremento anterior.

2.4. Código Fuente

El **código fuente** consolida el producto software desarrollado como parte del proyecto. Este directorio debe tener la estructura adecuada para organizar todo el código fuente y otros contenidos necesarios para desplegar y poner en funcionamiento el software construido, incluida todas las dependencias que pueda requerir.

Es aconsejable añadir, al menos, un fichero README que ayude a entender los contenidos del directorio y facilitar el despliegue del software.

```
21 string sInput;
22 int iLength, iPos;
23 double dTime;
24 bool again = true;
25
26
27 while (again) {
28     iPos = -1;
29     again = false;
30     while(iPos < sInput);
31     system("cls");
32     string sTemp(sInput);
33     iLength = sTemp.length();
34     if (iLength < 4) {
35         cout << "Error: ";
36         continue;
37     } else if (iLength > 10) {
38         cout << "Error: ";
39         continue;
40     }
41     while (iPos < sTemp.length()) {
42         cout << sTemp[iPos];
43         cout << " ";
44         iPos++;
45     }
46     cout << endl;
47     again = true;
48 }
```

3. Espacio de trabajo

El uso y organización del **espacio de trabajo (*workspace*)** está completamente abierto a tus necesidades, así que puedes utilizarlo de la forma que mejor encaje con las características del proyecto y con tu forma de trabajar.



Aunque nosotros no entraremos a revisar los contenidos del *workspace* (salvo que sea completamente necesario), sí es aconsejable que expliques su estructura y contenido, para mantener un orden y una organización aceptable. Por ejemplo, puedes explicarlo en esta misma página para que esté accesible para todos nosotros.

4. Herramientas digitales

4.1. Herramientas Digitales

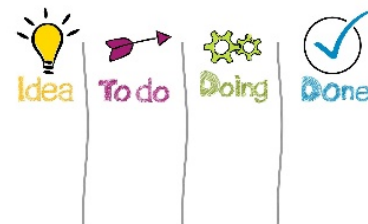
El uso de **herramientas digitales** simplifica las tareas habituales de un proyecto y, sobre todo, facilita la interacción entre cada uno de los miembros del equipo. Por este motivo, planteamos el uso de ciertas herramientas (de acuerdo con nuestra experiencia), aunque estamos abiertos a reemplazarlas o complementarlas de acuerdo con tus necesidades y conocimiento particulares. La única restricción es que sean herramientas *open source*, tengan una licencia gratuita o, en su defecto, la UVA disponga de licencia para poder usarlas en proyectos académicos.

Nuestra herramienta de referencia será **Microsoft Teams**, donde centralizaremos el desarrollo del proyecto y aprovecharemos varias de sus utilidades, así como sus posibilidades de integración con otras herramientas digitales.



4.2. Tablero de Proyecto

El **tablero de proyecto** es una herramienta de gestión que nos permite visualizar el estado global del proyecto de forma rápida y efectiva. Para ello, el tablero se divide en diferentes columnas, que representan los diferentes estados por los que pueden pasar las tareas planificadas para su realización en un determinado *sprint*.



Cada tarea se presenta como una tarjeta independiente, que tiene un **título** y una **descripción**, en la que se expone el trabajo a realizar y su encaje dentro del proyecto. Para ello, se asocia con el **objetivo** que aborda la tarea y se estima el **esfuerzo** que costará completarla. Finalmente, se hacen explícitos los **criterios de aceptación** que debe satisfacer el resultado de la tarea para considerar que esta se ha completado satisfactoriamente.

La tarea parte de un estado "Planificada" y evoluciona hacia "En desarrollo" cuando comienza su realización. Si todo va bien, la tarea alcanza el estado "Completada", una vez que se han validado los criterios de aceptación. En caso de bloqueo, la tarea debería pasar a estado "Bloqueada" hasta que se consiga revertir la situación que impide su avance.

Además de las tareas, el tablero suele contener una columna "Backlog" que recoge los objetivos del proyecto, indicando de forma visual cuáles se han conseguido ya, cuáles están en desarrollo y cuáles no han comenzado aún.

Para implementar el tablero de aprendizaje, utilizaremos **Trello**, dada su integración en Teams y la existencia de una licencia gratuita que nos permite utilizar sus servicios básicos de acuerdo con nuestras necesidades. Antes de poder acceder al tablero de proyecto, tendrás que registrarte en Trello y acceder a través del enlace que te suministraremos.



4.3. Canal de Comunicación

La **comunicación** en un equipo es el alma de un equipo, por lo que debe ser continua y fluida en todo momento, más allá de las diferentes ceremonias programadas a lo largo del *sprint*. Por este motivo, utilizaremos un chat que nos permitirá estar en contacto continuo y responder con agilidad a las cuestiones que se planteen, siempre dentro de las condiciones de disponibilidad acordadas en vuestro *working agreement*.

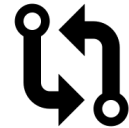


La comunicación también la delegaremos en **Teams**, dada la efectividad de sus canales de comunicación escritos y las facilidades que proporciona para la realización de videoconferencias.



4.4. Control de Versiones

Todos los productos que se generen en el proyecto deben estar sujetos a **control de versiones**, con el fin de poder mantener un histórico de su evolución que recoja todos los cambios que se realicen durante el proyecto.



- Por una parte, las versiones de los **documentos del proyecto** (memoria, presentaciones, etc.) puede realizarse de forma sencilla y efectiva utilizando los mecanismos que proporciona **Teams** a través de **SharePoint** y que permite acceder a todas las versiones de los documentos almacenados en el repositorio del proyecto.
- Sin embargo, esta utilidad no es suficiente para gestionar la evolución de los contenidos más técnicos del proyecto, principalmente su **código fuente**. En este caso, se sugiere utilizar **GitHub**, aprovechando la licencia (sin limitaciones) que ofrece a estudiantes.

4.5. Otras herramientas

Las **herramientas** anteriores son el núcleo de nuestro proyecto, pero estamos abiertos a incluir herramienta de gestión que consideres que puede ayudarnos a llevar a cabo el proyecto.



Cualquier propuesta será estudiada y, más que probablemente, aceptada siempre que no haya restricciones para su integración en el proyecto de forma **legal**.